The Journal of Machine Learning Research

Volume 13

Print-Archive Edition

Pages 1–1262

Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 13
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 2012.

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

JMLR-MLOSS Editors

JMLR Editorial Board

Journal of Machine Learning Research

Volume 13, 2012

Journal of Machine Learning Research 13 (2012) 1-26 Submitted 2/11; Revised 10/11; Published 1/12

Distance Metric Learning with Eigenvalue Optimization

Yiming Ying Y.YING@EXETER.AC.UK
College of Engineering, Mathematics and Physical Sciences
University of Exeter
Harrison Building, North Park Road
Exeter, EX4 4QF, UK

Peng Li LIPENG@IEEE.ORG
Department of Engineering Mathematics
University of Bristol
Merchant Venturers Building, Woodland Road
Bristol, BS8 1UB, UK

Editors: Sören Sonnenburg, Francis Bach and Cheng Soon Ong

Abstract

The main theme of this paper is to develop a novel eigenvalue optimization framework for learning a Mahalanobis metric. Within this context, we introduce a novel metric learning approach called *DML-eig* which is shown to be equivalent to a well-known eigenvalue optimization problem called minimizing the maximal eigenvalue of a symmetric matrix (Overton, 1988; Lewis and Overton, 1996). Moreover, we formulate *LMNN* (Weinberger et al., 2005), one of the state-of-the-art metric learning methods, as a similar eigenvalue optimization problem. This novel framework not only provides new insights into metric learning but also opens new avenues to the design of efficient metric learning algorithms. Indeed, first-order algorithms are developed for DML-eig and LMNN which only need the computation of the largest eigenvector of a matrix per iteration. Their convergence characteristics are rigorously established. Various experiments on benchmark data sets show the competitive performance of our new approaches. In addition, we report an encouraging result on a difficult and challenging face verification data set called Labeled Faces in the Wild (LFW).

Keywords: metric learning, convex optimization, semi-definite programming, first-order methods, eigenvalue optimization, matrix factorization, face verification

1. Introduction

Distance metrics are fundamental concepts in machine learning since a proper choice of a metric has crucial effects on the performance of both supervised and unsupervised learning algorithms. For example, the k-nearest neighbor (k-NN) classifier depends on a distance function to identify the nearest neighbors for classification. The k-means algorithm depends on the pairwise distance measurements between examples for clustering, and most information retrieval methods rely on a distance metric to identify the data points that are most similar to a given query. Recently, learning a distance metric from data has been actively studied in machine learning (Bar-Hillel et al., 2005; Davis et al., 2007; Goldberger et al., 2004; Rosales and Fung, 2006; Shen et al., 2009; Torresani and Lee, 2007; Weinberger et al., 2005; Weinberger and Saul, 2008; Xing et al., 2002; Ying et al., 2009). These methods have been successfully applied to many real-world application domains including information retrieval, face verification, image recognition (Chopra et al., 2005; Guillaumin et al., 2009; Hoi et al., 2006) and bioinformatics (Kato and Nagano, 2010; Vert et al., 2007).

Most metric learning methods attempt to learn a distance metric from side information which is often available in the form of pairwise constraints, that is, pairs of *similar* data points and pairs of *dissimilar* data points. The information of similarity or dissimilarity between a pair of examples can easily be collected from the label information in supervised classification. For example, we can reasonably let two samples in the same class be a similar pair and samples in the distinct classes be a dissimilar pair. In semi-supervised clustering, a small amount of knowledge is available concerning pairwise (must-link or cannot-link) constraints between data items. This side information delivers the message that a must-link pair of samples is a similar pair and a cannot-link one is a dissimilar pair. A common theme in metric learning is to learn a distance metric such that the distance between similar examples should be relatively smaller than that between dissimilar examples. Although the distance metric can be a general function, the most prevalent one is the Mahalanobis metric defined by $d_M(x_i,x_j) = \sqrt{(x_i - x_j)^\top M(x_i - x_j)}$ where M is a positive semi-definite (p.s.d.) matrix.

In this work we restrict our attention to learning a Mahalanobis metric for k-nearest neighbor (k-NN) classification. However, the proposed methods below can easily be adapted to metric learning for semi-supervised k-means clustering. Our main contribution is summarized as follows. Firstly, we introduce a novel approach called *DML-eig* mainly inspired by the original work of Xing et al. (2002). Although our ultimate target is similar to theirs, our methods are essentially different. In particular, we can show our approach is equivalent to a well-known eigenvalue optimization problem called *minimizing the maximal eigenvalue of a symmetric matrix* (Lewis and Overton, 1996; Overton, 1988). We further show that the above novel optimization formulation can also be extended to LMNN (Weinberger et al., 2005) and low-rank matrix factorization for collaborative filtering (Srebro et al., 2004). Secondly, in contrast to the full eigen-decomposition used in many existing approaches to metric learning, we will develop novel approximate semi-definite programming (SDP) algorithms for DML-eig and LMNN which only need the computation of the largest eigenvector of a matrix per iteration. The algorithms combine and develop the Frank-Wolfe algorithm (Frank and Wolfe, 1956; Hazan, 2008) and Nesterov's smoothing techniques (Nesterov, 2005). Finally, its rigorous convergence characteristics will also be established, and experiments on various UCI data sets and benchmark face data sets show the competitiveness of our new approaches. In addition, we report an encouraging result on a challenging face verification data set called Labeled Faces in the Wild (Huang et al., 2007).

The paper is organized as follows. In Section 2, we propose our new approach (DML-eig) for distance metric learning and show its equivalence to the well-known eigenvalue optimization problem. In addition, a generalized eigenvalue-optimization formulation will be established for LMNN and low-rank matrix factorization for collaborative filtering (Srebro et al., 2004). In Section 3, based on eigenvalue optimization formulations of DML-eig and LMNN, we develop novel first-order algorithms. Their convergence rates are successfully established. Section 4 discusses the related work. In Section 5, our proposed methods are compared with the state-of-the-art methods through extensive experiments. The last section concludes the paper.

2. Metric Learning Model and Equivalent Formulation

We begin by introducing useful notations. Let $\mathbb{N}_n = \{1,2,\ldots,n\}$ for any $n \in \mathbb{N}$. The space of symmetric d times d matrices will be denoted by $\mathbb{S}^d$ and the cone of p.s.d. matrices is denoted by

$\mathbb{S}^d_+$. For any $X,Y \in \mathbb{R}^{d\times n}$, we denote the inner product in $\mathbb{S}^d$ by $\langle X,Y\rangle := \mathbf{Tr}(X^\top Y)$ where $\mathbf{Tr}(\cdot)$ denotes the trace of a matrix. The standard norm in Euclidean space is denoted by $\|\cdot\|$.

Throughout the paper, the training data is given by $\mathbf{z} := \{(x_i,y_i) : i \in \mathbb{N}_n\}$ with input $x_i = (x_i^1, x_i^2, \ldots, x_i^d) \in \mathbb{R}^d$, class label y_i (not necessary binary) and later on we use the convention $X_{ij} = (x_i - x_j)(x_i - x_j)^\top$. Then, for any $M \in \mathbb{S}^d_+$, the associated Mahalanobis distance between x_i and x_j can be written as $d^2_M(x_i,x_j) = (x_i - x_j)^\top M(x_i - x_j) = \langle X_{ij}, M\rangle$. Let $\mathcal{S}$ index the similar pairs and $\mathcal{D}$ index the dissimilar pairs. For instance, if (x_i,x_j) is a similar pair we denote it by $\tau = (i,j) \in \mathcal{S}$, and write X_{ij} as X_τ for simplicity.

Given a set of pairwise distance constraints, the target of metric learning is to find a distance matrix M such that the distance between the dissimilar pairs is large and the distance between the similar pairs is small. There are many possible criteria to realize this intuition. Our model is mainly inspired by Xing et al. (2002) where the authors proposed to maximize the sum of distances between dissimilar pairs, while maintaining an upper bound on the sum of squared distances between similar pairs. Specifically, the following criterion was used in Xing et al. (2002):

$$\begin{array}{cl} \max_{M\in\mathbb{S}^d_+} & \sum_{(i,j)\in\mathcal{D}} d_M(x_i,x_j) \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} d^2_M(x_i,x_j) \leq 1. \end{array} \tag{1}$$

An iterative projection method was proposed to solve the above problem. However, it usually takes a long time to converge and the algorithm needs the computation of the full eigen-decomposition of a matrix in each iteration.

In this paper, we propose to maximize the minimal squared distances between dissimilar pairs while maintaining an upper bound for the sum of squared distances between similar pairs, that is,

$$\begin{array}{cl} \max_{M\in\mathbb{S}^d_+} & \min_{(i,j)\in\mathcal{D}} d^2_M(x_i,x_j) \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} d^2_M(x_i,x_j) \leq 1. \end{array} \tag{2}$$

Now, let $X_\mathcal{S} = \sum_{(i,j)\in\mathcal{S}} X_{ij}$ we can rewrite problem (2) as follows:

$$\begin{array}{cl} \max_{M\in\mathbb{S}^d_+} & \min_{\tau\in\mathcal{D}} \langle X_\tau, M\rangle \\ \text{s.t.} & \langle X_\mathcal{S}, M\rangle \leq 1. \end{array} \tag{3}$$

This problem is obviously a semi-definite programming (SDP) since it is equivalent to

$$\begin{array}{cl} \max_{M\in\mathbb{S}^d_+} & t \\ \text{s.t.} & \langle X_\tau, M\rangle \geq t,\ \forall \tau = (i,j) \in \mathcal{D}, \\ & \langle X_\mathcal{S}, M\rangle \leq 1. \end{array}$$

In contrast to problem (1), the objective function and the constraints in (3) are linear with respect to (w.r.t.) M. As shown in the next subsection, this simple but important property[1] plays a critical role in formulating problem (2) as an eigenvalue optimization problem. This equivalent formulation is key to the design of efficient algorithms in Section 3.

The generation of the pairwise constraints plays an important role in learning a metric. If labels are known then the learning setting is often referred to as supervised metric learning which can

[1] One might consider replacing the objective $\sum_{(i,j)\in\mathcal{D}} d_M(x_i,x_j)$ in problem (1) by $\sum_{(i,j)\in\mathcal{D}} d^2_M(x_i,x_j)$. This would also lead to a simple linear constraint and linear objective function. However, as mentioned in Xing et al. (2002), it would result in M always being rank 1 (i.e., the data are always projected onto a line).

further be divided into two categories: the *global metric learning* and the *local metric learning*. The global approach learns the distance metric in a global sense, that is, to satisfy all the pairwise constraints simultaneously. The original model in Xing et al. (2002) is a global method which used all the similar pairs (same labels) and dissimilar pairs (distinct labels). The local approach is to learn a distance metric only using local pairwise constraints which usually outperforms the global methods as observed in many previous studies. This is reasonable in the case of learning a metric for the k-NN classifiers since k-NN classifiers are influenced most by the data items that are close to the test/query examples. Since we are mainly concerned with learning a metric for k-NN classifier, the pairwise constraints for DML-eig are generated locally, that is, the similar/dissimilar pairs are k-nearest neighbors. The details can be found in the experimental section.

2.1 Equivalent Formulation as Eigenvalue Optimization

In this section we establish a min-max formulation of problem (3), which is finally shown to be equivalent to an eigenvalue optimization problem called *minimizing the maximal eigenvalue of symmetric matrices* (Lewis and Overton, 1996; Overton, 1988).

For simplicity of notation, for any $X \in \mathbb{S}^d$, we denote its maximum eigenvalue of $X \in \mathbb{S}^d$ by $\lambda_{\max}(X)$. Let D be the number of dissimilar pairs and the simplex is denoted by

$$\triangle = \{u \in \mathbb{R}^D : u_\tau \geq 0, \ \sum_{\tau \in \mathcal{D}} u_\tau = 1\}.$$

We also denote the *spectrahedron* by

$$\mathcal{P} = \{M \in \mathbb{S}^d_+ : \mathbf{Tr}(M) = 1\}.$$

Now we can show problem (3) is indeed an eigenvalue optimization problem.

Theorem 1. *Assume that $X_{\mathcal{S}}$ is invertible and, for any $\tau \in \mathcal{D}$, let $\widetilde{X}_\tau = X_{\mathcal{S}}^{-1/2} X_\tau X_{\mathcal{S}}^{-1/2}$. Then, problem (3) is equivalent to the following problem*

$$\max_{S \in \mathcal{P}} \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S \rangle, \tag{4}$$

which can further be written as an eigenvalue optimization problem:

$$\min_{u \in \triangle} \max_{S \in \mathcal{P}} \langle \sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau, S \rangle = \min_{u \in \triangle} \lambda_{max} \Big(\sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau \Big). \tag{5}$$

Proof. Let M^* be an optimal solution of problem (3) and $\widetilde{M}^* = \frac{M^*}{\langle X_{\mathcal{S}}, M^* \rangle}$. Then, we have $\langle X_{\mathcal{S}}, \widetilde{M}^* \rangle = 1$ and

$$\min_{\tau \in \mathcal{D}} \langle X_\tau, \widetilde{M}^* \rangle = \min_{\tau \in \mathcal{D}} \langle X_\tau, M^* \rangle / \langle X_{\mathcal{S}}, M^* \rangle \geq \min_{\tau \in \mathcal{D}} \langle X_\tau, M^* \rangle,$$

since $\langle X_{\mathcal{S}}, M^* \rangle \leq 1$. This implies that $\widetilde{M}^*$ is also an optimal solution. Consequently, problem (3) is equivalent to, up to a scaling constant,

$$\arg \max_{M \in \mathbb{S}^d_+} \{ \min_{\tau \in \mathcal{D}} \langle X_\tau, M \rangle : \langle X_{\mathcal{S}}, M \rangle = 1 \}. \tag{6}$$

Noting that $\min_{\tau \in \mathcal{D}} \langle X_\tau, M \rangle = \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \langle X_\tau, M \rangle$, the desired equivalence between (4) and (3) follows by changing variable $S = X_{\mathcal{S}}^{1/2} M X_{\mathcal{S}}^{1/2}$ in formulation (6).

Also, note from Overton (1988) that $\max_{M\in\mathcal{P}}\langle X,M\rangle = \lambda_{\max}(X)$. By the min-max theorem, problem (4) can further be written by a well-known eigenvalue optimization problem:

$$\min_{u\in\triangle}\max_{M\in\mathcal{P}}\langle \sum_{\tau\in\mathcal{D}} u_\tau \widetilde{X}_\tau, M\rangle = \min_{u\in\triangle}\lambda_{\max}\Big(\sum_{\tau\in\mathcal{D}} u_\tau\widetilde{X}_\tau\Big).$$

This completes the proof of the theorem. □

The problem of minimizing the maximal eigenvalue of a symmetric matrix is well-known which has important applications in engineering design, see Overton (1988); Lewis and Overton (1996). Hereafter, we refer to metric learning formulation (3) (equivalently (4) or (5)) as **DML-eig**.

We end this subsection with two remarks. Firstly, Theorem 1 assumes that $X_{\mathcal{S}}$ is invertible. In practice, this can be achieved by enforcing a small ridge to the diagonal of the matrix $X_{\mathcal{S}}$, that is, $X_{\mathcal{S}} \longleftarrow X_{\mathcal{S}} + \delta\mathbf{I}_d$ where $\mathbf{I}_d$ is the identity matrix and $\delta > 0$ is a very small ridge constant. Without loss of generality, we assume that $X_{\mathcal{S}}$ is positive definite throughout the paper. Secondly, when the dimension d of the input space is very large, the computation of $X_{\mathcal{S}}^{-1/2}$ could be time-consuming. Instead of directly inverting the matrix, one can use the Cholesky decomposition which is faster and numerically more stable. Indeed, the Cholesky decomposition tells us that $X_{\mathcal{S}} = LL^\top$ where L is a lower triangular matrix with strictly positive diagonal entries. Hence, in (6) we can let $S = L^\top ML$ (i.e., $M = (L^{-1})^\top SL^{-1}$) and Theorem 1 still holds true if we redefine, for any $\tau = (i,j) \in \mathcal{D}$, that $\widetilde{X}_\tau = (L^{-1}(x_i - x_j))(L^{-1}(x_i - x_j))^\top$. Therefore, it suffices to compute $\{L^{-1}x_i : i \in \mathbb{N}_n\}$ which can efficiently be obtained by solving linear system of equations (e.g., using the operation $L\setminus x_i$ in MATLAB).

2.2 Eigenvalue Optimization for LMNN

Weinberger et al. (2005) proposed the large margin nearest neighbor classification (LMNN) which is one of the state-of-the-art metric learning methods. In analogy to the above argument for DML-eig, we can also formulate LMNN as a generalized eigenvalue optimization problem.

Formulation (3) used the pairwise constraints in the form of similar/dissimilar pairs. In contrast, LMNN aims to learn a metric using the relative distance constraints which are presented in the form of triplets. With a little abuse of notation, we denote a triplet by $\tau = (i,j,k)$ which means that x_i is similar to x_j and x_j is dissimilar to x_k. Then, denote the set of triplets by $\mathcal{T}$ which can be specified based on label information (e.g., see Section 5). Given a set $\mathcal{S}$ of similar pairs and a set $\mathcal{T}$ of triplets, the target of LMNN is to learn a distance metric such that k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. In particular, let $X_{\mathcal{S}} = \sum_{(i,j)\in\mathcal{S}}(x_i - x_j)(x_i - x_j)^\top$ and $C_\tau = X_{jk} - X_{ij}$, then LMNN can be rewritten as

$$\begin{array}{ll} \min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) \\ \text{s.t.} & 1 - \langle C_\tau, M\rangle \le \xi_\tau, \\ & M \in \mathbb{S}^d_+, \xi_\tau \ge 0,\ \forall\tau = (i,j,k)\in\mathcal{T}, \end{array} \tag{7}$$

where $\gamma \in [0,1]$ is a trade-off parameter.

Let T be the number of triplets, that is, the cardinality of the triplet set $\mathcal{T}$. We can establish the following equivalent min-max formulation of LMNN. A similar result with a quite different proof has also been given in Baes and Bürgisser (2009) for a certain class of SDP problems.

Lemma 2. *LMNN formulation (7) is equivalent to*

$$\max_{M\in\mathbb{S}^d_+,\xi\geq 0}\big\{\min_{\tau}(\xi_\tau+\langle C_\tau,M\rangle):\gamma\mathbf{Tr}(X_{\mathcal{S}}M)+(1-\gamma)\sum_{\tau}\xi_\tau=1\big\}. \tag{8}$$

Proof. Write the LMNN formulation (7) as

$$\begin{array}{rl}\min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)\\ \text{s.t.} & \langle C_\tau,M\rangle+\xi_\tau\geq 1,\\ & M\in\mathbb{S}^d_+,\xi_\tau\geq 0,\ \forall\tau=(i,j,k)\in\mathcal{T}.\end{array} \tag{9}$$

The condition that $\langle C_\tau,M\rangle+\xi_\tau\geq 1$ for any $\tau\in\mathcal{T}$ is identical to $\min_{\tau\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau\geq 1$. Hence, problem (7) is further equivalent to

$$\begin{array}{rl}\min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)\\ \text{s.t.} & \min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau\geq 1,\\ & M\in\mathbb{S}^d_+,\xi\geq 0.\end{array} \tag{10}$$

Since the objective function and the constraints are linear w.r.t. variable (M,ξ), the optimal solution for (10) must be attained on the boundary of the feasible domain, that is, $\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau=1$. Consequently, problem (10) is identical to

$$\begin{array}{rl}\min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)\\ \text{s.t.} & \min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau=1,\\ & M\in\mathbb{S}^d_+,\xi\geq 0.\end{array} \tag{11}$$

Let $\Omega=\big\{(M,\xi):\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau\geq 0, M\in\mathbb{S}^d_+,\xi\geq 0\big\}$. We first claim that (11) is equivalent to

$$\min_{M,\xi}\left\{\frac{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)}{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau}:(M,\xi)\in\Omega\right\}. \tag{12}$$

To see this equivalence, let ϕ_1 be the optimal value of problem (11) and ϕ_2 be the optimal value of problem (12). Suppose that (M^*,ξ^*) be an optimal solution of problem (12). Let $\delta^*=\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M^*\rangle+\xi^*_\tau$ and denote $(\widetilde{M}^*,\widetilde{\xi}^*)=(M^*/\delta^*,\xi^*/\delta^*)$. Then, for any $M\in\mathbb{S}^d_+$ and $\xi\geq 0$ satisfying $\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau=1$,

$$\begin{array}{l}(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)=\frac{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)}{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau}\geq\phi_2=\frac{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi^*_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M^*)}{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M^*\rangle+\xi^*_\tau}\\ =(1-\gamma)\sum_{\tau\in\mathcal{T}}\widetilde{\xi}^*_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}\widetilde{M}^*)\geq\phi_1,\end{array}$$

where the last inequality follows from the fact that $\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,\widetilde{M}^*\rangle+\widetilde{\xi}^*_\tau=1$. Since the above inequality holds true for any $M\in\mathbb{S}^d_+$ and $\xi\geq 0$ satisfying $\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau=1$, we finally get that $\phi_1\geq\phi_2\geq\phi_1$, that is, $\phi_1=\phi_2$ and, moreover $(\widetilde{M}^*,\widetilde{\xi}^*)$ is an optimal solution of problem (11). This completes the equivalence between (11) and (12).

Now, rewrite problem (12) as

$$\min_{M,\xi}\left\{\Big(\frac{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,M\rangle+\xi_\tau}{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)}\Big)^{-1}:(M,\xi)\in\Omega\right\},$$

which is further equivalent to

$$\max_{M,\xi}\left\{\frac{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle+\xi_\tau}{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)}:(M,\xi)\in\Omega\right\}. \tag{13}$$

Using exactly the same argument of proving the equivalence between (11) and (12), one can show that the above problem (13) is equivalent to

$$\max_{M,\xi}\Big\{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle+\xi_\tau:(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)=1,(M,\xi)\in\Omega\Big\}. \tag{14}$$

Now consider problem (14) without the restriction $(M,\xi)\in\Omega$, that is,

$$\max_{M,\xi}\Big\{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle+\xi_\tau:(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)=1,M\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}. \tag{15}$$

Let $\tilde{M}=\mathbf{0}$ and $\tilde{\xi}_\tau=\frac{1}{(1-\gamma)T}$ for any τ which obviously satisfies the restriction condition of problem (15), that is, $(1-\gamma)\sum_{\tau\in\mathcal{T}}\tilde{\xi}_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}\tilde{M})=1$. Then,

$$\begin{aligned}&\max\nolimits_{M,\xi}\Big\{\min\nolimits_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle+\xi_\tau:(1-\gamma)\textstyle\sum_{\tau\in\mathcal{T}}\xi_\tau+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)=1,M\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}\\ &\geq\min\nolimits_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau,\tilde{M}\rangle+\tilde{\xi}_\tau=\tfrac{1}{(1-\gamma)T}>0,\end{aligned}$$

which means that any optimal solution for problem (15) automatically satisfies $(M,\xi)\in\Omega$. Consequently, problem (15) is equivalent to (14). Combining this with the equivalence between (11), (12), (13) and (14) finally yields the equivalence between problem (15) and the primal formulation (7) of LMNN. This completes the proof of the lemma. □

Using the above min-max representation for LMNN, it is now easy to reformulate LMNN as a generalized eigenvalue optimization as we will do below. With a little abuse of notation, denote the simplex by $\triangle=\{u\in\mathbb{R}^T:\sum_{\tau\in\mathcal{T}}u_\tau=1,\ u_\tau\geq 0\}$.

Theorem 3. *Assume that $X_{\mathcal{S}}$ is invertible and, for any $\tau\in\mathcal{T}$, let $\widetilde{C}_\tau=X_{\mathcal{S}}^{-1/2}C_\tau X_{\mathcal{S}}^{-1/2}$. Then, LMNN is equivalent to the following problem*

$$\max_{S,\xi}\Big\{\min_{u\in\triangle}\sum_{\tau\in\mathcal{T}}u_\tau\big(\xi_\tau+\langle\widetilde{C}_\tau,S\rangle\big):(1-\gamma)\xi^\top\mathbf{1}+\gamma\mathbf{Tr}(S)=1,S\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}, \tag{16}$$

where $\mathbf{1}$ is a column vector with all entries one. Moreover, it can further be written as a generalized eigenvalue optimization problem:

$$\min_{u\in\triangle}\max\Big(\frac{1}{1-\gamma}u_{max},\frac{1}{\gamma}\lambda_{max}\big(\sum_{\tau\in\mathcal{T}}u_\tau\widetilde{C}_\tau\big)\Big), \tag{17}$$

where u_{max} is the maximum element of the vector $(u_\tau:\tau\in\mathcal{T})$.

Proof. Note that $\min_{\tau=(i,j,k)\in\mathcal{T}}\big(\langle C_\tau,M\rangle+\xi_\tau\big)=\min_{u\in\triangle}u_\tau\big(\langle C_\tau,M\rangle+\xi_\tau\big)$. Combing this with Lemma 2 implies that LMNN is equivalent to

$$\max_{M,\xi}\Big\{\min_{u\in\triangle}\sum_{\tau\in\mathcal{D}}u_\tau\big(\xi_\tau+\langle C_\tau,M\rangle\big):(1-\gamma)\xi^\top\mathbf{1}+\gamma\mathbf{Tr}(X_{\mathcal{S}}M)=1,M\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}.$$

Letting $S = X_S^{1/2} M X_S^{1/2}$ yields the equivalence between (8) and (16).

By the min-max theorem, problem (16) is equivalent to

$$\min_{u\in\triangle}\Big\{\max_{S,\xi}\sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle\widetilde{C}_\tau, S\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(S) = 1, S\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}. \tag{18}$$

To see the equivalence between (18) and (17), observe that

$$\begin{aligned}&\max\Big\{\textstyle\sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle\widetilde{C}_\tau, S\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(S) = 1, S\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}\\ &= \max\Big\{\tfrac{1}{1-\gamma}\sum_{\tau\in\mathcal{D}} u_\tau\xi_\tau + \tfrac{1}{\gamma}\langle\sum_{\tau\in\mathcal{D}} u_\tau\widetilde{C}_\tau, S\rangle : \xi^\top\mathbf{1} + \mathbf{Tr}(S) = 1, S\in\mathbb{S}^d_+,\ \xi\geq 0\Big\}\\ &= \max\Big(\tfrac{1}{1-\gamma}u_{\max}, \tfrac{1}{\gamma}\lambda_{\max}\big(\sum_{\tau\in\mathcal{D}} u_\tau\widetilde{C}_\tau\big)\Big),\end{aligned}$$

where the last equality follows from the fact that the above maximization problem is a linear programming w.r.t. (S,ξ) and, for any $A\in\mathbb{S}^d$, $\max\{\langle A,B\rangle : B\in\mathbb{S}^d_+,\ \mathbf{Tr}(B)\leq 1\} = \lambda_{\max}(A)$. This completes the proof of the theorem. □

Since we have formulated LMNN as an eigenvalue optimization problem in the above theorem, hereafter we refer to formulation (16) (equivalently (17)) as **LMNN-eig**. The above eigenvalue optimization formulation is not restricted to metric learning problems. It can be extended to other machine learning tasks if their SDP formulation is similar to that of LMNN. Maximum-margin matrix factorization (Srebro et al., 2004) is one of such examples. Its eigenvalue optimization formulation can be found in Appendix A.

3. Eigenvalue Optimization Algorithms

In this section we develop efficient algorithms for solving DML-eig and LMNN-eig. We can directly employ the entropy smoothing techniques (Nesterov, 2007; Baes and Bürgisser, 2009) for eigenvalue optimization which, however, needs the computation of the full eigen-decomposition per iteration. Instead, we propose a new first-order method by developing and combining the smoothing techniques (Nesterov, 2005) and Frank-Wolfe algorithm (Frank and Wolfe, 1956; Hazan, 2008), which will only involve the computation of the largest eigenvector of a matrix.

3.1 Approximate Frank-Wolfe Algorithm for DML-eig

By Theorem 1, DML-eig is identical to problem:

$$\max_{S\in\mathcal{P}} f(S) = \max_{S\in\mathcal{P}}\min_{u\in\triangle}\sum_{\tau\in\mathcal{D}} u_\tau\langle\widetilde{X}_\tau, S\rangle. \tag{19}$$

To this end, for a smoothing parameter $\mu>0$, define

$$f_\mu(S) = \min_{u\in\triangle}\sum_{\tau\in\mathcal{D}} u_\tau\langle\widetilde{X}_\tau, S\rangle + \mu\sum_{\tau\in\mathcal{D}} u_\tau\ln u_\tau.$$

We use the smoothed problem $\max_{S\in\mathcal{P}} f_\mu(S)$ to approximate problem (19).

It is easy to see that

$$f_\mu(S) = -\mu\ln\big(\sum_{\tau\in\mathcal{D}} e^{-\langle\widetilde{X}_\tau, S\rangle/\mu}\big),$$

```
Input:
  · smoothing parameter $\mu > 0$ (e.g., $10^{-5}$)
  · tolerance value $tol$ (e.g., $10^{-5}$)
  · step sizes $\{\alpha_t \in (0,1) : t \in \mathbb{N}\}$
Initialization: $S_1^\mu \in \mathbb{S}_+^d$ with $\mathbf{Tr}(S_1^\mu) = 1$
for $t = 1, 2, 3, \ldots$ do
  · $Z_t^\mu = \arg\max\big\{f_\mu(S_t) + \langle Z, \nabla f_\mu(S_t^\mu)\rangle : Z \in \mathbb{S}_+^d,\ \mathbf{Tr}(Z) = 1\big\}$, that is, $Z_t^\mu = vv^\top$
        where $v$ is the maximal eigenvector of matrix $\nabla f_\mu(S_t^\mu)$
  · $S_{t+1}^\mu = (1-\alpha_t)S_t^\mu + \alpha_t Z_t^\mu$
  · if $|f_\mu(S_{t+1}^\mu) - f_\mu(S_t^\mu)| < tol$ then break
Output: $d \times d$ matrix $S_t^\mu \in \mathbb{S}_+^d$
```

Table 1: Approximate Frank-Wolfe Algorithm for DML-eig

and

$$\nabla f_\mu(S) = \frac{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\widetilde{X}_\tau}{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}}.$$

Since f_μ is a smooth function, we can prove that its gradient is Lipschitz continuous.

Lemma 4. *For any $S_1, S_2 \in \mathcal{P}$, then*

$$\|\nabla f_\mu(S_1) - \nabla f_\mu(S_2)\| \le C_\mu \|S_1 - S_2\|,$$

where $C_\mu = 2\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu$.

Proof. It suffices to see $\|\nabla^2 f_\mu(S)\| \le 2\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu$. To this end,

$$\nabla^2 f_\mu(S) = \frac{(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\widetilde{X}_\tau) \bigotimes (\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\widetilde{X}_\tau)}{\mu\left(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\right)^2} - \frac{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\widetilde{X}_\tau \bigotimes \widetilde{X}_\tau}{\mu\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}} := I + II,$$

where $X \bigotimes S$ denotes the tensor product of matrices X and S. We can estimate the term I as follows:

$$\|I\| \le \frac{\left(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\|\widetilde{X}_\tau\|\right)\left(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\|\widetilde{X}_\tau\|\right)}{\mu\left(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\right)^2} \le \frac{1}{\mu}\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2,$$

where, in the above inequality, we used the fact that $\|S \bigotimes X\| \le \|X\|\|S\|$ for any $X, S \in \mathbb{S}^d$. The second term II can be similarly estimated:

$$\|II\| \le \max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu.$$

Putting them together yields the desired result. □

The pseudo-code to solve DML-eig is described in Table 1 which is a generalization of Frank-wolfe algorithm (Frank and Wolfe, 1956) which originally applies to the context of minimizing a convex function over a feasible polytope. Hazan (2008) first extended the original Frank-Wolfe algorithm to solve SDP over the spectrahedron $\mathcal{P} = \{M : M \in \mathbb{S}_+^d, \mathbf{Tr}(M) = 1\}$. Recall that D is the cardinality of $\mathcal{D}$, that is, the number of dissimilar pairs. Then, we have the following convergence result.

Lemma 5. *For any $0 < \mu \leq 1$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by the algorithm in Table 1 and C_μ be defined in Lemma 4. Then we have that*

$$\max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_{t+1}^\mu) \leq C_\mu \alpha_t^2 + (1-\alpha_t)\big(\max_{S \in \mathcal{P}} f_\mu(S) - f(S_t^\mu)\big).$$

Proof. By the definition of C_μ in Lemma 4, we have

$$f_\mu(S_{t+1}^\mu) \geq f_\mu(S_t^\mu) + \alpha_t \langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu \rangle - C_\mu \alpha_t^2. \tag{20}$$

Since f is concave, for any $S \in \mathcal{P}$ there holds

$$\langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu \rangle \geq \langle \nabla f_\mu(S_t^\mu), S - S_t^\mu \rangle \geq f_\mu(S) - f(S_t^\mu),$$

which implies that

$$\langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu \rangle \geq \max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu).$$

Substituting the above inequality into (20) yields the desired result. □

For simplicity, let $R_t = \max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu)$. If $\alpha_t \in (0,1]$ for any $t \geq t_0$ with some $t_0 \in \mathbb{N}$, then by Lemma 5 and a simple induction, for any $t \geq t_0$ there holds

$$R_{t+1} \leq C_\mu \sum_{j=t_0}^{t} \prod_{k=j+1}^{t} (1-\alpha_k)\alpha_j^2 + \prod_{j=t_0}^{t} (1-\alpha_j) R_{t_0}. \tag{21}$$

Combining this inequality and some ideas in Ying and Zhou (2006), one can establish sufficient conditions on the stepsizes $\{\alpha_t : t \in \mathbb{N}\}$ such that $\lim_{t \to \infty} f_\mu(S_t^\mu) = \min_{S \in \mathcal{P}} f_\mu(S)$.

Theorem 6. *For any fixed $\mu > 0$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by the algorithm in Table 1. If the step sizes satisfy that*

$$\sum_{t \in \mathbb{N}} \alpha_t = \infty, \qquad \lim_{t \to \infty} \alpha_t = 0, \tag{22}$$

then

$$\lim_{t \to \infty} f_\mu(S_t^\mu) = \max_{S \in \mathcal{P}} f_\mu(S).$$

The detailed proof of the above theorem is given in Appendix B. Typical examples of step sizes satisfying condition (22) are $\{\alpha_t = t^{-\theta} : t \in \mathbb{N}\}$ with $0 < \theta \leq 1$. For the particular case $\theta = 1$, by Lemma 5 we can prove the following result.

Theorem 7. *For any $0 < \mu \leq 1$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by Table 1 with step sizes given by $\{\alpha_t = 2/(t+1) : t \in \mathbb{N}\}$. Then, for any $t \in \mathbb{N}$ we have that*

$$\max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu) \leq \frac{8 \max_{\tau \in \mathcal{D}} \|\widetilde{X}_\tau\|^2}{\mu t} + \frac{4 \ln D}{t}. \tag{23}$$

Furthermore,

$$\max_{S \in \mathcal{P}} f(S) - f(S_t^\mu) \leq 2\mu \ln D + \frac{8 \max_{\tau \in \mathcal{D}} \|\widetilde{X}_\tau\|^2}{\mu t} + \frac{8 \ln D}{t}.$$

Proof. It is easy to see, for any $S \in \mathcal{P}$ that

$$|f(S) - f_\mu(S)| \leq \mu \max_{u \in \triangle} \sum_{\tau \in \mathcal{D}} (-u_\tau \ln u_\tau) \leq \mu \ln D.$$

Let $S_* = \arg\max_{S \in \mathcal{P}} f(S)$ and $S^\mu_* = \arg\max_{S \in \mathcal{P}} f_\mu(S)$. Then, for any $t \in \mathbb{N}$,

$$\begin{array}{ll} \max_{S \in \mathcal{P}} f(S) - f(S^\mu_t) & = [f(S_*) - f_\mu(S_*)] + [f_\mu(S_*) - \max_{S \in \mathcal{P}} f_\mu(S)] \\ & +[f_\mu(S^\mu_*) - f_\mu(S^\mu_t)] + [f_\mu(S^\mu_t) - f(S^\mu_t)] \\ & \leq [f(S_*) - f_\mu(S_*)] + [f_\mu(S^\mu_*) - f_\mu(S^\mu_t)] + [f_\mu(S^\mu_t) - f(S^\mu_t)] \\ & \leq 2\mu \ln D + [f_\mu(S^\mu_*) - f_\mu(S^\mu_t)] \\ & = 2\mu \ln D + [\max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S^\mu_t)]. \end{array}$$

Hence, it suffices to prove (23) by induction. Indeed, for $t = 1$, we have that

$$\begin{array}{ll} \max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S^\mu_1) & \leq f_\mu(S^\mu_*) + \mu \sup_{u \in \triangle} (\sum_{\tau \in \mathcal{D}} (-u_\tau \ln u_\tau)) \\ & \leq \max_{S \in \mathcal{P}} \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S \rangle + \mu \ln D \\ & \leq \max_{S \in \mathcal{P}} \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \|\widetilde{X}_\tau\| \|S\| + \mu \ln D \\ & \leq \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \|\widetilde{X}_\tau\| + \mu \ln D \\ & \leq \min_{u \in \triangle} \big[\sum_{\tau \in \mathcal{D}} u_\tau + \sum_{\tau \in \mathcal{D}} u_\tau \|\widetilde{X}_\tau\|^2 \big] + \mu \ln D \\ & \leq 1 + \max_{\tau \in \mathcal{D}} \|\widetilde{X}_\tau\|^2 + \mu \ln D, \end{array}$$

which obviously satisfies (23) with $t = 1$. Suppose the inequality (23) holds true for some $t > 1$. Now by Lemma 5,

$$\begin{array}{ll} R_{t+1} & \leq C_\mu \alpha_t^2 + (1 - \alpha_t) R_t \\ & \leq \frac{4C_\mu}{(t+1)^2} + \frac{t-1}{t+1} \big(\frac{4C_\mu}{t} + \frac{4 \ln D}{t} \big) \\ & \leq 4(C_\mu + \ln D) \big(\frac{1}{(t+1)^2} + \frac{t-1}{(t+1)t} \big) \leq \frac{4(C_\mu + \ln D)}{t+1}, \end{array}$$

where the second inequality follows from the induction assumption. This proves the inequality (23) for all $t \in \mathbb{N}$ which completes the proof of the theorem. □

By the above theorem, for any $\varepsilon > 0$, then $\mu = \frac{\varepsilon}{4 \ln D}$ and the iteration number $t \geq 64(1 + \max_{\tau \in \mathcal{D}} \|\widetilde{X}_\tau\|^2) \ln D / \varepsilon^2$ yields that $\max_{S \in \mathcal{P}} f(S) - f(S^\mu_t) \leq \varepsilon$. The time complexity of the approximate first-order method for DML-eig is of $O\big(d^2/\varepsilon^2\big)$.

3.2 Approximate Frank-Wolfe Algorithm for LMNN-eig

We can easily extend the above approximate Frank-Wolfe algorithm to solve the eigenvalue optimization formulation of LMNN-eig (formulation (16) or (17)). To this end, let

$$f(S, \xi) = \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \big(\xi_\tau + \langle \widetilde{C}_\tau, S \rangle \big).$$

Then, problem (16) is identical to

$$\max \big\{ f(S, \xi) : (1 - \gamma) \sum_\tau \xi_\tau + \gamma \mathbf{Tr}(S) = 1, S \in \mathbb{S}^d_+,\ \xi \geq 0 \big\}.$$

```
Input:
    · smoothing parameter $\mu > 0$ (e.g., $10^{-5}$)
    · tolerance value $tol$ (e.g., $10^{-5}$)
    · step sizes $\{\alpha_t \in (0,1) : t \in \mathbb{N}\}$
Initialization: $S_1^\mu \in \mathbb{S}_+^d$ with $\mathbf{Tr}(S_1^\mu) = 1$ and $\xi_1^\mu \geq 0$
for $t = 1,2,3,\ldots$ do
    · $(Z_t^\mu, \beta_t^\mu) = \arg\max\big\{\langle Z, \partial_S f_\mu(S_t^\mu, \xi_t^\mu)\rangle + \xi^\top \partial_\xi f_\mu(S_t^\mu, \xi_t^\mu) : Z \in \mathbb{S}_+^d,\ \xi \geq 0$
                $(1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1\big\}$
    · $(S_{t+1}^\mu, \xi_{t+1}^\mu) = (1-\alpha_t)(S_t^\mu, \xi_t^\mu) + \alpha_t (Z_t^\mu, \beta_t^\mu)$
    · if $|f_\mu(S_{t+1}^\mu, \xi_{t+1}^\mu) - f_\mu(S_t^\mu, \xi_t^\mu)| < tol$ then break
Output: $d \times d$ matrix $S_t^\mu \in \mathbb{S}_+^d$ and slack variables $\xi_t^\mu$
```

Table 2: Approximate Frank-Wolfe Algorithm for LMNN-eig

In analogy to the smooth techniques applied to DML-eig, we approximate $f(S,\xi)$ by the following smooth function:

$$f_\mu(S,\xi) = \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau(\xi_\tau + \langle \widetilde{C}_\tau, S\rangle) + \mu \sum_{\tau\in\mathcal{D}} u_\tau \ln u_\tau.$$

One can easily see that

$$f_\mu(S,\xi) = -\mu \ln\big(\sum_{\tau\in\mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}\big).$$

and its gradient function is given by

$$\nabla_S f_\mu(S,\xi) = \frac{\sum_{\tau\in\mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu} \widetilde{C}_\tau}{\sum_{\tau\in\mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}},$$

and

$$\frac{\partial f_\mu(S,\xi)}{\partial \xi_\tau} = \frac{e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}}{\sum_{\tau\in\mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}}.$$

The approximate Frank-Wolfe algorithm for LMNN-eig is exactly the same as DML-eig in Table 1. The pseudo-code is listed in Table 2. The key step of the algorithm is to compute the following problem:

$$\begin{aligned}(Z_t^\mu, \beta_t^\mu) \quad &= \arg\max\big\{\langle Z, \partial_S f_\mu(S_t^\mu, \xi_t^\mu)\rangle + \xi^\top \partial_\xi f_\mu(S_t^\mu, \xi_t^\mu) : Z \in \mathbb{S}_+^d,\ \xi \geq 0 \\ &\qquad (1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1\big\}.\end{aligned}$$

Equivalently, one needs to solve, for any $A \in \mathbb{S}^d$ and $\beta \in \mathbb{R}^T$, the following problem:

$$(Z^*, \xi^*) \quad = \arg\max\big\{\langle Z, A\rangle + \xi^\top \beta : Z \in \mathbb{S}_+^d,\ \xi \geq 0, (1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1\big\}. \tag{24}$$

Let $\beta_{\max} = \beta_{\tau^*}$ with $\tau^* \in \mathcal{T}$ and v^* is the largest eigenvector of A. Then, problem (24) is a linear programming and its optimal value is either

$$\max\big\{\xi^\top \beta : (1-\gamma)\xi^\top \mathbf{1} = 1, \xi \geq 0\big\} = \frac{\beta\max}{1-\gamma},$$

or

$$\max\left\{\langle Z,A\rangle : \gamma\mathbf{Tr}(Z)=1, Z\in\mathbb{S}^d_+\right\}=\frac{\lambda_{\max}(A)}{\gamma}.$$

The optimal solution of problem (24) is given as follows. If $\frac{\lambda_{\max}(A)}{\gamma} >= \frac{\beta_{\max}}{1-\gamma}$, then $Z^* = \frac{v^*(v^*)^\top}{\gamma}$ where v^* is the largest eigenvector of matrix A and $\xi^* = \mathbf{0}$. Otherwise, $Z^* = \mathbf{0}$ and the τ^*-th element of ξ^* equals $\frac{1}{1-\gamma}$, that is, $(\xi^*)_{\tau^*} = \frac{1}{1-\gamma}$ and the other entries of ξ^* all zeros. In analogy to the arguments for Theorem 7, for step sizes $\{\alpha_t = \frac{2}{t+1} : t\in\mathbb{N}\}$ one can exactly prove the time complexity of LMNN-eig is $O(d^2/\varepsilon^2)$.

4. Related Work and Discussion

There is a large amount of work on metric learning including distance metric learning for k-means clustering (Xing et al., 2002), relevant component analysis (RCA) (Bar-Hillel et al., 2005), maximally collapsing metric learning (MCML) (Goldberger et al., 2004), neighborhood component analysis (NCA) (Goldberger et al., 2004) and an information-theoretic approach to metric learning (ITML) (Davis et al., 2007) etc. We refer the readers to Yang and Jin (2007) for a nice survey on metric learning. Below we discuss some specific metric learning models which are closely related to our work.

Xing et al. (2002) developed the metric learning model (2) to learn a Mahalanobis metric for k-means clustering. The main idea is to maximize the distance between points in the dissimilarity set under the constraint that the distance between points in the similarity set is upper-bounded. A projection gradient method is employed to obtain the optimal solution. Specifically, at each iteration the algorithm takes a gradient ascent step of the objective function and then projects it back to the set of constraints and the cone of the p.s.d. matrices. The projection to the p.s.d. cone needs the computation of the full eigen-decomposition with time complexity $O(d^3)$. The projection gradient method usually takes a large number of iterations to become convergent. It is worth mentioning that the metric learning model proposed in Xing et al. (2002) is a global method in the sense that the model aggregates all similarity constraints together as well as all dissimilarity constraints. In contrast to Xing et al. (2002), DML-eig aims to maximize the minimal distance between dissimilar pairs instead of maximizing the summation of their distances. Consequently, DML-eig would intuitively force the dissimilar samples to be far more separated from similar samples. This intuition may account for the superior performance of DML-eig which will be shown soon in the experimental section.

Weinberger et al. (2005) developed a large margin framework to learn a Mahalanobis distance metric for k-nearest neighbor (k-NN) classification (LMNN). The main intuition behind LMNN is that k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. In contrast to the global method (Xing et al., 2002), LMNN is a local method in the sense that only triplets from the k-nearest neighbors are used. Our method DML-eig is a local method which only uses the similar pairs and dissimilar pairs from k-nearest neighbors.

Since every $M\in\mathbb{S}^d_+$ can be factored as $M = AA^\top$ for some $A\in\mathbb{R}^{d\times d}$, LMNN becomes an unconstrained optimization problem with an unconstrained variable A. Weinberger et al. (2005) used this idea and proposed to use the sub-gradient method to obtain the optimal solution. Since the modified problem w.r.t. variable A is generally not convex, the sub-gradient method would lead to local minimizers. For some special SDP problems, it was shown in Burer and Monteiro (2003) that such dilemma will not happen. Specifically, Burer and Monteiro (2003) considered the following SDPs:

$$\min\Big\{\mathbf{Tr}(CM):\mathbf{Tr}(A_iM)=b_i, i=1,\ldots,m, M\in\mathbb{S}^d_+\Big\}. \tag{25}$$

It was proved that if A^* is a local minimum of the modified problem:

$$\min\Big\{\mathbf{Tr}(CAA^\top):\mathbf{Tr}(A_iAA^\top)=b_i, i=1,\ldots,m, A\in\mathbb{R}^{d\times d}\Big\},$$

then $M^*=A^*(A^*)^\top$ is a global minimum of the primal problem (25). However, since the hinge loss is not smooth, it is unclear how their proof can be adapted to the case of LMNN.

Rosales and Fung (2006) proposed the following element-sparse metric learning for high-dimensional data sets

$$\min_{M\in\mathbb{S}^d_+}\sum_{t=(i,j,k)\in\mathcal{T}}(1+x_{ij}^\top Mx_{ij}-x_{kj}^\top Mx_{kj})_+ +\gamma\sum_{\ell,k\in\mathbb{N}_d}|M_{\ell k}|. \tag{26}$$

In order to solve the optimization problem, they further proposed to restrict M to the space of *diagonal dominance* matrices which reduces formulation (26) to a linear programming problem. Such a restriction would only result in a sub-optimal solution.

Shalev-Shwartz et al. (2004) developed an appealing online learning model for learning a Mahalanobis distance metric. In each time, given a pair of examples the p.s.d. distance matrix is updated by a rank-one matrix which only needs the time complexity $O(d^2)$. However, since the pairs of similarly labeled and differently labeled examples are usually of order $O(n^2)$, the online learning procedure takes many rank-one matrix updates. Jin et al. (2009) established generalization bounds for large margin metric learning and proposed an adaptive way to adjust the step sizes of the online metric learning method in order to guarantee the output matrix in each step is positive semi-definite. Since the pairs of similarity and dissimilarity are usually of order $O(n^2)$ where n is the sample number, the online learning procedure generally needs many matrix updates.

Shen et al. (2009) recently employed the exponential loss for metric learning which can be written by

$$\min_{M\in\mathbb{S}^d_+}\sum_{\tau=(i,j,k)\in\mathcal{T}}e^{\langle C_\tau,M\rangle}+\mathbf{Tr}(M),$$

where $\mathcal{T}$ is the triplet set and $C_\tau=(x_i-x_j)(x_i-x_j)^\top-(x_j-x_k)(x_j-x_k)^\top$ for any $\tau=(i,j,k)\in\mathcal{T}$. A boosting-based algorithm called BoostMetric was developed which is based on the idea that each p.s.d. matrix can be decomposed into a linear positive combination of trace-one and rank-one matrices. The algorithm is essentially a column-generation scheme which iteratively finds the linear combination coefficients of the current basis set of rank-one matrices and then update the basis set of trace-one and rank-one matrices. The updating of rank-one and trace-one matrix only involves the computation of the largest eigenvector which is of time complexity $O(d^2)$. However, the number of linear combination for the p.s.d. matrix can be infinite and the convergence rate of this column-generation algorithm is not clear.

Recently, Guillaumin et al. (2009) proposed a metric learning model with logistic regression loss which is referred to as LDML. Promising results were reported in its application to face verification problems. LDML employed the gradient descent algorithm to obtain the optimal solution. However, in order to reduce the computational time, the algorithm ignored the positive semi-definiteness of the distance matrix which would only lead to a suboptimal solution.

Data	No.	n	d	♯class	♯ $\mathcal{T}$	♯$\mathcal{D}$
Wine	1	178	13	3	1134	378
Iris	2	150	4	3	954	315
Breast	3	569	30	2	3591	1197
Diabetes	4	768	8	2	4842	1614
Waveform	5	5000	21	3	3150	1050
Segment	6	2310	19	7	14553	4851
Optdigits	7	2680	64	10	24120	8040
Face	8	400	2576	40	2520	840
USPS	9	9298	256	10	58626	19542

Table 3: Description of data sets n is the number of samples and d is the dimensionality. For AT&T face data set, we use PCA to reduce its dimension to 64.

5. Experiments

In this section we compare our proposed method **DML-eig** and **LMNN-eig** with a few methods: the method proposed in Xing et al. (2002) denoted by **Xing**, **LMNN** (Weinberger et al., 2005) and its accelerated version **mLMNN** (Weinberger and Saul, 2008), **ITML** (Davis et al., 2007), **BoostMetric** (Shen et al., 2009) and the baseline algorithm that uses the standard Euclidean distance denoted by **Euc.** For all the data sets we have set $k = 3$ for nearest neighbor classification. The trade-off parameters in ITML, LMNN and LMNN-eig are tuned via three-fold cross validation. The smoothing parameter for DML-eig and LMNN-eig is set to be $\mu = 10^{-4}$ and the maximum iteration for DML-eig, BoostMetric, LMNN-eig is set to be 10^3.

We first run experiments on 9 data sets, that is, 1) wine, 2) iris, 3) Breast-Cancer, 4) the Indian Pima Diabetes, 5) Waveform, 6) Segment, 7) Optdigits, 8) AT&T Face data set [2] and 9) USPS. The statistics of data sets summarized in Table 3. All experimental results are obtained by averaging over 10 runs (except 1 run for USPS due to its large size). For each run, we randomly split the data sets 70% for training and 30% for test validation. We have used the same mechanism in Weinberger et al. (2005) to generate training triplets. Briefly speaking, for each training point x_i, k nearest neighbors that have same labels as y_i (targets) as well as k nearest neighbors that have different labels from y_i (imposers) are found. From x_i and its corresponding targets and imposers, we then construct the set of similar pairs $\mathcal{S}$ (same labels) and the set of dissimilar pairs $\mathcal{D}$ (distinct labels), and the set of triplets $\mathcal{T}$. As mentioned above, the original formulation in Xing et al. (2002) used all pairwise constraints. We emphasize here, for fairness of comparison (especially the running time comparison), that all methods including the Xing's method used the same set of similar/dissimilar pairs generated locally as above.

Finally we will apply the developed models and algorithms on a large and challenging face verification data set called *Labeled Faces in the Wild* (LFW).[3] It contains 13233 labeled faces of 5749 people, for 1680 people there are two or more faces. Furthermore, the data is challenging and difficult due to face variations in scale, pose, lighting, background, expression, hairstyle, and glasses, as the faces are detected in images in the wild, taken from Yahoo! News.

[2] Data sets can be found at `http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html`.

[3] Data set can be found at `http://vis-www.cs.umass.edu/lfw/index.html`.

	Euc.	Xing	LMNN	ITML	BoostMetric	DML-eig	LMNN-eig
1	3.46(3.60)	4.04(4.00)	3.08(2.07)	**1.15(2.07)**	2.31(2.18)	1.35(1.30)	2.88(1.87)
2	5.11(2.58)	6.67(3.11)	4.22(1.95)	4.44(2.57)	3.56(2.52)	**3.11(1.15)**	4.00(2.30)
3	6.47(1.33)	8.18 (1.58)	5.35(1.43)	6.82(1.57)	3.82(1.55)	**3.53(0.88)**	4.94(1.28)
4	31.09(2.03)	32.09 (3.56)	29.70(3.20)	29.96(2.97)	**26.78(2.12)**	27.71(3.93)	31.13(2.24)
5	18.87(0.65)	16.43(1.00)	18.61(0.72)	15.94(0.83)	16.86(0.90)	**15.33(0.80)**	18.49(0.21)
6	5.61(0.92)	5.26(0.60)	3.69(0.70)	5.02(0.70)	4.21(0.48)	**2.97(0.55)**	3.61(0.83)
7	1.67(0.24)	1.57(0.28)	**1.37(0.25)**	1.46(0.29)	1.38(0.33)	1.45(0.22)	1.43(0.42)
8	6.67(1.67)	7.75(0.69)	2.08(1.53)	2.42(2.17)	2.25(1.25)	**1.67(1.24)**	**1.67(1.76)**
9	3.05	-	**2.98**	3.92	3.34	3.66	3.13

Table 4: Average test error (%) of different metric learning methods (standard deviation are in parentheses). The best performance is denoted in bold type. The notation "–" means that the method does not converge in a reasonable time.

data	Xing	LMNN/mLMNN	ITML	BoostMetric	LMNN-eig	DML-eig
1	1.00	0.87/1.01	4.63	0.49	0.30	0.23
2	2.41	0.57/0.62	3.56	0.10	0.92	0.43
3	3.08	2.71/0.75	4.54	2.04	3.71	3.18
4	2.45	1.73/1.03	3.95	0.20	6.78	0.03
5	231.33	8.83/5.54	7.83	11.36	36.95	1.45
6	109.13	1.73/4.25	61.55	9.06	5.06	1.76
7	59.24	24.81/15.92	37.42	93.73	86.38	2.67
8	182.56	5.54/1.50	40.38	60.31	18.42	2.58
9	–	723.49/454.21	726.88	694.84	572.04	52.48

Table 5: Average running time (seconds) of different methods. The notation "–" means that the method does not converge in a reasonable time.

5.1 Generalization and Running Time

As we can see from Table 4, DML-eig consistently improves k-NN classification using Euclidean distance on most data sets. Hence, learning a Mahalanobis metric from training data does lead to improvements in k-NN classification. Also, we can see that DML-eig is competitive with the state-of-the-art methods: LMNN, ITML and BoostMetric. Indeed, DML-eig outperforms other algorithms on 5 out of 9 data sets. As expected, LMNN-eig performs similarly or slightly better than LMNN since these two models are essentially the same. In Table 5, we list the average CPU time of different algorithms. We can see that the method proposed in Xing et al. (2002) generally needs more time since it needs the full eigen-decomposition of a matrix per iteration. DML-eig, BoostMetric and LMNN are among the fastest algorithms while LMNN-eig is slower than LMNN and mLMNN in most cases. The accelerated version mLMNN is faster than LMNN.

On the left-hand side of Figure 1, we plot the running time versus the reduced dimension by principal component analysis (PCA) for AT&T data set. We can observe that LMNN, BoostMetric, LMNN-eig and DML-eig are faster than ITML and Xing's method. When the dimension is low, LMNN, BoostMetric, LMNN-eig and DML-eig are similar. As the dimension increases, DML-eig and mLMNN are faster. On this data set, LMNN-eig runs slower than mLMNN. The reason could be that mLMNN used the techniques of ball trees and employed only an active set of triplets per iteration. Our algorithms have not been combined with the techniques of ball trees and are implemented in MATLAB and better improvements are expected if used in C/C++. On the right-hand side of Figure 1, we also plot the test errors of various methods across different PCA dimensions. Almost every method performs better than the baseline method using the standard Euclidean distance metric. DML-eig performs slightly better than other methods. We observe that, with increasing PCA dimensions, DML-eig, BoostMetric and ITML yield relatively stable performance across different PCA dimensions. In contrast, the performance of other baseline methods such as LMNN and Xing's method varied as the PCA dimensions changed.

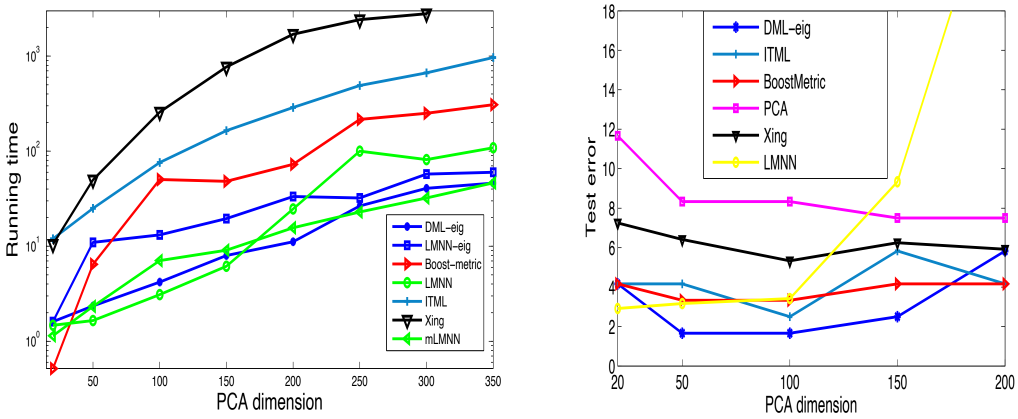


Figure 1: Performance on AT&T Face data set. Left figure: running time (seconds) versus PCA dimension. Right figure: test error (%) versus PCA dimension; the pink line is the performance of k-NN classifier ($k = 3$) using the standard Euclidean distance.

5.2 Application to Face Verification

In this experiment we investigate our proposed method (DML-eig) for face verification. The task of face verification is to determine whether two face images are from the same identity or not. It is a highly active area of research and finds application in access control, image search, security and many other areas. The large variation in lighting, pose, expression etc. of the face images poses great challenges to the face verification algorithms. Inference that is based on the raw pixels of the image data or features extracted from the images is usually unreliable as the data show large variation and are high-dimensional.

Metric learning provides a viable solution by comparing the image pairs based on the metric learnt from the face data. Here we evaluate our new metric learning method using a large scale face database—Labeled Faces in the Wild (LFW) (Huang et al., 2007). There are a total of 13233 images and 5749 people in the database. These face images are automatically captured from news articles on the web. Recently it has become a benchmark to test new face verification algorithms (Wolf et al., 2008; Guillaumin et al., 2009; Wolf et al., 2009; Taigman et al., 2009; Pinto et al., 2011).

The images we used are in gray scale and aligned in two ways. One is "funneled" (Huang et al., 2007) and the other is "aligned" using a commercial face alignment software by Taigman et al. (2009). These images are divided into ten folds where the subject identities are mutually exclusive. In each fold, there are 300 pairs of images from the same identity and another 300 pairs of images from different identities. We followed the standard procedure for training and test in the technical report of Huang et al. (2007). The performance of the algorithms is evaluated by average (and standard error of) correct verification rate and the ROC curve of the 10-fold cross validation test.

We investigated several descriptors (features) from face images in this experiment. As for the "funneled" images, we used SIFT descriptors computed at the fixed facial key-points (e.g., corners of eyes and nose). These data are available from Guillaumin et al. (2009). We focus on the SIFT descriptor to evaluate our algorithm as it provides a fair comparison to Guillaumin et al. (2009). To compare with the state-of-the-art methods in face verification, we further investigated three types of features for the "aligned" images: 1) raw pixel data by concatenating the intensity value of each pixel in the image; 2) Local Binary Patterns (LBP) (Ojala et al., 2002); and 3) LBP's variation, three-Patch Local Binary Patterns (TPLBP) (Wolf et al., 2008). The original dimensionality of the features is quite high ($3456 \sim 12000$) so we reduced the dimension using PCA. These descriptors were tested with both their original value and the square root of them (Wolf et al., 2008, 2009; Guillaumin et al., 2009).

There are two configuration for forming the training sets. One is "restricted configuration": only same/not-same labels are used during training and no information about the actual names of the people (class labels) in the image pairs should be used. In the past, most of the published work on this data set using the restricted protocol (e.g., Guillaumin et al., 2009; Wolf et al., 2009; Pinto et al., 2011). Another is "unrestricted configuration": all available information including the names of the people in the images can be used for training. So far there are only two published results on the unrestricted configuration (Guillaumin et al., 2009; Taigman et al., 2009). Here we mainly focus on the restricted configuration.

LMNN and BoostMetric are not applicable in this restricted configuration setting since they need label information to generate the triplet set. Therefore, we only compared our DML-eig method with LDML (Guillaumin et al., 2009) and ITML (Davis et al., 2007). For each of the ten-fold cross-validation test, we use the data from 2700 pairs of images from the same identities and another 2700 pairs of images from the different identities to learn a metric. Then test it using the other 600 image pairs. The performance is evaluated using accurate verification rate .

Table 6 illustrates the performances of our algorithm and ITML and LDML. The best verification rate of DML-eig is 81.27%. It outperforms LDML (77.50%) and ITML (76.20%) in their best settings. Note that the performance of DML-eig is consistently better than LDML and ITML in each PCA dimension.

By varying the dimension of principal components of the SIFT descriptor, the performance of DML-eig of the 10-fold cross validation test is plotted in Figure 2. The best performance is achieved when the dimension of principal components is 100. So we fix this dimension for SIFT feature in the following experiment. As mentioned in Guillaumin et al. (2009), the peak performance in a specific PCA dimension is due to the limit of training samples. The PCA dimension achieving the best performance is 35 for LDML and 55 for ITML. This number for DML-eig is 100 which is larger than that of both LDML and ITML. It shows that the DML-eig metric is less prone to overfitting than both LDML and ITML.

Method	PCA Dim.	Original	Square Root
ITML	35	0.7537 ± 0.0158	0.7627 ± 0.0161
LDML	35	0.7660 ± 0.0070	0.7750 ± 0.0050
DML-eig	35	0.7742 ± 0.0213	0.7793 ± 0.0214
ITML	40	0.7618 ± 0.0125	0.7643 ± 0.0121
LDML	40	–	–
DML-eig	40	0.7752 ± 0.0198	0.7838 ± 0.0195
ITML	55	0.7530 ± 0.0185	0.7557 ± 0.0187
LDML	55	0.7280 ± 0.0060	0.7280 ± 0.0040
DML-eig	55	0.7900 ± 0.0189	0.7938 ± 0.0163
ITML	100	0.7340 ± 0.0250	0.7403 ± 0.0216
LDML	100	–	–
DML-eig	100	**0.8055 ± 0.0171**	**0.8127 ± 0.0230**

Table 6: Performance comparison on LFW database in the restricted configuration (mean verification accuracy and standard error of the mean of 10-fold cross validation test) with only SIFT descriptors. “Square Root” means the features preprocessed by taking square root before fed into metric learning method. The result of LDML is cited from Guillaumin et al. (2009) where it was reported that the best result of LDML is achieved with PCA dimension 35. Our result of ITML is very similar to that reported in Guillaumin et al. (2009).

Method	Accuracy
High-Throughput Brain-Inspired Features, aligned (Pinto et al., 2011)	0.8813 ± 0.0058
LDML + Combined, funneled (Guillaumin et al., 2009)	0.7927 ± 0.0060
DML-eig + Combining four descriptors (this work)	0.8565 ± 0.0056

Table 7: Performance comparison of DML-eig and other state-of-the-art methods in the restricted configuration (mean verification rate and standard error of the mean of 10-fold cross validation test) based on combination of different types of descriptors. The descriptors vary in different study. The best result up to date is achieved using sophisticated large scale feature search (Pinto et al., 2011).

Besides the SIFT descriptor, we also investigated to combine it with other three types of descriptors aforementioned. Following Wolf et al. (2008); Guillaumin et al. (2009), we combine the distance scores from 4 different descriptors using a linear Support Vector Machine (SVM). The performance of DML-eig is compared to the other state-of-the-art methods in Table 7 and Figure 3. Note that each of these published results use its own learning technique and different feature extraction approaches which makes the conclusion hard to draw.

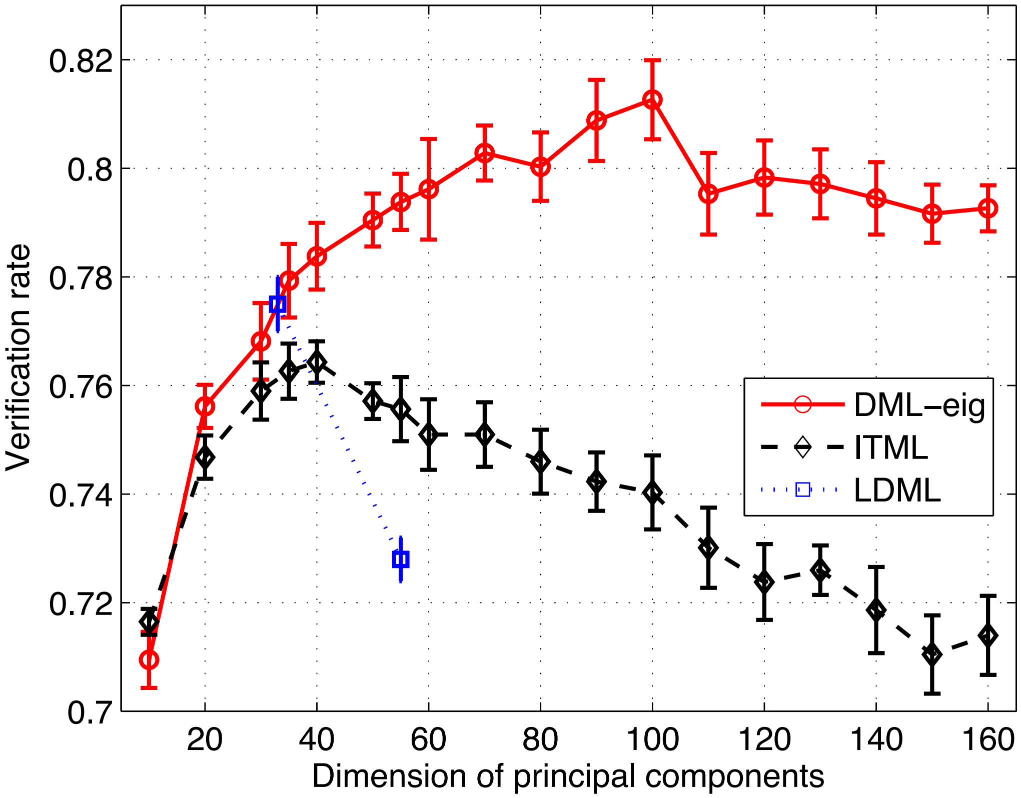


Figure 2: Performance of DML-eig, ITML and LDML metric by varying the dimension of the principal components using SIFT descriptor. The result of LDML is copied from Guillaumin et al. (2009).

The best result reported to date is 88.13% in restricted configuration which performs sophisticated large scale feature search (Pinto et al., 2011). This work used multiple complimentary representations which are derived through training set augmentation, alternative face comparison functions, and feature set searches with a varying number of model layers. These individual feature representations are then combined using kernel techniques. The results by other state-of-the-art methods are also based on different descriptors (Guillaumin et al., 2009; Wolf et al., 2009). The best result achieved by DML-eig is 85.65%, which is close to the other state-of-the-art approaches. In addition, we note that the performance of DML-eig based on the single SIFT descriptor (81.27% in Table 6) is better than that of LDML based on 4 types of descriptors (79.27% in Table 7). The ROC curves of different methods are depicted in Figure 3. We can see that DML-eig outperforms ITML and LDML while it is suboptimal to the best up-to-date method (Pinto et al., 2011) which, however, employed sophisticated feature search method.

Finally, the performance of DML-eig metric may be further improved by exploring different number of nearest neighbors and different types of descriptors such as those used in Pinto et al. (2011), making it a competitive candidate for the task of face verification.

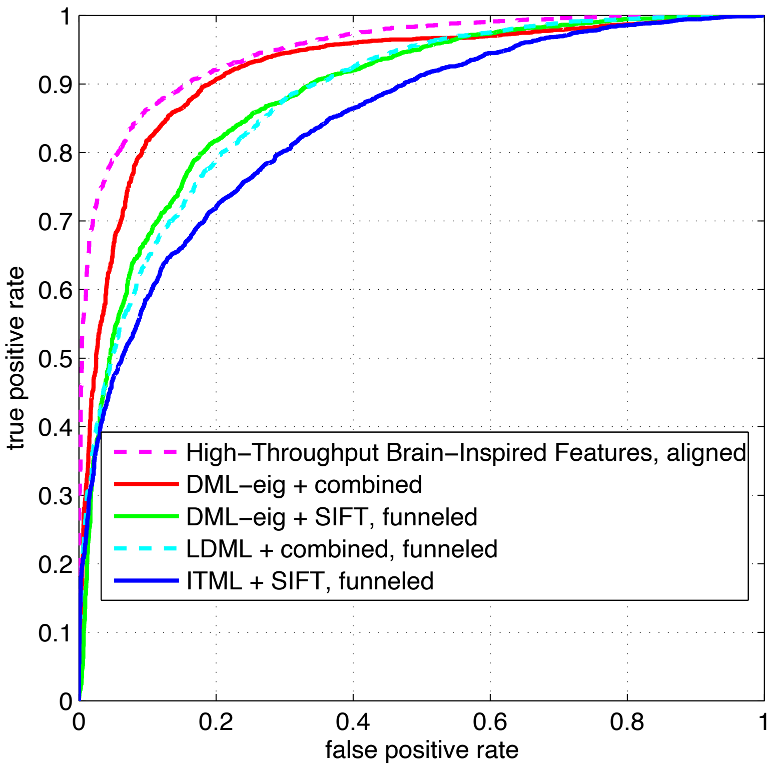


Figure 3: ROC curve of DML-eig and other the state of arts methods for face verification on LFW data set.

6. Conclusion

The main theme of this paper is to develop a new eigenvalue-optimization framework for metric learning. Within this context, we first proposed a novel metric learning model which was shown to be equivalent to a well-known eigenvalue optimization problem (Overton, 1988; Lewis and Overton, 1996). This appealing optimization formulation was further extended to LMNN (Weinberger et al., 2005) and maximum margin matrix factorization (Srebro et al., 2004). Then, we developed efficient first-order algorithms for metric learning which only involve the computation of the largest eigenvector of a matrix. Their convergence rates were rigorously established. Finally, experiments on various data sets have shown that our proposed approach is competitive with state-of-the-art metric learning methods. In particular, we reported promising results on the Labeled Faces in the Wild (LFW) data set.

In future we will exploit the extension of the above eigenvalue optimization framework to other machine learning tasks such as spectral graph cuts and semi-definite embedding (Weinberger et al., 2004). Another direction for investigation is to develop a kernelized version of DML-eig using the techniques in Jain et al. (2010). Finally, we will also investigate the performance of our methods on the LFW data set in the unrestricted configuration setting, and embed the technique of ball trees (Weinberger and Saul, 2008) into our algorithms to further increase the computational speed.

Acknowledgments

The authors would like to thank Colin Campbell, Massimiliano Pontil, and Charles Micchelli for stimulating discussion and invaluable comments on the preliminary version of this paper. The authors also sincerely thank the anonymous reviewers for their comments and suggestions which have led to valuable improvements of this paper. This work is supported by the EPSRC under grant EP/J001384/1. The second author would like to thank Cancer Research UK for the research grant.

Appendix A. Eigenvalue Optimization for Maximum-margin Matrix Factorization

Another important problem is low-rank matrix completion which recently has attracted much attention. This line of research involves computing a large matrix with a nuclear-norm (summation of singular values) regularization and the optimization problem here also consists of an SDP. Such tasks include multi-task feature learning (Argyriou et al., 2006) and low-rank matrix completion (Bach, 2008; Candes and Recht, 2008; Srebro et al., 2004). It has successful applications to collaborative filtering for predicting customers' preferences to products, where the matrix's rows and columns respectively identify the "customers" and "products", and a matrix entry encodes customers' preference of a product (e.g., Netflix data set, `http://www.netflixprize.com/`).

Similar eigenvalue optimization formulation can be developed for maximum-margin matrix factorization (MMMF) for collaborative filtering (Srebro et al., 2004). Given a partially labeled $Y_{ia} \in \{\pm 1\}$ with $ia \in S$, the target of MMMF is to learn a large matrix $X \in \mathbb{R}^{m\times n}$ where each entry X_{ia} indicates the preference of the customer i for product a. The following large margin model was proposed in Srebro et al. (2004) to learn X:

$$\begin{array}{ll} \min_X & \sum_{ia\in S} \xi_{ia} + \gamma \|X\|_* \\ \text{s.t.} & 1 - Y_{ia}X_{ia} \le \xi_{ia}, \\ & \xi_{ia} \ge 0, \ \forall ia \in S, \end{array}$$

where $\|X\|_*$ is the nuclear norm of X, that is, the summation of its singular values. The above model was further formulated as an SDP problem:

$$\begin{array}{ll} \min_M & \gamma \mathbf{Tr}(M) + \sum_{ia\in S} \xi_{ia} \\ & M = \begin{pmatrix} A & X \\ X^\top & B \end{pmatrix} \in \mathcal{S}_+^{(m+n)}, \\ & Y_{ia}X_{ia} + \xi_{ia} \ge 1, \ \forall ia \in S. \end{array} \tag{27}$$

Let e_i be a column vector with its i-th element one and all others zero, then we have $M_{i(m+a)} = X_{ia} = \langle C_{ia}, M\rangle$ with $C_{ia} = e_i e_{(m+a)}^\top$. Consequently, the constraint condition in problem (27) can be written as $\min_{ia\in S}\langle Y_{ia}, C_{ia}\rangle + \xi_{ia} \ge 1$. Using exact arguments for proving Theorem 3, we can formulate MMMF as an eigenvalue optimization problem.

Theorem 8. *MMMF formulation (27) is equivalent to*

$$\max\Big\{\min_{u\in\triangle} \sum_{ia\in S} u_{ia}\big(\xi_{ia} + \langle Y_{ia}C_{ia}, M\rangle\big) : \xi^\top \mathbf{1} + \gamma\mathbf{Tr}(M) = 1, M \in \mathcal{S}_+^{(m+n)}, \ \xi \ge 0\Big\}.$$

In particular it is equivalent to the following eigenvalue optimization problem:

$$\min_{u\in\triangle}\max\Big(u_{max},\frac{1}{\gamma}\lambda_{max}\big(\sum_{ia\in S}u_{ia}Y_{ia}C_{ia}\big)\Big). \tag{28}$$

As mentioned above, MMMF (27) is a standard SDP. Indeed, Srebro et al. (2004) proposed to employ standard SDP solvers (e.g., CSDP Borchers, 1999) to obtain the optimal solution. However, such generic solvers are only able to handle problems with about a hundred users and a hundred items. The eigenvalue-optimization formulation potentially provides more efficient algorithms for MMMF. Since the paper mainly focuses on metric learning, we leave its empirical implementation for future study.

Appendix B. Proof of Theorem 6

In this appendix we give the proof of Theorem 6. The spirit of the proof is very close to that of Theorem 1 in Ying and Zhou (2006) where similar conditions on step sizes were derived to guarantee the convergence of stochastic online learning algorithms in reproducing kernel Hilbert spaces.

Proof of Theorem 6. According to the assumption (22) on the step size, we can assume that, for any $t\geq t_0$, that $\alpha_t\leq 1/2$. Hence, the inequality (21) holds true. We will estimate the terms on the left-hand side of (21) one by one.

For the second term on the righthand side of (21), observe that $\prod_{j=t_0}^t(1-\alpha_j)\leq\exp\big\{-\sum_{j=t_0}^t\alpha_j\big\}\to 0$ as $t\to\infty$. Therefore, for any $\varepsilon>0$ there exists some $t_1\in\mathbb{N}$ such that the second term on the righthand side of (21) is bounded by ε whenever $t\geq t_1$.

To deal with the first term on the righthand side of (21), we use the assumption $\lim_{j\to\infty}\alpha_j=0$ and know that there exists some $j(\varepsilon)$ such that $\alpha_j\leq\varepsilon$ for every $j\geq j(\varepsilon)$. Write

$$\sum_{j=t_0}^{t}\alpha_j^2\prod_{k=j+1}^{t}(1-\alpha_k)=\sum_{j=t_0}^{j(\varepsilon)}\alpha_j^2\prod_{k=j+1}^{t}(1-\alpha_k)+\sum_{j=j(\varepsilon)+1}^{t}\alpha_j^2\prod_{k=j+1}^{t}(1-\alpha_k). \tag{29}$$

Since $j(\varepsilon)$ is fixed, we can find some $t_2\in\mathbb{N}$ such that for each $t\geq t_2$, there holds $\sum_{j=t(\varepsilon)+1}^t\alpha_j\geq\sum_{j=j(\varepsilon)+1}^{t_2}\alpha_j\geq\log\frac{j(\varepsilon)}{4\varepsilon}$. It follows that for each $1\leq j\leq j(\varepsilon)$, there holds $\prod_{k=j+1}^t(1-\alpha_k)\leq\exp\big\{-\sum_{k=j+1}^t\alpha_k\big\}\leq\exp\big\{-\sum_{k=j(\varepsilon)+1}^t\alpha_k\big\}\leq\frac{4\varepsilon}{j(\varepsilon)}$. This in connection with the bound $\alpha_j\leq 1/2$ for each $j\geq t_0$ tells us that the first term of (29) is bounded as

$$\sum_{j=t_0}^{t(\varepsilon)}\alpha_j^2\prod_{k=j+1}^{t}(1-\alpha_k)\leq\frac{4\varepsilon}{j(\varepsilon)}\sum_{j=t_0}^{j(\varepsilon)}\alpha_j^2\leq\varepsilon.$$

The second term on the righthand side of (29) is dominated by $\varepsilon\sum_{j=j(\varepsilon)+1}^{t-1}\alpha_j\prod_{k=j+1}^t(1-\alpha_k)$. Noting the fact that $\alpha_j=1-(1-\alpha_j)$ implies

$$\begin{aligned}\sum_{j=j(\varepsilon)+1}^{t}\alpha_j\prod_{k=j+1}^{t}(1-\alpha_k)&=\sum_{j=j(\varepsilon)+1}^{t}\Big[\prod_{k=j+1}^{t}(1-\alpha_k)-\prod_{k=j}^{t}(1-\alpha_k)\Big]\\&=\Big[1-\prod_{k=j(\varepsilon)+1}^{t}(1-\alpha_k)\Big]\leq 1.\end{aligned}$$

Therefore, when $t \geq \max\{t_1, t_2\}$, combining the estimation with inequality (21), we have $R_{t+1} \leq (1 + C_\mu)\varepsilon$. This proves the theorem. □

References

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, 2006.

F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019-1048, 2008.

M. Baes and M. Bürgisser. Smoothing techniques for solving semi-definite programs with many constraints. IFOR Internal report, ETH, 2009. Available electronically via `http://www.optimization-online.org/DB-FILE/2009/10/2414.pdf`.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937-965, 2005.

B. Borchers. CSDP, a C library for semi-definite programming. *Optimization Methods and Software*, 11:613-623, 1999.

S. Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semi-definite programs via low-rank factorization. *Mathematical Programming*, 95:329-357, 2003.

E.J. Candes and B. Recht. Exact matrix completion via convex optimisation. *arXiv:0805.4471*, 2008. Available electronically via `http://arxiv.org/abs/0805.4471`.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 209–216, 2007.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quaterly*, 3:149–154, 1956.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems 17*, 2004.

M. Guillaumin, J. Verbeek and C. Schmid. Is that you? Metric learning approaches for face identification. In *IEEE 12th International Conference on Computer Vision*, pages 498–505, 2009.

E. Hazan. Sparse approximation solutions to semi-definite programs. *LATIN: Proceedings of the 8th Latin American Conference on Theoretical informatics*, 2008.

C. Helmberg and F. Rendl. A spectral bundle method for semi-definite programming. *SIAM Journal on Optimization*, 10(3):673-696, 1999.

S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2072–2078, 2006.

G. B. Huang, M. Ramesh, T. Berg and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report* 07–49, October, 2007.

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

P. Jain, B. Kulis, I. S. Dhillon. Inductive regularized learning of kernel functions. In *Advances in Neural Information Processing Systems 23*, 2010.

R. Jin, S. Wang and Y. Zhou. Regularized distance metric learning: theory and algorithm. In *Advances in Neural Information Processing Systems 22*, 2009.

T. Kato and N. Nagano. Metric learning for enzyme active-site search. *Bioinformatics*, 26:2698-2704, 2010.

A. S. Lewis and M. L. Overton. Eigenvalue optimization. *Acta Numerica*, 5:149–190, 1996.

A. Nemirovski. *Efficient methods in convex programming*. Lecture Notes, 1994.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2003.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127-152, 2005.

Y. Nesterov. Smoothing technique and its applications in semi-definite optimization. *Mathematical Programming*, 110:245–259, 2007.

T. Ojala, M. Pietikainen and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM. J. Matrix Anal. & Appl.* 9:256–268, 1988.

N. Pinto and D. Cox. Beyond simple features: a large-scale feature search approach to unconstrained face recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2011.

R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

C. Shen, J. Kim, L. Wang and A. Hengel. Positive semi-definite metric learning with boosting. In *Advances in Neural Information Processing Systems 22*, 2009.

S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

N. Srebro, J.D.M. Rennie, and T.S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2004.

Y. Taigman and L. Wolf and T. Hassner. Multiple one-shots for utilizing class label information. In *The British Machine Vision Conference*, Sept. 2009.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems 19*, 2007.

J.-P. Vert, J. Qiu and W. S. Noble. A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, 8(Suppl 10), 2007.

K. Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbour classification. In *Advances in Neural Information Processing Systems 18*, 2005.

K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, pages 1160–1167, 2008.

K. Q. Weinberger, F. Sha and L. K. Saul. Learning the kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

L. Wolf, T. Hassner and Y. Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, 2009.

L. Wolf, T. Hassner and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision*, October, 2008.

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. In *Advances in Neural Information Processing Systems 15*, 2003.

L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Technical report, Department of Computer Science and Engineering, Michigan State University*, 2007.

L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. Hoi, and M. Satyanarayanan. A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32:30–44, 2010.

Y. Ying, K. Huang and C. Campbell. Sparse metric learning via smooth optimization. In *Advances in Neural Information Processing Systems 22*, 2009.

Y. Ying and D.X. Zhou, Online regularized classification algorithms. *IEEE Trans. Inform. Theory*, 11:4775-4788, 2006.

Journal of Machine Learning Research 13 (2012) 27-66 Submitted 12/10; Revised 6/11; Published 1/12

Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection

Gavin Brown GAVIN.BROWN@CS.MANCHESTER.AC.UK
Adam Pocock ADAM.POCOCK@CS.MANCHESTER.AC.UK
Ming-Jie Zhao MING-JIE.ZHAO@CS.MANCHESTER.AC.UK
Mikel Luján MIKEL.LUJAN@CS.MANCHESTER.AC.UK
School of Computer Science
University of Manchester
Manchester M13 9PL, UK

Editor: Isabelle Guyon

Abstract

We present a unifying framework for information theoretic feature selection, bringing almost two decades of research on heuristic filter criteria under a single theoretical interpretation. This is in response to the question: *"what are the implicit statistical assumptions of feature selection criteria based on mutual information?"*. To answer this, we adopt a different strategy than is usual in the feature selection literature—instead of trying to *define* a criterion, we *derive* one, directly from a clearly specified objective function: the conditional likelihood of the training labels. While many hand-designed heuristic criteria try to optimize a definition of feature 'relevancy' and 'redundancy', our approach leads to a probabilistic framework which naturally incorporates these concepts. As a result we can unify the numerous criteria published over the last two decades, and show them to be low-order approximations to the exact (but intractable) optimisation problem. The primary contribution is to show that *common heuristics for information based feature selection (including Markov Blanket algorithms as a special case) are approximate iterative maximisers of the conditional likelihood*. A large empirical study provides strong evidence to favour certain classes of criteria, in particular those that balance the relative size of the relevancy/redundancy terms. Overall we conclude that the JMI criterion (Yang and Moody, 1999; Meyer et al., 2008) provides the best tradeoff in terms of accuracy, stability, and flexibility with small data samples.

Keywords: feature selection, mutual information, conditional likelihood

1. Introduction

High dimensional data sets are a significant challenge for Machine Learning. Some of the most practically relevant and high-impact applications, such as *gene expression* data, may easily have more than 10,000 features. Many of these features may be completely *irrelevant* to the task at hand, or *redundant* in the context of others. Learning in this situation raises important issues, for example, over-fitting to irrelevant aspects of the data, and the computational burden of processing many similar features that provide redundant information. It is therefore an important research direction to automatically identify meaningful smaller subsets of these variables, that is, *feature selection*.

Feature selection techniques can be broadly grouped into approaches that are classifier-dependent ('wrapper' and 'embedded' methods), and classifier-independent ('filter' methods). Wrapper methods search the space of feature subsets, using the training/validation accuracy of a particular classifier as the measure of utility for a candidate subset. This may deliver significant advantages in generalisation, though has the disadvantage of a considerable computational expense, and may produce subsets that are overly specific to the classifier used. As a result, any change in the learning model is likely to render the feature set suboptimal. Embedded methods (Guyon et al., 2006, Chapter 3) exploit the structure of specific classes of learning models to *guide* the feature selection process. While the defining component of a wrapper method is simply the search procedure, the defining component of an embedded method is a criterion derived through fundamental knowledge of a specific class of functions. An example is the method introduced by Weston et al. (2001), selecting features to minimize a generalisation bound that holds for Support Vector Machines. These methods are less computationally expensive, and less prone to overfitting than wrappers, but still use quite strict model structure assumptions. In contrast, *filter* methods (Duch, 2006) separate the classification and feature selection components, and define a heuristic *scoring criterion* to act as a proxy measure of the classification accuracy. Filters evaluate statistics of the data *independently* of any particular classifier, thereby extracting features that are generic, having incorporated few assumptions.

Each of these three approaches has its advantages and disadvantages, the primary distinguishing factors being speed of computation, and the chance of overfitting. In general, in terms of speed, filters are faster than embedded methods which are in turn faster than wrappers. In terms of overfitting, wrappers have higher learning capacity so are more likely to overfit than embedded methods, which in turn are more likely to overfit than filter methods. All of this of course changes with extremes of data/feature availability—for example, embedded methods will likely outperform filter methods in generalisation error as the number of datapoints increases, and wrappers become more computationally unfeasible as the number of features increases. A primary advantage of filters is that they are relatively cheap in terms of computational expense, and are generally more amenable to a theoretical analysis of their design. Such theoretical analysis is the focus of this article.

The defining component of a filter method is the *relevance index* (also known as a *selection/scoring criterion*), quantifying the ‘utility’ of including a particular feature in the set. Numerous hand-designed heuristics have been suggested (Duch, 2006), all attempting to maximise feature ‘relevancy’ and minimise ‘redundancy’. However, few of these are motivated from a solid theoretical foundation. It is preferable to start from a more principled perspective—the desired approach is outlined eloquently by Guyon:

> *“It is important to start with a clean mathematical statement of the problem addressed [...] It should be made clear how optimally the chosen approach addresses the problem stated. Finally, the eventual approximations made by the algorithm to solve the optimisation problem stated should be explained. An interesting topic of research would be to ‘retrofit’ successful heuristic algorithms in a theoretical framework.”* (Guyon et al., 2006, pg. 21)

In this work we adopt this approach—instead of trying to *define* feature relevance indices, we *derive* them starting from a clearly specified objective function. The objective we choose is a well accepted statistical principle, *the conditional likelihood of the class labels given the features*. As a result we are able to provide deeper insight into the feature selection problem, and achieve precisely the goal above, to retrofit numerous hand-designed heuristics into a theoretical framework.

2. Background

In this section we give a brief introduction to information theoretic concepts, followed by a summary of how they have been used to tackle the feature selection problem.

2.1 Entropy and Mutual Information

The fundamental unit of information is the *entropy* of a random variable, discussed in several standard texts, most prominently (Cover and Thomas, 1991). The entropy, denoted $H(X)$, quantifies the uncertainty present in the distribution of X. It is defined as,

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x),$$

where the lower case x denotes a possible value that the variable X can adopt from the alphabet $\mathcal{X}$. To compute[1] this, we need an estimate of the distribution $p(X)$. When X is discrete this can be estimated by frequency counts from data, that is $\hat{p}(x) = \frac{\#x}{N}$, the fraction of observations taking on value x from the total N. We provide more discussion on this issue in Section 3.3. If the distribution is highly biased toward one particular event $x \in \mathcal{X}$, that is, little uncertainty over the outcome, then the entropy is low. If all events are equally likely, that is, maximum uncertainty over the outcome, then $H(X)$ is maximal.[2] Following the standard rules of probability theory, entropy can be *conditioned* on other events. The *conditional entropy* of X given Y is denoted,

$$H(X|Y) = -\sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log p(x|y).$$

This can be thought of as the amount of uncertainty remaining in X after we learn the outcome of Y. We can now define the *Mutual Information* (Shannon, 1948) between X and Y, that is, the amount of information *shared* by X and Y, as follows:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(xy) \log \frac{p(xy)}{p(x)p(y)}. \end{aligned}$$

This is the difference of two entropies—the uncertainty *before* Y is known, $H(X)$, and the uncertainty *after* Y is known, $H(X|Y)$. This can also be interpreted as the amount of uncertainty in X which is removed by knowing Y, thus following the intuitive meaning of mutual information as the amount of information that one variable provides about another. It should be noted that the Mutual Information is symmetric, that is, $I(X;Y) = I(Y;X)$, and is zero if and only if the variables are statistically independent, that is $p(xy) = p(x)p(y)$. The relation between these quantities can be seen in Figure 1. The Mutual Information can also be conditioned—the *conditional information* is,

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) - H(X|YZ) \\ &= \sum_{z \in \mathcal{Z}} p(z) \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(xy|z) \log \frac{p(xy|z)}{p(x|z)p(y|z)}. \end{aligned}$$

[1] The base of the logarithm is arbitrary, but decides the 'units' of the entropy. When using base 2, the units are 'bits', when using base e, the units are 'nats.'

[2] In general, $0 \leq H(X) \leq \log(|\mathcal{X}|)$.

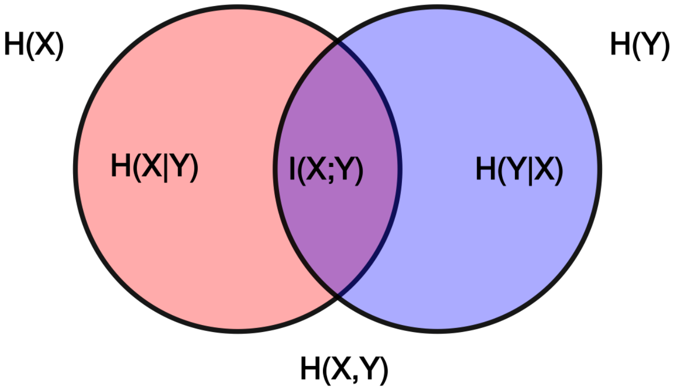


Figure 1: Illustration of various information theoretic quantities.

This can be thought of as the information still shared between X and Y after the value of a third variable, Z, is revealed. The conditional mutual information will emerge as a particularly important property in understanding the results of this work.

This section has briefly covered the principles of information theory; in the following section we discuss motivations for using it to solve the feature selection problem.

2.2 Filter Criteria Based on Mutual Information

Filter methods are defined by a criterion J, also referred to as a 'relevance index' or 'scoring' criterion (Duch, 2006), which is intended to measure how potentially useful a feature or feature subset may be when used in a classifier. An intuitive J would be some measure of correlation between the feature and the class label—the intuition being that a stronger correlation between these should imply a greater predictive ability when using the feature. For a class label Y, the *mutual information* score for a feature X_k is

$$J_{mim}(X_k) = I(X_k;Y). \tag{1}$$

This heuristic, which considers a score for each feature independently of others, has been used many times in the literature, for example, Lewis (1992). We refer to this feature scoring criterion as 'MIM', standing for *Mutual Information Maximisation*. To use this measure we simply rank the features in order of their MIM score, and select the top K features, where K is decided by some predefined need for a certain number of features or some other stopping criterion (Duch, 2006). A commonly cited justification for this measure is that the mutual information can be used to write both an upper and lower bound on the Bayes error rate (Fano, 1961; Hellman and Raviv, 1970). An important limitation is that this assumes that each feature is independent of all other features—and effectively ranks the features in descending order of their individual mutual information content. However, where features may be interdependent, this is known to be suboptimal. In general, it is widely accepted that a useful and parsimonious set of features should not only be individually *relevant*, but also should not be *redundant* with respect to each other—features should not be highly correlated. The reader is warned that while this statement seems appealingly intuitive, it is *not strictly correct*, as will be expanded upon in later sections. In spite of this, several criteria have been proposed that attempt to pursue this ‘relevancy-redundancy’ goal. For example, Battiti (1994) presents the *Mutual Information Feature Selection* (MIFS) criterion:

$$J_{mifs}(X_k) = I(X_k;Y) - \beta \sum_{X_j \in S} I(X_k;X_j),$$

where S is the set of currently selected features. This includes the $I(X_k;Y)$ term to ensure feature *relevance*, but introduces a penalty to enforce low correlations with features already selected in S. Note that this assumes we are selecting features *sequentially*, iteratively constructing our final feature subset. For a survey of other search methods than simple sequential selection, the reader is referred to Duch (2006); however it should be noted that all theoretical results presented in this paper will be generally applicable to any search procedure, and based solely on properties of the criteria themselves. The β in the MIFS criterion is a configurable parameter, which must be set experimentally. Using $\beta = 0$ would be equivalent to $J_{mim}(X_k)$, selecting features independently, while a larger value will place more emphasis on reducing inter-feature dependencies. In experiments, Battiti found that $\beta = 1$ is often optimal, though with no strong theory to explain why. The MIFS criterion focuses on reducing *redundancy*; an alternative approach was proposed by Yang and Moody (1999), and also later by Meyer et al. (2008) using the *Joint Mutual Information* (JMI), to focus on increasing *complementary* information between features. The JMI score for feature X_k is

$$J_{jmi}(X_k) = \sum_{X_j \in S} I(X_k X_j;Y).$$

This is the information between the targets and a *joint* random variable $X_k X_j$, defined by pairing the candidate X_k with each feature previously selected. The idea is if the candidate feature is ‘complementary’ with existing features, we should include it.

The MIFS and JMI schemes were the first of many criteria that attempted to manage the relevance-redundancy tradeoff with various heuristic terms, however it is clear they have very different motivations. The criteria identified in the literature 1992-2011 are listed in Table 1. The practice in this research problem has been to *hand-design* criteria, piecing criteria together as a jigsaw of information theoretic terms—the overall aim to manage the relevance-redundancy trade-off, with each new criterion motivated from a different direction. Several questions arise here: Which criterion should we believe? What do they assume about the data? Are there other useful criteria, as yet undiscovered? In the following section we offer a novel perspective on this problem.

3. A Novel Approach

In the following sections we formulate the feature selection task as a conditional likelihood problem. We will demonstrate that precise links can be drawn between the well-accepted statistical framework of likelihood functions, and the current feature selection heuristics of mutual information criteria.

3.1 A Conditional Likelihood Problem

We assume an underlying i.i.d. process $p : X \rightarrow Y$, from which we have a sample of N observations. Each observation is a pair $(\mathbf{x}, y)$, consisting of a d-dimensional feature vector $\mathbf{x} = [x_1, ..., x_d]^T$, and a target class y, drawn from the underlying random variables $X = \{X_1, ..., X_d\}$ and Y. Furthermore, we assume that $p(y|\mathbf{x})$ is defined by a *subset* of the d features in $\mathbf{x}$, while the remaining features are

Criterion	*Full name*	*Authors*
MIM	Mutual Information Maximisation	Lewis (1992)
MIFS	Mutual Information Feature Selection	Battiti (1994)
KS	Koller-Sahami metric	Koller and Sahami (1996)
JMI	Joint Mutual Information	Yang and Moody (1999)
MIFS-U	MIFS-'Uniform'	Kwak and Choi (2002)
IF	Informative Fragments	Vidal-Naquet and Ullman (2003)
FCBF	Fast Correlation Based Filter	Yu and Liu (2004)
AMIFS	Adaptive MIFS	Tesmer and Estevez (2004)
CMIM	Conditional Mutual Info Maximisation	Fleuret (2004)
MRMR	Max-Relevance Min-Redundancy	Peng et al. (2005)
ICAP	Interaction Capping	Jakulin (2005)
CIFE	Conditional Infomax Feature Extraction	Lin and Tang (2006)
DISR	Double Input Symmetrical Relevance	Meyer and Bontempi (2006)
MINRED	Minimum Redundancy	Duch (2006)
IGFS	Interaction Gain Feature Selection	El Akadi et al. (2008)
SOA	Second Order Approximation	Guo and Nixon (2009)
CMIFS	Conditional MIFS	Cheng et al. (2011)

Table 1: Various information-based criteria from the literature. Sections 3 and 4 will show how these can all be interpreted in a single theoretical framework.

irrelevant. Our modeling task is therefore two-fold: firstly to identify the features that play a functional role, and secondly to use these features to perform predictions. In this work we concentrate on the first stage, that of selecting the relevant features.

We adopt a d-dimensional binary vector θ: a 1 indicating the feature is selected, a 0 indicating it is discarded. Notation $\mathbf{x}_\theta$ indicates the vector of selected features, that is, the full vector $\mathbf{x}$ projected onto the dimensions specified by θ. Notation $\mathbf{x}_{\tilde{\theta}}$ is the complement, that is, the unselected features. The full feature vector can therefore be expressed as $\mathbf{x} = \{\mathbf{x}_\theta, \mathbf{x}_{\tilde{\theta}}\}$. As mentioned, we assume the process p is defined by a subset of the features, so for some unknown optimal vector θ^*, we have that $p(y|\mathbf{x}) = p(y|\mathbf{x}_{\theta^*})$. We approximate p using a hypothetical predictive model q, with two layers of parameters: θ representing which features are selected, and τ representing parameters used to predict y. Our problem statement is to identify the minimal subset of features such that we *maximize the conditional likelihood of the training labels, with respect to these parameters*. For i.i.d. data $\mathcal{D} = \{(\mathbf{x}^i, y^i); i = 1..N\}$ the conditional likelihood of the labels given parameters $\{\theta, \tau\}$ is

$$\mathcal{L}(\theta, \tau|\mathcal{D}) = \prod_{i=1}^{N} q(y^i|\mathbf{x}_\theta^i, \tau).$$

The (scaled) conditional *log*-likelihood is

$$\ell = \frac{1}{N}\sum_{i=1}^{N} \log q(y^i|\mathbf{x}_\theta^i, \tau). \tag{2}$$

This is the error function we wish to optimize with respect to the parameters $\{\tau, \theta\}$; the scaling term has no effect on the optima, but simplifies exposition later. Using conditional likelihood has become popular in so-called *discriminative* modelling applications, where we are interested only in the classification performance; for example Grossman and Domingos (2004) used it to learn Bayesian Network classifiers. We will expand upon this link to discriminative models in Section 9.3. Maximising conditional likelihood corresponds to minimising KL-divergence between the true and predicted class posterior probabilities—for classification, we often only require the *correct* class, and not precise estimates of the posteriors, hence Equation (2) is a proxy lower bound for classification accuracy.

We now introduce the quantity $p(y|\mathbf{x}_\theta)$: this is the true distribution of the class labels given the selected features $\mathbf{x}_\theta$. It is important to note the distinction from $p(y|\mathbf{x})$, the true distribution given *all* features. Multiplying and dividing q by $p(y|\mathbf{x}_\theta)$, we can re-write the above as,

$$\ell \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\log\frac{q(y^i|\mathbf{x}^i_\theta,\tau)}{p(y^i|\mathbf{x}^i_\theta)}+\frac{1}{N}\sum_{i=1}^{N}\log p(y^i|\mathbf{x}^i_\theta). \tag{3}$$

The second term in (3) can be similarly expanded, introducing the probability $p(y|\mathbf{x})$:

$$\ell \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\log\frac{q(y^i|\mathbf{x}^i_\theta,\tau)}{p(y^i|\mathbf{x}^i_\theta)}+\frac{1}{N}\sum_{i=1}^{N}\log\frac{p(y^i|\mathbf{x}^i_\theta)}{p(y^i|\mathbf{x}^i)}+\frac{1}{N}\sum_{i=1}^{N}\log p(y^i|\mathbf{x}^i).$$

These are finite sample approximations, drawing datapoints i.i.d. with respect to the distribution $p(\mathbf{x}y)$. We use $E_{\mathbf{x}y}\{\cdot\}$ to denote statistical expectation, and for convenience we negate the above, turning our maximisation problem into a minimisation. This gives us,

$$-\ell \quad \approx \quad E_{\mathbf{x}y}\Big\{\log\frac{p(y|\mathbf{x}_\theta)}{q(y|\mathbf{x}_\theta,\tau)}\Big\}+E_{\mathbf{x}y}\Big\{\log\frac{p(y|\mathbf{x})}{p(y|\mathbf{x}_\theta)}\Big\}-E_{\mathbf{x}y}\Big\{\log p(y|\mathbf{x})\Big\}. \tag{4}$$

These three terms have interesting properties which together define the feature selection problem. It is particularly interesting to note that the second term is *precisely* that introduced by Koller and Sahami (1996) in their definitions of optimal feature selection. In their work, the term was adopted ad-hoc as a sensible objective to follow—here we have shown it to be a direct and natural consequence of adopting the conditional likelihood as an objective function. Remembering $\mathbf{x}=\{\mathbf{x}_\theta,\mathbf{x}_{\tilde\theta}\}$, this second term can be developed:

$$\begin{aligned}
\Delta_{KS} &= E_{\mathbf{x}y}\Big\{\log\frac{p(y|\mathbf{x})}{p(y|\mathbf{x}_\theta)}\Big\}\\
&= \sum_{\mathbf{x}y}p(\mathbf{x}y)\log\frac{p(y|\mathbf{x}_\theta\mathbf{x}_{\tilde\theta})}{p(y|\mathbf{x}_\theta)}\\
&= \sum_{\mathbf{x}y}p(\mathbf{x}y)\log\frac{p(y|\mathbf{x}_\theta\mathbf{x}_{\tilde\theta})}{p(y|\mathbf{x}_\theta)}\frac{p(\mathbf{x}_{\tilde\theta}|\mathbf{x}_\theta)}{p(\mathbf{x}_{\tilde\theta}|\mathbf{x}_\theta)}\\
&= \sum_{\mathbf{x}y}p(\mathbf{x}y)\log\frac{p(\mathbf{x}_{\tilde\theta}y|\mathbf{x}_\theta)}{p(\mathbf{x}_{\tilde\theta}|\mathbf{x}_\theta)p(y|\mathbf{x}_\theta)}\\
&= I(X_{\tilde\theta};Y|X_\theta).
\end{aligned} \tag{5}$$

This is the conditional mutual information between the class label and the remaining features, given the selected features. We can note also that the third term in (4) is another information theoretic quantity, the conditional entropy $H(Y|X)$. In summary, we see that our objective function can be decomposed into three distinct terms, each with its own interpretation:

$$\lim_{N\to\infty} -\ell \;=\; E_{\mathbf{x}y}\left\{ \log \frac{p(y|\mathbf{x}_\theta)}{q(y|\mathbf{x}_\theta,\tau)} \right\} + I(X_{\tilde{\theta}};Y|X_\theta) + H(Y|X). \qquad (6)$$

The first term is a likelihood ratio between the true and the predicted class distributions given the selected features, averaged over the input space. The size of this term will depend on how well the model q can approximate p, given the supplied features.[3] When θ takes on the true value θ^* (or consists of a superset of θ^*) this becomes a KL-divergence $p||q$. The second term is $I(X_{\tilde{\theta}};Y|X_\theta)$, the conditional mutual information between the class label and the unselected features, given the selected features. The size of this term depends solely on the choice of features, and will decrease as the selected feature set X_θ explains more about Y, until eventually becoming zero when the remaining features $X_{\tilde{\theta}}$ contain no additional information about Y in the context of X_θ. It can be noted that due to the chain rule, we have

$$I(X;Y) = I(X_\theta;Y) + I(X_{\tilde{\theta}};Y|X_\theta),$$

hence minimizing $I(X_{\tilde{\theta}};Y|X_\theta)$ is equivalent to maximising $I(X_\theta;Y)$. The final term is $H(Y|X)$, the conditional entropy of the labels given *all features*. This term quantifies the uncertainty still remaining in the label even when we know *all possible* features; it is an irreducible constant, independent of all parameters, and in fact forms a bound on the Bayes error (Fano, 1961).

These three terms make explicit the effect of the feature selection parameters θ, separating them from the effect of the parameters τ in the model that *uses* those features. If we somehow had the optimal feature subset θ^*, which perfectly captured the underlying process p, then $I(X_{\tilde{\theta}};Y|X_\theta)$ would be zero. The remaining (reducible) error is then down to the KL divergence $p||q$, expressing how well the predictive model q can *make use* of the provided features. Of course, different models q will have different predictive ability: a good feature subset will not necessarily be put to good use if the model is too simple to express the underlying function. This perspective was also considered by Tsamardinos and Aliferis (2003), and earlier by Kohavi and John (1997)—the above results place these in the context of a precise objective function, the conditional likelihood. For the remainder of the paper we will use the same assumption as that made implicitly by *all* filter selection methods. For completeness, here we make the assumption explicit:

Definition 1 : Filter assumption
Given an objective function for a classifier, we can address the problems of optimizing the feature set and optimizing the classifier in two stages: first picking good features, then building the classifier to use them.

This implies that the second term in (6) can be optimized independently of the first. In this section we have formulated the feature selection task as a conditional likelihood problem. In the following, we consider how this problem statement relates to the existing literature, and discuss how to solve it in practice: including how to optimize the feature selection parameters, and the estimation of the necessary distributions.

[3] In fact, if q is a *consistent* estimator, this term will approach zero with large N.

3.2 Optimizing the Feature Selection Parameters

Under the filter assumption in Definition 1, Equation (6) demonstrates that the optima of the conditional likelihood coincide with that of the conditional mutual information:

$$\arg\max_{\theta} \mathcal{L}(\theta|\mathcal{D}) = \arg\min_{\theta} I(X_{\tilde{\theta}};Y|X_\theta). \tag{7}$$

There may of course be multiple global optima, in addition to the trivial minimum of selecting all features. With this in mind, we can introduce a minimality constraint on the size of the feature set, and define our problem:

$$\theta^* = \arg\min_{\theta'}\{|\theta'| : \theta' = \arg\min_{\theta} I(X_{\tilde{\theta}};Y|X_\theta)\}. \tag{8}$$

This is the smallest feature set X_θ, such that the mutual information $I(X_{\tilde{\theta}};Y|X_\theta)$ is minimal, and thus the conditional likelihood is maximal. It should be remembered that the likelihood is only our proxy for classification error, and the minimal feature set in terms of classification could be smaller than that which optimises likelihood. In the following paragraphs, we consider how this problem is implicitly tackled by methods already in the literature.

A common heuristic approach is a sequential search considering features one-by-one for addition/removal; this is used for example in Markov Blanket learning algorithms such as IAMB (Tsamardinos et al., 2003). We will now demonstrate that this sequential search heuristic is in fact equivalent to a greedy iterative optimisation of Equation (8). To understand this we must time-index the feature sets. Notation $X_{\theta^t}/X_{\tilde{\theta}^t}$ indicates the selected and unselected feature sets at timestep t—with a slight abuse of notation treating these interchangeably as sets and random variables.

Definition 2 : Forward Selection Step with Mutual Information
The forward selection step adds the feature with the maximum mutual information in the context of the currently selected set X_{θ^t}. The operations performed are:

$$\begin{aligned} X_k &= \arg\max_{X_k \in X_{\tilde{\theta}^t}} I(X_k;Y|X_{\theta^t}), \\ X_{\theta^{t+1}} &\leftarrow X_{\theta^t} \cup X_k, \\ X_{\tilde{\theta}^{t+1}} &\leftarrow X_{\tilde{\theta}^t} \backslash X_k. \end{aligned}$$

A subtle (but important) implementation point for this selection heuristic is that it should *not* add another feature if $\forall X_k,\ I(X_k;Y|X_\theta) = 0$. This ensures we will not unnecessarily increase the size of the feature set.

Theorem 3 *The forward selection mutual information heuristic adds the feature that generates the largest possible increase in the conditional likelihood—a greedy iterative maximisation.*

Proof With the definitions above and the chain rule of mutual information, we have that:

$$I(X_{\tilde{\theta}^{t+1}};Y|X_{\theta^{t+1}}) = I(X_{\tilde{\theta}^t};Y|X_{\theta^t}) - I(X_k;Y|X_{\theta^t}).$$

The feature X_k that *maximises* $I(X_k;Y|X_{\theta^t})$ is the same that *minimizes* $I(X_{\tilde{\theta}^{t+1}};Y|X_{\theta^{t+1}})$; therefore the forward step is a greedy *minimization* of our objective $I(X_{\tilde{\theta}};Y|X_\theta)$, and therefore maximises the conditional likelihood. ■

Definition 4 : Backward Elimination Step with Mutual Information
In a backward step, a feature is removed—the utility of a feature X_k is considered as its mutual information with the target, conditioned on all other elements of the selected set without X_k. The operations performed are:

$$\begin{aligned} X_k &= \arg\min_{X_k \in X_{\theta^t}} I(X_k;Y|\{X_{\theta^t} \backslash X_k\}). \\ X_{\theta^{t+1}} &\leftarrow X_{\theta^t} \backslash X_k \\ X_{\widetilde{\theta}^{t+1}} &\leftarrow X_{\widetilde{\theta}^t} \cup X_k \end{aligned}$$

Theorem 5 *The backward elimination mutual information heuristic removes the feature that causes the minimum possible decrease in the conditional likelihood.*

Proof With these definitions and the chain rule of mutual information, we have that:

$$I(X_{\widetilde{\theta}^{t+1}};Y|X_{\theta^{t+1}}) = I(X_{\widetilde{\theta}^t};Y|X_{\theta^t}) + I(X_k;Y|X_{\theta^{t+1}}).$$

The feature X_k that *minimizes* $I(X_k;Y|X_{\theta^{t+1}})$ is that which keeps $I(X_{\widetilde{\theta}^{t+1}};Y|X_{\theta^{t+1}})$ as close as possible to $I(X_{\widetilde{\theta}^t};Y|X_{\theta^t})$; therefore the backward elimination step removes a feature while attempting to maintain the likelihood as close as possible to its current value. ■

To strictly achieve our optimization goal, a backward step should *only* remove a feature if $I(X_k;Y|\{X_{\theta^t} \backslash X_k\}) = 0$. In practice, working with real data, there will likely be estimation errors (see the following section) and thus very rarely the strict zero will be observed. This brings us to an interesting corollary regarding IAMB (Tsamardinos and Aliferis, 2003).

Corollary 6 *Since the IAMB algorithm uses precisely these forward/backward selection heuristics, it is a greedy iterative maximisation of the conditional likelihood. In IAMB, a backward elimination step is only accepted if $I(X_k;Y|\{X_{\theta^t} \backslash X_k\}) \approx 0$, and otherwise the procedure terminates.*

In Tsamardinos and Aliferis (2003) it is shown that IAMB returns the Markov Blanket of any target node in a Bayesian network, and that this set coincides with the strongly relevant features in the definitions from Kohavi and John (1997). The precise links to this literature are explored further in Section 7. The IAMB family of algorithms adopt a common assumption, that the data is *faithful* to some unknown Bayesian Network. In the cases where this assumption holds, the procedure was proven to identify the unique Markov Blanket. Since IAMB uses precisely the forward/backward steps we have derived, we can conclude that *the Markov Blanket coincides with the (unique) maximum of the conditional likelihood function*. A more recent variation of the IAMB algorithm, called MMMB (Min-Max Markov Blanket) uses a series of optimisations to mitigate the requirement of exponential amounts of data to estimate the relevant statistical quantities. These optimisations do not change the underlying behaviour of the algorithm, as it still maximises the conditional likelihood for the selected feature set, however they do slightly obscure the strong link to our framework.

3.3 Estimation of the Mutual Information Terms

In considering the forward/backward heuristics, we must take account of the fact that we do not have perfect knowledge of the mutual information. This is because we have implicitly assumed we have access to the true distributions $p(\mathbf{x}y)$, $p(y|\mathbf{x}_\theta)$, etc. In practice we have to estimate these from data. The problem calculating mutual information reduces to that of *entropy estimation*, and is fundamental in statistics (Paninski, 2003). The mutual information is defined as the expected logarithm of a ratio:

$$I(X;Y) = E_{xy}\Big\{ \log \frac{p(xy)}{p(x)p(y)} \Big\}.$$

We can estimate this, since the Strong Law of Large Numbers assures us that the sample estimate using $\hat{p}$ converges *almost surely* to the expected value—for a dataset of N i.i.d. observations (x^i, y^i),

$$I(X;Y) \approx \hat{I}(X;Y) = \frac{1}{N}\sum_{i=1}^{N} \log \frac{\hat{p}(x^i y^i)}{\hat{p}(x^i)\hat{p}(y^i)}.$$

In order to calculate this we need the estimated distributions $\hat{p}(xy)$, $\hat{p}(x)$, and $\hat{p}(y)$. The computation of entropies for continuous or ordinal data is highly non-trivial, and requires an assumed model of the underlying distributions—to simplify experiments throughout this article, we use discrete data, and estimate distributions with *histogram estimators* using fixed-width bins. The probability of any particular event $p(X = x)$ is estimated by maximum likelihood, the frequency of occurrence of the event $X = x$ divided by the total number of events (i.e., datapoints). For more information on alternative entropy estimation procedures, we refer the reader to Paninski (2003).

At this point we must note that the approximation above holds *only* if N is large *relative to the dimension of the distributions over x and y*. For example if x, y are binary, $N \approx 100$ should be more than sufficient to get reliable estimates; however if x, y are multinomial, this will likely be insufficient. In the context of the sequential selection heuristics we have discussed, we are approximating $I(X_k;Y|X_\theta)$ as,

$$I(X_k;Y|X_\theta) \approx \hat{I}(X_k;Y|X_\theta) = \frac{1}{N}\sum_{i=1}^{N} \log \frac{\hat{p}(x_k^i y^i|\mathbf{x}_\theta^i)}{\hat{p}(x_k^i|\mathbf{x}_\theta^i)\hat{p}(y^i|\mathbf{x}_\theta^i)}. \tag{9}$$

As the dimension of the variable X_θ grows (i.e., as we add more features) then the necessary probability distributions become more high dimensional, and hence our estimate of the mutual information becomes less reliable. This in turn causes increasingly poor judgements for the inclusion/exclusion of features. For precisely this reason, the research community have developed various low-dimensional approximations to (9). In the following sections, we will investigate the implicit statistical assumptions and empirical effects of these approximations.

In the remainder of this paper, we use $I(X;Y)$ to denote the ideal case of being able to compute the mutual information, though in practice on real data we use the finite sample estimate $\hat{I}(X;Y)$.

3.4 Summary

In these sections we have in effect *reverse-engineered* a mutual information-based selection scheme, starting from a clearly defined conditional likelihood problem, and discussed estimation of the various quantities involved. In the following sections we will show that we can retrofit numerous existing relevancy-redundancy heuristics from the feature selection literature into this probabilistic framework.

4. Retrofitting Successful Heuristics

In the previous section, starting from a clearly defined conditional likelihood problem, we derived a greedy optimization process which assesses features based on a simple scoring criterion on the utility of including a feature $X_k \in X_{\tilde{\theta}}$. The score for a feature X_k is,

$$J_{cmi}(X_k) = I(X_k;Y|S), \tag{10}$$

where *cmi* stands for conditional mutual information, and for notational brevity we now use $S = X_\theta$ for the currently selected set. An important question is, how does (10) relate to existing heuristics in the literature, such as MIFS? We will see that MIFS, and certain other criteria, can be phrased cleanly as *linear combinations* of Shannon entropy terms, while some are non-linear combinations, involving *max* or *min* operations.

4.1 Criteria as Linear Combinations of Shannon Information Terms

Repeating the MIFS criterion for clarity,

$$J_{mifs}(X_k) = I(X_k;Y) - \beta \sum_{X_j \in S} I(X_k;X_j). \tag{11}$$

We can see that we first need to rearrange (10) into the form of a simple relevancy term between X_k and Y, plus some additional terms, before we can compare it to MIFS. Using the identity $I(A;B|C) - I(A;B) = I(A;C|B) - I(A;C)$, we can re-express (10) as,

$$J_{cmi}(X_k) = I(X_k;Y|S) = I(X_k;Y) - I(X_k;S) + I(X_k;S|Y). \tag{12}$$

It is interesting to see terms in this expression corresponding to the concepts of 'relevancy' and 'redundancy', that is, $I(X_k;Y)$ and $I(X_k;S)$. The score will be increased if the relevancy of X_k is large and the redundancy with existing features is small. This is in accordance with a common view in the feature selection literature, observing that we wish to avoid redundant variables. However, we can also see an important additional term $I(X_k;S|Y)$, which is not traditionally accounted for in the feature selection literature—we call this the *conditional redundancy*. This term has the opposite sign to the redundancy $I(X_k;S)$, hence J_{cmi} will be increased when this is large, that is, a strong class-conditional dependence of X_k with the existing set S. Thus, we come to the important conclusion that *the inclusion of correlated features can be useful*, provided the correlation *within classes* is stronger than the overall correlation. We note that this is a similar observation to that of Guyon et al. (2006), that "correlation does not imply redundancy"—Equation (12) effectively embodies this statement in information theoretic terms.

The sum of the last two terms in (12) represents the three-way interaction between the existing feature set S, the target Y, and the candidate feature X_k being considered for inclusion in S. To further understand this, we can note the following property:

$$I(X_kS;Y) = I(S;Y) + I(X_k;Y|S) = I(S;Y) + I(X_k;Y) - I(X_k;S) + I(X_k;S|Y).$$

We see that if $I(X_k;S) > I(X_k;S|Y)$, then the total utility when including X_k, that is $I(X_kS;Y)$, is *less* than the sum of the individual relevancies $I(S;Y) + I(X_k;Y)$. This can be interpreted as X_k having unnecessary duplicated information. In the opposite case, when $I(X_k;S) < I(X_k;S|Y)$, then X_k and

S combine well and provide more information *together* than by the sum of their parts, $I(S;Y)$, and $I(X_k;Y)$.

The important point to take away from this expression is that the terms are in a *trade-off*—we do not require a feature with low redundancy for its own sake, but instead require a feature that best trades off the three terms so as to maximise the score overall. Much like the bias-variance dilemma, attempting to decrease one term is likely to increase another.

The relation of (10) and (11) can be seen with assumptions on the underlying distribution $p(\mathbf{x}y)$. Writing the latter two terms of (12) as entropies:

$$\begin{aligned} J_{cmi}(X_k) \;=\;& I(X_k;Y) \\ & - H(S) + H(S|X_k) \\ & + H(S|Y) - H(S|X_kY). \end{aligned} \tag{13}$$

To develop this further, we require an assumption.

Assumption 1 *For all unselected features $X_k \in X_{\tilde{\theta}}$, assume the following,*

$$\begin{aligned} p(\mathbf{x}_\theta|x_k) &= \prod_{j\in S} p(x_j|x_k) \\ p(\mathbf{x}_\theta|x_ky) &= \prod_{j\in S} p(x_j|x_ky). \end{aligned}$$

This states that the selected features X_θ are independent and class-conditionally independent given the unselected feature X_k under consideration.

Using this, Equation (13) becomes,

$$\begin{aligned} J'_{cmi}(X_k) \;=\;& I(X_k;Y) \\ & - H(S) + \sum_{j\in S} H(X_j|X_k) \\ & + H(S|Y) - \sum_{j\in S} H(X_j|X_kY). \end{aligned}$$

where the prime on J indicates we are making assumptions on the distribution. Now, if we introduce $\sum_{j\in S} H(X_j) - \sum_{j\in S} H(X_j)$, and $\sum_{j\in S} H(X_j|Y) - \sum_{j\in S} H(X_j|Y)$, we recover mutual information terms, between the candidate feature and each member of the set S, plus some additional terms,

$$\begin{aligned} J'_{cmi}(X_k) \;=\;& I(X_k;Y) \\ & - \sum_{j\in S} I(X_j;X_k) + \sum_{j\in S} H(X_j) - H(S) \\ & + \sum_{j\in S} I(X_j;X_k|Y) - \sum_{j\in S} H(X_j|Y) + H(S|Y). \end{aligned} \tag{14}$$

Several of the terms in (14) are constant with respect to X_k—as such, removing them will have *no effect on the choice of feature*. Removing these terms, we have an equivalent criterion,

$$J'_{cmi}(X_k) = I(X_k;Y) - \sum_{j\in S} I(X_j;X_k) + \sum_{j\in S} I(X_j;X_k|Y). \tag{15}$$

This has in fact already appeared in the literature as a filter criterion, originally proposed by Lin and Tang (2006), as Conditional Infomax Feature Extraction (CIFE), though it has been repeatedly rediscovered by other authors (El Akadi et al., 2008; Guo and Nixon, 2009). It is particularly interesting as it represents a sort of 'root' criterion, from which several others can be derived. For example, the link to MIFS can be seen with one further assumption, that the features are pairwise class-conditionally independent.

Assumption 2 *For all features i, j, assume $p(x_i x_j|y) = p(x_i|y)p(x_j|y)$. This states that the features are pairwise class-conditionally independent.*

With this assumption, the term $\sum I(X_j;X_k|Y)$ will be zero, and (15) becomes (11), the MIFS criterion, with $\beta = 1$. The β parameter in MIFS can be interpreted as encoding a strength of belief in another assumption, that of unconditional independence.

Assumption 3 *For all features i, j, assume $p(x_i x_j) = p(x_i)p(x_j)$. This states that the features are pairwise independent.*

A β close to zero implies very strong belief in the independence statement, indicating that any measured association $I(X_j;X_k)$ is in fact spurious, possibly due to noise in the data. A β value closer to 1 implies a lesser belief, that any measured dependency $I(X_j;X_k)$ should be incorporated into the feature score exactly as observed. Since MIM is produced by setting $\beta = 0$, we can see that MIM also adopts Assumption 3. The same line of reasoning can be applied to a very similar criterion proposed by Peng et al. (2005), the *Minimum-Redundancy Maximum-Relevance* criterion,

$$J_{mrmr}(X_k) = I(X_k;Y) - \frac{1}{|S|}\sum_{j \in S} I(X_k;X_j).$$

Since mRMR omits the conditional redundancy term entirely, it is implicitly using Assumption 2. The β coefficient has been set inversely proportional to the size of the current feature set. If we have a large set S, then β will be extremely small. The interpretation is then that as the set S grows, mRMR adopts a stronger belief in Assumption 3. In the original paper, (Peng et al., 2005, Section 2.3) it was claimed that mRMR is equivalent to (10). In this section, through making explicit the intrinsic assumptions of the criterion, we have clearly illustrated that this claim is incorrect.

Balagani and Phoha (2010) present an analysis of the three criteria mRMR, MIFS and CIFE, arriving at similar results to our own: that these criteria make highly restrictive assumptions on the underlying data distributions. Though the conclusions are similar, our approach includes their results as a special case, and makes explicit the link to a likelihood function.

The relation of the MIFS/mRMR to Equation (15) is relatively straightforward. It is more challenging to consider how closely other criteria might be re-expressed in this form. Yang and Moody (1999) propose using *Joint Mutual Information* (JMI),

$$J_{jmi}(X_k) = \sum_{j \in S} I(X_k X_j;Y). \tag{16}$$

Using some relatively simple manipulations (see appendix) this can be re-written as,

$$J_{jmi}(X_k) = I(X_k;Y) - \frac{1}{|S|}\sum_{j \in S}\Big[I(X_k;X_j) - I(X_k;X_j|Y)\Big]. \tag{17}$$

This criterion (17) returns *exactly* the same set of features as the JMI criterion (16); however in this form, we can see the relation to our proposed framework. The JMI criterion, like mRMR, has a stronger belief in the pairwise independence assumptions as the feature set S grows. Similarities can of course be observed between JMI, MIFS and mRMR—the differences being the scaling factor and the conditional term—and their subsequent relation to Equation (15). It is in fact possible to identify numerous criteria from the literature that can all be re-written into a common form, corresponding to variations upon (15). A *space* of potential criteria can be imagined, where we parameterize criterion (15) as so:

$$J'_{cmi} = I(X_k;Y) - \beta \sum_{j\in S} I(X_j;X_k) + \gamma \sum_{j\in S} I(X_j;X_k|Y). \tag{18}$$

Figure 2 shows how the criteria we have discussed so far can all be fitted inside this unit square corresponding to β/γ parameters. MIFS sits on the left hand axis of the square—with $\gamma = 0$ and $\beta \in [0,1]$. The MIM criterion, Equation (1), which simply assesses each feature individually without any regard of others, sits at the bottom left, with $\gamma = 0, \beta = 0$. The top right of the square corresponds to $\gamma = 1, \beta = 1$, which is the CIFE criterion (Lin and Tang, 2006), also suggested by El Akadi et al. (2008) and Guo and Nixon (2009). A very similar criterion, using an assumption to approximate the terms, was proposed by Cheng et al. (2011).

The JMI and mRMR criteria are unique in that they *move linearly* within the space as the feature set S grows. As the size of the set S increases they move closer towards the origin and the MIM criterion. The particularly interesting point about this property is that the *relative magnitude* of the relevancy term to the redundancy terms stays approximately constant as S grows, whereas with MIFS, the redundancy term will in general be $|S|$ times bigger than the relevancy term. The consequences of this will be explored in the experimental section of this paper. Any criterion expressible in the unit square has made independence Assumption 1. In addition, any criteria that sit at points other than $\beta = 1, \gamma = 1$ have adopted varying degrees of belief in Assumptions 2 and 3.

A further interesting point about this square is simply that it is sparsely populated. An obvious unexplored region is the bottom right, the corner corresponding to $\beta = 0, \gamma = 1$; though there is no clear intuitive justification for this point, for completeness in the experimental section we will evaluate it, as the *conditional redundancy* or ‘condred’ criterion. In previous work (Brown, 2009) we explored this unit square, though derived from an expansion of the mutual information function rather than directly from the conditional likelihood. While this resulted in an identical expression to (18), the probabilistic framework we present here is far more expressive, allowing exact specification of the underlying assumptions.

The unit square of Figure 2 describes *linear* criteria, named as so since they are linear combinations of the relevance/redundancy terms. There exist other criteria that follow a similar form, but involving other operations, making them *non-linear*.

4.2 Criteria as Non-Linear Combinations of Shannon Information Terms

Fleuret (2004) proposed the *Conditional Mutual Information Maximization* criterion,

$$J_{cmim}(X_k) = \min_{X_j\in S} \Big[I(X_k;Y|X_j) \Big].$$

This can be re-written,

$$J_{cmim}(X_k) = I(X_k;Y) - \max_{X_j\in S} \Big[I(X_k;X_j) - I(X_k;X_j|Y) \Big]. \tag{19}$$

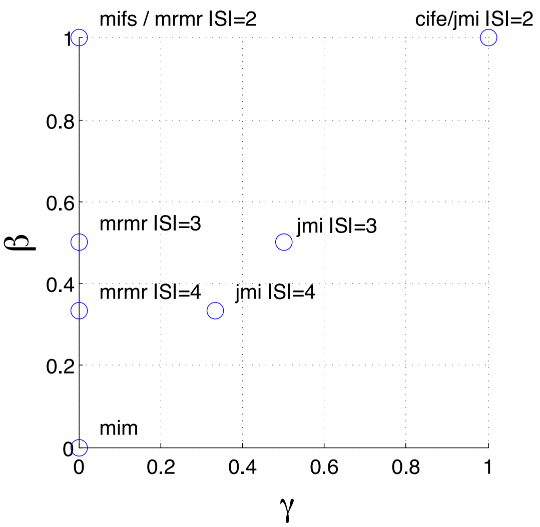


Figure 2: The full space of *linear* filter criteria, describing several examples from Table 1. Note that *all* criteria in this space adopt Assumption 1. Additionally, the γ and β axes represent the criteria belief in Assumptions 2 and 3, respectively. The left hand axis is where the mRMR and MIFS algorithms sit. The bottom left corner, MIM, is the assumption of completely independent features, using just marginal mutual information. Note that some criteria are equivalent at particular sizes of the current feature set $|S|$.

The proof is again available in the appendix. Due to the *max* operator, the probabilistic interpretation is a little less straightforward. It is clear however that CMIM adopts Assumption 1, since it evaluates only pairwise feature statistics.

Vidal-Naquet and Ullman (2003) propose another criterion used in Computer Vision, which we refer to as *Informative Fragments*,

$$J_{if}(X_k) = \min_{X_j \in S} \Big[I(X_k X_j; Y) - I(X_j; Y) \Big].$$

The authors motivate this criterion by noting that it measures the gain of combining a new feature X_k with each existing feature X_j, over simply using X_j by itself. The X_j with the least 'gain' from being paired with X_k is taken as the score for X_k. Interestingly, using the chain rule $I(X_k X_j; Y) = I(X_j; Y) + I(X_k; Y|X_j)$, therefore IF is equivalent to CMIM, that is, $J_{if}(X_k) = J_{cmim}(X_k)$, making the same assumptions. Jakulin (2005) proposed the criterion,

$$J_{icap}(X_k) = I(X_k; Y) - \sum_{X_j \in S} \max \Big[0, \{ I(X_k; X_j) - I(X_k; X_j|Y) \} \Big].$$

Again, this adopts Assumption 1, using the same redundancy and *conditional* redundancy terms, yet the exact probabilistic interpretation is unclear.

An interesting class of criteria use a normalisation term on the mutual information to offset the inherent bias toward high arity features (Duch, 2006). An example of this is *Double Input*

Symmetrical Relevance (Meyer and Bontempi, 2006), a modification of the JMI criterion:

$$J_{disr}(X_k) = \sum_{X_j \in S} \frac{I(X_k X_j ; Y)}{H(X_k X_j Y)}.$$

The inclusion of this normalisation term breaks the strong theoretical link to a likelihood function, but again for completeness we will include this in our empirical investigations. While the criteria in the unit square can have their probabilistic assumptions made explicit, the nonlinearity in the CMIM, ICAP and DISR criteria make such an interpretation far more difficult.

4.3 Summary of Theoretical Findings

In this section we have shown that numerous criteria published over the past two decades of research can be 'retro-fitted' into the framework we have proposed—the criteria are approximations to (10), each making different assumptions on the underlying distributions. Since in the previous section we saw that accepting the top ranked feature according to (10) provides the maximum possible increase in the likelihood, we see now that the criteria are *approximate* maximisers of the likelihood. Whether or not they indeed provide the maximum increase at each step will depend on how well the implicit assumptions on the data can be trusted. Also, it should be remembered that even if we used (10), it is not guaranteed to find the global optimum of the likelihood, since (a) it is a greedy search, and (b) finite data will mean distributions cannot be accurately modelled. In this case, we have reached the limit of what a theoretical analysis can tell us about the criteria, and we must close the remaining 'gaps' in our understanding with an experimental study.

5. Experiments

In this section we empirically evaluate some of the criteria in the literature against one another. Note that we are not pursuing an exhaustive analysis, attempting to identify the 'winning' criterion that provides best performance overall[4]—rather, we primarily observe how the theoretical properties of criteria relate to the similarity of the returned feature sets. While these properties are interesting, we of course must acknowledge that classification performance is the ultimate evaluation of a criterion—hence we also include here classification results on UCI data sets and in Section 6 on the well-known benchmark NIPS Feature Selection Challenge.

In the following sections, we ask the questions: "how stable is a criterion to small changes in the training data set?", "how similar are the criteria to each other?", "how do the different criteria behave in limited and extreme small-sample situations?", and finally, "what is the relation between stability and accuracy?".

To address these questions, we use the 15 data sets detailed in Table 2. These are chosen to have a wide variety of example-feature ratios, and a range of multi-class problems. The features within each data set have a variety of characteristics—some binary/discrete, and some continuous. Continuous features were discretized, using an equal-width strategy into 5 bins, while features already with a categorical range were left untouched. The 'ratio' statistic quoted in the final column is an indicator of the difficulty of the feature selection for each data set. This uses the number of datapoints (N), the median arity of the features (m), and the number of classes (c)—the ratio quoted in

[4] In any case, the No Free Lunch Theorem applies here also (Tsamardinos and Aliferis, 2003).

the table for each data set is $\frac{N}{mc}$, hence a smaller value indicates a more challenging feature selection problem.

A key point of this work is to understand the statistical assumptions on the data imposed by the feature selection criteria—if our classification model were to make even more assumptions, this is likely to obscure the experimental observations relating performance to theoretical properties. For this reason, in all experiments we use a simple nearest neighbour classifier ($k = 3$), this is chosen as it makes few (if any) assumptions about the data, and we avoid the need for parameter tuning. For the feature selection search procedure, the filter criteria are applied using a simple forward selection, to select a fixed number of features, specified in each experiment, before being used with the classifier.

Data	*Features*	*Examples*	*Classes*	Ratio
breast	30	569	2	57
congress	16	435	2	72
heart	13	270	2	34
ionosphere	34	351	2	35
krvskp	36	3196	2	799
landsat	36	6435	6	214
lungcancer	56	32	3	4
parkinsons	22	195	2	20
semeion	256	1593	10	80
sonar	60	208	2	21
soybeansmall	35	47	4	6
spect	22	267	2	67
splice	60	3175	3	265
waveform	40	5000	3	333
wine	13	178	3	12

Table 2: Data sets used in experiments. The final column indicates the difficulty of the data in feature selection, a smaller value indicating a more challenging problem.

5.1 How Stable are the Criteria to Small Changes in the Data?

The set of features selected by any procedure will of course depend on the data provided. It is a plausible complaint if the set of returned features varies wildly with only slight variations in the supplied data. This is an issue reminiscent of the *bias-variance dilemma*, where the sensitivity of a classifier to its initial conditions causes high variance responses. However, while the bias-variance decomposition is well-defined and understood, the corresponding issue for feature selection, the 'stability', has only recently been studied. The stability of a feature selection criterion requires a measure to quantify the 'similarity' between two selected feature sets. This was first discussed by Kalousis et al. (2007), who investigated several measures, with the final recommendation being the Tanimoto distance between sets. Such set-intersection measures seem appropriate, but have limitations; for example, if two criteria selected identical feature sets of size 10, we might be less surprised if we knew the overall pool of features was of size 12, than if it was size 12,000. To account for this, Kuncheva (2007) presents a *consistency index*, based on the hypergeometric distribution with a correction for chance.

Definition 7 *The consistency for two subsets* $A,B \subset X$*, such that* $|A| = |B| = k$*, and* $r = |A \cap B|$*, where* $0 < k < |X| = n$*, is*

$$C(A,B) = \frac{rn - k^2}{k(n-k)}.$$

The consistency takes values in the range $[-1,+1]$, with a positive value indicating similar sets, a zero value indicating a purely random relation, and a negative value indicating a strong anti-correlation between the features sets.

One problem with the consistency index is that it does not take feature *redundancy* into account. That is, two procedures could select features which have different array indices, so are identified as ‘different’, but in fact are so highly correlated that they are effectively identical. A method to deal with this situation was proposed by Yu et al. (2008). This method constructs a weighted complete bipartite graph, where the two node sets correspond to two different feature sets, and weights are assigned to the arcs are the normalized mutual information between the features at the nodes, also sometimes referred to as the symmetrical uncertainty. The weight between node i in set A, and node j in set B, is

$$w(A(i),B(j)) = \frac{I(X_{A(i)};X_{B(j)})}{H(X_{A(i)}) + H(X_{B(j)})}.$$

The Hungarian algorithm is then applied to identify the maximum weighted matching between the two node sets, and the overall similarity between sets A and B is the final matching cost. This is the *information consistency* of the two sets. For more details, we refer to Yu et al. (2008).

We now compare these two measures on the criteria from the previous sections. For each data set, we take a bootstrap sample and select a set of features using each feature selection criterion. The (information) stability of a single criterion is quantified as the average pairwise (information) consistency across 50 bootstraps from the training data.

Figure 3 shows Kuncheva’s stability measure on average over 15 data sets, selecting feature sets of size 10; note that the criteria have been displayed ordered left-to-right by their median value of stability over the 15 data sets. The marginal mutual information, MIM, is as expected the most stable, given that it has the lowest dimensional distribution to approximate. The next most stable is JMI which includes the relevancy/redundancy terms, but *averages* over the current feature set; this averaging process might therefore be interpreted empirically as a form of ‘smoothing’, enabling the criteria overall to be resistant to poor estimation of probability distributions. It can be noted that the far right of Figure 3 consists of the MIFS, ICAP and CIFE criteria, all of which do not attempt to average the redundancy terms.

Figure 4 shows the same data sets, but instead the *information stability* is computed; as mentioned, this should take into account the fact that some features are highly correlated. Interestingly, the two box-plots show broadly similar results. MIM is the most stable, and CIFE is the least stable, though here we see that JMI, DISR, and MRMR are actually more stable than Kuncheva’s stability index can reflect. An interesting line of future research might be to combine the best of these two stability measures—one that can take into account both feature redundancy and a correction for random chance.

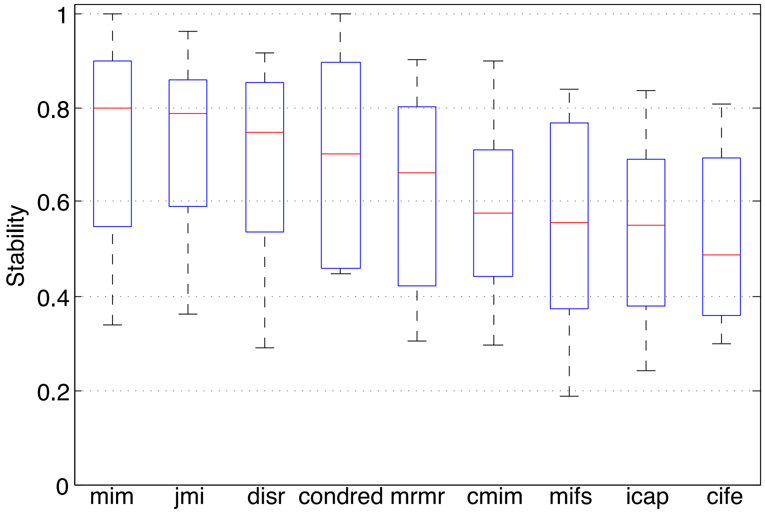


Figure 3: Kuncheva's Stability Index across 15 data sets. The box indicates the upper/lower quartiles, the horizontal line within each shows the median value, while the dotted crossbars indicate the maximum/minimum values. For convenience of interpretation, criteria on the x-axis are ordered by their median value.

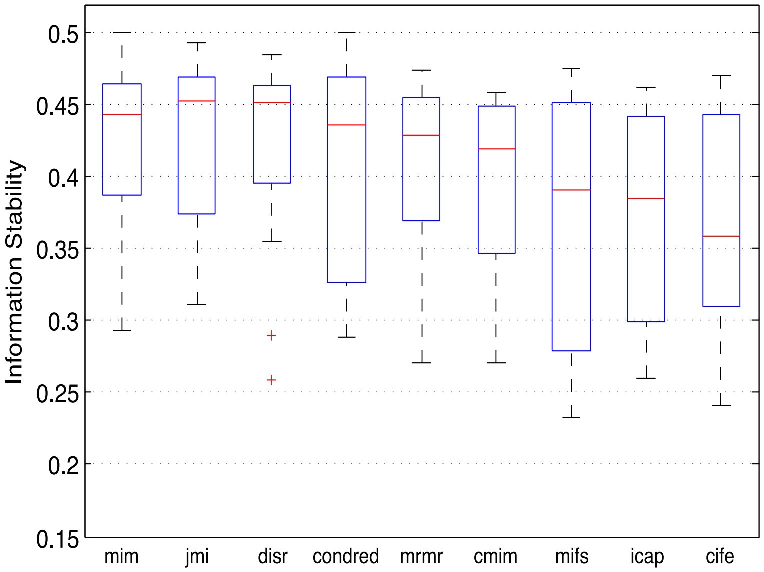
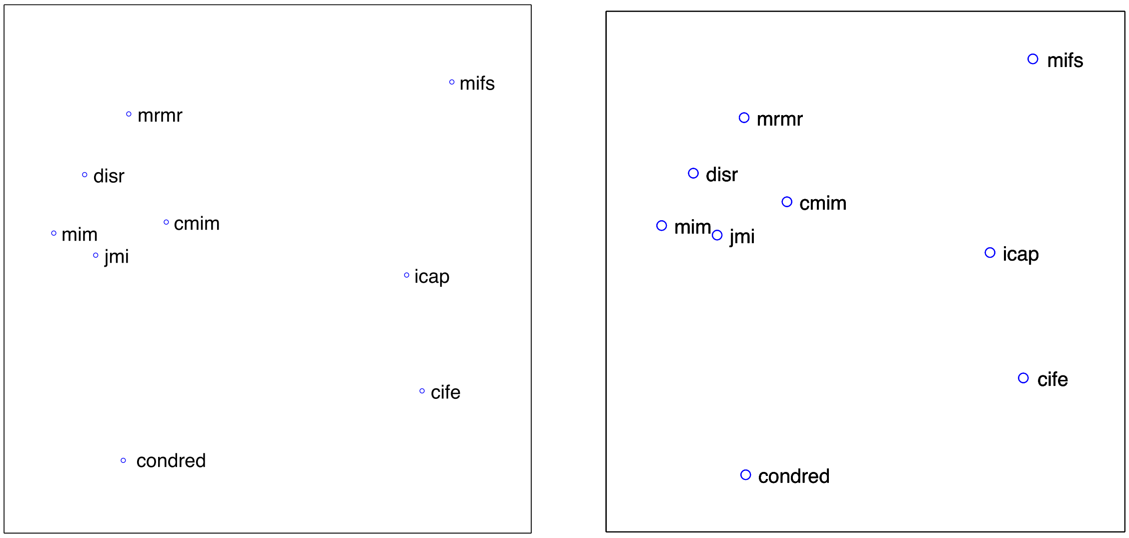


Figure 4: Yu et al's Information Stability Index across 15 data sets. For comparison, criteria on the x-axis are ordered identically to Figure 3. The general picture emerges similarly, though the information stability index is able to take feature redundancy into account, showing that some criteria are slightly more stable than expected.



(a) Kuncheva's Consistency Index.

(b) Yu et al's Information Stability Index.

Figure 5: Relations between feature sets generated by different criteria, on average over 15 data sets. 2-D visualisation generated by classical multi-dimensional scaling.

5.2 How Similar are the Criteria?

Two criteria can be directly compared with the same methodology: by measuring the consistency and information consistency between selected feature subsets on a common set of data. We calculate the mean consistencies between two feature sets of size 10, repeatedly selected over 50 bootstraps from the original data. This is then arranged in a similarity matrix, and we use classical multi-dimensional scaling to visualise this as a 2-d map, shown in Figures 5a and 5b. Note again that while the indices may return different absolute values (one is a normalized mean of a hypergeometric distribution and the other is a pairwise sum of mutual information terms) they show very similar relative 'distances' between criteria.

Both diagrams show a cluster of several criteria, and 4 clear outliers: MIFS, CIFE, ICAP and CondRed. The 5 criteria clustering in the upper left of the space appear to return relatively similar feature sets. The 4 outliers appear to return quite significantly different feature sets, both from the clustered set, and from each other. A common characteristic of these 4 outliers is that they do not scale the redundancy or conditional redundancy information terms. In these criteria, the upper bound on the redundancy term $\sum_{j \in S} I(X_k;X_j)$ grows linearly with the number of selected features, whilst the upper bound on the relevancy term $I(X_k;Y)$ remains constant. When this happens the relevancy term is overwhelmed by the redundancy term and thus the criterion selects features with minimal redundancy, rather than trading off between the two terms. This leads to strongly divergent feature sets being selected, which is reflected in the stability of the criteria. Each of the outliers are different from each other as they have different combinations of redundancy and conditional redundancy. We will see this 'balance' between relevancy and redundancy emerge as a common theme in the experiments over the next few sections.

5.3 How do Criteria Behave in Limited and Extreme Small-sample Situations?

To assess how criteria behave in data poor situations, we vary the number of datapoints supplied to perform the feature selection. The procedure was to randomly select 140 datapoints, then use the remaining data as a hold-out set. From this 140, the number provided to each criterion was increased in steps of 10, from a minimal set of size 20. To allow a reasonable testing set size, we limited this assessment to only data sets with at least 200 datapoints total; this gives us 11 data sets from the 15, omitting *lungcancer*, *parkinsons*, *soybeansmall*, and *wine*. For each data set we select 10 features and apply the 3-nn classifier, recording the rank-order of the criteria in terms of their generalisation error. This process was repeated and averaged over 50 trials, giving the results in Figure 6.

To aid interpretation we label MIM with a simple point marker, MIFS, CIFE, CondRed, and ICAP with a circle, and the remaining criteria (DISR, JMI, mRMR and CMIM) with a star. The criteria labelled with a star balance the relative magnitude of the relevancy and redundancy terms, those with a circle do not attempt to balance them, and MIM contains no redundancy term. There is a clear separation between those criteria with a star outperforming those with a circle, and MIM varying in performance between the two groups as we allow more training datapoints.

Notice that the highest ranked criteria coincide with those in the cluster at the top left of Figures 5a and 5b. We suggest that the relative difference in performance is due to the same reason noted in Section 5.2, that the redundancy term grows with the size of the selected feature set. In this case, the redundancy term eventually grows to outweigh the relevancy by a large degree, and the new features are selected solely on the basis of redundancy, ignoring the relevance, thus leading to poor classification performance.

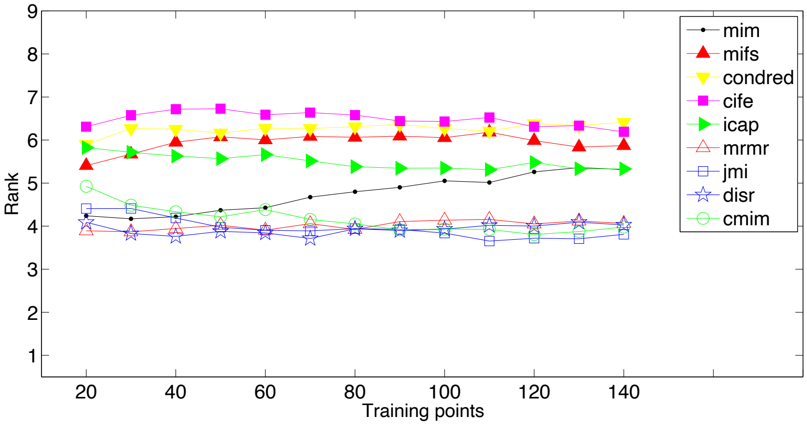


Figure 6: Average ranks of criteria in terms of test error, selecting 10 features, across 11 data sets. Note the clear dominance of criteria which do not allow the redundancy term to overwhelm the relevancy term (unfilled markers) over those that allow redundancy to grow with the size of the feature set (filled markers).

Data	*Features*	*Examples*	*Classes*
Colon	2000	62	2
Leukemia	7070	72	2
Lung	325	73	7
Lymph	4026	96	9
NCI9	9712	60	9

Table 3: Data sets from Peng et al. (2005), used in experiments.

5.4 Extreme Small-Sample Experiments

In the previous sections we discussed two theoretical properties of information-based feature selection criteria: whether it balances the relative magnitude of relevancy against redundancy, and whether it includes a class-conditional redundancy term. Empirically on the UCI data sets, we see that the balancing is far more important than the inclusion of the conditional redundancy term—for example, MRMR succeeds in many cases, while MIFS performs poorly. Now, we consider whether same property may hold in extreme small-sample situations, when the number of examples is so low that reliable estimation of distributions becomes extremely difficult. We use data sourced from Peng et al. (2005), detailed in Table 3. Results are shown in Figure 7, selecting 50 features from each data set and plotting leave-one-out classification error. It should of course be remembered that on such small data sets, making just one additional datapoint error can result in seemingly large changes in accuracy. For example, the difference between the best and worst criteria on Leukemia was just 3 datapoints. In contrast to the UCI results, the picture is less clear. On Colon, the criteria all perform similarly; this is the least complex of all the data sets, having the smallest number of classes with a (relatively) small number of features. As we move through the data sets with increasing numbers of features/classes, we see that MIFS, CONDRED, CIFE and ICAP start to break away, performing poorly compared to the others. Again, we note that these do not attempt to balance relevancy/redundancy. This difference is clearest on the NCI9 data, the most complex with 9 classes and 9712 features. However, as we may expect with such high dimensional and challenging problems, there are some exceptions—the Colon data as mentioned, and also the Lung data where ICAP/MIFS perform well.

5.5 What is the Relation Between Stability and Accuracy?

An important question is whether we can find a good balance between the stability of a criterion and the classification accuracy. This was considered by Gulgezen et al. (2009), who studied the stability/accuracy trade-off for the MRMR criterion. In the following, we consider this trade-off in the context of *Pareto-optimality*, across the 9 criteria, and the 15 data sets from Table 2. Experimental protocol was to take 50 bootstraps from the data set, each time calculating the out-of-bag error using the 3-nn. The stability measure was Kuncheva's stability index calculated from the 50 feature sets, and the accuracy was the mean out-of-bag accuracy across the 50 bootstraps. The experiments were also repeated using the Information Stability measure, revealing almost identical results. Results using Kuncheva's stability index are shown in Figure 8.

The *Pareto-optimal set* is defined as the set of criteria for which no other criterion has both a higher accuracy and a higher stability, hence the members of the Pareto-optimal set are said to be *non-dominated* (Fonseca and Fleming, 1996). Thus, each of the subfigures of Figure 8, criteria

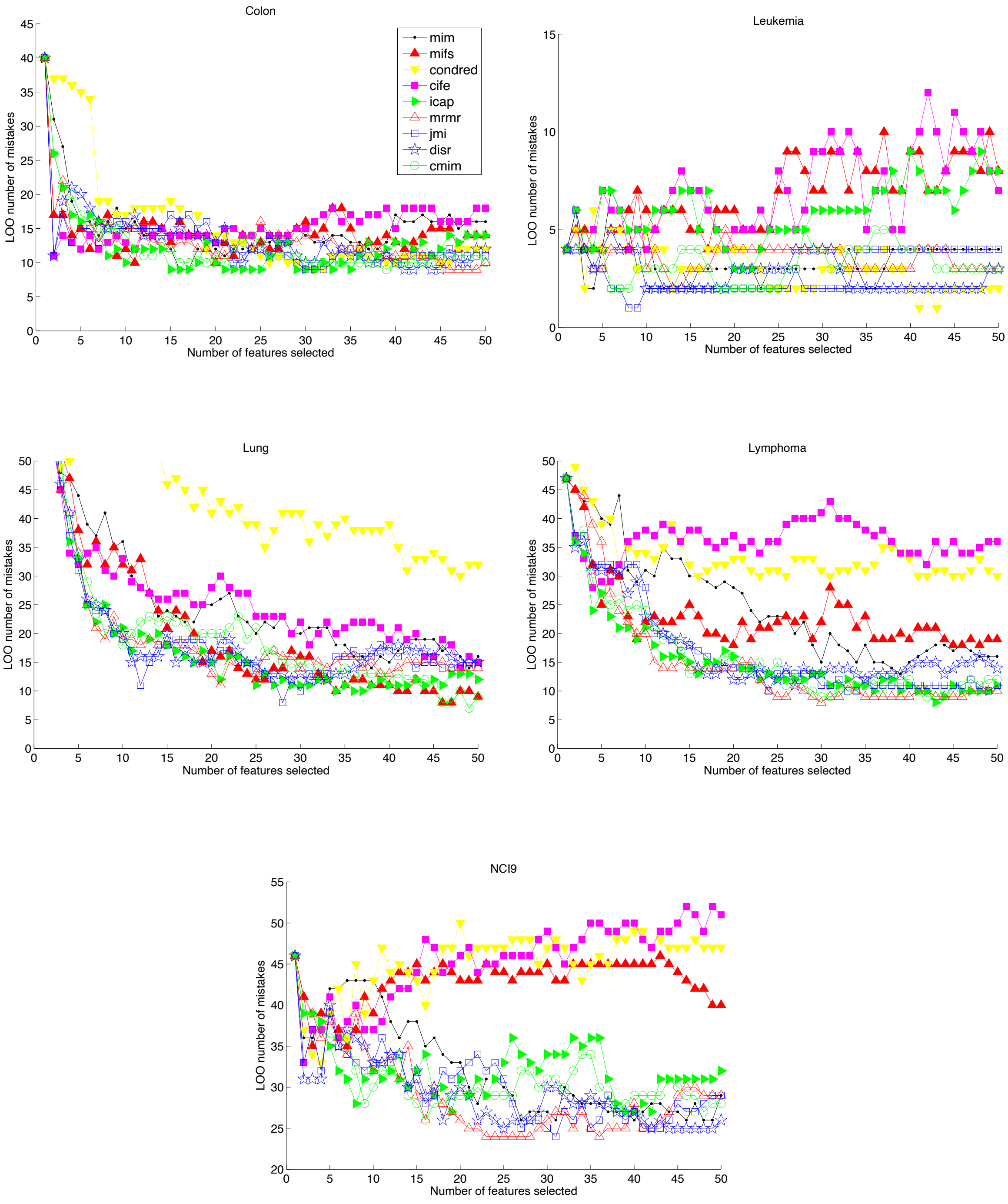


Figure 7: LOO results on Peng's data sets : Colon, Lymphoma, Leukemia, Lung, NCI9.

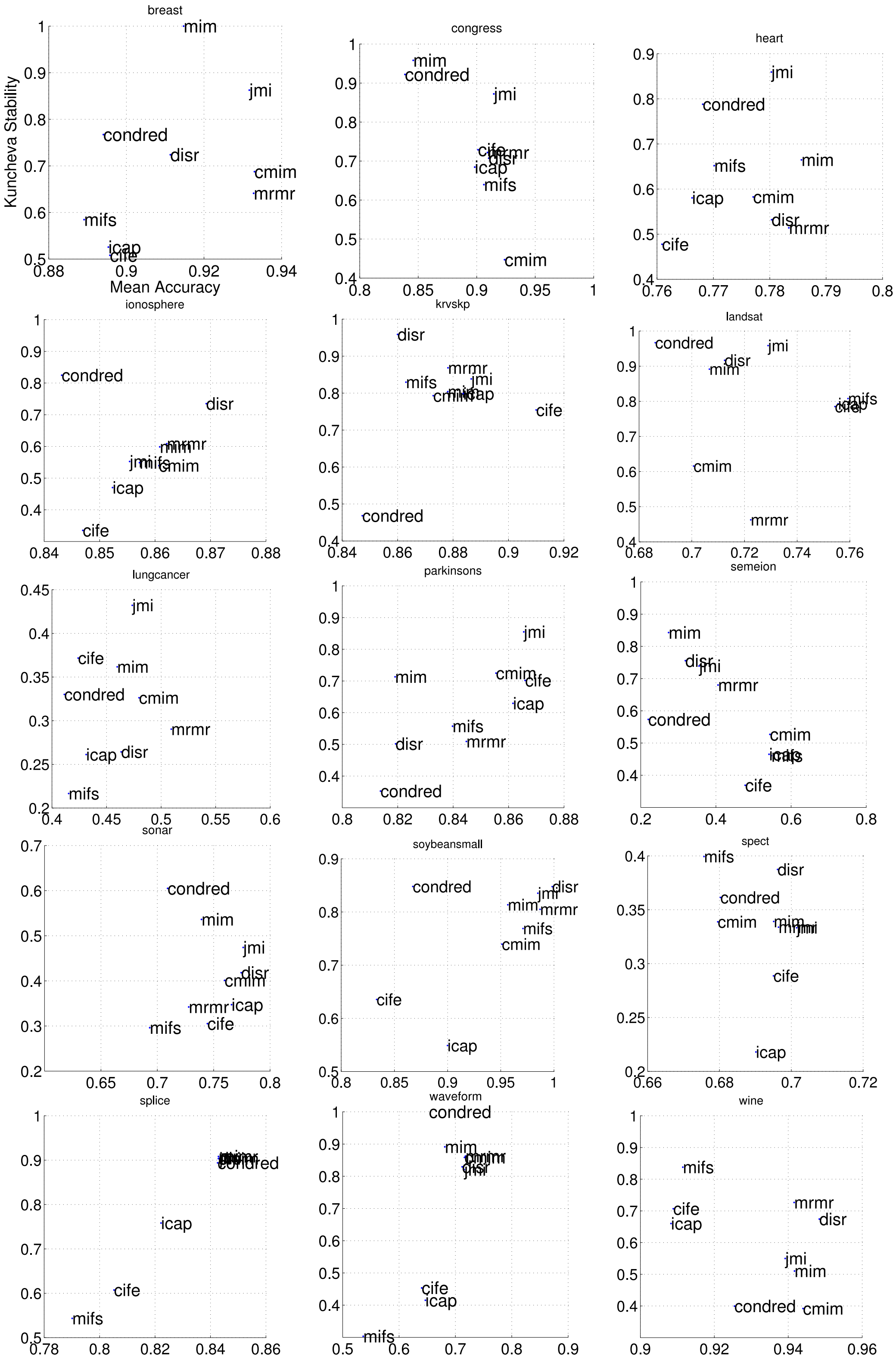


Figure 8: Stabilty (y-axes) versus Accuracy (x-axes) over 50 bootstraps for each of the UCI data sets. The pareto-optimal rankings are summarised in Table 4.

Accuracy/Stability(Yu)	Accuracy/Stability(Kuncheva)	Accuracy
JMI (1.6)	JMI (1.5)	JMI (2.6)
DISR (2.3)	DISR (2.2)	MRMR (3.6)
MIM (2.4)	MIM (2.3)	DISR (3.7)
MRMR (2.5)	MRMR (2.5)	CMIM (4.5)
CMIM (3.3)	CONDRED (3.2)	ICAP (5.3)
ICAP (3.6)	CMIM (3.4)	MIM (5.4)
CONDRED (3.7)	ICAP (4.3)	CIFE (5.9)
CIFE (4.3)	CIFE (4.8)	MIFS (6.5)
MIFS (4.5)	MIFS (4.9)	CONDRED (7.4)

Table 4: *Column 1:* Non-dominated Rank of different criteria for the trade-off of accuracy/stability. Criteria with a higher rank (closer to 1.0) provide a better tradeoff than those with a lower rank. *Column 2:* As column 1 but using Kuncheva's Stability Index. *Column 3:* Average ranks for accuracy alone.

that appear further to the top-right of the space *dominate* those toward the bottom left—in such a situation there is no reason to choose those at the bottom left, since they are dominated on both objectives by other criteria.

A summary (for both stability and information stability) is provided in the first two columns of Table 4, showing the *non-dominated rank* of the different criteria. This is computed per data set as the number of other criteria which dominate a given criterion, in the Pareto-optimal sense, then averaged over the 15 data sets. We can see that these rankings are similar to the results earlier, with MIFS, ICAP, CIFE and CondRed performing poorly. We note that JMI, (which both balances the relevancy and redundancy terms and includes the conditional redundancy) outperforms all other criteria.

We present the average accuracy ranks across the 50 bootstraps in column 3. These are similar to the results from Figure 6 but use a bootstrap of the full data set, rather than a small sample from it. Following Demšar (2006) we analysed these ranks using a Friedman test to determine which criteria are statistically significantly different from each other. We then used a Nemenyi post-hoc test to determine which criteria differed, with statistical significances at 90%, 95%, and 99% confidences. These give a partial ordering for the criteria which we present in Figure 9, showing a *Significant Dominance Partial Order* diagram. Note that this style of diagram encapsulates the same information as a Critical Difference diagram (Demšar, 2006), but allows us to display multiple levels of statistical significance. A bold line connecting two criteria signifies a difference at the 99% confidence level, a dashed line at the 95% level, and a dotted line at the 90% level. Absence of a link signifies that we do not have the statistical power to determine the difference one way or another. Reading Figure 9, we see that with 99% confidence JMI is significantly superior to CondRed, and MIFS, but not statistically significantly different from the other criteria. As we lower our confidence level, more differences appear, for example MRMR and MIFS are only significantly different at the 90% confidence level.

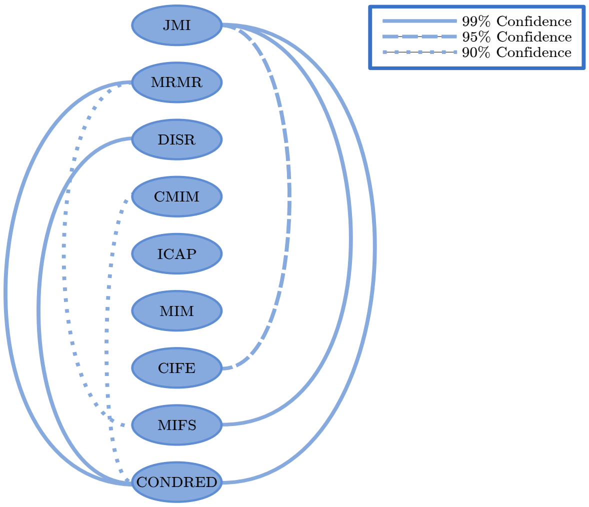


Figure 9: Significant dominance partial-order diagram. Criteria are placed top to bottom in the diagram by their rank taken from column 3 of Table 4. A link joining two criteria means a statistically significant difference is observed with a Nemenyi post-hoc test at the specified confidence level. For example JMI is significantly superior to MIFS ($\beta = 1$) at the 99% confidence level. Note that the absence of a link does not signify the lack of a statistically significant difference, but that the Nemenyi test does not have sufficient power (in terms of number of data sets) to determine the outcome (Demšar, 2006). It is interesting to note that the four bottom ranked criteria correspond to the corners of the unit square in Figure 2; while the top three (JMI/MRMR/DISR) are all very similar, scaling the redundancy terms by the size of the feature set. The middle ranks belong to CMIM/ICAP, which are similar in that they use the min/max strategy instead of a linear combination of terms.

5.6 Summary of Empirical Findings

From experiments in this section, we conclude that the balance of relevancy/redundancy terms is extremely important, while the inclusion of a class conditional term seems to matter less. We find that some criteria are inherently more *stable* than others, and that the trade-off between accuracy (using a simple k-nn classifier) and stability of the feature sets differs between criteria. The best overall trade-off for accuracy/stability was found in the JMI and MRMR criteria. In the following section we re-assess these findings, in the context of two problems posed for the NIPS Feature Selection Challenge.

6. Performance on the NIPS Feature Selection Challenge

In this section we investigate performance of the criteria on data sets taken from the NIPS Feature Selection Challenge (Guyon, 2003).

6.1 Experimental Protocols

We present results using GISETTE (a handwriting recognition task), and MADELON (an artificially generated data set).

Data	*Features*	*Examples (Tr/Val)*	*Classes*
GISETTE	5000	6000/1000	2
MADELON	500	2000/600	2

Table 5: Data sets from the NIPS challenge, used in experiments.

To apply the mutual information criteria, we estimate the necessary distributions using histogram estimators: features were discretized independently into 10 equal width bins, with bin boundaries determined from training data. After the feature selection process the original (undiscretised) data sets were used to classify the validation data. Each criterion was used to generate a ranking for the top 200 features in each data set. We show results using the full top 200 for GISETTE, but only the top 20 for MADELON as after this point all criteria demonstrated severe overfitting. We use the Balanced Error Rate, for fair comparison with previously published work on the NIPS data sets. We accept that this does not necessarily share the same optima as the classification error (to which the conditional likelihood relates), and leave investigations of this to future work.

Validation data results are presented in Figure 10 (GISETTE) and Figure 11 (MADELON). The minimum of the validation error was used to select the best performing feature set size, the training data alone used to classify the testing data, and finally test labels were submitted to the challenge website. Test results are provided in Table 6 for GISETTE, and Table 7 for MADELON.[5]

Unlike in Section 5, the data sets we have used from the NIPS Feature Selection Challenge have a greater number of datapoints (GISETTE has 6000 training examples, MADELON has 2000) and thus we can present results using a direct implementation of Equation (10) as a criterion. We refer to this criterion as CMI, as it is using the conditional mutual information to score features. Unfortunately there are still estimation errors in this calculation when selecting a large number of

[5] We do not provide classification confidences as we used a nearest neighbour classifier and thus the AUC is equal to $1-$ BER.

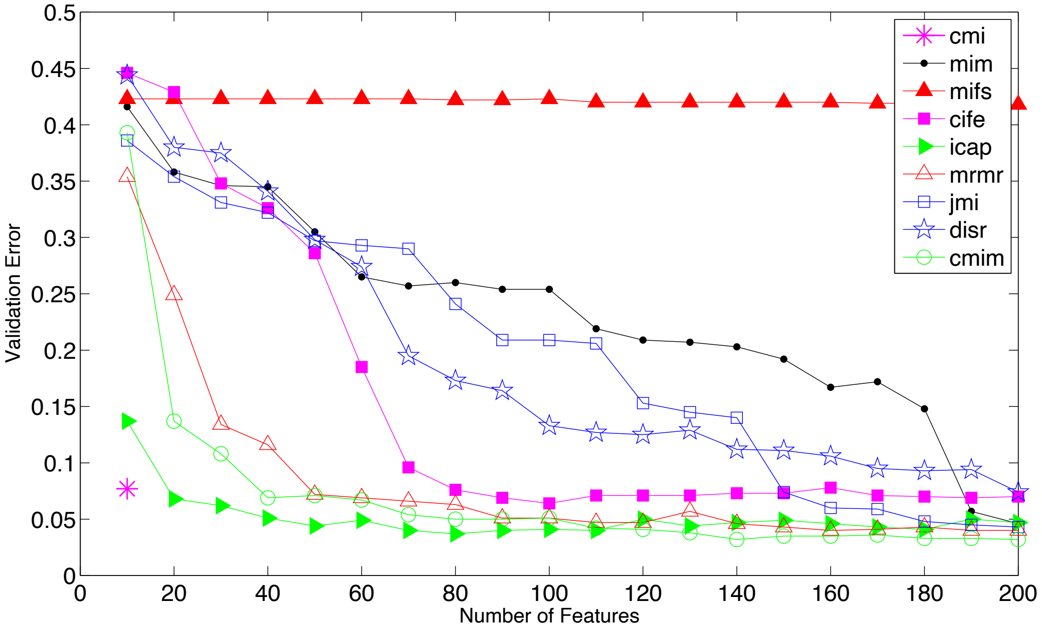


Figure 10: Validation Error curve using GISETTE.

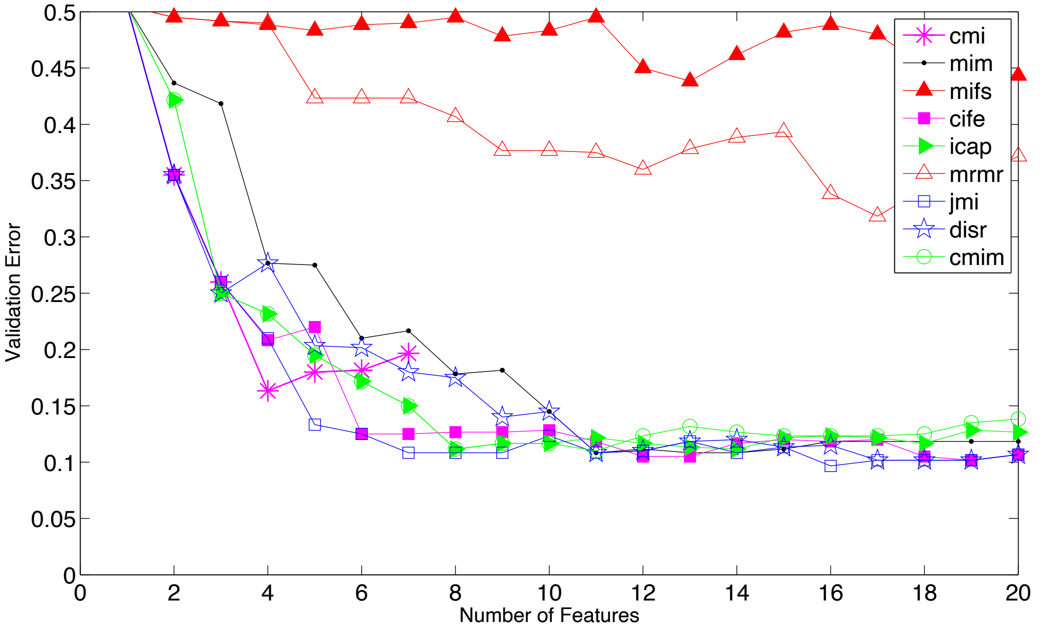


Figure 11: Validation Error curve using MADELON.

features, even given the large number of datapoints and so the criterion fails to select features after a certain point, as each feature appears equally irrelevant. In GISETTE, CMI selected 13 features, and so the top 10 features were used and thus one result is shown. In MADELON, CMI selected 7 features and so 7 results are shown.

6.2 Results on Test Data

In Table 6 there are several distinctions between the criteria, the most striking of which is the failure of MIFS to select an informative feature set. The importance of balancing the magnitude of the relevancy and the redundancy can be seen whilst looking at the other criteria in this test. Those criteria which balance the magnitudes, (CMIM, JMI, & mRMR) perform better than those which do not (ICAP,CIFE). The DISR criterion forms an outlier here as it performs poorly when compared to JMI. The only difference between these two criteria is the normalization in DISR—as such, this is the likely cause of the observed poor performance, the introduction of more variance by estimating the normalization $H(X_kX_jY)$.

We can also see how important the low dimensional approximation is, as even with 6000 training examples CMI cannot estimate the required joint distribution to avoid selecting probes, despite being a direct iterative maximisation of the conditional likelihood in the limit of datapoints.

Criterion	*BER*	*AUC*	*Features (%)*	*Probes (%)*
MIM	4.18	95.82	4.00	0.00
MIFS	42.00	58.00	4.00	58.50
CIFE	6.85	93.15	2.00	0.00
ICAP	4.17	95.83	1.60	0.00
CMIM	**2.86**	**97.14**	**2.80**	**0.00**
CMI	8.06	91.94	0.20	20.00
mRMR	2.94	97.06	3.20	0.00
JMI	3.51	96.49	4.00	0.00
DISR	8.03	91.97	4.00	0.00
Winning Challenge Entry	**1.35**	**98.71**	**18.3**	**0.0**

Table 6: NIPS FS Challenge Results: GISETTE.

The MADELON results (Table 7) show a particularly interesting point—the top performers (in terms of BER) are JMI and CIFE. Both these criteria include the class-conditional redundancy term, but CIFE does not balance the influence of relevancy against redundancy. In this case, it appears the 'balancing' issue, so important in our previous experiments seems to have little importance—instead, the presence of the conditional redundancy term is the differentiating factor between criteria (note the poor performance of MIFS/MRMR). This is perhaps not surprising given the nature of the MADELON data, constructed precisely to require features to be evaluated jointly.

It is interesting to note that the challenge organisers benchmarked a 3-NN using the optimal feature set, achieving a 10% test error (Guyon, 2003). Many of the criteria managed to select feature sets which achieved a similar error rate using a 3-NN, and it is likely that a more sophisticated classifier is required to further improve performance.

This concludes our experimental study—in the following, we make further links to the literature for the theoretical framework, and discuss implications for future work.

Criterion	*BER*	*AUC*	*Features (%)*	*Probes (%)*
MIM	10.78	89.22	2.20	0.00
MIFS	46.06	53.94	2.60	92.31
CIFE	**9.50**	**90.50**	**3.80**	**0.00**
ICAP	11.11	88.89	1.60	0.00
CMIM	11.83	88.17	2.20	0.00
CMI	21.39	78.61	0.80	0.00
mRMR	35.83	64.17	3.40	82.35
JMI	**9.50**	**90.50**	**3.20**	**0.00**
DISR	9.56	90.44	3.40	0.00
Winning Challenge Entry	**7.11**	**96.95**	**1.6**	**0.0**

Table 7: NIPS FS Challenge Results: MADELON.

7. Related Work: Strong and Weak Relevance

Kohavi and John (1997) proposed definitions of *strong* and *weak* feature relevance. The definitions are formed from statements about the conditional probability distributions of the variables involved. We can re-state the definitions of Kohavi and John (hereafter KJ) in terms of mutual information, and see how they can fit into our conditional likelihood maximisation framework. In the notation below, notation X_i indicates the ith feature in the overall set X, and notation $X_{\backslash i}$ indicates the set $\{X \backslash X_i\}$, all features *except* the ith.

Definition 8 : Strongly Relevant Feature (Kohavi and John, 1997)
Feature X_i is strongly relevant *to Y iff there exists an assignment of values x_i, y, $x_{\backslash i}$ for which $p(X_i = x_i, X_{\backslash i} = x_{\backslash i}) > 0$ and $p(Y = y|X_i = x_i, X_{\backslash i} = x_{\backslash i}) \neq p(Y = y|X_{\backslash i} = x_{\backslash i})$.*

Corollary 9 *A feature X_i is* strongly relevant *iff $I(X_i;Y|X_{\backslash i}) > 0$.*

Proof The KL divergence $D_{KL}(p(y|xz) \mid\mid p(y|z)) > 0$ iff $p(y|xz) \neq p(y|z)$ for some assignment of values x, y, z. A simple re-application of the manipulations leading to Equation (5) demonstrates that the expected KL-divergence $E_{xz}\{p(y|xz)||p(y|z)\}$ is equal to the mutual information $I(X;Y|Z)$. In the definition of strong relevance, if there exists a single assignment of values $x_i, y, x_{\backslash i}$ that satisfies the inequality, then $E_x\{p(y|x_i x_{\backslash i})||p(y|x_{\backslash i})\} > 0$ and therefore $I(X_i;Y|X_{\backslash i}) > 0$. ■

Given the framework we have presented, we can note that this strong relevance comes from a combination of *three terms*,

$$I(X_i;Y|X_{\backslash i}) = I(X_i;Y) - I(X_i;X_{\backslash i}) + I(X_i;X_{\backslash i}|Y).$$

This view of strong relevance demonstrates explicitly that a feature may be individually irrelevant (i.e., $p(y|x_i) = p(y)$ and thus $I(X_i;Y) = 0$), but still strongly relevant if $I(X_i;X_{\backslash i}|Y) - I(X_i;X_{\backslash i}) > 0$.

Definition 10 : Weakly Relevant Feature (Kohavi and John, 1997)
Feature X_i is weakly relevant *to Y iff it is not strongly relevant and there exists a subset $Z \subset X_{\backslash i}$, and an assignment of values x_i, y, z for which $p(X_i = x_i, Z = z) > 0$ such that $p(Y = y|X_i = x_i, Z = z) \neq p(Y = y|Z = z)$.*

Corollary 11 *A feature X_i is weakly relevant to Y iff it is not strongly relevant and $I(X_i;Y|Z) > 0$ for some $Z \subset X_{\backslash i}$.*

Proof This follows immediately from the proof for the strong relevance above. ■

It is interesting, and somewhat non-intuitive, that there can be cases where there are *no* strongly relevant features, but *all* are weakly relevant. This will occur for example in a data set where all features have exact duplicates: we have $2M$ features and $\forall i,\ X_{M+i} = X_i$. In this case, for any X_k (such that $k < M$) we will have $I(X_k;Y|X_{\backslash i}) = 0$ since its duplicate feature X_{M+k} will carry the same information. In this case, for any feature X_k (such that $k < M$) that is strongly relevant in the data set $\{X_1,...,X_M\}$, it is *weakly* relevant in the data set $\{X_1,...,X_{2M}\}$.

This issue can be dealt with by refining our definition of relevance with respect to a subset of the full feature space. A particular subset about which we have some information is the currently selected set S. We can relate our framework to KJ's definitions in this context. Following KJ's formulations,

Definition 12 : Relevance with respect to the current set S.
Feature X_i is relevant *to Y with respect to S iff there exists an assignment of values x_i, y, s for which $p(X_i = x_i, S = s) > 0$ and $p(Y = y|X_i = x_i, S = s) \neq p(Y = y|S = s)$.*

Corollary 13 *Feature X_i is* relevant *to Y with respect to S, iff $I(X_i;Y|S) > 0$.*

A feature that is relevant with respect to S is either strongly or weakly relevant (in the KJ sense) but it is not possible to determine in which class it lies, as we have not conditioned on $X_{\backslash i}$. Notice that the definition coincides exactly with the forward selection heuristic (Definition 2), which we have shown is a hill-climber on the conditional likelihood. As a result, we see *that hill-climbing on the conditional likelihood corresponds to adding the* most *relevant feature with respect to the current set S*. Again we re-emphasize, that the resultant gain in the likelihood comes from a combination of *three sources*:

$$I(X_i;Y|S) = I(X_i;Y) - I(X_i;S) + I(X_i;S|Y).$$

It could easily be the case that $I(X_i;Y) = 0$, that is a feature is entirely irrelevant when considered on its own—but the sum of the two redundancy terms results in a positive value for $I(X_i;Y|S)$. We see that if a criterion does not attempt to model both of the redundancy terms, even if only using low dimensional approximations, it runs the risk of evaluating the relevance of X_i incorrectly.

Definition 14 : Irrelevance with respect to the current set S.
Feature X_i is irrelevant *to Y with respect to S iff $\forall$ x_i, y, s for which $p(X_i = x_i, S = s) > 0$ and $p(Y = y|X_i = x_i, S = s) = p(Y = y|S = s)$.*

Corollary 15 *Feature X_i is* irrelevant *to Y with respect to S, iff $I(X_i;Y|S) = 0$.*

In a forward step, if a feature X_i is irrelevant with respect to S, adding it alone to *S will not increase the conditional likelihood*. However, there may be further additions to S in the future, giving us a selected set S'; we may then find that X_i is then *relevant* with respect to S'. In a backward step we check whether a feature is irrelevant with respect to $\{S \backslash X_i\}$, using the test $I(X_i;Y|\{S \backslash X_i\}) = 0$. In this case, removing this feature *will not decrease the conditional likelihood*.

8. Related Work: Structure Learning in Bayesian Networks

The framework we have described also serves to highlight a number of important links to the literature on structure learning of directed acyclic graphical (DAG) models (Korb, 2011). The problem of DAG learning from observed data is known to be NP-hard (Chickering et al., 2004), and as such there exist two main families of approximate algorithms. *Metric* or *Score-and-Search* learners construct a graph by searching the space of DAGs directly, assigning a score to each based on properties of the graph in relation to the observed data; probably the most well-known score is the BIC measure (Korb, 2011). However, the space of DAGs is superexponential in the number of variables, and hence an exhaustive search rapidly becomes computationally infeasible. Grossman and Domingos (2004) proposed a greedy hill-climbing search over structures, using conditional likelihood as a scoring criterion. Their work found significant advantage from using this 'discriminative' learning objective, as opposed to the traditional 'generative' joint likelihood. The potential of this discriminative model perspective will be expanded upon in Section 9.3.

Constraint learners approach the problem from a constructivist point of view, adding and removing arcs from a single DAG according to conditional independence tests given the data. When the candidate DAG passes all conditional independence statements observed in the data, it is considered to be a good model. In the current paper, for a feature to be eligible for inclusion, we required that $I(X_k;Y|S) > 0$. This is equivalent to a conditional independence test $X_k \not\perp\!\!\!\perp Y \mid S$. One well-known problem with constraint learners is that if a test gives an incorrect result, the error can 'cascade', causing the algorithm to draw further incorrect conclusions on the network structure. This problem is also true of the popular greedy-search heuristics that we have described in this work.

In Section 3.2, we showed that Markov Blanket algorithms (Tsamardinos et al., 2003) are an example of the framework we propose. Specifically, the solution to Equation (7) is a (possibly non-unique) Markov Blanket, and the solution to Equation (8) is exactly the Markov *boundary*, that is, a minimal, unique blanket. It is interesting to note that these algorithms, which are a restricted class of structure learners, assume *faithfulness* of the data distribution. We can see straightforwardly that all criteria we have considered, when combined with a greedy forward selection, also make this assumption.

9. Conclusion

This work has presented a unifying framework for information theoretic feature selection, bringing almost two decades of research on heuristic scoring criteria under a single theoretical interpretation. This is achieved via a novel interpretation of information theoretic feature selection as *an optimization of the conditional likelihood*—this is in contrast to the current view of mutual information, as a heuristic measure of feature relevancy.

9.1 Summary of Contributions

In Section 3 we showed how to decompose the conditional likelihood into three terms, each with their own interpretation in relation to the feature selection problem. One of these emerges as a *conditional mutual information*. This observation allows us to answer the following question:

What are the implicit statistical assumptions of mutual information criteria? The investigations have revealed that the various criteria published over the past two decades are all *approximate iterative maximisers of the conditional likelihood*. The approximations are due to implicit assumptions on the data distribution: some are more restrictive than others, and are detailed in Section 4. The approximations, while heuristic, are necessary due to the need to estimate high dimensional probability distributions. The popular Markov Blanket learning algorithm IAMB is included in this class of procedures, hence can also bee seen as an iterative maximiser of the conditional likelihood.

The main differences between criteria are whether they include a *class-conditional* term, and whether they provide a mechanism to *balance* the relative size of the redundancy terms against the relevancy term. To ascertain how these differences impact the criteria in practice, we conducted an empirical study of 9 different heuristic mutual information criteria across 22 data sets. We analyzed how the criteria behave in large/small sample situations, how the stability of returned feature sets varies between criteria, and how similar criteria are in the feature sets they return. In particular, the following questions were investigated:

How do the theoretical properties translate to classifier accuracy? Summarising the performance of the criteria under the above conditions, including the class-conditional term is *not* always necessary. Various criteria, for example MRMR, are successful without this term. However, without this term criteria are blind to certain classes of problems, for example, the MADELON data set, and will perform poorly in these cases. Balancing the relevancy and redundancy terms is however *extremely* important—criteria like MIFS, or CIFE, that allow redundancy to swamp relevancy, are ranked lowest for accuracy in almost all experiments. In addition, this imbalance tends to cause large instability in the returned feature sets—being highly sensitive to the supplied data.

How stable are the criteria to small changes in the data? Several criteria return wildly different feature sets with just small changes in the data, while others return similar sets each time, hence are 'stable' procedures. The most stable was the univariate mutual information, followed closely by JMI (Yang and Moody, 1999; Meyer et al., 2008); while among the least stable are MIFS (Battiti, 1994) and ICAP (Jakulin, 2005). As visualised by multi-dimensional scaling in Figure 5, several criteria appear to return quite similar sets, while there are some outliers.

How do criteria behave in limited and extreme small-sample situations? In extreme small-sample situations, it appears the above rules (regarding the conditional term and the balancing of relevancy-redundancy) can be broken—the poor estimation of distributions means the theoretical properties do not translate immediately to performance.

9.2 Advice for the Practitioner

From our investigations we have identified three desirable characteristics of an information based selection criterion. The first is whether it includes reference to a conditional redundancy term—criteria that do not incorporate it are effectively blind to an entire class of problems, those with strong class-conditional dependencies. The second is whether it keeps the relative size of the redundancy term from swamping the relevancy term. We find this to be *essential*—without this control, the relevancy of the kth feature can easily be ignored in the selection process due to the $k-1$ redundancy terms. The third is simply whether the criterion is a low-dimensional approximation, hence making it usable with small sample sizes. On GISETTE with 6000 examples, we were unable to select more than 13 features with any kind of reliability. Therefore, low dimensional approximations, the focus of this article, are essential.

A summary of the criteria is shown in Table 8. Overall we find only 3 criteria that satisfy these properties: CMIM, JMI and DISR. We recommend the JMI criterion, as from empirical investigations it has the best trade-off (in the Pareto-optimal sense) of accuracy and stability. DISR is a normalised variant of JMI—in practice we found little need for this normalisation and the extra computation involved. If higher stability is required—the MIM criterion, as expected, displayed the highest stability with respect to variations in the data—therefore in extreme data-poor situations we would recommend this as a first step. If speed is required, the CMIM criterion admits an fast exact implementation giving orders of magnitude speed-up over a straightforward implementation—refer to Fleuret (2004) for details.

To aid replicability of this work, implementations of all criteria we have discussed are provided at: `http://www.cs.man.ac.uk/∼gbrown/fstoolbox/`

	MIM	mRMR	MIFS	CMIM	JMI	DISR	ICAP	CIFE	CMI
Cond Redund term?	✗	✗	✗	✔	✔	✔	✔	✔	✔
Balances rel/red?	✔	✔	✗	✔	✔	✔	✗	✗	✔
Estimable?	✔	✔	✔	✔	✔	✔	✔	✔	✗

Table 8: Summary of criteria. They have been arranged left to right in order of ascending estimation difficulty. *Cond Redund term*: does it include the conditional redundancy term? *Balances rel/red*: does it balance the relevance and redundancy terms? *Estimable*: does it use a low dimensional approximation, making it usable with small samples?

9.3 Future Work

While advice on the suitability of existing criteria is of course useful, perhaps a more interesting result of this work is the perspective it brings to the feature selection problem. We were able to *explicitly* state an objective function, and derive an appropriate information-based criterion to maximise it. This begs the question, what selection criteria would result from different objective functions? Dmochowski et al. (2010) study a weighted conditional likelihood, and its suitability for cost-sensitive problems—it is possible (though outside the scope of this paper) to derive information-based criteria in this context. The reverse question is equally interesting, what objective functions are implied by other existing criteria, such as the Gini Index? The KL-divergence (which defines the mutual information) is a special case of a wider family of measures, based on the f-divergence—could we obtain similar efficient criteria that pursue these measures, and what overall objectives do they imply?

In this work we explored criteria that use pairwise (i.e., $I(X_k;X_j)$) approximations to the derived objective. These approximations are commonly used as they provide a reasonable heuristic while still being (relatively) simple to estimate. There has been work which suggests relaxing this pairwise approximation, and thus increasing the number of terms (Brown, 2009; Meyer et al., 2008), but there is little exploration of how much data is required to estimate these multivariate information terms. A theoretical analysis of the tradeoff between estimation accuracy and additional information provided by these more complex terms could provide interesting directions for improving the power of filter feature selection techniques.

A very interesting direction concerns the motivation behind the conditional likelihood as an objective. It can be noted that the conditional likelihood, though a well-accepted objective function in its own right, can be derived from a probabilistic discriminative model, as follows. We approximate the true distribution p with our model q, with three distinct parameter sets: θ for feature selection,

τ for classification, and λ modelling the input distribution $p(\mathbf{x})$. Following Minka (2005), in the construction of a discriminative model, our joint likelihood is

$$\mathcal{L}(\mathcal{D},\theta,\tau,\lambda) = p(\theta,\tau)p(\lambda)\prod_{i=1}^{N} q(y^i|\mathbf{x}^i,\theta,\tau)q(\mathbf{x}^i|\lambda).$$

In this type of model, we wish to maximize $\mathcal{L}$ with respect to θ (our feature selection parameters) and τ (our model parameters), and are not concerned with the generative parameters λ. Excluding the generative terms gives

$$\mathcal{L}(\mathcal{D},\theta,\tau,\lambda) \propto p(\theta,\tau)\prod_{i=1}^{N} q(y^i|\mathbf{x}^i,\theta,\tau).$$

When we have no particular bias or prior knowledge over which subset of features or parameters are more likely (i.e., a flat prior $p(\theta,\tau)$), this reduces to the conditional likelihood:

$$\mathcal{L}(\mathcal{D},\theta,\tau,\lambda) \propto \prod_{i=1}^{N} q(y^i|\mathbf{x}^i,\theta,\tau),$$

which was exactly our starting point for the current paper. An obvious extension here is to take a non-uniform prior over features. An important direction for machine learning is to incorporate *domain knowledge*. A non-uniform prior would mean influencing the search procedure to incorporate our background knowledge of the features. This is applicable for example in gene expression data, when we may have information about the metabolic pathways in which genes participate, and therefore which genes are likely to influence certain biological functions. This is outside the scope of this paper but is the focus of our current research.

Acknowledgments

This research was conducted with support from the UK Engineering and Physical Sciences Research Council, on grants EP/G000662/1 and EP/F023855/1. Mikel Luján is supported by a Royal Society University Research Fellowship. Gavin Brown would like to thank James Neil, Sohan Seth, and Fabio Roli for invaluable commentary on drafts of this work.

Appendix A.

The following proofs make use of the identity, $I(A;B|C) - I(A;B) = I(A;C|B) - I(A;C)$.

A.1 Proof of Equation (17)

The *Joint Mutual Information* criterion (Yang and Moody, 1999) can be written,

$$\begin{aligned} J_{jmi}(X_k) &= \sum_{X_j \in S} I(X_kX_j;Y), \\ &= \sum_{X_j \in S} \Big[I(X_j;Y) + I(X_k;Y|X_j) \Big]. \end{aligned}$$

The term $\sum_{X_j \in S} I(X_j;Y)$ in the above is constant with respect to the X_k argument that we are interested in, so can be omitted. The criterion therefore reduces to (17) as follows,

$$\begin{aligned} J_{jmi}(X_k) &= \sum_{X_j \in S} \Big[I(X_k;Y|X_j) \Big] \\ &= \sum_{X_j \in S} \Big[I(X_k;Y) - I(X_k;X_j) + I(X_k;X_j|Y) \Big] \\ &= |S| \times I(X_k;Y) - \sum_{X_j \in S} \Big[I(X_k;X_j) - I(X_k;X_j|Y) \Big] \\ &\propto I(X_k;Y) - \frac{1}{|S|} \sum_{X_j \in S} \Big[I(X_k;X_j) - I(X_k;X_j|Y) \Big]. \end{aligned}$$

A.2 Proof of Equation (19)

The rearrangement of the Conditional Mutual Information criterion (Fleuret, 2004) follows a very similar procedure. The original, and its rewriting are,

$$\begin{aligned} J_{cmim}(X_k) &= \min_{X_j \in S} \Big[I(X_k;Y|X_j) \Big] \\ &= \min_{X_j \in S} \Big[I(X_k;Y) - I(X_k;X_j) + I(X_k;X_j|Y) \Big] \\ &= I(X_k;Y) + \min_{X_j \in S} \Big[I(X_k;X_j|Y) - I(X_k;X_j) \Big] \\ &= I(X_k;Y) - \max_{X_j \in S} \Big[I(X_k;X_j) - I(X_k;X_j|Y) \Big], \end{aligned}$$

which is exactly Equation (19).

References

K. S. Balagani and V. V. Phoha. On the feature selection criterion based on an approximation of multidimensional mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1342–1343, 2010. ISSN 0162-8828.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

G. Brown. A new perspective for information theoretic feature selection. In *International Conference on Artificial Intelligence and Statistics*, volume 5, pages 49–56, 2009.

H. Cheng, Z. Qin, C. Feng, Y. Wang, and F. Li. Conditional mutual information-based feature selection analyzing for synergy and redundancy. *Electronics and Telecommunications Research Institute (ETRI) Journal*, 33(2), 2011.

D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience New York, 1991.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

J. P. Dmochowski, P. Sajda, and L. C. Parra. Maximum likelihood in cost-sensitive learning: model specification, approximations, and upper bounds. *Journal of Machine Learning Research*, 11: 3313–3332, 2010.

W. Duch. *Feature Extraction: Foundations and Applications*, chapter 3, pages 89–117. Studies in Fuzziness & Soft Computing. Springer, 2006. ISBN 3-540-35487-5.

A. El Akadi, A. El Ouardighi, and D. Aboutajdine. A powerful feature selection approach based on mutual information. *International Journal of Computer Science and Network Security*, 8(4):116, 2008.

R. M. Fano. *Transmission of Information: Statistical Theory of Communications*. New York: Wiley, 1961.

F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

C. Fonseca and P. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. *Parallel Problem Solving from Nature*, pages 584–593, 1996.

D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *International Conference on Machine Learning*. ACM, 2004.

G. Gulgezen, Z. Cataltepe, and L. Yu. Stable and accurate feature selection. *Machine Learning and Knowledge Discovery in Databases*, pages 455–468, 2009.

B. Guo and M. S. Nixon. Gait feature subset selection by mutual information. *IEEE Trans Systems, Man and Cybernetics*, 39(1):36–46, January 2009.

I. Guyon. *Design of experiments for the NIPS 2003 variable selection benchmark*. http://www.nipsfsc.ecs.soton.ac.uk/papers/NIPS2003-Datasets.pdf, 2003.

I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction: Foundations and Applications*. Springer, 2006. ISBN 3-540-35487-5.

M. Hellman and J. Raviv. Probability of error, equivocation, and the chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970.

A. Jakulin. *Machine Learning Based on Attribute Interactions*. PhD thesis, University of Ljubljana, Slovenia, 2005.

A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*, 12(1):95–116, 2007. ISSN 0219-1377.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2): 273–324, 1997. ISSN 0004-3702.

D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, 1996.

K. Korb. *Encyclopedia of Machine Learning*, chapter Learning Graphical Models, page 584. Springer, 2011.

L. I. Kuncheva. A stability index for feature selection. In *IASTED International Multi-Conference: Artificial Intelligence and Applications*, pages 390–395, 2007.

N. Kwak and C. H. Choi. Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13(1):143–159, 2002.

D. D. Lewis. Feature selection and feature extraction for text categorization. In *Proceedings of the workshop on Speech and Natural Language*, pages 212–217. Association for Computational Linguistics Morristown, NJ, USA, 1992.

D. Lin and X. Tang. Conditional infomax learning: An integrated framework for feature extraction and fusion. In *European Conference on Computer Vision*, 2006.

P. Meyer and G. Bontempi. On the use of variable complementarity for feature selection in cancer classification. In *Evolutionary Computation and Machine Learning in Bioinformatics*, pages 91–102, 2006.

P. E. Meyer, C. Schretter, and G. Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3): 261–274, 2008.

T. Minka. Discriminative models, not discriminative training. *Microsoft Research Cambridge, Tech. Rep. TR-2005-144*, 2005.

L. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003. ISSN 0899-7667.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27(3): 379–423, 1948.

M. Tesmer and P. A. Estevez. Amifs: Adaptive feature selection by using mutual information. In *IEEE International Joint Conference on Neural Networks*, volume 1, 2004.

I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2003.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In *16th International FLAIRS Conference*, volume 103, 2003.

M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. *Advances in Neural Information Processing Systems*, pages 668–674, 2001. ISSN 1049-5258.

H. Yang and J. Moody. Data visualization and feature selection: New algorithms for non-gaussian data. *Advances in Neural Information Processing Systems*, 12, 1999.

L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

L. Yu, C. Ding, and S. Loscalzo. Stable feature selection via dense feature groups. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 803–811, 2008.

Plug-in Approach to Active Learning

Stanislav Minsker SMINSKER@MATH.GATECH.EDU
686 Cherry Street
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332-0160, USA

Editor: Sanjoy Dasgupta

Abstract

We present a new active learning algorithm based on nonparametric estimators of the regression function. Our investigation provides probabilistic bounds for the rates of convergence of the generalization error achievable by proposed method over a broad class of underlying distributions. We also prove minimax lower bounds which show that the obtained rates are almost tight.

Keywords: active learning, selective sampling, model selection, classification, confidence bands

1. Introduction

Let $(S,\mathcal{B})$ be a measurable space and let $(X,Y) \in S \times \{-1,1\}$ be a random couple with unknown distribution P. The marginal distribution of the design variable X will be denoted by Π. Let $\eta(x) := \mathbb{E}(Y|X=x)$ be the regression function. The goal of *binary classification* is to predict label Y based on the observation X. Prediction is based on a *classifier* - a measurable function $f : S \mapsto \{-1,1\}$. The quality of a classifier is measured in terms of its generalization error, $R(f) = \Pr(Y \neq f(X))$. In practice, the distribution P remains unknown but the learning algorithm has access to the *training data* - the i.i.d. sample $(X_i,Y_i),\ i = 1\dots n$ from P. It often happens that the cost of obtaining the training data is associated with labeling the observations X_i while the pool of observations itself is almost unlimited. This suggests to measure the performance of a learning algorithm in terms of its *label complexity*, the number of labels Y_i required to obtain a classifier with the desired accuracy. *Active learning* theory is mainly devoted to design and analysis of the algorithms that can take advantage of this modified framework. Most of these procedures can be characterized by the following property: at each step k, observation X_k is sampled from a distribution $\hat{\Pi}_k$ that depends on previously obtained $(X_i,Y_i),\ i \leq k-1$(while passive learners obtain all available training data at the same time). $\hat{\Pi}_k$ is designed to be supported on a set where classification is difficult and requires more labeled data to be collected. The situation when active learners outperform passive algorithms might occur when the so-called *Tsybakov's low noise assumption* is satisfied: there exist constants $B,\gamma > 0$ such that

$$\forall\, t > 0,\ \Pi(x : |\eta(x)| \leq t) \leq Bt^{\gamma}. \tag{1}$$

This assumption provides a convenient way to characterize the noise level of the problem and will play a crucial role in our investigation.

The topic of active learning is widely present in the literature; see Balcan et al. (2009), Hanneke (2011), Castro and Nowak (2008) for review. It was discovered that in some cases the generalization error of a resulting classifier can converge to zero exponentially fast with respect to its label complexity(while the best rate for passive learning is usually polynomial with respect to the cardinality of the training data set). However, available algorithms that adapt to the unknown parameters of the problem(γ in Tsybakov's low noise assumption, regularity of the decision boundary) involve empirical risk minimization with binary loss, along with other computationally hard problems, see Balcan et al. (2008), Dasgupta et al. (2008), Hanneke (2011) and Koltchinskii (2010). On the other hand, the algorithms that can be effectively implemented, as in Castro and Nowak (2008), are not adaptive.

The majority of the previous work in the field was done under standard complexity assumptions on the set of possible classifiers(such as polynomial growth of the covering numbers). Castro and Nowak (2008) derived their results under the regularity conditions on the decision boundary and the noise assumption which is slightly more restrictive then (1). Essentially, they proved that if the decision boundary is a graph of the Hölder smooth function $g \in \Sigma(\beta, K, [0,1]^{d-1})$ (see Section 2 for definitions) and the noise assumption is satisfied with $\gamma > 0$, then the minimax lower bound for the expected excess risk of the active classifier is of order $C \cdot N^{-\frac{\beta(1+\gamma)}{2\beta+\gamma(d-1)}}$ and the upper bound is $C(N/\log N)^{-\frac{\beta(1+\gamma)}{2\beta+\gamma(d-1)}}$, where N is the label budget. However, the construction of the classifier that achieves an upper bound assumes β and γ to be known.

In this paper, we consider the problem of active learning under classical nonparametric assumptions on the regression function - namely, we assume that it belongs to a certain Hölder class $\Sigma(\beta, K, [0,1]^d)$ and satisfies to the low noise condition (1) with some positive γ. In this case, the work of Audibert and Tsybakov (2005) showed that plug-in classifiers can attain optimal rates in the *passive* learning framework, namely, that the expected excess risk of a classifier $\hat{g} = \text{sign}\, \hat{\eta}$ is bounded above by $C \cdot N^{-\frac{\beta(1+\gamma)}{2\beta+d}}$ (which is the optimal rate), where $\hat{\eta}$ is the local polynomial estimator of the regression function and N is the size of the training data set. We were able to partially extend this claim to the case of active learning: first, we obtain minimax lower bounds for the excess risk of an active classifier in terms of its label complexity. Second, we propose a new algorithm that is based on plug-in classifiers, attains almost optimal rates over a broad class of distributions and possesses adaptivity with respect to β, γ(within the certain range of these parameters).

The paper is organized as follows: the next section introduces remaining notations and specifies the main assumptions made throughout the paper. This is followed by a qualitative description of our learning algorithm. The second part of the work contains the statements and proofs of our main results - minimax upper and lower bounds for the excess risk.

2. Preliminaries

Our *active learning* framework is governed by the following rules:

1. Observations are sampled sequentially: X_k is sampled from the modified distribution $\hat{\Pi}_k$ that depends on $(X_1, Y_1), \ldots, (X_{k-1}, Y_{k-1})$.

2. Y_k is sampled from the conditional distribution $P_{Y|X}(\cdot | X = x)$. Labels are conditionally independent given the feature vectors $X_i, \ i \le n$.

Usually, the distribution $\hat{\Pi}_k$ is supported on a set where classification is difficult.

Given the probability measure $\mathbb{Q}$ on $S \times \{-1, 1\}$, we denote the integral with respect to this measure by $\mathbb{Q}g := \int g d\mathbb{Q}$. Let $\mathcal{F}$ be a class of bounded, measurable functions. The risk and the excess risk of $f \in \mathcal{F}$ with respect to the measure $\mathbb{Q}$ are defined by

$$R_{\mathbb{Q}}(f) := \mathbb{Q} I_{y \neq \text{sign } f(x)}$$
$$\mathcal{E}_{\mathbb{Q}}(f) := R_{\mathbb{Q}}(f) - \inf_{g \in \mathcal{F}} R_{\mathbb{Q}}(g),$$

where $I_{\mathcal{A}}$ is the indicator of event $\mathcal{A}$. We will omit the subindex $\mathbb{Q}$ when the underlying measure is clear from the context. Recall that we denoted the distribution of (X,Y) by P. The minimal possible risk with respect to P is

$$R^* = \inf_{g:S \mapsto [-1,1]} \Pr\left(Y \neq \text{sign } g(X)\right),$$

where the infimum is taken over all measurable functions. It is well known that it is attained for any g such that sign $g(x) =$ sign $\eta(x)$ Π - a.s. Given $g \in \mathcal{F}$, $A \in \mathcal{B}$, $\delta > 0$, define

$$\mathcal{F}_{\infty,A}(g;\delta) := \left\{ f \in \mathcal{F} : \ \|f-g\|_{\infty,A} \leq \delta \right\},$$

where $\|f-g\|_{\infty,A} = \sup_{x \in A} |f(x) - g(x)|$. For $A \in \mathcal{B}$, define the function class

$$\mathcal{F}|_A := \left\{ f|_A, \ f \in \mathcal{F} \right\},$$

where $f|_A(x) := f(x) I_A(x)$. From now on, we restrict our attention to the case $S = [0,1]^d$. Let $K > 0$.

Definition 1 *We say that $g : \mathbb{R}^d \mapsto \mathbb{R}$ belongs to $\Sigma(\beta, K, [0,1]^d)$, the $(\beta, K, [0,1]^d)$ - Hölder class of functions, if g is $\lfloor \beta \rfloor$ times continuously differentiable and for all $x, x_1 \in [0,1]^d$ satisfies*

$$|g(x_1) - T_x(x_1)| \leq K \|x - x_1\|_\infty^\beta,$$

where T_x is the Taylor polynomial of degree $\lfloor \beta \rfloor$ of g at the point x.

Definition 2 *$\mathcal{P}(\beta,\gamma)$ is the class of probability distributions on $[0,1]^d \times \{-1,+1\}$ with the following properties:*

1. $\forall\, t > 0$, $\Pi(x : |\eta(x)| \leq t) \leq Bt^\gamma$;

2. $\eta(x) \in \Sigma(\beta, K, [0,1]^d)$.

We do not mention the dependence of $\mathcal{P}(\beta,\gamma)$ on the fixed constants B, K explicitly, but this should not cause any uncertainty.

Finally, let us define $\mathcal{P}_U^*(\beta,\gamma)$ and $\mathcal{P}_U(\beta,\gamma)$, the subclasses of $\mathcal{P}(\beta,\gamma)$, by imposing two additional assumptions. Along with the formal descriptions of these assumptions, we shall try to provide some motivation behind them. The first deals with the marginal Π. For an integer $M \geq 1$, let

$$\mathcal{G}_M := \left\{ \left(\frac{k_1}{M}, \dots, \frac{k_d}{M} \right), \ k_i = 1 \dots M, \ i = 1 \dots d \right\}$$

be the regular grid on the unit cube $[0,1]^d$ with mesh size M^{-1}. It naturally defines a partition into a set of M^d open cubes $R_i, \ i = 1 \dots M^d$ with edges of length M^{-1} and vertices in $\mathcal{G}_M$. Below, we consider the nested sequence of grids $\{\mathcal{G}_{2^m}, \ m \geq 1\}$ and corresponding dyadic partitions of the unit cube.

Definition 3 *We will say that Π is (u_1, u_2)-regular with respect to $\{\mathcal{G}_{2^m}\}$ if for any $m \geq 1$, any element of the partition R_i, $i \leq 2^{dm}$ such that $R_i \cap \operatorname{supp}(\Pi) \neq \emptyset$, we have*

$$u_1 \cdot 2^{-dm} \leq \Pi(R_i) \leq u_2 \cdot 2^{-dm},$$

where $0 < u_1 \leq u_2 < \infty$.

Assumption 1 Π *is (u_1, u_2) - regular.*

In particular, (u_1, u_2)-regularity holds for the distribution with a density p on $[0,1]^d$ such that $0 < u_1 \leq p(x) \leq u_2 < \infty$.

Let us mention that our definition of regularity is of rather technical nature; for most of the paper, the reader might think of Π as being uniform on $[0,1]^d$(however, we need slightly more complicated marginal to construct the minimax lower bounds for the excess risk). It is known that estimation of regression function in sup-norm is sensitive to the geometry of design distribution, mainly because the quality of estimation depends on the *local* amount of data at every point; conditions similar to our *Assumption* 1 were used in the previous works where this problem appeared, for example, *strong density assumption* in Audibert and Tsybakov (2005) and *Assumption D* in Gaïffas (2007).

A useful characteristic of (u_1, u_2) - regular distribution Π is that this property is stable with respect to restrictions of Π to certain subsets of its support. This fact fits the active learning framework particularly well.

Definition 4 *We say that $\mathbb{Q}$ belongs to $\mathcal{P}_U(\beta, \gamma)$ if $\mathbb{Q} \in \mathcal{P}(\beta, \gamma)$ and Assumption 1 is satisfied for some u_1, u_2.*

The second assumption is crucial in derivation of the upper bounds. The space of piecewise-constant functions which is used to construct the estimators of $\eta(x)$ is defined via

$$\mathcal{F}_m = \left\{ \sum_{i=1}^{2^{dm}} \lambda_i I_{R_i}(\cdot) : \ |\lambda_i| \leq 1, \ i = 1 \ldots 2^{dm} \right\},$$

where $\{R_i\}_{i=1}^{2^{dm}}$ forms the dyadic partition of the unit cube. Note that $\mathcal{F}_m$ can be viewed as a $\|\cdot\|_\infty$-unit ball in the linear span of first 2^{dm} Haar basis functions in $[0,1]^d$. Moreover, $\{\mathcal{F}_m, \ m \geq 1\}$ is a nested family, which is a desirable property for the model selection procedures. By $\bar{\eta}_m(x)$ we denote the $L_2(\Pi)$ - projection of the regression function onto $\mathcal{F}_m$.

We will say that the set $A \subset [0,1]^d$ *approximates the decision boundary* $\{x : \eta(x) = 0\}$ if there exists $t > 0$ such that

$$\{x : |\eta(x)| \leq t\}_\Pi \subseteq A_\Pi \subseteq \{x : |\eta(x)| \leq 3t\}_\Pi, \tag{2}$$

where for any set A we define $A_\Pi := A \cap \operatorname{supp}(\Pi)$. The most important example we have in mind is the following: let $\hat{\eta}$ be some estimator of η with $\|\hat{\eta} - \eta\|_{\infty, \operatorname{supp}(\Pi)} \leq t$, and define the $2t$ - band around η by

$$\hat{F} = \left\{ f : \ \hat{\eta}(x) - 2t \leq f(x) \leq \hat{\eta}(x) + 2t \ \forall x \in [0,1]^d \right\}.$$

Take $A = \left\{x : \ \exists f_1, f_2 \in \hat{F} \text{ s.t. } \operatorname{sign} f_1(x) \neq \operatorname{sign} f_2(x)\right\}$, then it is easy to see that A satisfies (2). Modified design distributions used by our algorithm are supported on the sets with similar structure.

Let $\sigma(\mathcal{F}_m)$ be the sigma-algebra generated by $\mathcal{F}_m$ and $A \in \sigma(\mathcal{F}_m)$.

Assumption 2 *There exists $B_2 > 0$ such that for all $m \geq 1$, $A \in \sigma(\mathcal{F}_m)$ satisfying (2) and such that $A_\Pi \neq \emptyset$ the following holds true:*

$$\int\limits_{[0,1]^d} (\eta - \bar{\eta}_m)^2 \, \Pi(dx|x \in A_\Pi) \geq B_2 \|\eta - \bar{\eta}_m\|^2_{\infty, A_\Pi}.$$

Appearance of *Assumption 2* is motivated by the structure of our learning algorithm - namely, it is based on adaptive confidence bands for the regression function. Nonparametric confidence bands is a big topic in statistical literature, and the review of this subject is not our goal. We just mention that it is impossible to construct adaptive confidence bands of optimal size over the whole $\bigcup\limits_{\beta \leq 1} \Sigma\left(\beta, K, [0,1]^d\right)$. Low (1997); Hoffmann and Nickl (to appear) discuss the subject in details. However, it is possible to construct adaptive L_2 - confidence balls (see an example following Theorem 6.1 in Koltchinskii, 2011). For functions satisfying *Assumption 2*, this fact allows to obtain confidence bands of desired size. In particular,

(a) functions that are differentiable, with gradient being bounded away from 0 in the vicinity of decision boundary;

(b) Lipschitz continuous functions that are convex in the vicinity of decision boundary

satisfy *Assumption* 2. For precise statements, see Propositions 15, 16 in Appendix A. A different approach to adaptive confidence bands in case of one-dimensional density estimation is presented in Giné and Nickl (2010). Finally, we define $\mathcal{P}^*_U(\beta,\gamma)$:

Definition 5 *We say that $\mathbb{Q}$ belongs to $\mathcal{P}^*_U(\beta,\gamma)$ if $\mathbb{Q} \in \mathcal{P}_U(\beta,\gamma)$ and Assumption 2 is satisfied for some $B_2 > 0$.*

2.1 Learning Algorithm

Now we give a brief description of the algorithm, since several definitions appear naturally in this context. First, let us emphasize that *the marginal distribution Π is assumed to be known to the learner.* This is not a restriction, since we are not limited in the use of unlabeled data and Π can be estimated to any desired accuracy. Our construction is based on so-called *plug-in* classifiers of the form $\hat{f}(\cdot) = \text{sign}\,\hat{\eta}(\cdot)$, where $\hat{\eta}$ is a piecewise-constant estimator of the regression function. As we have already mentioned above, it was shown in Audibert and Tsybakov (2005) that in the passive learning framework plug-in classifiers attain optimal rate for the excess risk of order $N^{-\frac{\beta(1+\gamma)}{2\beta+d}}$, with $\hat{\eta}$ being the local polynomial estimator.

Our active learning algorithm iteratively improves the classifier by constructing shrinking confidence bands for the regression function. On every step k, the piecewise-constant estimator $\hat{\eta}_k$ is obtained via the model selection procedure which allows adaptation to the unknown smoothness(for Hölder exponent ≤ 1). The estimator is further used to construct a confidence band $\hat{\mathcal{F}}_k$ for $\eta(x)$. The *active set* associated with $\hat{\mathcal{F}}_k$ is defined as

$$\hat{A}_k = A(\hat{\mathcal{F}}_k) := \left\{ x \in \text{supp}(\Pi) : \exists f_1, f_2 \in \hat{\mathcal{F}}_k, \text{sign}\, f_1(x) \neq \text{sign}\, f_2(x) \right\}.$$

Clearly, this is the set where the confidence band crosses zero level and where classification is potentially difficult. $\hat{A}_k$ serves as a support of the modified distribution $\hat{\Pi}_{k+1}$: on step $k+1$, label Y is requested only for observations $X \in \hat{A}_k$, forcing the labeled data to concentrate in the domain where higher precision is needed. This allows one to obtain a tighter confidence band for the regression function restricted to the active set. Since $\hat{A}_k$ approaches the decision boundary, its size is controlled by the low noise assumption. The algorithm does not require a priori knowledge of the noise and regularity parameters, being adaptive for $\gamma > 0, \beta \le 1$. Further details are given in Section 3.2.

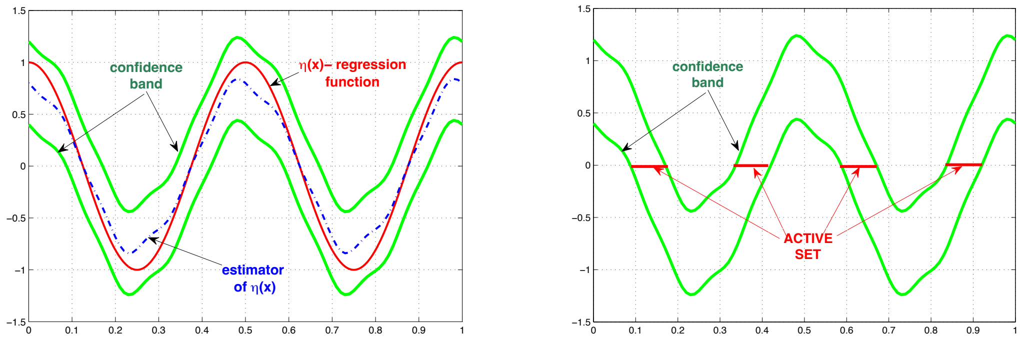


Figure 1: Active Learning Algorithm

2.2 Comparison Inequalities

Before proceeding with the main results, let us recall the well-known connections between the binary risk and the $\|\cdot\|_\infty, \|\cdot\|_{L_2(\Pi)}$ - norm risks:

Proposition 6 *Under the low noise assumption,*

$$R_P(f) - R^* \le D_1 \|(f-\eta) I\{\text{sign } f \ne \text{sign } \eta\}\|_\infty^{1+\gamma}; \tag{3}$$

$$R_P(f) - R^* \le D_2 \|(f-\eta) I\{\text{sign } f \ne \text{sign } \eta\}\|_{L_2(\Pi)}^{\frac{2(1+\gamma)}{2+\gamma}}; \tag{4}$$

$$R_P(f) - R^* \ge D_3 \Pi(\text{sign } f \ne \text{sign } \eta)^{\frac{1+\gamma}{\gamma}}. \tag{5}$$

Proof For (3) and (4), see Audibert and Tsybakov (2005), Lemmas 5.1 and 5.2 respectively, and for (5)—Koltchinskii (2011), Lemma 5.2. ■

3. Main Results

The question we address below is: what are the best possible rates that can be achieved by active algorithms in our framework and how these rates can be attained.

3.1 Minimax Lower Bounds For the Excess Risk

The goal of this section is to prove that for $P \in \mathcal{P}(\beta,\gamma)$, no active learner can output a classifier with expected excess risk converging to zero faster than $N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}}$. Our result builds upon the minimax bounds of Audibert and Tsybakov (2005), Castro and Nowak (2008).

Remark The theorem below is proved for a smaller class $\mathcal{P}^*_U(\beta,\gamma)$, which implies the result for $\mathcal{P}(\beta,\gamma)$.

Theorem 7 *Let β,γ,d be such that $\beta\gamma \le d$. Then there exists $C > 0$ such that for all n large enough and for any active classifier $\hat{f}_n(x)$ we have*

$$\sup_{P\in\mathcal{P}^*_U(\beta,\gamma)} \mathbb{E}R_P(\hat{f}_n) - R^* \ge CN^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}}.$$

Proof We proceed by constructing the appropriate family of classifiers $f_\sigma(x) = \text{sign}\,\eta_\sigma(x)$, in a way similar to Theorem 3.5 in Audibert and Tsybakov (2005), and then apply Theorem 2.5 from Tsybakov (2009). We present it below for reader's convenience.

Theorem 8 *Let Σ be a class of models, $\mathrm{d} : \Sigma\times\Sigma \mapsto \mathbb{R}$ - the pseudometric and $\{P_f,\ f\in\Sigma\}$ - a collection of probability measures associated with Σ. Assume there exists a subset $\{f_0,\dots,f_M\}$ of Σ such that*

1. *$d(f_i,f_j) \ge 2s > 0$ for all $0 \le i < j \le M$;*
2. *$P_{f_j} \ll P_{f_0}$ for every $1 \le j \le M$;*
3. *$\frac{1}{M}\sum\limits_{j=1}^{M} \mathrm{KL}(P_{f_j},P_{f_0}) \le \alpha\log M, \quad 0<\alpha<\frac{1}{8}$.*

Then

$$\inf_{\hat{f}}\sup_{f\in\Sigma} P_f\left(\mathrm{d}(\hat{f},f) \ge s\right) \ge \frac{\sqrt{M}}{1+\sqrt{M}}\left(1-2\alpha-\sqrt{\frac{2\alpha}{\log M}}\right),$$

where the infimum is taken over all possible estimators of f based on a sample from P_f and $\mathrm{KL}(\cdot,\cdot)$ is the Kullback-Leibler divergence.

Going back to the proof, let $q = 2^l,\ l \ge 1$ and

$$G_q := \left\{\left(\frac{2k_1-1}{2q},\dots,\frac{2k_d-1}{2q}\right),\ k_i = 1\dots q,\ i = 1\dots d\right\}$$

be the grid on $[0,1]^d$. For $x\in[0,1]^d$, let

$$n_q(x) = \text{argmin}\ \left\{\|x-x_k\|_2 :\ x_k\in G_q\right\}.$$

If $n_q(x)$ is not unique, we choose a representative with the smallest $\|\cdot\|_2$ norm. The unit cube is partitioned with respect to G_q as follows: x_1,x_2 belong to the same subset if $n_q(x_1) = n_q(x_2)$. Let $'\succ'$ be some order on the elements of G_q such that $x\succ y$ implies $\|x\|_2 \ge \|y\|_2$. Assume that the elements of the partition are enumerated with respect to the order of their centers induced by $'\succ'$: $[0,1]^d = \bigcup\limits_{i=1}^{q^d} R_i$. Fix $1\le m\le q^d$ and let

$$S := \bigcup_{i=1}^{m} R_i$$

Note that the partition is ordered in such a way that there always exists $1 \le k \le q\sqrt{d}$ with

$$B_+\left(0, \frac{k}{q}\right) \subseteq S \subseteq B_+\left(0, \frac{k+3\sqrt{d}}{q}\right), \tag{6}$$

where $B_+(0,R) := \left\{x \in \mathbb{R}^d_+ : \ \|x\|_2 \le R\right\}$. In other words, (6) means that that the difference between the radii of inscribed and circumscribed spherical sectors of S is of order $C(d)q^{-1}$.

Let $v > r_1 > r_2$ be three integers satisfying

$$2^{-v} < 2^{-r_1} < 2^{-r_1}\sqrt{d} < 2^{-r_2}\sqrt{d} < 2^{-1}. \tag{7}$$

Define $u(x) : \mathbb{R} \mapsto \mathbb{R}_+$ by

$$u(x) := \frac{\int_x^\infty U(t)dt}{\int\limits_{2^{-v}}^{1/2} U(t)dt}, \tag{8}$$

where

$$U(t) := \begin{cases} \exp\left(-\frac{1}{(1/2-x)(x-2^{-v})}\right), & x \in (2^{-v}, \frac{1}{2}) \\ 0 & \text{else.} \end{cases}$$

Note that $u(x)$ is an infinitely differentiable function such that $u(x) = 1, \ x \in [0, 2^{-v}]$ and $u(x) = 0, \ x \ge \frac{1}{2}$. Finally, for $x \in \mathbb{R}^d$ let

$$\Phi(x) := Cu(\|x\|_2),$$

where $C := C_{L,\beta}$ is chosen such that $\Phi \in \Sigma(\beta, L, \mathbb{R}^d)$.

Let $r_S := \inf\{r > 0 : \ B_+(0,r) \supseteq S\}$ and

$$A_0 := \left\{\bigcup_i R_i : \ R_i \cap B_+\left(0, r_S + q^{-\frac{\beta\gamma}{d}}\right) = \emptyset\right\}.$$

Note that

$$r_S \le c\frac{m^{1/d}}{q}, \tag{9}$$

since $\mathrm{Vol}\,(S) = mq^{-d}$.

Define $\mathcal{H}_m = \{P_\sigma : \sigma \in \{-1,1\}^m\}$ to be the hypercube of probability distributions on $[0,1]^d \times \{-1,+1\}$. The marginal distribution Π of X is independent of σ: define its density p by

$$p(x) = \begin{cases} \frac{2^{d(r_1-1)}}{2^{d(r_1-r_2)}-1}, & x \in B_\infty\left(z, \frac{2^{-r_2}}{q}\right) \setminus B_\infty\left(z, \frac{2^{-r_1}}{q}\right), \ z \in G_q \cap S, \\ c_0, & x \in A_0, \\ 0 & \text{else.} \end{cases}$$

where $B_\infty(z,r) := \{x : \ \|x-z\|_\infty \le r\}$, $c_0 := \frac{1-mq^{-d}}{\mathrm{Vol}(A_0)}$(note that $\Pi(R_i) = q^{-d} \quad \forall i \le m$) and r_1, r_2 are defined in (7). In particular, Π satisfies *Assumption* 1 since it is supported on the union of dyadic cubes and has bounded above and below on $\mathrm{supp}(\Pi)$ density. Let

$$\Psi(x) := u\left(1/2 - q^{\frac{\beta\gamma}{d}}\,\mathrm{dist}_2(x, B_+(0, r_S))\right),$$

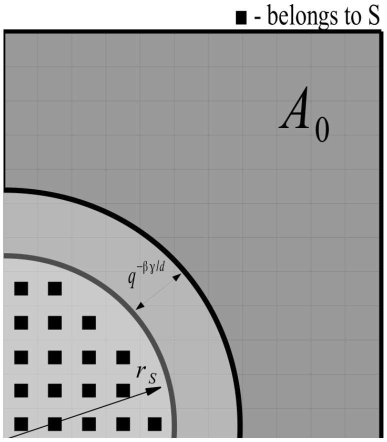


Figure 2: Geometry of the support

where $u(\cdot)$ is defined in (8) and $\mathrm{dist}_2(x,A) := \inf\{\|x-y\|_2,\ y \in A\}$.

Finally, the regression function $\eta_\sigma(x) = \mathbb{E}_{P_\sigma}(Y|X=x)$ is defined via

$$\eta_\sigma(x) := \begin{cases} \sigma_i q^{-\beta}\Phi(q[x-n_q(x)]), & x \in R_i,\ 1 \le i \le m \\ \frac{1}{C_{L,\beta}\sqrt{d}}\,\mathrm{dist}_2(x, B_+(0,r_S))^{\frac{d}{\gamma}} \cdot \Psi(x), & x \in [0,1]^d \setminus S. \end{cases}$$

The graph of η_σ is a surface consisting of small ”bumps” spread around S and tending away from 0 monotonically with respect to $\mathrm{dist}_2(\cdot, B_+(0,r_S))$ on $[0,1]^d \setminus S$. Clearly, $\eta_\sigma(x)$ satisfies smoothness requirement, [1] since for $x \in [0,1]^d$

$$\mathrm{dist}_2(x, B_+(0,r_S)) = (\|x\|_2 - r_S) \vee 0.$$

Let’s check that it also satisfies the low noise condition. Since $|\eta_\sigma| \ge Cq^{-\beta}$ on the support of Π, it is enough to consider $t = Czq^{-\beta}$ for $z > 1$:

$$\begin{aligned} \Pi(|\eta_\sigma(x)| \le Czq^{-\beta}) &\le mq^{-d} + \Pi\left(\mathrm{dist}_2(x, B_+(0,r_S)) \le Cz^{\gamma/d} q^{-\frac{\beta\gamma}{d}}\right) \le \\ &\le mq^{-d} + C_2\left(r_S + Cz^{\gamma/d} q^{-\frac{\beta\gamma}{d}}\right)^d \le \\ &\le mq^{-d} + C_3 mq^{-d} + C_4 z^{\gamma} q^{-\beta\gamma} \le \\ &\le \widehat{C} t^{\gamma}, \end{aligned}$$

if $mq^{-d} = O(q^{-\beta\gamma})$. Here, the first inequality follows from considering η_σ on S and A_0 separately, and second inequality follows from (9) and direct computation of the sphere volume.

Finally, η_σ satisfies *Assumption* 2 with some $B_2 := B_2(q)$ since on $\mathrm{supp}(\Pi)$

$$0 < c_1(q) \le \|\nabla \eta_\sigma(x)\|_2 \le c_2(q) < \infty.$$

The next step in the proof is to choose the subset of $\mathcal{H}$ which is “well-separated”: this can be done due to the following fact (see Tsybakov, 2009, Lemma 2.9):

[1] $\Psi(x)$ is introduced to provide extra smoothness at the boundary of $B_+(0,r_S)$.

Proposition 9 (Gilbert-Varshamov) *For $m \geq 8$, there exists*

$$\{\sigma_0, \dots, \sigma_M\} \subset \{-1, 1\}^m$$

such that $\sigma_0 = \{1, 1, \dots, 1\}$, $\rho(\sigma_i, \sigma_j) \geq \frac{m}{8}\ \forall\ 0 \leq i < k \leq M$ *and* $M \geq 2^{m/8}$ *where* ρ *stands for the Hamming distance.*

Let $\mathcal{H}' := \{P_{\sigma_0}, \dots, P_{\sigma_M}\}$ be chosen such that $\{\sigma_0, \dots, \sigma_M\}$ satisfies the proposition above. Next, following the proof of Theorems 1 and 3 in Castro and Nowak (2008), we note that $\forall \sigma \in \mathcal{H}'$, $\sigma \neq \sigma_0$

$$\mathrm{KL}(P_{\sigma,N} \| P_{\sigma_0,N}) \leq 8N \max_{x \in [0,1]} (\eta_\sigma(x) - \eta_{\sigma_0}(x))^2 \leq 32 C_{L,\beta}^2 N q^{-2\beta}, \tag{10}$$

where $P_{\sigma,N}$ is the joint distribution of $(X_i, Y_i)_{i=1}^N$ under hypothesis that the distribution of couple (X, Y) is P_σ. Let us briefly sketch the derivation of (10); see also the proof of Theorem 1 in Castro and Nowak (2008). Denote

$$\begin{aligned} \bar{X}_k &:= (X_1, \dots, X_k), \\ \bar{Y}_k &:= (Y_1, \dots, Y_k). \end{aligned}$$

Then $dP_{\sigma,N}$ admits the following factorization:

$$dP_{\sigma,N}(\bar{X}_N, \bar{Y}_N) = \prod_{i=1}^N P_\sigma(Y_i | X_i) dP(X_i | \bar{X}_{i-1}, \bar{Y}_{i-1}),$$

where $dP(X_i | \bar{X}_{i-1}, \bar{Y}_{i-1})$ does not depend on σ but only on the active learning algorithm. As a consequence,

$$\begin{aligned} \mathrm{KL}(P_{\sigma,N} \| P_{\sigma_0,N}) &= \mathbb{E}_{P_{\sigma,N}} \log \frac{dP_{\sigma,N}(\bar{X}_N, \bar{Y}_N)}{dP_{\sigma_0,N}(\bar{X}_n, \bar{Y}_N)} = \mathbb{E}_{P_{\sigma,N}} \log \frac{\prod_{i=1}^N P_\sigma(Y_i|X_i)}{\prod_{i=1}^N P_{\sigma_0}(Y_i|X_i)} = \\ &= \sum_{i=1}^N \mathbb{E}_{P_{\sigma,N}} \left[\mathbb{E}_{P_\sigma} \left(\log \frac{P_\sigma(Y_i|X_i)}{P_{\sigma_0}(Y_i|X_i)} | X_i \right) \right] \leq \\ &\leq N \max_{x \in [0,1]^d} \mathbb{E}_{P_\sigma} \left(\log \frac{P_\sigma(Y_1|X_1)}{P_{\sigma_0}(Y_1|X_1)} | X_1 = x \right) \leq \\ &\leq 8N \max_{x \in [0,1]^d} (\eta_\sigma(x) - \eta_{\sigma_0}(x))^2, \end{aligned}$$

where the last inequality follows from Lemma 1 (Castro and Nowak, 2008). Also, note that we have $\max_{x \in [0,1]^d}$ in our bounds rather than the average over x that would appear in the passive learning framework.

It remains to choose q, m in appropriate way: set $q = \lfloor C_1 N^{\frac{1}{2\beta + d - \beta\gamma}} \rfloor$ and $m = \lfloor C_2 q^{d - \beta\gamma} \rfloor$ where C_1, C_2 are such that $q^d \geq m \geq 1$ and $32 C_{L,\beta}^2 N q^{-2\beta} < \frac{m}{64}$ which is possible for N big enough. In particular, $m q^{-d} = O(q^{-\beta\gamma})$. Together with the bound (10), this gives

$$\frac{1}{M} \sum_{\sigma \in \mathcal{H}'} \mathrm{KL}(P_\sigma \| P_{\sigma^0}) \leq 32 C_u^2 N q^{-2\beta} < \frac{m}{8^2} = \frac{1}{8} \log |\mathcal{H}'|,$$

so that conditions of Theorem 8 are satisfied. Setting

$$f_\sigma(x) := \text{sign}\ \eta_\sigma(x),$$

we finally have $\forall \sigma_1 \neq \sigma_2 \in \mathcal{H}'$

$$d(f_{\sigma_1}, f_{\sigma_2}) := \Pi(\text{sign}\ \eta_{\sigma_1}(x) \neq \text{sign}\ \eta_{\sigma_2}(x)) \geq \frac{m}{8q^d} \geq C_4 N^{-\frac{\beta\gamma}{2\beta+d-\beta\gamma}},$$

where the lower bound just follows by construction of our hypotheses. Since under the low noise assumption $R_P(\hat{f}_n) - R^* \geq c\Pi(\hat{f}_n \neq \text{sign}\ \eta)^{\frac{1+\gamma}{\gamma}}$ (see (5)), we conclude by Theorem 8 that

$$\inf_{\hat{f}_N} \sup_{P \in \mathcal{P}_U^*(\beta,\gamma)} \Pr\left(R_P(\hat{f}_n) - R^* \geq C_4 N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}} \right) \geq$$
$$\geq \inf_{\hat{f}_N} \sup_{P \in \mathcal{P}_U^*(\beta,\gamma)} \Pr\left(\Pi(\hat{f}_n(x) \neq \text{sign}\ \eta_P(x)) \geq \frac{C_4}{2} N^{-\frac{\beta\gamma}{2\beta+d-\beta\gamma}} \right) \geq \tau > 0.$$

■

3.2 Upper Bounds For the Excess Risk

Below, we present a new active learning algorithm which is computationally tractable, adaptive with respect to β, γ(in a certain range of these parameters) and can be applied in the nonparametric setting. We show that the classifier constructed by the algorithm attains the rates of Theorem 7, up to polylogarithmic factor, if $0 < \beta \leq 1$ and $\beta\gamma \leq d$ (the last condition covers the most interesting case when the regression function hits or crosses the decision boundary in the interior of the support of Π; for detailed statement about the connection between the behavior of the regression function near the decision boundary with parameters β, γ, see Proposition 3.4 in Audibert and Tsybakov, 2005). The problem of adaptation to higher order of smoothness ($\beta > 1$) is still awaiting its complete solution; we address these questions below in our final remarks.

For the purpose of this section, the regularity assumption reads as follows: there exists $0 < \beta \leq 1$ such that $\forall x_1, x_2 \in [0,1]^d$

$$|\eta(x_1) - \eta(x_2)| \leq B_1 \|x_1 - x_2\|_\infty^\beta. \tag{11}$$

Since we want to be able to construct non-asymptotic confidence bands, some estimates on the size of constants in (11) and *Assumption 2* are needed. Below, we will additionally assume that

$$B_1 \leq \log N,$$
$$B_2 \geq \log^{-1} N,$$

where N is the label budget. This can be replaced by any known bounds on B_1, B_2.

Let $A \in \sigma(\mathcal{F}_m)$ with $A_\Pi := A \cap \text{supp}(\Pi) \neq \emptyset$. Define

$$\hat{\Pi}_A(dx) := \Pi(dx | x \in A_\Pi)$$

and $d_m := \dim \mathcal{F}_m|_{A_\Pi}$. Next, we introduce a simple estimator of the regression function on the set A_Π. Given the resolution level m and an iid sample $(X_i, Y_i),\ i \le N$ with $X_i \sim \hat{\Pi}_A$, let

$$\hat{\eta}_{m,A}(x) := \sum_{i:R_i \cap A_\Pi \neq \emptyset} \frac{\sum_{j=1}^N Y_j I_{R_i}(X_j)}{N \cdot \hat{\Pi}_A(R_i)} I_{R_i}(x). \tag{12}$$

Since we assumed that the marginal Π is known, the estimator is well-defined. The following proposition provides the information about concentration of $\hat{\eta}_m$ around its mean:

Proposition 10 *For all $t > 0$,*

$$\Pr\left(\max_{x \in A_\Pi} |\hat{\eta}_{m,A}(x) - \bar{\eta}_m(x)| \ \geq t\sqrt{\frac{2^{dm}\Pi(A)}{u_1 N}}\right) \leq$$
$$\leq 2d_m \exp\left(\frac{-t^2}{2(1+\frac{t}{3}\sqrt{2^{dm}\Pi(A)/u_1 N)}}\right),$$

Proof This is a straightforward application of the Bernstein's inequality to the random variables

$$S_N^i := \sum_{j=1}^N Y_j I_{R_i}(X_j),\ i \in \{i : R_i \cap A_\Pi \neq \emptyset\},$$

and the union bound: indeed, note that $\mathbb{E}(Y I_{R_i}(X_j))^2 = \hat{\Pi}_A(R_i)$, so that

$$\Pr\left(\left|S_N^i - N\int_{R_i} \eta d\hat{\Pi}_A\right| \geq tN\hat{\Pi}_A(R_i)\right) \leq 2\exp\left(-\frac{N\hat{\Pi}_A(R_i)t^2}{2+2t/3}\right),$$

and the rest follows by simple algebra using that $\hat{\Pi}_A(R_i) \geq \frac{u_1}{2^{dm}\Pi(A)}$ by the (u_1, u_2)-regularity of Π. ■

Given a sequence of hypotheses classes $\mathcal{G}_m,\ m \geq 1$, define the index set

$$\mathcal{J}(N) := \left\{m \in \mathbb{N} :\ 1 \leq \dim \mathcal{G}_m \leq \frac{N}{\log^2 N}\right\} \tag{13}$$

- the set of possible "resolution levels" of an estimator based on N classified observations(an upper bound corresponds to the fact that we want the estimator to be consistent). When talking about model selection procedures below, we will implicitly assume that the model index is chosen from the corresponding set $\mathcal{J}$. The role of $\mathcal{G}_m$ will be played by $\mathcal{F}_m|_A$ for appropriately chosen set A. We are now ready to present the active learning algorithm followed by its detailed analysis(see Table 1).

Remark Note that on every iteration, **Algorithm 1a** uses the whole sample to select the resolution level $\hat{m}_k$ and to build the estimator $\hat{\eta}_k$. While being suitable for practical implementation, this is not convenient for theoretical analysis. We will prove the upper bounds for a slighly modified version: namely, on every iteration k labeled data is divided into two subsamples $S_{k,1}$ and $S_{k,2}$ of approximately equal size, $|S_{k,1}| \simeq |S_{k,2}| \simeq \left\lfloor \frac{1}{2} N_k \cdot \Pi(\hat{A}_k) \right\rfloor$. Then $S_{1,k}$ is used to select the resolution level $\hat{m}_k$ and $S_{k,2}$ - to construct $\hat{\eta}_k$. We will call this modified version **Algorithm 1b**.

```
**Algorithm 1a**
**input** label budget $N$; confidence $\alpha$;
$\hat{m}_0 = 0,\ \hat{\mathcal{F}}_0 := \mathcal{F}_{\hat{m}_0},\ \hat{\eta}_0 \equiv 0$;
$LB := N$;                    // label budget
$N_0 := 2^{\lfloor \log_2 \sqrt{N} \rfloor}$;
$s^{(k)}(m,N,\alpha) := s(m,N,\alpha) := m(\log N + \log \frac{1}{\alpha})$;
$k := 0$;
**while** $LB \geq 0$ **do**
$k := k+1$;
$N_k := 2N_{k-1}$;
$\hat{A}_k := \left\{ x \in [0,1]^d :\ \exists f_1, f_2 \in \hat{\mathcal{F}}_{k-1}, \text{sign}\,(f_1(x)) \neq \text{sign}\,(f_2(x)) \right\}$;
**if** $\hat{A}_k \cap \text{supp}(\Pi) = \emptyset$ **or** $LB < \lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor$ **then**
        **break; output** $\hat{g} := \text{sign}\, \hat{\eta}_{k-1}$
        **else**
**for** $i = 1 \dots \lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor$
**sample i.i.d** $\left( X_i^{(k)}, Y_i^{(k)} \right)$ **with** $X_i^{(k)} \sim \hat{\Pi}_k := \Pi(dx | x \in \hat{A}_k)$;
**end for**;
$LB := LB - \lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor$;
$\hat{P}_k := \frac{1}{\lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor} \sum_i \delta_{X_i^{(k)}, Y_i^{(k)}}$      // "active" empirical measure
$\hat{m}_k := \text{argmin}_{\ m \geq \hat{m}_{k-1}} \left[ \inf_{f \in \mathcal{F}_m} \hat{P}_k (Y - f(X))^2 + K_1 \frac{2^{dm}\Pi(\hat{A}_k) + s(m - \hat{m}_{k-1}, N, \alpha)}{\lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor} \right]$
$\hat{\eta}_k := \hat{\eta}_{\hat{m}_k, \hat{A}_k}$        // see (12)
$\delta_k := \tilde{D} \cdot \log^2 \frac{N}{\alpha} \sqrt{\frac{2^{d\hat{m}_k}}{N_k}}$;
$\hat{\mathcal{F}}_k := \left\{ f \in \mathcal{F}_{\hat{m}_k} :\ f|_{\hat{A}_k} \in \mathcal{F}_{\infty, \hat{A}_k}(\hat{\eta}_k; \delta_k),\ \ f|_{[0,1]^d \setminus \hat{A}_k} \equiv \hat{\eta}_{k-1}|_{[0,1]^d \setminus \hat{A}_k} \right\}$;
**end;**
```

Table 1: Active Learning Algorithm

As a first step towards the analysis of **Algorithm 1b**, let us prove the useful fact about the general model selection scheme. Given an iid sample $(X_i, Y_i),\ i \leq N$, set $s_m = m(s + \log\log_2 N),\ m \geq 1$ and

$$\hat{m} := \hat{m}(s) = \text{argmin}_{\ m \in \mathcal{J}(N)} \left[\inf_{f \in \mathcal{F}_m} P_N (Y - f(X))^2 + K_1 \frac{2^{dm} + s_m}{N} \right],$$

$$\bar{m} := \min \left\{ m \geq 1 :\ \inf_{f \in \mathcal{F}_m} \mathbb{E}(f(X) - \eta(X))^2 \leq K_2 \frac{2^{dm}}{N} \right\}.$$

Theorem 11 *There exist an absolute constant K_1 big enough such that, with probability $\geq 1 - e^{-s}$,*

$$\hat{m} \leq \bar{m}.$$

Proof See Appendix B. ■

Straightforward application of this result immediately yields the following:

Corollary 12 *Suppose $\eta(x) \in \Sigma(\beta, L, [0,1]^d)$. Then, with probability $\geq 1 - e^{-s}$,*

$$2^{\hat{m}} \leq C_1 \cdot N^{\frac{1}{2\beta+d}}$$

Proof By definition of $\bar{m}$, we have

$$\bar{m} \leq 1 + \max\left\{m : \inf_{f \in \mathcal{F}_m} \mathbb{E}(f(X) - \eta(X))^2 > K_2 \frac{2^{dm}}{N}\right\} \leq$$
$$\leq 1 + \max\left\{m : L^2 2^{-2\beta m} > K_2 \frac{2^{dm}}{N}\right\},$$

and the claim follows. ■

With this bound in hand, we are ready to formulate and prove the main result of this section:

Theorem 13 *Suppose that $P \in \mathcal{P}_U^*(\beta, \gamma)$ with $B_1 \leq \log N$, $B_2 \geq \log^{-1} N$ and $\beta\gamma \leq d$. Then, with probability $\geq 1 - 3\alpha$, the classifier $\hat{g}$ returned by* **Algorithm 1b** *with label budget N satisfies*

$$R_P(\hat{g}) - R^* \leq \text{Const} \cdot N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}} \log^p \frac{N}{\alpha},$$

where $p \leq \frac{2\beta\gamma(1+\gamma)}{2\beta+d-\beta\gamma}$ and B_1, B_2 are the constants from (11) and Assumption 2.

Remarks

1. Note that when $\beta\gamma > \frac{d}{3}$, $N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}}$ is a *fast rate*, that is, faster than $N^{-\frac{1}{2}}$; at the same time, the passive learning rate $N^{-\frac{\beta(1+\gamma)}{2\beta+d}}$ is guaranteed to be fast only when $\beta\gamma > \frac{d}{2}$, see Audibert and Tsybakov (2005).

2. For $\hat{\alpha} \simeq N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}}$ **Algorithm 1b** returns a classifier $\hat{g}_{\hat{\alpha}}$ that satisfies

$$\mathbb{E}R_P(\hat{g}_{\hat{\alpha}}) - R^* \leq \text{Const} \cdot N^{-\frac{\beta(1+\gamma)}{2\beta+d-\beta\gamma}} \log^p N.$$

This is a direct corollary of Theorem 13 and the inequality

$$\mathbb{E}|Z| \leq t + \|Z\|_\infty \Pr(|Z| \geq t).$$

Proof Our main goal is to construct high probability bounds for the size of the active sets defined by **Algorithm 1b**. In turn, these bounds depend on the size of the confidence bands for $\eta(x)$, and the previous result(Theorem 11) is used to obtain the required estimates. Suppose L is the number of steps performed by the algorithm before termination; clearly, $L \leq N$.

Let $N_k^{\text{act}} := \lfloor N_k \cdot \Pi(\hat{A}_k) \rfloor$ be the number of labels requested on k-th step of the algorithm: this choice guarantees that the "density" of labeled examples doubles on every step.

Claim: the following bound for the size of the active set holds uniformly for all $2 \leq k \leq L$ with probability at least $1 - 2\alpha$:

$$\Pi(\hat{A}_k) \leq C N_k^{-\frac{\beta\gamma}{2\beta+d}} \left(\log \frac{N}{\alpha}\right)^{2\gamma}. \tag{14}$$

It is not hard to finish the proof assuming (14) is true: indeed, it implies that the number of labels requested on step k satisfies

$$N_k^{\text{act}} = \lfloor N_k \Pi(\hat{A}_k) \rfloor \le C \cdot N_k^{\frac{2\beta+d-\beta\gamma}{2\beta+d}} \left(\log \frac{N}{\alpha} \right)^{2\gamma}$$

with probability $\ge 1-2\alpha$. Since $\sum_k N_k^{\text{act}} \le N$, one easily deduces that on the last iteration L we have

$$N_L \ge c \left(\frac{N}{\log^{2\gamma}(N/\alpha)} \right)^{\frac{2\beta+d}{2\beta+d-\beta\gamma}} \tag{15}$$

To obtain the risk bound of the theorem from here, we apply [2] inequality (3) from Proposition 6:

$$R_P(\hat{g}) - R^* \le D_1 \|(\hat{\eta}_L - \eta) \cdot I\{\text{sign}\, \hat{\eta}_L \ne \text{sign}\, \eta\}\|_\infty^{1+\gamma}. \tag{16}$$

It remains to estimate $\|\hat{\eta}_L - \eta\|_{\infty, \hat{A}_L}$: we will show below while proving (14) that

$$\|\hat{\eta}_L - \eta\|_{\infty, \hat{A}_L} \le C \cdot N_L^{-\frac{\beta}{2\beta+d}} \log^2 \frac{N}{\alpha}.$$

Together with (15) and (16), it implies the final result.

To finish the proof, it remains to establish (14). Recall that $\bar{\eta}_k$ stands for the $L_2(\Pi)$ - projection of η onto $\mathcal{F}_{\hat{m}_k}$. An important role in the argument is played by the bound on the $L_2(\hat{\Pi}_k)$ - norm of the "bias" $(\bar{\eta}_k - \eta)$: together with *Assumption* 2, it allows to estimate $\|\bar{\eta}_k - \eta\|_{\infty, \hat{A}_k}$. The required bound follows from the following oracle inequality: there exists an event $\mathcal{B}$ of probability $\ge 1-\alpha$ such that on this event for every $1 \le k \le L$

$$\begin{aligned} \|\bar{\eta}_k - \eta\|^2_{L_2(\hat{\Pi}_k)} \le \inf_{m \ge \hat{m}_{k-1}} \Bigg[& \inf_{f \in \mathcal{F}_m} \|f - \eta\|^2_{L_2(\hat{\Pi}_k)} + \\ & + K_1 \frac{2^{dm}\Pi(\hat{A}_k) + (m - \hat{m}_{k-1}) \log(N/\alpha)}{N_k \Pi(\hat{A}_k)} \Bigg]. \end{aligned} \tag{17}$$

It general form, this inequality is given by Theorem 6.1 in Koltchinskii (2011) and provides the estimate for $\|\hat{\eta}_k - \eta\|_{L_2(\hat{\Pi}_k)}$, so it automatically implies the weaker bound for the bias term only. To deduce (17), we use the mentioned general inequality L times(once for every iteration) and the union bound. The quantity $2^{dm}\Pi(\hat{A}_k)$ in (17) plays the role of the dimension, which is justified below. Let $k \ge 1$ be fixed. For $m \ge \hat{m}_{k-1}$, consider hypothesis classes

$$\mathcal{F}_m|_{\hat{A}_k} := \left\{ f I_{\hat{A}_k},\ f \in \mathcal{F}_m \right\}.$$

An obvious but important fact is that for $P \in \mathcal{P}_U(\beta, \gamma)$, the dimension of $\mathcal{F}_m|_{\hat{A}_k}$ is bounded by $u_1^{-1} \cdot 2^m \Pi(\hat{A}_k)$: indeed,

$$\Pi(\hat{A}_k) = \sum_{j: R_j \cap \hat{A}_k \ne \emptyset} \Pi(R_j) \ge u_1 2^{-dm} \cdot \#\left\{ j : R_j \cap \hat{A}_k \ne \emptyset \right\},$$

[2] alternatively, inequality (4) can be used but results in a slightly inferior logarithmic factor.

hence

$$\dim \mathcal{F}_m|_{\hat{A}_k} = \#\left\{j : R_j \cap \hat{A}_k \neq \emptyset\right\} \leq u_1^{-1} \cdot 2^m \Pi(\hat{A}_k). \tag{18}$$

Theorem 11 applies conditionally on $\left\{X_i^{(j)}\right\}_{i=1}^{N_j}$, $j \leq k-1$ with sample of size N_k^{act} and $s = \log(N/\alpha)$: to apply the theorem, note that, by definition of $\hat{A}_k$, it is independent of $X_i^{(k)}$, $i = 1 \dots N_k^{\text{act}}$. Arguing as in Corollary 12 and using (18), we conclude that the following inequality holds with probability $\geq 1 - \frac{\alpha}{N}$ for every fixed k:

$$2^{\hat{m}_k} \leq C \cdot N_k^{\frac{1}{2\beta+d}}. \tag{19}$$

Let $\mathcal{E}_1$ be an event of probability $\geq 1-\alpha$ such that on this event bound (19) holds for every step k, $k \leq L$ and let $\mathcal{E}_2$ be an event of probability $\geq 1-\alpha$ on which inequalities (17) are satisfied. Suppose that event $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs and let k_0 be a fixed arbitrary integer $2 \leq k_0 \leq L+1$. It is enough to assume that $\hat{A}_{k_0-1}$ is nonempty(otherwise, the bound trivially holds), so that it contains at least one cube with sidelength $2^{-\hat{m}_{k_0-2}}$ and

$$\Pi(\hat{A}_{k_0-1}) \geq u_1 2^{-d\hat{m}_{k_0-1}} \geq cN_{k_0}^{-\frac{d}{2\beta+d}}. \tag{20}$$

Consider inequality (17) with $k = k_0 - 1$ and $2^m \simeq N_{k_0-1}^{\frac{1}{2\beta+d}}$. By (20), we have

$$\|\bar{\eta}_{k_0-1} - \eta\|^2_{L_2(\hat{\Pi}_{k_0-1})} \leq CN_{k_0-1}^{-\frac{2\beta}{2\beta+d}} \log^2 \frac{N}{\alpha}. \tag{21}$$

For convenience and brevity, denote $\Omega := \operatorname{supp}(\Pi)$. Now *Assumption 2* comes into play: it implies, together with (21) that

$$CN_{k_0-1}^{-\frac{\beta}{2\beta+d}} \log \frac{N}{\alpha} \geq \|\bar{\eta}_{k_0-1} - \eta\|_{L_2(\hat{\Pi}_{k_0-1})} \geq B_2 \|\bar{\eta}_{k_0-1} - \eta\|_{\infty, \Omega \cap \hat{A}_{k_0-1}}. \tag{22}$$

To bound

$$\|\hat{\eta}_{k_0-1}(x) - \bar{\eta}_{k_0-1}(x)\|_{\infty, \Omega \cap \hat{A}_{k_0-1}}$$

we apply Proposition 10. Recall that $\hat{m}_{k_0-1}$ depends only on the subsample $S_{k_0-1,1}$ but not on $S_{k_0-1,2}$. Let

$$\mathcal{T}_k := \left\{ \left\{X_i^{(j)}, Y_i^{(j)}\right\}_{i=1}^{N_j^{\text{act}}}, \ j \leq k-1 \right\} \cup S_{k,1}$$

be the random vector that defines $\hat{A}_k$ and resolution level $\hat{m}_k$. Note that for any x,

$$\mathbb{E}(\hat{\eta}_{k_0-1}(x) | \mathcal{T}_{k_0-1}) \overset{\text{a.s.}}{=} \bar{\eta}_{\hat{m}_{k_0-1}}(x).$$

Proposition 10 thus implies

$$\Pr\left(\max_{x \in \Omega \cap \hat{A}_{k_0-1}} |\hat{\eta}_{k_0-1}(x) - \bar{\eta}_{\hat{m}_{k_0-1}}(x)| \geq Kt \sqrt{\frac{2^{d\hat{m}_{k_0-1}}}{N_{k_0-1}}} \,\middle|\, \mathcal{T}_{k_0-1} \right) \leq$$
$$\leq N \exp\left(\frac{-t^2}{2(1 + \frac{t}{3}C_3)} \right).$$

Choosing $t = c\log(N/\alpha)$ and taking expectation, the inequality(now unconditional) becomes

$$\Pr\left(\max_{x\in\Omega\cap\hat{A}_{k_0-1}} |\hat{\eta}_{\hat{m}_{k_0-1}}(x) - \bar{\eta}_{\hat{m}_{k_0-1}}(x)| \le K\sqrt{\frac{2^{d\hat{m}_{k_0-1}}\log^2(N/\alpha)}{N_{k_0-1}}}\right) \ge 1-\alpha. \tag{23}$$

Let $\mathcal{E}_3$ be the event on which (23) holds true. Combined, the estimates (19),(22) and (23) imply that on $\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3$

$$\begin{aligned}\|\eta - \hat{\eta}_{k_0-1}\|_{\infty,\Omega\cap\hat{A}_{k_0-1}} &\le \|\eta - \bar{\eta}_{k_0-1}\|_{\infty,\Omega\cap\hat{A}_{k_0-1}} + \|\bar{\eta}_{k_0-1} - \hat{\eta}_{k_0-1}\|_{\infty,\Omega\cap\hat{A}_{k_0-1}} \\ &\le \frac{C}{B_2}N_{k_0-1}^{-\frac{\beta}{2\beta+d}}\log\frac{N}{\alpha} + K\sqrt{\frac{2^{d\hat{m}_{k_0-1}}\log^2(N/\alpha)}{N_{k_0-1}}} \le \\ &\le (K+C)\cdot N_{k_0-1}^{-\frac{\beta}{2\beta+d}}\log^2\frac{N}{\alpha},\end{aligned} \tag{24}$$

where we used the assumption $B_2 \ge \log^{-1} N$. Now the width of the confidence band is defined via

$$\delta_k := 2(K+C)\cdot N_{k_0-1}^{-\frac{\beta}{2\beta+d}}\log^2\frac{N}{\alpha} \tag{25}$$

(in particular, $\tilde{D}$ from **Algorithm 1a** is equal to $2(K+C)$). With the bound (24) available, it is straightforward to finish the proof of the claim. Indeed, by (25) and the definition of the active set, the necessary condition for $x \in \Omega\cap\hat{A}_{k_0}$ is

$$|\eta(x)| \le 3(K+C)\cdot N_{k_0-1}^{-\frac{\beta}{2\beta+d}}\log^2\frac{N}{\alpha},$$

so that

$$\begin{aligned}\Pi(\hat{A}_{k_0}) = \Pi(\Omega\cap\hat{A}_{k_0}) &\le \Pi\left(|\eta(x)| \le 3(K+C)\cdot N_{k_0-1}^{-\frac{\beta}{2\beta+d}}\log^2\frac{N}{\alpha}\right) \le \\ &\le \tilde{B}N_{k_0-1}^{-\frac{\beta\gamma}{2\beta+d}}\log^{2\gamma}\frac{N}{\alpha}.\end{aligned}$$

by the low noise assumption. This completes the proof of the claim since $\Pr(\mathcal{E}_1\cap\mathcal{E}_2\cap\mathcal{E}_3) \ge 1-3\alpha$. ■

We conclude this section by discussing running time of the active learning algorithm. Assume that the algorithm has access to the sampling subroutine that, given $A \subset [0,1]^d$ with $\Pi(A) > 0$, generates i.i.d. (X_i, Y_i) with $X_i \sim \Pi(dx|x\in A)$.

Proposition 14 *The running time of* **Algorithm 1a(1b)** *with label budget N is*

$$O(dN\log^2 N).$$

Remark In view of Theorem 13, the running time required to output a classifier $\hat{g}$ such that $R_P(\hat{g}) - R^* \le \varepsilon$ with probability $\ge 1-\alpha$ is

$$O\left(\left(\frac{1}{\varepsilon}\right)^{\frac{2\beta+d-\beta\gamma}{\beta(1+\gamma)}}\text{poly}\left(\log\frac{1}{\varepsilon\alpha}\right)\right).$$

Proof We will use the notations of Theorem 13. Let N_k^{act} be the number of labels requested by the algorithm on step k. The resolution level $\hat{m}_k$ is always chosen such that $\hat{A}_k$ is partitioned into at most N_k^{act} dyadic cubes, see (13). This means that the estimator $\hat{\eta}_k$ takes at most N_k^{act} distinct values. The key observation is that for any k, the active set $\hat{A}_{k+1}$ is always represented as the union of a finite number(at most N_k^{act}) of dyadic cubes: to determine if a cube $R_j \subset \hat{A}_{k+1}$, it is enough to take a point $x \in R_j$ and compare $\text{sign}(\hat{\eta}_k(x) - \delta_k)$ with $\text{sign}(\hat{\eta}_k(x) + \delta_k)$: $R_j \in \hat{A}_{k+1}$ only if the signs are different(so that the confidence band crosses zero level). This can be done in $O(N_k^{\text{act}})$ steps.

Next, resolution level $\hat{m}_k$ can be found in $O(N_k^{\text{act}} \log^2 N)$ steps: there are at most $\log_2 N_k^{\text{act}}$ models to consider; for each m, $\inf_{f \in \mathcal{F}_m} \hat{P}_k(Y - f(X))^2$ is found explicitly and is achieved for the piecewise-constant $\hat{f}(x) = \frac{\sum_i Y_i^{(k)} I_{R_j}(X_i^{(k)})}{\sum_i I_{R_j}(X_i^{(k)})}$, $x \in R_j$. Sorting of the data required for this computation is done in $O(dN_k^{\text{act}} \log N)$ steps for each m, so the whole k-th iteration running time is $O(dN_k^{\text{act}} \log^2 N)$. Since $\sum\limits_k N_k^{\text{act}} \leq N$, the result follows. ■

4. Conclusion and Open Problems

We have shown that active learning can significantly improve the quality of a classifier over the passive algorithm for a large class of underlying distributions. Presented method achieves fast rates of convergence for the excess risk, moreover, it is adaptive(in the certain range of smoothness and noise parameters) and involves minimization only with respect to quadratic loss(rather than the $0-1$ loss).

The natural question related to our results is:

- Can we implement adaptive smooth estimators in the learning algorithm to extend our results beyond the case $\beta \leq 1$?

The answer to this second question is so far an open problem. Our conjecture is that the correct rate of convergence for the excess risk is, up to logarithmic factors, $N^{-\frac{\beta(1+\gamma)}{2\beta+d-\gamma(\beta\wedge 1)}}$, which coincides with presented results for $\beta \leq 1$. This rate can be derived from an argument similar to the proof of Theorem 13 under the assumption that on every step k one could construct an estimator $\hat{\eta}_k$ with

$$\|\eta - \hat{\eta}_k\|_{\infty, \hat{A}_k} \lesssim N_k^{-\frac{\beta}{2\beta+d}}.$$

At the same time, the active set associated to $\hat{\eta}_k$ should maintain some structure which is suitable for the iterative nature of the algorithm. Transforming these ideas into a rigorous proof is a goal of our future work.

Acknowledgments

I want to express my sincere gratitude to my Ph.D. advisor, Dr. Vladimir Koltchinskii, for his support and numerous helpful discussions.

I am grateful to the anonymous reviewers for carefully reading the manuscript. Their insightful and wise suggestions helped to improve the quality of presentation and results.

This project was supported by NSF Grant DMS-0906880 and by the Algorithms and Randomness Center, Georgia Institute of Technology, through the ARC Fellowship.

Appendix A. Functions Satisfying Assumption 2

In the propositions below, we will assume for simplicity that the marginal distribution Π is absolutely continuous with respect to Lebesgue measure with density $p(x)$ such that

$$0 < p_1 \le p(x) \le p_2 < \infty \text{ for all } x \in [0,1]^d.$$

Given $t \in (0,1]$, define $A_t := \{x : \ |\eta(x)| \le t\}$.

Proposition 15 *Suppose η is Lipschitz continuous with Lipschitz constant S. Assume also that for some $t_* > 0$ we have*

(a) $\Pi\left(A_{t_*/3}\right) > 0$*;*

(b) η *is twice differentiable for all* $x \in A_{t_*}$*;*

(c) $\inf_{x \in A_{t_*}} \|\nabla\eta(x)\|_1 \ge s > 0$;

(d) $\sup_{x\in A_{t_*}} \|D^2\eta(x)\| \le C < \infty$ *where* $\|\cdot\|$ *is the operator norm.*

Then η *satisfies Assumption 2.*

Proof By intermediate value theorem, for any cube $R_i, \ 1 \le i \le 2^{dm}$ there exists $x_0 \in R_i$ such that $\bar\eta_m(x) = \eta(x_0), \ x \in R_i$. This implies

$$\begin{aligned}|\eta(x) - \bar\eta_m(x)| = |\eta(x) - \eta(x_0)| = |\nabla\eta(\xi)\cdot(x-x_0)| \le \\ \le \|\nabla\eta(\xi)\|_1\|x-x_0\|_\infty \le S\cdot 2^{-m}.\end{aligned}$$

On the other hand, if $R_i \subset A_{t_*}$ then

$$\begin{aligned}|\eta(x) - \bar\eta_m(x)| &= |\eta(x) - \eta(x_0)| = \\ &= |\nabla\eta(x_0)\cdot(x-x_0) + \frac{1}{2}[D^2\eta(\xi)](x-x_0)\cdot(x-x_0)| \ge \\ &\ge |\nabla\eta(x_0)\cdot(x-x_0)| - \frac{1}{2}\sup_{\xi}\|D^2\eta(\xi)\|\max_{x\in R_i}\|x-x_0\|_2^2 \ge \qquad (26)\\ &\ge |\nabla\eta(x_0)\cdot(x-x_0)| - C_1 2^{-2m}.\end{aligned}$$

Note that a strictly positive continuous function

$$h(y,u) = \int_{[0,1]^d} (u\cdot(x-y))^2 dx$$

achieves its minimal value $h_* > 0$ on a compact set $[0,1]^d \times \left\{u \in \mathbb{R}^d : \|u\|_1 = 1\right\}$. This implies(using (26) and the inequality $(a-b)^2 \ge \frac{a^2}{2} - b^2$)

$$\begin{aligned}&\Pi^{-1}(R_i)\int_{R_i}(\eta(x) - \bar\eta_m(x))^2 p(x)dx \ge \\ &\ge \frac{1}{2}(p_2 2^{dm})^{-1}\int_{R_i}(\nabla\eta(x_0)\cdot(x-x_0))^2 p_1 dx - C_1^2 2^{-4m} \ge \\ &\ge \frac{1}{2}\frac{p_1}{p_2}\|\nabla\eta(x_0)\|_1^2 2^{-2m}\cdot h_* - C_1^2 2^{-4m} \ge c_2 2^{-2m} \qquad \text{for } m \ge m_0.\end{aligned}$$

Now take a set $A \in \sigma(\mathcal{F}_m)$, $m \geq m_0$ from *Assumption* 2. There are 2 possibilities: either $A \subset A_{t_*}$ or $A \supset A_{t_*/3}$. In the first case the computation above implies

$$\int\limits_{[0,1]^d} (\eta - \bar{\eta}_m)^2 \Pi(dx|x \in A) \geq c_2 2^{-2m} = \frac{c_2}{S^2} S^2 2^{-2m} \geq$$
$$\geq \frac{c_2}{S^2} \|\eta - \bar{\eta}_m\|^2_{\infty,A}.$$

If the second case occurs, note that, since $\left\{x : 0 < |\eta(x)| < \frac{t_*}{3}\right\}$ has nonempty interior, it must contain a dyadic cube R_* with edge length 2^{-m_*}. Then for any $m \geq \max(m_0, m_*)$

$$\int\limits_{[0,1]^d} (\eta - \bar{\eta}_m)^2 \Pi(dx|x \in A) \geq$$
$$\geq \Pi^{-1}(A) \int\limits_{R_*} (\eta - \bar{\eta}_m)^2 \Pi(dx) \geq \frac{c_2}{4} 2^{-2m} \Pi(R_*) \geq$$
$$\geq \frac{c_2}{S^2} \Pi(R_*) \|\eta - \bar{\eta}_m\|^2_{\infty,A}$$

and the claim follows. ■

The next proposition describes conditions which allow functions to have vanishing gradient on decision boundary but requires convexity and regular behaviour of the gradient.

Everywhere below, $\nabla\eta$ denotes the subgradient of a convex function η.

For $0 < t_1 < t_2$, define $G(t_1, t_2) := \frac{\sup\limits_{x \in A_{t_2} \setminus A_{t_1}} \|\nabla\eta(x)\|_1}{\inf\limits_{x \in A_{t_2} \setminus A_{t_1}} \|\nabla\eta(x)\|_1}$. In case when $\nabla\eta(x)$ is not unique, we choose a representative that makes $G(t_1, t_2)$ as small as possible.

Proposition 16 *Suppose $\eta(x)$ is Lipschitz continuous with Lipschitz constant S. Moreover, assume that there exists $t_* > 0$ and $q : (0,\infty) \mapsto (0,\infty)$ such that $A_{t_*} \subset (0,1)^d$ and*

(a) $b_1 t^\gamma \leq \Pi(A_t) \leq b_2 t^\gamma \ \forall t < t_*$;

(b) For all $0 < t_1 < t_2 \leq t_*$, $G(t_1, t_2) \leq q\left(\frac{t_2}{t_1}\right)$;

(c) Restriction of η to any convex subset of A_{t_*} is convex.

Then η satisfies Assumption 2.

Remark *The statement remains valid if we replace η by $|\eta|$ in (c).*

Proof Assume that for some $t \leq t_*$ and $k > 0$

$$R \subset A_t \setminus A_{t/k}$$

is a dyadic cube with edge length 2^{-m} and let x_0 be such that $\bar{\eta}_m(x) = \eta(x_0)$, $x \in R$. Note that η is convex on R due to (c). Using the subgradient inequality $\eta(x) - \eta(x_0) \geq \nabla\eta(x_0) \cdot (x - x_0)$, we obtain

$$\int\limits_R (\eta(x) - \eta(x_0))^2 d\Pi(x) \geq \int\limits_R (\eta(x) - \eta(x_0))^2 I\left\{\nabla\eta(x_0) \cdot (x - x_0) \geq 0\right\} d\Pi(x)$$
$$\geq \int\limits_R \left(\nabla\eta(x_0) \cdot (x - x_0)\right)^2 I\left\{\nabla\eta(x_0) \cdot (x - x_0) \geq 0\right\} d\Pi(x). \tag{27}$$

The next step is to show that under our assumptions x_0 can be chosen such that

$$\mathrm{dist}_\infty(x_0, \partial R) \geq \nu 2^{-m}, \tag{28}$$

where $\nu = \nu(k)$ is independent of m. In this case any part of R cut by a hyperplane through x_0 contains half of a ball $B(x_0, r_0)$ of radius $r_0 = \nu(k)2^{-m}$ and the last integral in (27) can be further bounded below to get

$$\begin{aligned}\int_R (\eta(x) - \eta(x_0))^2 d\Pi(x) \geq \frac{1}{2} \int_{B(x_0,r_0)} (\nabla\eta(x_0)\cdot(x-x_0))^2 p_1 dx \geq \\ \geq c(k)\|\nabla\eta(x_0)\|_1^2 2^{-2m}2^{-dm}.\end{aligned} \tag{29}$$

It remains to show (28). Assume that for all y such that $\eta(y) = \eta(x_0)$ we have

$$\mathrm{dist}_\infty(y, \partial R) \leq \delta 2^{-m}$$

for some $\delta > 0$. This implies that the boundary of the convex set

$$\{x \in R : \eta(x) \leq \eta(x_0)\}$$

is contained in $R_\delta := \{x \in R : \mathrm{dist}_\infty(x, \partial R) \leq \delta 2^{-m}\}$. There are two possibilities: either $\{x \in R : \eta(x) \leq \eta(x_0)\} \supseteq R \setminus R_\delta$ or $\{x \in R : \eta(x) \leq \eta(x_0)\} \subset R_\delta$.

We consider the first case only(the proof in the second case is similar). First, note that by (b) for all $x \in R_\delta$ $\|\nabla\eta(x)\|_1 \leq q(k)\|\nabla\eta(x_0)\|_1$ and

$$\begin{aligned}\eta(x) &\leq \eta(x_0) + \|\nabla\eta(x)\|_1 \delta 2^{-m} \leq \\ &\leq \eta(x_0) + q(k)\|\nabla\eta(x_0)\|_1 \delta 2^{-m}.\end{aligned} \tag{30}$$

Let x_c be the center of the cube R and u - the unit vector in direction $\nabla\eta(x_c)$. Observe that

$$\begin{aligned}\eta(x_c + (1-3\delta)2^{-m}u) - \eta(x_c) \geq \nabla\eta(x_c)\cdot(1-3\delta)2^{-m}u = \\ = (1-3\delta)2^{-m}\|\nabla\eta(x_c)\|_2.\end{aligned}$$

On the other hand, $x_c + (1-3\delta)2^{-m}u \in R \setminus R_\delta$ and

$$\eta(x_c + (1-3\delta)2^{-m}u) \leq \eta(x_0),$$

hence $\eta(x_c) \leq \eta(x_0) - c(1-3\delta)2^{-m}\|\nabla\eta(x_c)\|_1$. Consequently, for all

$$x \in B(x_c, \delta) := \left\{x : \|x - x_c\|_\infty \leq \frac{1}{2}c2^{-m}(1-3\delta)\right\}$$

we have

$$\begin{aligned}\eta(x) &\leq \eta(x_c) + \|\nabla\eta(x_c)\|_1\|x - x_c\|_\infty \leq \\ &\leq \eta(x_0) - \frac{1}{2}c2^{-m}(1-3\delta)\|\nabla\eta(x_c)\|_1.\end{aligned} \tag{31}$$

Finally, recall that $\eta(x_0)$ is the average value of η on R. Together with (30) and (31) this gives

$$\begin{aligned}\Pi(R)\eta(x_0) &= \int\limits_R \eta(x)d\Pi = \int\limits_{R_\delta} \eta(x)d\Pi + \int\limits_{R\setminus R_\delta} \eta(x)d\Pi \le \\ &\le (\eta(x_0) + q(k)\|\nabla\eta(x_0)\|_1 \delta 2^{-m})\Pi(R_\delta) + \\ &+ (\eta(x_0) - c_2 2^{-m}(1-3\delta)\|\nabla\eta(x_0)\|_1)\Pi\left(B(x_c,\delta)\right) + \\ &+ \eta(x_0)\Pi(R\setminus(R_\delta \cup B(x_c,\delta))) = \\ &= \Pi(R)\eta(x_0) + q(k)\|\nabla\eta(x_0)\|_1\delta 2^{-m}\Pi(R_\delta) - \\ &- c_2 2^{-m}(1-3\delta)\|\nabla\eta(x_0)\|_1\Pi\left(B(x_c,\delta)\right).\end{aligned}$$

Since $\Pi(R_\delta) \le p_2 2^{-dm}$ and $\Pi(B(x_c,\delta)) \ge c_3 2^{-dm}(1-3\delta)^d$, the inequality above implies

$$c_4 q(k)\delta \ge (1-3\delta)^{d+1}$$

which is impossible for small δ (e.g., for $\delta < \frac{c}{q(k)(3d+4)}$).

Let A be a set from condition 2. If $A \supseteq A_{t_*/3}$, then there exists a dyadic cube R_* with edge length 2^{-m_*} such that $R_* \subset A_{t_*/3} \setminus A_{t_*/k}$ for some $k > 0$, and the claim follows from (29) as in Proposition 15.

Assume now that $A_t \subset A \subset A_{3t}$ and $3t \le t_*$. Condition (a) of the proposition implies that for any $\varepsilon > 0$ we can choose $k(\varepsilon) > 0$ large enough so that

$$\Pi(A\setminus A_{t/k}) \ge \Pi(A) - b_2(t/k)^\gamma \ge \Pi(A) - \frac{b_2}{b_1}k^{-\gamma}\Pi(A_t) \ge (1-\varepsilon)\Pi(A). \tag{32}$$

This means that for any partition of A into dyadic cubes R_i with edge length 2^{-m} at least half of them satisfy

$$\Pi(R_i \setminus A_{t/k}) \ge (1-c\varepsilon)\Pi(R_i). \tag{33}$$

Let I be the index set of cardinality $|I| \ge c\Pi(A)2^{dm-1}$ such that (33) is true for $i \in I$. Since $R_i \cap A_{t/k}$ is convex, there exists[3] $z = z(\varepsilon) \in \mathbb{N}$ such that for any such cube R_i there exists a dyadic sub-cube with edge length $2^{-(m+z)}$ entirely contained in $R_i \setminus A_{t/k}$:

$$T_i \subset R_i \setminus A_{t/k} \subset A_{3t} \setminus A_{t/k}.$$

It follows that $\Pi\big(\bigcup\limits_i T_i\big) \ge \tilde{c}(\varepsilon)\Pi(A)$. Recall that condition (b) implies

$$\frac{\sup\limits_{x\in\cup_i T_i} \|\nabla\eta(x)\|_1}{\inf\limits_{x\in\cup_i T_i} \|\nabla\eta(x)\|_1} \le q(3k).$$

Finally, $\sup\limits_{x\in A_{3t}} \|\nabla\eta(x)\|_2$ is attained at the boundary point, that is for some $x_* : |\eta(x_*)| = 3t$, and by (b)

$$\sup_{x\in A_{3t}} \|\nabla\eta(x)\|_1 \le \sqrt{d}\|\nabla\eta(x_*)\|_1 \le q(3k)\sqrt{d} \inf_{x\in A_{3t}\setminus A_{t/k}} \|\nabla\eta(x)\|_1.$$

[3] If, on the contrary, every sub-cube with edge length $2^{-(m+z)}$ contains a point from $A_{t/k}$, then $A_{t/k}$ must contain the convex hull of these points which would contradict (32) for large z.

Application of (29) to every cube T_i gives

$$\sum_{i\in I}\int_{T_i}(\eta(x)-\bar\eta_{m+z}(x))^2 d\Pi(x)\geq c_1(k)\Pi(A)|I|\inf_{x\in A_{3t}\setminus A_{t/k}}\|\nabla\eta(x)\|_1^2 2^{-2m}2^{-dm}\geq$$

$$\geq c_2(k)\Pi(A)\sup_{x\in A_{3t}}\|\nabla\eta(x)\|_1^2 2^{-2m}\geq c_3(k)\Pi(A)\|\eta-\bar\eta(m)\|^2_{\infty,A}$$

concluding the proof. ■

Appendix B. Proof of Theorem 11

The main ideas of this proof, which significantly simplifies and clarifies initial author's version, are due to V. Koltchinskii. For convenience and brevity, let us introduce additional notations. Recall that

$$s_m = m(s+\log\log_2 N).$$

Let

$$\tau_N(m,s) := K_1\frac{2^{dm}+s_m}{N},$$
$$\pi_N(m,s) := K_2\frac{2^{dm}+s+\log\log_2 N}{N}.$$

By $\mathcal{E}_P(\mathcal{F},f)$ (or $\mathcal{E}_{P_N}(\mathcal{F},f)$) we denote the excess risk of $f\in\mathcal{F}$ with respect to the true (or empirical) measure:

$$\mathcal{E}_P(\mathcal{F},f) := P(y-f(x))^2-\inf_{g\in\mathcal{F}}P(y-g(x))^2,$$
$$\mathcal{E}_{P_N}(\mathcal{F},f) := P_N(y-f(x))^2-\inf_{g\in\mathcal{F}}P_N(y-g(x))^2.$$

It follows from Theorem 4.2 in Koltchinskii (2011) and the union bound that there exists an event $\mathcal{B}$ of probability $\geq 1-e^{-s}$ such that on this event the following holds for all m such that $dm\leq\log N$:

$$\begin{aligned}&\mathcal{E}_P(\mathcal{F}_m,\hat f_{\hat m})\leq\pi_N(m,s),\\ \forall f\in\mathcal{F}_m,\quad&\mathcal{E}_P(\mathcal{F}_m,f)\leq 2(\mathcal{E}_{P_N}(\mathcal{F}_m,f)\vee\pi_N(m,s)),\\ \forall f\in\mathcal{F}_m,\quad&\mathcal{E}_{P_N}(\mathcal{F}_m,f)\leq\frac{3}{2}(\mathcal{E}_P(\mathcal{F}_m,f)\vee\pi_N(m,s)).\end{aligned}\tag{34}$$

We will show that on $\mathcal{B}$, $\{\hat m\leq\bar m\}$ holds. Indeed, assume that, on the contrary, $\hat m>\bar m$; by definition of $\hat m$, we have

$$P_N(Y-\hat f_{\hat m})^2+\tau_N(\hat m,s)\leq P_N(Y-\hat f_{\bar m})^2+\tau_N(\bar m,s),$$

which implies

$$\mathcal{E}_{P_N}(\mathcal{F}_{\hat m},\hat f_{\bar m})\geq\tau_N(\hat m,s)-\tau_N(\bar m,s)>3\pi_N(\hat m,s)$$

for K_1 big enough. By (34),

$$\mathcal{E}_{P_N}(\mathcal{F}_{\hat m},\hat f_{\bar m})=\inf_{f\in\mathcal{F}_{\bar m}}\mathcal{E}_{P_N}(\mathcal{F}_{\hat m},f)\leq\frac{3}{2}\left(\inf_{f\in\mathcal{F}_{\bar m}}\mathcal{E}_P(\mathcal{F}_{\hat m},f)\vee\pi_N(\hat m,s)\right),$$

and combination the two inequalities above yields

$$\inf_{f \in \mathcal{F}_{\hat{m}}} \mathcal{E}_P(\mathcal{F}_{\hat{m}}, f) > \pi_N(\hat{m}, s). \tag{35}$$

Since for any m $\mathcal{E}_P(\mathcal{F}_m, f) \le \mathbb{E}(f(X) - \eta(X))^2$, the definition of $\bar{m}$ and (35) imply that

$$\pi_N(\bar{m}, s) \ge \inf_{f \in \mathcal{F}_{\hat{m}}} \mathbb{E}(f(X) - \eta(X))^2 > \pi_N(\hat{m}, s),$$

contradicting our assumption, hence proving the claim.

References

J.-Y. Audibert and A. B. Tsybakov. Fast learning rates for plug-in classifiers. *Preprint*, 2005. Available at: `http://imagine.enpc.fr/publications/papers/05preprint_AudTsy.pdf`.

M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *Proceedings of the Conference on Learning Theory*, pages 45–56, 2008.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *J. Comput. System Sci.*, 75(1):78–89, 2009.

R. M. Castro and R. D. Nowak. Minimax bounds for active learning. *IEEE Trans. Inform. Theory*, 54(5):2339–2353, 2008.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20*, pages 353–360. MIT Press, 2008.

S. Gaïffas. Sharp estimation in sup norm with random design. *Statist. Probab. Lett.*, 77(8):782–794, 2007.

E. Giné and R. Nickl. Confidence bands in density estimation. *Ann. Statist.*, 38(2):1122–1170, 2010.

S. Hanneke. Rates of convergence in active learning. *Ann. Statist.*, 39(1):333–361, 2011.

M. Hoffmann and R. Nickl. On adaptive inference and confidence bands. *The Annals of Statistics*, to appear.

V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *J. Mach. Learn. Res.*, 11:2457–2485, 2010.

V. Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems*. Springer, 2011. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008, École d'Été de Probabilités de Saint-Flour.

M. G. Low. On nonparametric confidence intervals. *Ann. Statist.*, 25(6):2547–2554, 1997.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Journal of Machine Learning Research 13 (2012) 91-136 Submitted 12/10; Revised 8/11; Published 1/12

Refinement of Operator-valued Reproducing Kernels

Haizhang Zhang[*] ZHHAIZH2@SYSU.EDU.CN
School of Mathematics and Computational Science
Sun Yat-sen University
Guangzhou 510275, P. R. China

Yuesheng Xu[†] YXU06@SYR.EDU
Department of Mathematics
Syracuse University
Syracuse, NY 13244, USA

Qinghui Zhang ZHQINGH@MAIL2.SYSU.EDU.CN
School of Mathematics and Computational Science
Sun Yat-sen University
Guangzhou 510275, P. R. China

Editor: John Shawe-Taylor

Abstract

This paper studies the construction of a *refinement* kernel for a given operator-valued reproducing kernel such that the vector-valued reproducing kernel Hilbert space of the refinement kernel contains that of the given kernel as a subspace. The study is motivated from the need of updating the current operator-valued reproducing kernel in multi-task learning when underfitting or overfitting occurs. Numerical simulations confirm that the established refinement kernel method is able to meet this need. Various characterizations are provided based on feature maps and vector-valued integral representations of operator-valued reproducing kernels. Concrete examples of refining translation invariant and finite Hilbert-Schmidt operator-valued reproducing kernels are provided. Other examples include refinement of Hessian of scalar-valued translation-invariant kernels and transformation kernels. Existence and properties of operator-valued reproducing kernels preserved during the refinement process are also investigated.

Keywords: vector-valued reproducing kernel Hilbert spaces, operator-valued reproducing kernels, refinement, embedding, translation invariant kernels, Hessian of Gaussian kernels, Hilbert-Schmidt kernels, numerical experiments

1. Introduction

Machine learning designs algorithms for the purpose of inferring from finite empirical data a function dependency which can then be used to understand or predict generation of new data. Past research has mainly focused on single task learning problems where the function to be learned is scalar-valued. Built upon the theory of scalar-valued reproducing kernels (Aronszajn, 1950), kernel methods have proven useful in single task learning (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004; Vapnik, 1998). The approach might be justified in three ways. Firstly, as inputs for

[*] Also in the Guangdong Province Key Laboratory of Computational Science.

[†] Also at Guangdong Province Key Laboratory of Computational Science, Sun Yat-sen University, Guangzhou 510275, P. R. China.

learning algorithms are sample data, requiring the sampling process to be stable seems inevitable. Thanks to the existence of an inner product, Hilbert spaces are the class of normed vector spaces that we can handle best. These two considerations lead immediately to the notion of reproducing kernel Hilbert spaces (RKHS). Secondly, a reasonable learning scheme is expected to make use of the similarity between a new input and the existing inputs for prediction. Inner products provide a natural measurement of similarities. It is well-known that a bivariate function is a scalar-valued reproducing kernel if and only if it is representable as some inner product of the feature of inputs (Schölkopf and Smola, 2002). Finally, finding a feature map and taking the inner product of the feature of two inputs are equivalent to choosing a scalar-valued reproducing kernel and performing function evaluations of it. This brings computational efficiency and gives birth to the important "kernel trick" (Schölkopf and Smola, 2002) in machine learning. For references on single task learning and scalar-valued RKHS, we recommend Aronszajn (1950), Cucker and Smale (2002), Cucker and Zhou (2007), Evgeniou et al. (2000), Schölkopf and Smola (2002), Shawe-Taylor and Cristianini (2004) and Vapnik (1998); Zhang et al. (2009).

In this paper, we are concerned with multi-task learning where the function to be reconstructed from finite sample data takes range in a finite-dimensional Euclidean space, or more generally, a Hilbert space. Motivated by the success of kernel methods in single task learning, it was proposed in Evgeniou et al. (2005) and Micchelli and Pontil (2005) to develop algorithms for multi-task learning in the framework of vector-valued RKHS. We attempt to contribute to the theory of vector-valued RKHS by studying a special embedding relationship between two vector-valued RKHS. We shall briefly review existing work on vector-valued RKHS and the associated operator-valued reproducing kernels. The study of vector-valued RKHS dates back to Pedrick (1957). The notion of matrix-valued or operator-valued reproducing kernels was also obtained in Burbea and Masani (1984). References Mukherjee and Wu (2006), Mukherjee and Zhou (2006) and Ying and Campbell (2008) were devoted to learning a multi-variate function and its gradient simultaneously. Reference Carmeli et al. (2006) established the Mercer theorem for vector-valued RKHS and characterized those spaces with elements being p-integrable vector-valued functions. Various characterizations and examples of universal operator-valued reproducing kernels were provided in Caponnetto et al. (2008) and Carmeli et al. (2010). The latter (Carmeli et al., 2010) also examined basic operations of operator-valued reproducing kernels and extended the Bochner characterization of translation invariant reproducing kernels to the operator-valued case.

The purpose of this paper is to study the refinement relationship of two vector-valued reproducing kernels. We say that a vector-valued reproducing kernel is a refinement of another kernel of such type if the RKHS of the first kernel contains that of the latter one as a linear subspace and their norms coincide on the smaller space. The precise definition will be given in the next section after we provide necessary preliminaries on vector-valued RKHS. The study is motivated by the need of updating a vector-valued reproducing kernel for multi-task machine learning when underfitting or overfitting occurs. Detailed explanations of this motivation will be presented in the next section. Mathematically, a thorough understanding of the refinement relationship is essential to the establishment of a multi-scale decomposition of vector-valued RKHS, which in turn is the foundation for extending multi-scale analysis (Daubechies, 1992; Mallat, 1989) to kernel methods. In fact, a special refinement method by a bijective mapping from the input space to itself provides such a decomposition. As the procedure is similar to the scalar-valued case, we refer interested authors to Xu and Zhang (2007) for the details. The notion of refinement of scalar-valued kernels was initiated and extensively investigated by the first two authors (Xu and Zhang, 2007, 2009). Therefore, a general principle we shall follow is to briefly mention or even completely omit arguments that are not essentially different from the scalar-valued case. As we proceed with the study, it will become clear that nontrivial obstacles in extending the scalar-valued theory to vector-valued RKHS are mainly caused by the complexity in the vector-valued integral representation of the operator-valued reproducing kernels under investigation, by the complicated form of the feature map involved, which is also operator-valued, and by the infinite-dimensionality of the output space in some occasions.

To be more specific, we would personally regard the following results to be mathematically nontrivial: Theorem 11 of characterizing the refinement of kernels defined by the integral of scalar-valued kernels with respect to an operator-valued measure, Proposition 10 of studying the refinement of positive operators, Lemma 13 of proving the disjointness of the RKHS of translation-invariant kernels of different types, and Theorem 21 about the refinement of finite Hilbert-Schmidt kernels. Besides, compared to the scalar-valued case in Xu and Zhang (2009), Sections 5.2 and 5.3 about the refinement of Hessian kernels and transformation kernels are unique, and Section 7 of numerical experiments is novel. By contrast, the discussion of general characterizations and finite-dimensional RKHS in Section 3, refinement of kernels defined by the integral of operator-value kernels with respect to a scalar-valued measure in Section 4.1, and Section 6 about the existence of refinement and properties preserved by the refinement process can be viewed as either trivial extensions or not of sufficient mathematical depth. We also remark that every vector-valued RKHS is isometrically isomorphic to a scalar-valued RKHS on an extended input space (see Proposition 6 below). However, this does not mean that the question of studying refinement of operator-valued kernels can be trivially reduced to that about scalar-valued kernels. The isometry procedure will usually make the resulting scalar-valued kernel and extended input space complex and difficult to analyze. Moreover, favorable properties such as translation invariance and Hilbert-Schmidt structure of the original kernels are generally lost in the process. Therefore, an independent study of the refinement of operator-valued kernels is necessary and challenging.

This paper is organized as follows. We shall introduce necessary preliminaries on vector-valued RKHS and motivate our study from multi-tasking learning in the next section. In Section 3, we shall present three general characterizations of the refinement relationship by examining the difference of two given kernels, the feature map representation of kernels, and the associated kernels on the extended input space. Recall that most scalar-valued reproducing kernels are represented by integrals. In the operator-valued case, we have two types of integral representations: the integral of operator-valued reproducing kernels with respect to a scalar-valued measure, and the integral of scalar-valued reproducing kernels with respect to an operator-valued measure. As a key part of this paper, we shall investigate in Section 4 specifications of the general characterizations when the operator-valued reproducing kernels are given by such integrals. In Section 5, we present concrete examples of refinement by looking into translation-invariant operator-valued kernels, Hessian of a scalar-valued kernels, Hilbert-Schmidt kernels, etc. Section 6 treats specially the existence of nontrivial refinements and desirable properties of operator-valued reproducing kernels that can be preserved during the refinement process. In Section 7, we perform three numerical simulations to show the effect of the refinement kernel method in updating operator-valued reproducing kernels for multi-task learning. Finally, we conclude the paper in Section 8.

2. Kernel Refinement

To explain our motivation from multi-task learning in details, we first recall the definition of operator-valued reproducing kernels. Throughout the paper, we let X and Λ denote a prescribed set and a separable Hilbert space, respectively. We shall call X the input space and Λ the output space. To avoid confusion, elements in X and Λ will be denoted by x,y, and ξ,η, respectively. Unless specifically mentioned, all the normed vector spaces in the paper are over the field $\mathbb{C}$ of complex numbers. Let $\mathcal{L}(\Lambda)$ be the set of all the bounded linear operators from Λ to Λ, and $\mathcal{L}_+(\Lambda)$ its subset of those linear operators A that are self-adjoint and positive, namely,

$$(A\xi,\xi)_\Lambda \geq 0 \text{ for all } \xi \in \Lambda,$$

where $(\cdot,\cdot)_\Lambda$ is the inner product on Λ. The adjoint of $A \in \mathcal{L}(\Lambda)$ is denoted by A^*. An $\mathcal{L}(\Lambda)$*-valued reproducing kernel* on X is a function $K : X \times X \to \mathcal{L}(\Lambda)$ such that $K(x,y) = K(y,x)^*$ for all $x,y \in X$, and such that for all $x_j \in X$, $\xi_j \in \Lambda$, $j \in \mathbb{N}_n := \{1,2,\ldots,n\}$, $n \in \mathbb{N}$,

$$\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda \geq 0. \tag{1}$$

For each $\mathcal{L}(\Lambda)$-valued reproducing kernel K on X, there exists a unique Hilbert space, denoted by $\mathcal{H}_K$, consisting of Λ-valued functions on X such that

$$K(x,\cdot)\xi \in \mathcal{H}_K \text{ for all } x \in X \text{ and } \xi \in \Lambda \tag{2}$$

and

$$(f(x),\xi)_\Lambda = (f,K(x,\cdot)\xi)_{\mathcal{H}_K} \text{ for all } f \in \mathcal{H}_K,\ x \in X, \text{ and } \xi \in \Lambda. \tag{3}$$

It is implied by the above two properties that the point evaluation at each $x \in X$:

$$\delta_x(f) := f(x), \quad f \in \mathcal{H}_K$$

is continuous from $\mathcal{H}_K$ to Λ. In other words, $\mathcal{H}_K$ is a Λ-valued RKHS. We call it the RKHS of K. Conversely, for each Λ-valued RKHS on X, there exists a unique $\mathcal{L}(\Lambda)$-valued reproducing kernel K on X that satisfies (2) and (3). For this reason, we also call K the reproducing kernel (or kernel for short) of $\mathcal{H}_K$. The bijective correspondence between $\mathcal{L}(\Lambda)$-valued reproducing kernels and Λ-valued RKHS is central to the theory of vector-valued RKHS.

Given two $\mathcal{L}(\Lambda)$-valued reproducing kernels K,G on X, we shall investigate in this paper the fundamental embedding relationship $\mathcal{H}_K \preceq \mathcal{H}_G$ in the sense that $\mathcal{H}_K \subseteq \mathcal{H}_G$ and for all $f \in \mathcal{H}_K$, $\|f\|_{\mathcal{H}_K} = \|f\|_{\mathcal{H}_G}$. Here, $\|\cdot\|_{\mathcal{W}}$ denotes the norm of a normed vector space $\mathcal{W}$. We call G a *refinement* of K if there does hold $\mathcal{H}_K \preceq \mathcal{H}_G$. Such a refinement is said to be nontrivial if $G \neq K$.

We motivate this study from the kernel methods for multi-task learning and from the multi-scale decomposition of vector-valued RKHS. Let $\mathbf{z} := \{(x_j,\xi_j) : j \in \mathbb{N}_n\} \subseteq X \times \Lambda$ be given sample data. A typical kernel method infers from $\mathbf{z}$ the minimizer $f_{\mathbf{z}}$ of

$$\min_{f \in \mathcal{H}_K} \frac{1}{n}\sum_{j=1}^{n} C(x_j,\xi_j,f(x_j)) + \sigma\phi(\|f\|_{\mathcal{H}_K}), \tag{4}$$

where K is a selected $\mathcal{L}(\Lambda)$-valued reproducing kernel on X, C a prescribed loss function, σ a positive regularization parameter, and ϕ a regularizer. The ideal predictor $f_0 : X \to \Lambda$ that we are pursuing is the one that minimizes

$$\mathcal{E}(f) := \int_{X \times \Lambda} C(x, \xi, f(\xi)) dP$$

among all possible functions f from X to Λ. Here P is an unknown probability measure on $X \times \Lambda$ that dominates the generation of data from $X \times \Lambda$. We wish that $\mathcal{E}(f_z) - \mathcal{E}(f_0)$ can converge to zero in probability as the number n of sampling points tends to infinity. Whether this will happen depends heavily on the choice of the kernel K. The error $\mathcal{E}(f_z) - \mathcal{E}(f_0)$ can be decomposed into the sum of the *approximation error* and *sampling error* (Schölkopf and Smola, 2002; Vapnik, 1998). The approximation error occurs as we search the minimizer in a restricted set of candidate functions, namely, $\mathcal{H}_K$. It becomes smaller as $\mathcal{H}_K$ enlarges. The sampling error is caused by replacing the expectation $\mathcal{E}(f)$ of the loss function $C(x, \xi, f(\xi))$ with the sample mean

$$\frac{1}{n} \sum_{j=1}^{n} C(x_j, \xi_j, f(x_j)).$$

By the law of large numbers, the sample mean converges to the expectation in probability as $n \to \infty$ for a fixed $f \in \mathcal{H}_K$. However, as $f_{\mathbf{z}}$ varies according to changes in the sample data $\mathbf{z}$, we need a uniform version of the law of large number on $\mathcal{H}_K$ in order to well control the sampling error. Therefore, the sampling error usually increases as $\mathcal{H}_K$ enlarges, or to be more precisely, as the *capacity* of $\mathcal{H}_K$ increases.

By the above analysis, we might encounter two situations after the choice of an $\mathcal{L}(\Lambda)$-valued reproducing kernel K:

— overfitting, which occurs when the capacity of $\mathcal{H}_K$ is too large, forcing the minimizer obtained from (4) to imitate artificial function dependency in the sample data, and thus causing the sampling error to be out of control;

— underfitting, which occurs when $\mathcal{H}_K$ is too small for the minimizer of (4) to describe the desired function dependency implied in the data, and thus failing in bounding the approximation error.

When one of the above situations happens, a remedy is to modify the reproducing kernel. Specifically, one might want to find another $\mathcal{L}(\Lambda)$-valued reproducing kernel G such that $\mathcal{H}_K \preceq \mathcal{H}_G$ when there is underfitting, or such that $\mathcal{H}_G \preceq \mathcal{H}_K$ when there is overfitting. We see that in either case, we need to make use of the refinement relationship. We shall verify in the last section through extensive numerical simulations that the refinement kernel method is indeed able to provide an appropriate update of an operator-valued reproducing kernel when underfitting or overfitting occurs.

Before moving on to the characterization of refinement of operator-valued reproducing kernels, we collect here notations that will be frequently used in the rest of the paper. They will also be (or have been) defined when first used.

- X: a general input space,

- Λ: a Hilbert space, serving as the output space,

- $\|\cdot\|_\Lambda$: the norm on a Hilbert or Banach space Λ,
- $\mathcal{W}$: a Hilbert space, usually serving as the feature space of reproducing kernels,
- $\mathcal{L}(\Lambda)$: the space of bounded linear operators from Λ to Λ,
- $\mathcal{L}_+(\Lambda)$: the set of self-adjoint and positive bounded linear operators from Λ to Λ,
- $\mathcal{L}(\Lambda,\mathcal{W})$: the space of bounded linear operators from Λ to $\mathcal{W}$,
- K,G: $\mathcal{L}(\Lambda)$-valued reproducing kernels,
- $\mathcal{H}_K,\mathcal{H}_G$: the RKHS of kernels K,G, respectively,
- $\mathcal{H}_K\preceq\mathcal{H}_G$: G is a refinement of K, namely, $\mathcal{H}_K\subseteq\mathcal{H}_G$ and $\|f\|_{\mathcal{H}_K}=\|f\|_{\mathcal{H}_G}$ for all $f\in\mathcal{H}_K$,
- $\tilde{X}$: the extended input space $X\times\Lambda$,
- $\tilde{K}$: the scalar-valued kernel (11) associated with an $\mathcal{L}(\Lambda)$-valued kernel K,
- μ,ν: scalar-valued or operator-valued measures,
- $|\mu|$: the variation (19) of a measure μ,
- $(\Omega,\mathcal{F},\mu)$: a measure space,
- $\mu\preceq\nu$: means that μ is the restriction of ν on some measurable set,
- $L^2(\Omega,\mathcal{B},d\mu)$: the Hilbert space (16) of square integrable $\mathcal{B}$-valued functions on Ω with respect to the measure μ,
- $L^2(\Omega,d\mu)$: the Hilbert space of scalar-valued square integrable functions on Ω with respect to the measure μ,
- $L^\infty(\Omega,d\mu)$: the Banach space of essentially bounded measurable functions on Ω with respect to the measure μ,
- $A\preceq B$: see (29) for this refinement relation of two positive operators,
- $\mathcal{B}(\mathbb{R}^d,\Lambda)$: the set of all the $\mathcal{L}_+(\Lambda)$-valued measures of bounded variation on the σ-algebra of Borel subsets in $\mathbb{R}^d$,
- γ_c,γ_s: the continuous part γ_c and singular part γ_s in the Lebesgue decomposition (38) of a Borel measure γ,
- L_c,L_s: the continuous and singular parts (39) of a translation-invariant kernel,
- $\Lambda\otimes\mathcal{W}$: the tensor product of two Hilbert spaces Λ and $\mathcal{W}$,
- $\sqrt{A}$: the square root of a positive bounded linear operator A.

3. General Characterizations

The relationship between the RKHS of the sum of two operator-valued reproducing kernels and those of the summand kernels has been made clear in Theorem 1 on page 44 of Pedrick (1957). Our first characterization of refinement is a direct consequence of this result.

Proposition 1 *Let K,G be two $\mathcal{L}(\Lambda)$-valued reproducing kernels on X. Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $G-K$ is an $\mathcal{L}(\Lambda)$-valued reproducing kernel on X and $\mathcal{H}_K \cap \mathcal{H}_{G-K} = \{0\}$. If $\mathcal{H}_K \preceq \mathcal{H}_G$ then $\mathcal{H}_{G-K}$ is the orthogonal complement of $\mathcal{H}_K$ in $\mathcal{H}_G$.*

Every reproducing kernel has a feature map representation. Specifically, K is an $\mathcal{L}(\Lambda)$-valued reproducing kernel on X if and only if there exists a Hilbert space $\mathcal{W}$ and a mapping $\Phi : X \to \mathcal{L}(\Lambda, \mathcal{W})$ such that

$$K(x,y) = \Phi(y)^*\Phi(x), \quad x,y \in X, \tag{5}$$

where $\mathcal{L}(\Lambda, \mathcal{W})$ denotes the set of bounded linear operators from Λ to $\mathcal{W}$, and $\Phi(y)^*$ is the adjoint operator of $\Phi(y)$. We call Φ a *feature map* of K. The following lemma is useful in identifying the RKHS of a reproducing kernel given by a feature map representation (5).

Lemma 2 *If K is an $\mathcal{L}(\Lambda)$-valued reproducing kernel on X given by (5) then*

$$\mathcal{H}_K = \{\Phi(\cdot)^* u : u \in \mathcal{W}\}$$

with inner product

$$(\Phi(\cdot)^*u, \Phi(\cdot)^*v)_{\mathcal{H}_K} := (P_\Phi u, P_\Phi v)_{\mathcal{W}}, \quad u,v \in \mathcal{W},$$

where P_Φ is the orthogonal projection of $\mathcal{W}$ onto

$$\mathcal{W}_\Phi := \overline{span}\{\Phi(x)\xi : x \in X, \xi \in \Lambda\}.$$

The second characterization can be proved using Lemma 2 and the same arguments with those for the scalar-valued case (Xu and Zhang, 2007).

Theorem 3 *Suppose that $\mathcal{L}(\Lambda)$-valued reproducing kernels K and G are given by the feature maps $\Phi : X \to \mathcal{L}(\Lambda, \mathcal{W})$ and $\Phi' : X \to \mathcal{L}(\Lambda, \mathcal{W}')$, respectively. Assume that $\mathcal{W}_\Phi = \mathcal{W}$ and $\mathcal{W}'_{\Phi'} = \mathcal{W}'$. Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if there exists a bounded linear operator T from $\mathcal{W}'$ to $\mathcal{W}$ such that*

$$T\Phi'(x) = \Phi(x) \text{ for all } x \in X,$$

and the adjoint operator $T^* : \mathcal{W} \to \mathcal{W}'$ is isometric. In this case, G is a nontrivial refinement of K if and only if T is not injective.

To illustrate the above useful results, we shall present a concrete example aiming at refining $\mathcal{L}(\Lambda)$-valued reproducing kernels K with a finite-dimensional RKHS. A simple observation is made regarding such a kernel.

Proposition 4 *A Λ-valued RKHS $\mathcal{H}_K$ is of finite dimension $n \in \mathbb{N}$ if and only if there exists an $n \times n$ hermitian and strictly positive-definite matrix A and n linearly independent functions $\phi_j : X \to \Lambda$, $j \in \mathbb{N}_n$ such that*

$$K(x,y)\xi = \sum_{j=1}^{n}\sum_{k=1}^{n} A_{jk}(\xi, \phi_j(x))_\Lambda \phi_k(y), \quad x,y \in X, \ \xi \in \Lambda. \tag{6}$$

Proof Assume that $\mathcal{H}_K$ is n dimensional with orthogonal basis $\{\phi_j : j \in \mathbb{N}_n\}$. As $K(x,\cdot)\xi \in \mathcal{H}_K$ for all $x \in X, \xi \in \Lambda$, there exist functions $c_j : X \times \Lambda \to \mathbb{C}$ such that

$$K(x,y)\xi = \sum_{j=1}^{n} c_j(\xi,x)\phi_j(y), \;\; x,y \in X, \; \xi \in \Lambda.$$

Since $\{\phi_j : j \in \mathbb{N}_n\}$ is a basis for $\mathcal{H}_K$, each function $f \in \mathcal{H}_K$ has the form

$$f = \sum_{j=1}^{n} d_j\phi_j, \; d_j \in \mathbb{C}, \; j \in \mathbb{N}_n.$$

Clearly, $\|f\| := \left(\sum_{j=1}^{n} |d_j|^2\right)^{1/2}$ is a norm on $\mathcal{H}_K$. It is equivalent to the original one on $\mathcal{H}_K$ as $\dim \mathcal{H}_K < \infty$. It is implied that there exists some $C > 0$ such that

$$\sum_{j=1}^{n} |c_j(\xi,x)|^2 \leq C\|K(x,\cdot)\xi\|^2_{\mathcal{H}_K} = C(K(x,x)\xi,\xi)_\Lambda \leq C\|\xi\|^2_\Lambda\|K(x,x)\|. \tag{7}$$

Obviously, for each $x \in X$ and $j \in \mathbb{N}_n$, $c_j(\cdot,x)$ is a linear functional on Λ. This together with (7) implies that $c_j(\cdot,x)$ are bounded linear functionals on Λ. By the Riesz representation theorem, there exists $\psi_j : X \to \Lambda$, $j \in \mathbb{N}_n$ such that

$$c_j(\xi,x) = (\xi,\psi_j(x))_\Lambda.$$

We conclude that K has the form

$$K(x,y)\xi = \sum_{j=1}^{n} (\xi,\psi_j(x))_\Lambda\phi_j(y), \;\; x,y \in X, \; \xi \in \Lambda. \tag{8}$$

Since $\{\phi_j : j \in \mathbb{N}_n\}$ is an orthogonal basis for $\mathcal{H}_K$, by (3),

$$(\xi,\psi_j(x))_\Lambda = (K(x,\cdot)\xi,\phi_j)_{\mathcal{H}_K} = (\xi,\phi_j(x))_\Lambda, \; \xi \in \Lambda, x \in X.$$

It follows that $\psi_j = \phi_j, \; j \in \mathbb{N}_n$. Substituting this into (8) yields that

$$K(x,y)\xi = \sum_{j=1}^{n} (\xi,\phi_j(x))_\Lambda\phi_j(y), \;\; x,y \in X, \; \xi \in \Lambda,$$

which indeed is a special form of (6).

Conversely, assume that K has the form (6). We set $\mathcal{W}_A := l^2_A(\mathbb{N}_n) := \{c = (c_j : j \in \mathbb{N}_n) \in \mathbb{C}^n\}$ with inner product

$$(c,d)_{l^2_A(\mathbb{N}_n)} := \sum_{j=1}^{n}\sum_{k=1}^{n} c_j\bar{d}_k A_{jk}.$$

Introduce $\Phi : X \to \mathcal{L}(\Lambda, \mathcal{W}_A)$ by setting $\Phi(x)\xi := ((\xi,\phi_j(x))_\Lambda : j \in \mathbb{N}_n)$. Direct computations show that

$$\Phi^*(x)u = \sum_{j=1}^{n}\sum_{k=1}^{n} \phi_j(x)u_k A_{jk}, \; u = (u_j : j \in \mathbb{N}_n) \in \mathcal{W}_A.$$

Thus, we see that $K(x,y) = \Phi(y)^*\Phi(x)$, $x, y \in X$, implying that K is an $\mathcal{L}(\Lambda)$-valued reproducing kernel. By the linear independence of ϕ_j, $j \in \mathbb{N}_n$, $\mathrm{span}\{\Phi(x)\xi : x \in X, \xi \in \Lambda\} = \mathcal{W}_A$. We hence apply Lemma 2 to get that

$$\mathcal{H}_K = \{\Phi(\cdot)^*u : u \in \mathcal{W}_A\} = \mathrm{span}\{\phi_j : j \in \mathbb{N}_n\},$$

which is of dimension n. ■

By the above proposition, we let ϕ_j, $j \in \mathbb{N}_m$ be linearly independent functions from X to Λ, where $m \geq n$ are fixed positive integers. Let A and B be $n \times n$ and $m \times m$ hermitian and strictly positive-definite matrices, respectively. We define K by (6) in terms of matrix A and G by

$$G(x,y)\xi := \sum_{j=1}^{m}\sum_{k=1}^{m} B_{jk}(\xi, \phi_j(x))_\Lambda \phi_k(y),\ x, y \in X \tag{9}$$

and shall investigate conditions for G to be a refinement of K.

Proposition 5 *Let K, G be defined by (6) and (9), respectively. Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if B^{-1} is an augmentation of A^{-1}, namely, $B^{-1}_{jk} = A^{-1}_{jk}$, $j, k \in \mathbb{N}_n$. In particular, if K, G have the form*

$$K(x,y)\xi = \sum_{j\in\mathbb{N}_n} a_j(\xi, \phi_j(x))_\Lambda \phi_j(y),\ G(x,y)\xi = \sum_{k\in\mathbb{N}_m} b_k(\xi, \phi_k(x))_\Lambda \phi_k(y)$$

for some positive constants a_j, b_k, then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $a_j = b_j$, $j \in \mathbb{N}_n$. In both cases if $\mathcal{H}_K \preceq \mathcal{H}_G$ then G is a nontrivial refinement of K if and only if $m > n$.

Proof It suffices to prove the first claim. We observe that K, G have the feature spaces $\mathcal{W} = l^2_A(\mathbb{N}_n)$ and $\mathcal{W}' = l^2_B(\mathbb{N}_m)$, respectively, with feature maps

$$\Phi(x)\xi := ((\xi, \phi_j(x))_\Lambda : j \in \mathbb{N}_n),\ \Phi'(x)\xi := ((\xi, \phi_k(x))_\Lambda : k \in \mathbb{N}_m),\ x \in X, \xi \in \Lambda.$$

Suppose that $\mathcal{H}_K \preceq \mathcal{H}_G$, then by Theorem 3, there exists a bounded linear operator $T : \mathcal{W}' \to \mathcal{W}$ with properties as described there. It can be represented by an $n \times m$ matrix D as

$$(T\Phi'(x)\xi)_j = \sum_{k=1}^{m} D_{jk}(\xi, \phi_k(x))_\Lambda = (\xi, \phi_j(x))_\Lambda,\ \ x \in X, \xi \in \Lambda, \tag{10}$$

which implies that $D = [I_n, 0]$, where I_n denotes the $n \times n$ identity matrix. The adjoint operator T^* of T is then represented by

$$T^*u = B^{-1}\begin{bmatrix} A \\ 0 \end{bmatrix} u,\ u \in \mathbb{C}^n.$$

Since T^* is isometric, we get that

$$(T^*u, T^*v)_{\mathcal{W}'} = (u,v)_{\mathcal{W}},$$

which has the form

$$v^*[A,0]B^{-1}BB^{-1}\begin{bmatrix} A \\ 0 \end{bmatrix} u = v^*Au,\ u, v \in \mathbb{C}^n.$$

We derive from the above equation that

$$[A,0]B^{-1}\begin{bmatrix} A \\ 0 \end{bmatrix} = A.$$

Therefore, B^{-1} is an augmentation of A^{-1}. Conversely, if this is true then $T : \mathcal{W}' \to \mathcal{W}$ defined by

$$Tu' := [I_n, 0]u',\ u' \in \mathbb{C}^m$$

satisfies the two properties in Theorem 3. As a result, $\mathcal{H}_K \preceq \mathcal{H}_G$. ■

It is worthwhile to point out that the above characterization is independent of the Hilbert space Λ.

Unlike the previous two characterizations, the third one comes as a surprise, telling us that theoretically we are able to reduce our consideration to the scalar-valued case.

Introduce for each $\mathcal{L}(\Lambda)$-valued reproducing kernel K on X a scalar-valued reproducing kernel $\tilde{K}$ on the *extended input space* $\tilde{X} := X \times \Lambda$ by setting

$$\tilde{K}((x,\xi),(y,\eta)) := (K(x,y)\xi,\eta)_\Lambda,\ \ x,y \in X,\ \xi,\eta \in \Lambda. \tag{11}$$

By (1), $\tilde{K}$ is indeed positive-definite.

Proposition 6 *There holds $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$. Furthermore, G is a nontrivial refinement of K on X if and only if $\tilde{G}$ is a nontrivial refinement of $\tilde{K}$ on $\tilde{X}$.*

Proof We first explore the close relationship between $\mathcal{H}_K$ and $\mathcal{H}_{\tilde{K}}$. By (3),

$$\tilde{K}((x,\xi),(y,\eta)) = (K(x,y)\xi,\eta)_\Lambda = (K(x,\cdot)\xi, K(y,\cdot)\eta)_{\mathcal{H}_K},$$

which provides a natural feature map $\Phi : \tilde{X} \to \mathcal{H}_K$ of $\tilde{K}$

$$\Phi((x,\xi)) := K(x,\cdot)\xi,\ \ x \in X,\ \xi \in \Lambda.$$

The density condition $\mathcal{W}_\Phi = \mathcal{H}_K$ is clearly satisfied by (3). We hence obtain by (2) that every function $\tilde{f}$ in $\mathcal{H}_{\tilde{K}}$ is of the form

$$\tilde{f}(x,\xi) := (f(x),\xi)_\Lambda \text{ for some } f \in \mathcal{H}_K$$

with

$$\|\tilde{f}\|_{\mathcal{H}_{\tilde{K}}} = \|f\|_{\mathcal{H}_K}.$$

Similar observations can be made about $\mathcal{H}_{\tilde{G}}$.

It follows immediately that $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$ if $\mathcal{H}_K \preceq \mathcal{H}_G$. On the other hand, suppose that $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$. Then for each $f \in \mathcal{H}_K$ there exists some $g \in \mathcal{H}_G$ such that

$$(f(x),\xi)_\Lambda = \tilde{f}(x,\xi) = \tilde{g}(x,\xi) = (g(x),\xi)_\Lambda \text{ for all } x \in X,\ \xi \in \Lambda \tag{12}$$

and

$$\|f\|_{\mathcal{H}_K} = \|\tilde{f}\|_{\mathcal{H}_{\tilde{K}}} = \|\tilde{g}\|_{\mathcal{H}_{\tilde{G}}} = \|g\|_{\mathcal{H}_G}.$$

Equation (12) implies that $f = g$. Therefore, $\mathcal{H}_K \preceq \mathcal{H}_G$. ■

It appears by Proposition 6 that we do not have to bother studying refinement of operator-valued reproducing kernels. Although the strategy sometimes does simplify the problem, the difficulty is generally not reduced significantly. Instead, the result might be viewed as transferring the complexity to the input space. Moreover, desirable properties such as translation invariance of the original kernels might be lost in the process. As a result, an independent study of the operator-valued case remains necessary and challenging.

4. Integral Representations

We shall characterize in this section the refinement of operator-valued kernels defined by two kinds of integral representations: the integral of operator-valued kernels with respect to a scalar-valued measure, and the integral of scalar-valued kernels with respect to an operator-valued measure. The characterizations to be established are crucial to the study of this paper as many useful operator-valued kernels are of an integral representation. Typical examples include the important translation-invariant operator-valued kernels and hessian kernels to be considered in the next section. We also point out in advance the difference in the refinement for the two kinds of integral representations. Firstly, the first refinement corresponds to the same feature map and different measures, while the other when the Radon-Nikodym property is engaged has different feature maps and the same measure. The arguments of the proofs and the obtained results are essential different. The characterization of the first kind of refinement can be viewed as a straightforward generalization of that obtained in Xu and Zhang (2009), while the other one is mathematically nontrivial.

This section will be built on the theory of vector-valued measures and integrals (Berberian, 1966; Diestel and Uhl, 1977). Necessary preliminaries on the subjects will be explained in sufficient details.

4.1 Operator-valued Kernels With Respect to Scalar-valued Measures

Let us first introduce integration of a vector-valued function with respect to a scalar-valued measure. Let $\mathcal{F}$ be a σ-algebra of subsets of a fixed set Ω, μ a finite nonnegative measure on $\mathcal{F}$, and $\mathcal{B}$ a Banach space. We are concerned with $\mathcal{B}$-valued functions on Ω. A function $f : \Omega \to \mathcal{B}$ is said to be *simple* if

$$f = \sum_{j=1}^{n} a_j \chi_{E_j} \tag{13}$$

for some finitely many $a_j \in \mathcal{B}$ and pairwise disjoint subsets $E_j \in \mathcal{F}$, $j \in \mathbb{N}_n$. A function $f : \Omega \to \mathcal{B}$ is called *μ-measurable* if there exists a sequence of $\mathcal{B}$-valued simple functions f_n on Ω such that

$$\lim_{n\to\infty} \|f_n(t) - f(t)\|_{\mathcal{B}} = 0 \text{ for } \mu - \text{a.e. } t \in \Omega,$$

where $\mu -$ a.e. stands for "everywhere except for a set of zero μ measure". Finally, a $\mathcal{B}$-valued function f on Ω is called *μ-Bochner integrable* if there exists a sequence of simple functions $f_n : \Omega \to \mathcal{B}$ such that

$$\lim_{n\to\infty} \int_{\Omega} \|f_n(t) - f(t)\|_{\mathcal{B}} d\mu(t) = 0. \tag{14}$$

The integral of a simple function f of the form (13) on any $E\in\mathcal{F}$ with respect to μ is defined by

$$\int_E fd\mu := \sum_{j=1}^n a_j\mu(E_j\cap E).$$

In general, suppose that f is a μ-Bochner integrable function from Ω to $\mathcal{B}$, that is, (14) holds true. Then it is obvious that for each $E\in\mathcal{F}$, $\int_E f_n d\mu$, $n\in\mathbb{N}$ form a Cauchy sequence in $\mathcal{B}$. Therefore,

$$\int_E fd\mu := \lim_{n\to\infty}\int_E f_n d\mu.$$

The resulting integral $\int_E fd\mu$ is an element in $\mathcal{B}$.

It is known that a μ-measurable function $f:\Omega\to\mathcal{B}$ is Bochner integrable if and only if

$$\int_\Omega \|f(t)\|_{\mathcal{B}}d\mu(t) < +\infty.$$

This provides a way for us to comprehend the integral $\int_E fd\mu$ in the most needed case when f is $\mathcal{L}(\Lambda)$-valued. If $\mathcal{B}=\mathcal{L}(\Lambda)$ then we have for each $E\in\mathcal{F}$ that

$$\left(\int_E fd\mu\xi,\eta\right)_\Lambda = \int_E (f(t)\xi,\eta)_\Lambda d\mu(t),\ \ \xi,\eta\in\Lambda. \tag{15}$$

Clearly, the right hand side above defines a sesquilinear form on $\Lambda\times\Lambda$ which is bounded as

$$\left|\int_E (f(t)\xi,\eta)_\Lambda d\mu(t)\right| \le \int_E \|f(t)\|_{\mathcal{L}(\Lambda)}d\mu(t)\ \|\xi\|_\Lambda\|\eta\|_\Lambda,$$

where $\|\cdot\|_{\mathcal{L}(\Lambda)}$ is the operator norm on $\mathcal{L}(\Lambda)$. As a result, (15) gives an equivalent way of defining the integral $\int_E fd\mu$ as a bounded linear operator on Λ (Conway, 1990).

We introduce another notation before returning to reproducing kernels. Denote by $L^2(\Omega,\mathcal{B},d\mu)$ the Banach space of all the μ-measurable functions $f:\Omega\to\mathcal{B}$ such that

$$\|f\|_{L^2(\Omega,\mathcal{B},d\mu)} := \left(\int_\Omega \|f(t)\|_{\mathcal{B}}^2 d\mu(t)\right)^{1/2} < +\infty. \tag{16}$$

When $\mathcal{B}=\mathbb{C}$, $L^2(\Omega,\mathbb{C},d\mu)$ will be abbreviated as $L^2(\Omega,d\mu)$. When $\mathcal{B}$ is a Hilbert space, $L^2(\Omega,\mathcal{B},d\mu)$ is also a Hilbert space with the inner product

$$(f,g)_{L^2(\Omega,\mathcal{B},d\mu)} := \int_\Omega (f(t),g(t))_{\mathcal{B}}d\mu(t),\ \ f,g\in L^2(\Omega,\mathcal{B},d\mu).$$

The discussion in this section by far can be found in Diestel and Uhl (1977).

Let μ,ν be two finite nonnegative measures on a σ-algebra $\mathcal{F}$ of subsets of Ω. To introduce our $\mathcal{L}(\Lambda)$-valued reproducing kernels, we also let $\mathcal{W}$ be a Hilbert space and ϕ a mapping from $X\times\Omega$ to $\mathcal{L}(\Lambda,\mathcal{W})$ such that for each $x\in X$, $\phi(x,\cdot)$ belongs to both $L^2(\Omega,\mathcal{L}(\Lambda,\mathcal{W}),d\mu)$ and $L^2(\Omega,\mathcal{L}(\Lambda,\mathcal{W}),d\nu)$. We shall investigate conditions that ensure $\mathcal{H}_K\preceq\mathcal{H}_G$ where

$$K(x,y) = \int_\Omega \phi(y,t)^*\phi(x,t)d\mu(t),\ \ x,y\in X \tag{17}$$

and

$$G(x,y)=\int_\Omega \phi(y,t)^*\phi(x,t)d\nu(t),\ \ x,y\in X, \tag{18}$$

where $\phi(y,t)^*$ is the adjoint operator of $\phi(y,t)$. Note that K,G are well-defined as the integrand is Bochner integrable with respect to both μ and ν. For instance, we observe by the Cauchy-Schwartz inequality for all $x,y\in X$ that

$$\begin{aligned}\int_\Omega \|\phi(y,t)^*\phi(x,t)\|_{\mathcal{L}(\Lambda)}d\mu(t) &\le \int_\Omega \|\phi(y,t)^*\|_{\mathcal{L}(\mathcal{W},\Lambda)}\|\phi(x,t)\|_{\mathcal{L}(\Lambda,\mathcal{W})}d\mu(t)\\ &= \int_\Omega \|\phi(y,t)\|_{\mathcal{L}(\Lambda,\mathcal{W})}\|\phi(x,t)\|_{\mathcal{L}(\Lambda,\mathcal{W})}d\mu(t)\\ &\le \|\phi(y,\cdot)\|_{L^2(\Omega,\mathcal{L}(\Lambda,\mathcal{W}),d\mu)}\|\phi(x,\cdot)\|_{L^2(\Omega,\mathcal{L}(\Lambda,\mathcal{W}),d\mu)}.\end{aligned}$$

An alternative of expressing K,G is for all $x,y\in X$, $\xi,\eta\in\Lambda$ that

$$\tilde{K}((x,\xi),(y,\eta))=(K(x,y)\xi,\eta)_\Lambda=\int_\Omega(\phi(x,t)\xi,\phi(y,t)\eta)_{\mathcal{W}}d\mu(t)$$

and

$$\tilde{G}((x,\xi),(y,\eta))=(G(x,y)\xi,\eta)_\Lambda=\int_\Omega(\phi(x,t)\xi,\phi(y,t)\eta)_{\mathcal{W}}d\nu(t).$$

When $\Lambda=\mathcal{W}=\mathbb{C}$, a characterization of $\mathcal{H}_K\preceq\mathcal{H}_G$ in terms of μ,ν has been established in Xu and Zhang (2009). The relation, between the two measures, which we shall need is absolute continuity. We say that μ is *absolutely continuous* with respect to ν if for all $E\in\mathcal{F}$, $\nu(E)=0$ implies $\mu(E)=0$. In this case, by the Radon-Nikodym theorem (see, Rudin, 1987, page 121) for scalar-valued measures, there exists a nonnegative ν-integrable function, denoted by $d\mu/d\nu$, such that

$$\mu(E)=\int_E\frac{d\mu}{d\nu}(t)d\nu(t)\text{ for all }E\in\mathcal{F}.$$

We write $\mu\preceq\nu$ if μ is absolutely continuous with respect to ν and $d\mu/d\nu\in\{0,1\}$ $\nu-$a.e. Thus, $\mu\preceq\nu$ if and only if μ is the restriction of ν on some measurable set in $\mathcal{F}$.

When $\Lambda=\mathcal{W}=\mathbb{C}$, it was proved in Theorem 8 of Xu and Zhang (2009) that if span$\{\phi(x,\cdot):x\in X\}$ is dense in both $L^2(\Omega,d\mu)$ and $L^2(\Omega,d\nu)$ then G is a refinement of K if and only if $\mu\preceq\nu$. If $\mu\preceq\nu$ then G is a nontrivial refinement of K if and only if $\nu(\Omega)>\mu(\Omega)$.

Theorem 7 *Let K,G be given by (17) and (18). If* span$\{\phi(x,\cdot)\xi:x\in X,\ \xi\in\Lambda\}$ *is dense in both $L^2(\Omega,\mathcal{W},d\mu)$ and $L^2(\Omega,\mathcal{W},d\nu)$ then $\mathcal{H}_K\preceq\mathcal{H}_G$ if and only if $\mu\preceq\nu$. In this case, the refinement G of K is nontrivial if and only if $\nu(\Omega)-\mu(\Omega)>0$.*

Proof When $\mathcal{W}=\mathbb{C}$, as a direct consequence of Theorem 8 in Xu and Zhang (2009), $\mathcal{H}_{\tilde{K}}\preceq\mathcal{H}_{\tilde{G}}$ if and only if $\mu\preceq\nu$. The result hence follows from Proposition 6. When $\mathcal{W}$ is a general Hilbert space, it can be proved by arguments similar to those in Xu and Zhang (2009). ■

4.2 Scalar-valued Kernels with Respect to Operator-valued Measures

Again, $\mathcal{B}$ is a Banach space and $\mathcal{F}$ denotes a σ-algebra consisting of subsets of a fixed set Ω. A $\mathcal{B}$-valued measure on $\mathcal{F}$ is a function from $\mathcal{F}$ to $\mathcal{B}$ that is countably additive in the sense that for every sequence of pairwise disjoint sets $E_j \in \mathcal{F}$, $j \in \mathbb{N}$

$$\mu\left(\bigcup_{j=1}^{\infty} E_j\right) = \sum_{j=1}^{\infty} \mu(E_j),$$

where the series converges in the norm of $\mathcal{B}$. Every $\mathcal{B}$-valued measure μ on $\mathcal{F}$ comes with a scalar-valued measure $|\mu|$ on $\mathcal{F}$ defined by

$$|\mu|(E) := \sup_{\mathcal{P}} \sum_{F \in \mathcal{P}} \|\mu(F)\|_{\mathcal{B}}, \ \ E \in \mathcal{F}, \tag{19}$$

where the supremum is taken over all partitions $\mathcal{P}$ of E into countably many pairwise disjoint members of $\mathcal{F}$. We call $|\mu|$ the *variation* of μ and shall only work with these vector-valued measures μ that are of *bounded variation*, that is, $|\mu|(\Omega) < +\infty$. We note that μ vanishes on sets of zero $|\mu|$ measure. It implies that μ is absolutely continuous with respect to $|\mu|$ in the sense that

$$\lim_{|\mu(E)| \to 0} \mu(E) = 0.$$

The only type of integration that we shall need is to integrate a bounded $\mathcal{F}$-measurable scalar-valued function with respect to a $\mathcal{B}$-valued measure of bounded variation. Denote by $L^\infty(\Omega, d|\mu|)$ the Banach space of essentially bounded $\mathcal{F}$-measurable functions on Ω with the norm

$$\|f\|_{L^\infty(\Omega,d|\mu|)} := \inf\{a \geq 0 : |\mu|(\{|f| > a\}) = 0\}.$$

For a simple function $f : \Omega \to \mathbb{C}$ of the form

$$f = \sum_{j=1}^{n} \alpha_j \chi_{E_j},$$

where $\alpha_j \in \mathbb{C}$ and E_j are pairwise disjoint members in $\mathcal{F}$, we define

$$\int_E f d\mu := \sum_{j=1}^{n} \alpha_j \mu(E_j \cap E), \ \ E \in \mathcal{F}.$$

Clearly,

$$\left\| \int_E f d\mu \right\|_{\mathcal{B}} \leq \|f\|_{L^\infty(\Omega,d|\mu|)} |\mu|(E).$$

Therefore, the map sending a simple function f to $\int_E f d\mu$ can be uniquely extended to a bounded linear operator from $L^\infty(\Omega, d|\mu|)$ to $\mathcal{B}$. The outcome of the application of the resulting operator on a general $f \in L^\infty(\Omega, d|\mu|)$ is still denoted by $\int_E f d\mu$. This is how the $\mathcal{B}$-valued integral is defined.

It is time to present the second type of reproducing kernels defined by integration:

$$K(x,y) := \int_\Omega \Psi(x,y,t) d\mu(t), \ \ x,y \in X, \tag{20}$$

where μ is an $\mathcal{L}_+(\Lambda)$-valued measure on $\mathcal{F}$ of bounded variation, and Ψ is a scalar-valued function such that $\Psi(\cdot,\cdot,t)$ is a scalar-valued reproducing kernel on X for all $t \in \Omega$ and for all $x, y \in X$, $\Psi(x,y,\cdot)$ is bounded and $\mathcal{F}$-measurable. We verify that (20) indeed defines an $\mathcal{L}(\Lambda)$-valued reproducing kernel.

Proposition 8 *With the above assumptions on Ψ and μ, the function K defined by (20) is an $\mathcal{L}(\Lambda)$-valued reproducing kernel on X.*

Proof Fix finite $x_j \in X$ and $\xi_j \in \Lambda$, $j \in \mathbb{N}_n$. For any $\varepsilon > 0$, there exist simple functions

$$f_{j,k} := \sum_{l=1}^{m} \alpha_{j,k,l}\chi_{E_l}, \quad j,k \in \mathbb{N}_n$$

such that

$$\|\Psi(x_j,x_k,\cdot) - f_{j,k}\|_{L^\infty(\Omega,d|\mu|)} < \varepsilon, \quad j,k \in \mathbb{N}_n. \tag{21}$$

Here, $\alpha_{j,k,l} \in \mathbb{C}$ and E_l are pairwise disjoint sets in $\mathcal{F}$ with $|\mu|(E_l) > 0$, $l \in \mathbb{N}_m$. By (21) and the definition of integration in this section,

$$\left|\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda - \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\left(\int_\Omega f_{j,k}d\mu\right)\xi_j,\xi_k\right)_\Lambda\right| \le \varepsilon|\mu|(\Omega)\left(\sum_{j=1}^{n}\|\xi_j\|_\Lambda\right)^2. \tag{22}$$

We may choose by (21) for each $l \in \mathbb{N}_m$ some $t_l \in E_l$ such that

$$\left|\Psi(x_j,x_k,t_l) - \alpha_{j,k,l}\right| \le \varepsilon.$$

Letting

$$S := \sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{m}\Psi(x_j,x_k,t_l)(\mu(E_l)\xi_j,\xi_k)_\Lambda,$$

we get by the above equation that

$$\begin{aligned}\left|\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\left(\int_\Omega f_{j,k}d\mu\right)\xi_j,\xi_k\right)_\Lambda - S\right| &\le \left|\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{m}|\alpha_{j,k,l} - \Psi(x_j,x_k,t_l)|(\mu(E_l)\xi_j,\xi_k)_\Lambda\right| \\ &\le \varepsilon\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{m}\|\mu(E_l)\|_{\mathcal{L}(\Lambda)}\|\xi_j\|_\Lambda\|\xi_k\|_\Lambda \le \varepsilon|\mu|(\Omega)\left(\sum_{j=1}^{n}\|\xi_j\|_\Lambda\right)^2.\end{aligned} \tag{23}$$

Combining (22) and (23) yields that

$$\left|\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda - S\right| \le 2\varepsilon|\mu|(\Omega)\left(\sum_{j=1}^{n}\|\xi_j\|_\Lambda\right)^2. \tag{24}$$

Since $\Psi(\cdot,\cdot,t_l)$ is a scalar-valued reproducing kernel on X, $[\Psi(x_j,x_k,t_l) : j,k \in \mathbb{N}_n]$ is a positive semi-definite matrix for each $l \in \mathbb{N}_m$. So are $[(\mu(E_l)\xi_j,\xi_k)_\Lambda : j,k \in \mathbb{N}_n]$, $l \in \mathbb{N}_m$ as $\mu(E_l) \in \mathcal{L}_+(\Lambda)$. By the Schur product theorem (see, for example, Horn and Johnson, 1991, page 309), the Hadamard product of two positive semi-definite matrices remains positive semi-definite. We obtain by this fact that $S > 0$, which together with (24), and the fact that ε can be arbitrarily small, proves (1). ■

To investigate the refinement relationship, we shall consider a simplified version of (20) that covers a large class of operator-valued reproducing kernels. Let $\phi : X \times \Omega \to \mathbb{C}$ be such that $\phi(x,\cdot)$ is a bounded $\mathcal{F}$-measurable function for every $x \in X$ and such that

$$\overline{\operatorname{span}}\{\phi(x,\cdot) : x \in X\} = L^2(\Omega,d\gamma) \text{ for any finite nonnegative measure } \gamma \text{ on } \mathcal{F}. \tag{25}$$

We shall see by the concrete examples in the next section that the denseness requirement (25) is not too restricted in applications. The kernels we shall consider are

$$K(x,y) := \int_{\Omega} \phi(x,t)\overline{\phi(y,t)}d\mu(t), \;\; x,y \in X \tag{26}$$

and

$$G(x,y) := \int_{\Omega} \phi(x,t)\overline{\phi(y,t)}d\nu(t), \;\; x,y \in X, \tag{27}$$

where μ,ν are two $\mathcal{L}_+(\Lambda)$-valued measures on $\mathcal{F}$ of bounded variation. By Proposition 8, K,G are $\mathcal{L}(\Lambda)$-valued reproducing kernels on X. Our idea is to use the Radon-Nikodym property of vector-valued measures to study the refinement property.

Let $\mathcal{B}$ be a Banach space and γ a finite nonnegative measure on $\mathcal{F}$. We say that a $\mathcal{B}$-valued measure ρ on $\mathcal{F}$ of bounded variation has the *Radon-Nikodym property* with respect to γ if there is a γ-Bochner integrable function $\Gamma : \Omega \to \mathcal{L}_+(\Lambda)$ such that for all $E \in \mathcal{F}$

$$\rho(E) = \int_E \Gamma d\gamma.$$

Apparently, this could only be true when ρ is absolutely continuous with respect to γ. For this reason, we also say that the space $\mathcal{B}$ has the Radon-Nikodym property with respect to γ if every $\mathcal{B}$-valued measure of bounded variation that is absolutely continuous with respect to γ has the Radon-Nikodym property with respect to γ. Moreover, $\mathcal{B}$ is said to have the Radon-Nikodym property if it has it with respect to any finite nonnegative measure on any measure space $\mathcal{F}$.

Strikingly different from the scalar-valued case, a Banach space $\mathcal{B}$ may not have the Radon-Nikodym property. For instance, the Banach space c_0 of all sequences $\alpha := (\alpha_j \in \mathbb{C} : j \in \mathbb{N})$ with

$$\lim_{j\to\infty} |\alpha_j| = 0$$

under the norm $\|\alpha\|_{c_0} := \sup\{|\alpha_j| : j \in \mathbb{N}\}$ does not have the property with respect to the Lebesgue measure (see, Diestel and Uhl, 1977, page 60). Consequently, the space $\mathcal{L}(\Lambda)$ does not have the Radon-Nikodym property when Λ is infinite-dimensional. To see this, since Λ is separable we let $\{e_j : j \in \mathbb{N}\}$ be an orthonormal basis for Λ. Denote by $\mathcal{L}_0(\Lambda)$ the set of all the operators $T \in \mathcal{L}(\Lambda)$ such that

$$Te_j = \alpha_j e_j, \;\; j \in \mathbb{N}$$

for some $\alpha \in c_0$. One sees that $\|T\|_{\mathcal{L}(\Lambda)} = \|\alpha\|_{c_0}$ (Conway, 1990). As a result, $\mathcal{L}_0(\Lambda)$ is a closed subspace of $\mathcal{L}(\Lambda)$ that is isometrically isomorphic to c_0. Since c_0 does not have the Radon-Nikodym property, neither does $\mathcal{L}_0(\Lambda)$. A Banach space has the Radon-Nikodym property if and only if each of its closed linear subspaces does (Diestel and Uhl, 1977). By this fact, $\mathcal{L}(\Lambda)$ does not have Radon-Nikodym property.

We shall focus on the situation where this desired property holds. For example, reflexive Banach spaces have the Radon-Nikodym property (Diestel and Uhl, 1977). In applications, Λ is usually finite-dimensional. In this case, $\mathcal{L}(\Lambda)$ is of finite dimension as well. Any two norms on a finite-dimensional Banach space are equivalent and a finite-dimensional $\mathcal{L}(\Lambda)$ can be endowed with a norm that makes it a Hilbert space. It yields that $\mathcal{L}(\Lambda)$ is reflexive. The conclusion is that when Λ is finite-dimensional, $\mathcal{L}(\Lambda)$ does have the Radon-Nikodym property. Another way of overcoming the difficulty is to confine to a subclass of $\mathcal{L}(\Lambda)$, for example, to the Schatten class (Birman and

Solomjak, 1987). Denote for each compact operator $T \in \mathcal{L}(\Lambda)$ by $s_j(T)$, $j \in \mathbb{N}$, the nonnegative square root of the j-th largest eigenvalue of T^*T. It is called the j-th *singular number* of T. For $p \in (1,+\infty)$, the p-th Schatten class $\mathcal{S}_p(\Lambda)$ consists of all the compact linear operators $T \in \mathcal{L}(\Lambda)$ with the norm

$$\|T\|_{\mathcal{S}_p(\Lambda)} := \left(\sum_{j=1}^{\infty}(s_j(T))^p\right)^{1/p} < +\infty.$$

The p-th Schatten class $S_p(\Lambda)$ is a reflexive Banach space and hence has the Radon-Nikodym property. When $p = 2, S_2(\Lambda)$ is the class of Hilbert-Schmidt operators and

$$\|T\|_{\mathcal{S}_2(\Lambda)} = \left(\sum_{j=1}^{\infty}\|Te_j\|_{\Lambda}\right)^{1/2}.$$

We shall not go into further details about the Radon-Nikodym property. Interested readers are referred to Chapter III of Diestel and Uhl (1977) and the references therein.

The assumption we shall need is that there exists a finite nonnegative measure γ on $\mathcal{F}$ such that both μ and ν have the Radon-Nikodym property with respect to γ. In other words, there exist γ-Bochner integrable functions $\Gamma_\mu, \Gamma_\nu : \Omega \to \mathcal{L}_+(\Lambda)$ such that

$$\mu(E) = \int_E \Gamma_\mu d\gamma \quad \text{and} \quad \nu(E) = \int_E \Gamma_\nu d\gamma \quad \text{for all } E \in \mathcal{F}. \tag{28}$$

Such two functions exist if $\gamma := |\mu| + |\nu|$ and μ, ν take values in the p-th Schatten class of $\mathcal{L}(\Lambda)$, $1 < p < +\infty$.

Suppose that K, G are given by (26) and (27), where ϕ, μ, ν satisfy (25) and (28). Our purpose is to investigate $\mathcal{H}_K \preceq \mathcal{H}_G$. To this end, let us first identify $\mathcal{H}_{\tilde{K}}$ and $\mathcal{H}_{\tilde{G}}$. We shall only present results for $\mathcal{H}_{\tilde{K}}$ as those for $\mathcal{H}_{\tilde{G}}$ have a similar form.

Lemma 9 *The RKHS $\mathcal{H}_{\tilde{K}}$ consists of functions F_f of the form*

$$F_f(x,\xi) := \int_\Omega (\Gamma_\mu(t)f(t),\xi)_\Lambda \overline{\phi(x,t)} d\gamma(t), \;\; x \in X, \; \xi \in \Lambda,$$

where f can be an arbitrary element from the Hilbert space $\mathcal{W}_\mu$ of γ-measurable functions from Ω to Λ such that

$$\|f\|_{\mathcal{W}_\mu} := \left(\int_\Omega (\Gamma_\mu(t)f(t),f(t))_\Lambda d\gamma(t)\right)^{1/2} < +\infty.$$

Moreover, $\|F_f\|_{\mathcal{H}_{\tilde{K}}} = \|f\|_{\mathcal{W}_\mu}$ for all $f \in \mathcal{W}_\mu$.

Proof We observe for all $x, y \in X$ and $\xi, \eta \in \Lambda$ that

$$\tilde{K}((x,\xi),(y,\eta)) = \int_\Omega \phi(x,t)\overline{\phi(y,t)}(\Gamma_\mu(t)\xi,\eta)_\Lambda d\gamma(t).$$

Thus, we may choose $\mathcal{W}_\mu$ as a feature space for $\tilde{K}$. The associated feature map $\Phi_\mu : X \times \Lambda \to \mathcal{W}_\mu$ is then selected as

$$\Phi_\mu(x,\xi)(t) := \phi(x,t)\xi, \;\; t \in \Omega.$$

We next verify the denseness condition that $\overline{\text{span}}\{\Phi_\mu(x,\xi):\ x\in X,\ \xi\in\Lambda\} = \mathcal{W}_\mu$. Suppose that $f\in\mathcal{W}_\mu$ is orthogonal to $\Phi_\mu(x,\xi)$ for all $x\in X$ and $\xi\in\Lambda$, that is,

$$\int_\Omega (\Gamma_\mu f(t),\xi)_\Lambda \overline{\phi(x,t)}d\gamma(t) = 0 \text{ for all } x\in X,\ \xi\in\Lambda.$$

By (25),

$$(\Gamma_\mu(t)f(t),\xi)_\Lambda = 0\ \gamma - \text{a.e.}$$

As this holds for an arbitrary $\xi\in\Lambda$, $\Gamma_\mu(t)f(t)=0\ \gamma-\text{a.e.}$ It implies that $\|f\|_{\mathcal{W}_\mu}=0$. The result now follows immediately from Lemma 2. ■

For two operators $A,B\in\mathcal{L}_+(\Lambda)$, we write $A\preceq B$ if for all $\xi\in\Lambda$ there exists some $\eta\in\Lambda$ such that

$$A\xi = B\eta \text{ and } (A\xi,\xi)_\Lambda = (B\eta,\eta)_\Lambda. \tag{29}$$

We make a simple observation about this special relationship between two linear operators.

Let $\ker(A)$ and $\text{ran}\,(A)$ be the kernel and range of A, respectively. If $\text{ran}\,(A)$ is closed then as A is self-adjoint, there holds the direct sum decomposition

$$\Lambda = \ker(A)\oplus\text{ran}\,(A). \tag{30}$$

Thus, A is bijective and bounded from $\text{ran}\,(A)$ to $\text{ran}\,(A)$. By the open mapping theorem, it has a bounded inverse on $\text{ran}\,(A)$, which we denote by A^{-1}.

Proposition 10 *Suppose that $A,B\in\mathcal{L}_+(\Lambda)$ have closed range. Then $A\preceq B$ if and only if*

$$\text{ran}\,(A)\subseteq\text{ran}\,(B) \tag{31}$$

and

$$P_{B,A}B^{-1} = A^{-1} \text{ on } \text{ran}\,(A), \tag{32}$$

where $P_{B,A}$ denotes the orthogonal projection from $\text{ran}\,(B)$ *to* $\text{ran}\,(A)$. *Particularly, if A is onto then $A\preceq B$ if and only if $A=B$.*

Proof Let A,B have closed range. Suppose first that $A\preceq B$. Then (31) clearly holds true. Set for each $\xi\in\text{ran}\,(A)$

$$\eta_\xi := B^{-1}A\xi.$$

Clearly, the mapping $\xi\to\eta_\xi$ is linear from $\text{ran}\,(A)$ to $\text{ran}\,(B)$. Thus, we have for arbitrary $\xi,\xi'\in\Lambda$ that

$$(A\xi'+A\xi,\xi'+\xi)_\Lambda = (B\eta_{\xi'+\xi},\eta_{\xi'+\xi})_\Lambda = (B\eta_{\xi'}+B\eta_\xi,\eta_{\xi'}+\eta_\xi)_\Lambda,$$

which implies that

$$\text{Re}\,(A\xi',\xi)_\Lambda = \text{Re}\,(B\eta_{\xi'},\eta_\xi)_\Lambda.$$

A textbook trick yields that for all $\xi,\xi'\in\text{ran}\,(A)$,

$$(A\xi',\xi)_\Lambda = (B\eta_{\xi'},\eta_\xi)_\Lambda = (A\xi',\eta_\xi)_\Lambda.$$

We hence obtain that $\xi - \eta_\xi \in \ker(A)$ for all $\xi \in \operatorname{ran}(A)$. Consequently,

$$A\xi - AB^{-1}A\xi = A\xi - A\eta_\xi = 0 \text{ for all } \xi \in \operatorname{ran}(A),$$

from which (32) follows.

On the other hand, suppose that (31) and (32) hold true. Then we choose for each $\xi \in \Lambda$

$$\eta := B^{-1}A\xi$$

and verify that $B\eta = A\xi$ and

$$(B\eta,\eta)_\Lambda = (A\xi, B^{-1}A\xi)_\Lambda = (A\xi, P_{B,A}B^{-1}A\xi)_\Lambda = (A\xi, A^{-1}A\xi)_\Lambda = (A\xi,\xi)_\Lambda.$$

Finally, if A is onto then by (31), $\operatorname{ran}(A) = \operatorname{ran}(B) = \Lambda$. According to (30), both A and B are injective. Therefore, they possess a bounded inverse on Λ. It implies that $P_{B,A}$ is the identity operator on Λ. By Equation (32), $A = B$. The proof is complete. ■

We are ready to present the main result of this section.

Theorem 11 *Let K, G be given by (26) and (27), where ϕ, μ, ν satisfy (25) and (28). Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\Gamma_\mu \preceq \Gamma_\nu$ $\gamma-a.e.$*

Proof By Proposition 6 and Lemma 9, $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if for all $f \in \mathcal{W}_\mu$, there exists some $g \in \mathcal{W}_\nu$ such that

$$\int_\Omega (\Gamma_\mu(t)f(t),\xi)_\Lambda \overline{\phi(x,t)}d\gamma(t) = \int_\Omega (\Gamma_\nu(t)g(t),\xi)_\Lambda \overline{\phi(x,t)}d\gamma(t) \text{ for all } x \in X,\ \xi \in \Lambda \tag{33}$$

and

$$\int_\Omega (\Gamma_\mu(t)f(t),f(t))_\Lambda d\gamma(t) = \int_\Omega (\Gamma_\nu(t)g(t),g(t))_\Lambda d\gamma(t). \tag{34}$$

By the denseness condition (25), (33) holds true if and only if

$$(\Gamma_\mu(t)f(t),\xi)_\Lambda = (\Gamma_\nu(t)g(t),\xi)_\Lambda \ \text{ for } \gamma-\text{a.e. } t \in \Omega \text{ and all } \xi \in \Lambda,$$

which is equivalent to

$$\Gamma_\mu(t)f(t) = \Gamma_\nu(t)g(t) \text{ for } \gamma-\text{a.e. } t \in \Omega. \tag{35}$$

We conclude that $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if for every $f \in \mathcal{W}_\mu$, there exists some $g \in \mathcal{W}_\nu$ such that Equations (34) and (35) hold true.

Suppose that $\Gamma_\mu \preceq \Gamma_\nu$ $\gamma-$a.e. Then clearly, for each $f \in \mathcal{W}_\mu$, we can find a function $g : \Omega \to \Lambda$ which is defined γ-almost everywhere and satisfies (35) and

$$(\Gamma_\mu(t)f(t),f(t))_\Lambda = (\Gamma_\nu(t)g(t),g(t))_\Lambda \text{ for } \gamma-\text{a.e. } t \in \Omega.$$

The above equation implies (34). Therefore, $\mathcal{H}_K \preceq \mathcal{H}_G$.

On the other hand, suppose that we can find for every $f \in \mathcal{W}_\mu$ some $g_f \in \mathcal{W}_\nu$ satisfying (34) and (35). The function g_f can be chosen so that $f \to g_f$ is linear from $\mathcal{W}_\mu$ to $\mathcal{W}_\nu$. A trick similar to that used in Lemma 9 enables us to obtain from (34) and (35) that

$$\int_\Omega (\Gamma_\mu(t)f'(t), f(t) - g_f(t))_\Lambda d\gamma(t) = 0 \text{ for all } f' \in \mathcal{W}_\mu.$$

Letting $f' := \phi(x,\cdot)\xi$ for arbitrary $x \in X$ and $\xi \in \Lambda$ in the above equation and invoking (25), we have that

$$\Gamma_\mu(t)(f(t) - g_f(t)) = 0 \text{ for } \gamma - \text{a.e. } t \in \Omega.$$

By the above equation and (35), we get for γ-almost every $t \in \Omega$ that

$$(\Gamma_\nu(t)g_f(t), g_f(t))_\Lambda = (\Gamma_\mu(t)f(t), g_f(t))_\Lambda = (f(t), \Gamma_\mu(t)g_f(t))_\Lambda = (f(t), \Gamma_\mu(t)f(t))_\Lambda = (\Gamma_\mu(t)f(t), f(t))_\Lambda.$$

Since (35) and the above equation are true for an arbitrary $f \in \mathcal{W}_\mu$, $\Gamma_\mu \preceq \Gamma_\nu \ \gamma -$ a.e. ■

5. Examples

We present in this section several concrete examples of refinement of operator-valued reproducing kernels. They are built on the general characterizations established in the last two sections.

5.1 Translation Invariant Reproducing Kernels

Let $d \in \mathbb{N}$ and K be an $\mathcal{L}(\Lambda)$-valued reproducing kernel on $\mathbb{R}^d$. We say that K is *translation invariant* if for all $x, y, a \in \mathbb{R}^d$

$$K(x-a, y-a) = K(x,y).$$

A celebrated characterization due to Bochner (1959) states that every continuous scalar-valued translation invariant reproducing kernel on $\mathbb{R}^d$ must be the Fourier transform of a finite nonnegative Borel measure on $\mathbb{R}^d$, and vice versa. This result has been generalized to the operator-valued case (Berberian, 1966; Carmeli et al., 2010; Fillmore, 1970). Specifically, a continuous function K from $\mathbb{R}^d \times \mathbb{R}^d$ to $\mathcal{L}(\Lambda)$ is a translation invariant reproducing kernel if and only if it has the form

$$K(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\mu(t), \ \ x, y \in \mathbb{R}^d, \tag{36}$$

for some $\mu \in \mathcal{B}(\mathbb{R}^d, \Lambda)$, the set of all the $\mathcal{L}_+(\Lambda)$-valued measures of bounded variation on the σ-algebra of Borel subsets in $\mathbb{R}^d$. Let G be the kernel given by

$$G(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\nu(t), \ \ x, y \in \mathbb{R}^d, \tag{37}$$

where $\nu \in \mathcal{B}(\mathbb{R}^d, \Lambda)$. The purpose of this subsection is to characterize $\mathcal{H}_K \preceq \mathcal{H}_G$ in terms of μ, ν. To this end, we first investigate the structure of the RKHS of a translation invariant $\mathcal{L}(\Lambda)$-valued reproducing kernel.

Let γ be an arbitrary measure in $\mathcal{B}(\mathbb{R}^d, \Lambda)$ and L the associated translation invariant reproducing kernel defined by

$$L(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma(t), \ \ x, y \in \mathbb{R}^d.$$

There exists a decomposition of γ with respect to the Lebesgue measure dx on $\mathbb{R}^d$ (Diestel and Uhl, 1977) as follows:

$$\gamma = \gamma_c + \gamma_s, \tag{38}$$

where γ_c, γ_s are the unique measures in $\mathcal{B}(\mathbb{R}^d, \Lambda)$ such that γ_c is absolutely continuous with respect to dx, and for each continuous linear functional λ on $\mathcal{L}(\Lambda)$, the scalar-valued measure $\lambda\gamma_s$ and dx are mutually singular. It follows from this decomposition of measures a decomposition of L:

$$L = L_c + L_s,$$

where

$$L_c(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma_c(t), \;\; L_s(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma_s(t), \quad x, y \in \mathbb{R}^d. \tag{39}$$

Our first observation is that $\mathcal{H}_L$ is the orthogonal direct sum of $\mathcal{H}_{L_c}$ and $\mathcal{H}_{L_s}$. Two lemmas are needed to prove this useful fact.

Lemma 12 *Let L_c, L_s be given by (39). Then for all $\xi \in \Lambda$ and $x, y \in \mathbb{R}^d$*

$$(L_a(x,y)\xi, \xi)_\Lambda = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma_{a,\xi}(t), \;\; a = c \text{ or } s, \tag{40}$$

where $\gamma_{a,\xi}$ is a scalar-valued Borel measure on $\mathbb{R}^d$ defined for each Borel set $E \subseteq \mathbb{R}^d$ by

$$\gamma_{a,\xi}(E) := (\gamma_a(E)\xi, \xi)_\Lambda, \;\; a = c \text{ or } s.$$

Proof Let $a \in \{c, s\}$, $\xi \in \Lambda$, $x, y \in \mathbb{R}^d$, and s_n be a sequence of simple functions on $\mathbb{R}^d$ that converges to $e^{i(x-y)\cdot t}$ in $L^\infty(\mathbb{R}^d, dx)$. Then

$$\lim_{n\to\infty} \left(\left(\int_{\mathbb{R}^d} s_n d\gamma_a \right) \xi, \xi \right)_\Lambda = (L_a(x,y)\xi, \xi)_\Lambda.$$

By definition, we have for each $n \in \mathbb{N}$ that

$$\lim_{n\to\infty} \left(\left(\int_{\mathbb{R}^d} s_n d\gamma_a \right) \xi, \xi \right)_\Lambda = \int_{\mathbb{R}^d} s_n d\gamma_{a,\xi}.$$

As

$$\lim_{n\to\infty} \int_{\mathbb{R}^d} s_n d\gamma_{a,\xi} = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma_{a,\xi}(t),$$

we conclude from the previous two equations that (40) holds true. ■

Lemma 13 *There holds $\mathcal{H}_{L_c} \cap \mathcal{H}_{L_s} = \{0\}$.*

Proof We introduce for each $\xi \in \Lambda$ two scalar-valued translation invariant reproducing kernels on $\mathbb{R}^d$ by setting

$$A_a(x,y) := (L_a(x,y)\xi, \xi)_\Lambda, \;\; x, y \in \mathbb{R}^d, \;\; a \in \{c, s\}.$$

By Lemma 12, we have the alternative representations for A_c and A_s

$$A_a(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t} d\gamma_{a,\xi}(t), \;\; x, y \in \mathbb{R}^d, \;\; a = c \text{ or } s.$$

By the Lebesgue decomposition of γ, $\gamma_{c,\xi}$ is absolutely continuous with respect to dx while $\gamma_{s,\xi}$ and dx are mutually singular. As a consequence, $\mathcal{H}_{A_c} \cap \mathcal{H}_{A_s} = \{0\}$ by Lemma 17 in Xu and Zhang (2009).

Let $a \in \{c,s\}$. By (3),

$$A_a(x,y) = (L_a(x,\cdot)\xi, L_a(y,\cdot)\xi)_{\mathcal{H}_{L_a}},\ \ x,y \in \mathbb{R}^d.$$

A feature map for A_a may hence be chosen as

$$\Phi_a(x) := L_a(x,\cdot)\xi,\ \ x \in \mathbb{R}^d$$

with the feature space being $\mathcal{H}_{L_a}$. We identify by Lemma 2 that

$$\mathcal{H}_{A_a} = \{(\tilde{f}(\cdot),\xi)_\Lambda : \tilde{f} \in \mathcal{H}_{L_a}\}. \tag{41}$$

Assume that $\mathcal{H}_{L_c} \cap \mathcal{H}_{L_s} \neq \{0\}$. Then there exist nontrivial functions $\tilde{f} \in \mathcal{H}_{L_c}$ and $\tilde{g} \in \mathcal{H}_{L_s}$ such that $\tilde{f} = \tilde{g}$. As a result, there exists some $\xi \in \Lambda$ such that $(\tilde{f}(\cdot),\xi)_\Lambda$ is not the trivial function. By equation (41)

$$(\tilde{f}(\cdot),\xi)_\Lambda = (\tilde{g}(\cdot),\xi)_\Lambda \in \mathcal{H}_{A_c} \cap \mathcal{H}_{A_s},$$

contradicting the fact that $\mathcal{H}_{A_c} \cap \mathcal{H}_{A_s} = \{0\}$. ■

Theorem 14 *The space $\mathcal{H}_L$ is the orthogonal direct sum of $\mathcal{H}_{L_c}$ and $\mathcal{H}_{L_s}$, namely, $\mathcal{H}_L = \mathcal{H}_{L_c} \bigoplus \mathcal{H}_{L_s}$.*

Proof The result follows directly from Lemma 13 and Proposition 1. ■

We are now in a position to study the refinement relationship $\mathcal{H}_K \preceq \mathcal{H}_G$, where K, G are defined by (36) and (37). Firstly, the task can be separated into two related ones according to the Lebesgue decomposition of measures μ, ν.

Proposition 15 *There holds $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_{K_c} \preceq \mathcal{H}_{G_c}$ and $\mathcal{H}_{K_s} \preceq \mathcal{H}_{G_s}$.*

Proof By Theorem 14, $\mathcal{H}_K = \mathcal{H}_{K_c} \bigoplus \mathcal{H}_{K_s}$ and $\mathcal{H}_G = \mathcal{H}_{G_c} \bigoplus \mathcal{H}_{G_s}$. Therefore, if $\mathcal{H}_{K_c} \preceq \mathcal{H}_{G_c}$ and $\mathcal{H}_{K_s} \preceq \mathcal{H}_{G_s}$ then $\mathcal{H}_K \preceq \mathcal{H}_G$.

On the other hand, suppose that $\mathcal{H}_K \preceq \mathcal{H}_G$. Let $f \in \mathcal{H}_{K_c}$. Then $f \in \mathcal{H}_K$ and $\|f\|_{\mathcal{H}_{K_c}} = \|f\|_{\mathcal{H}_K}$. Since $\mathcal{H}_K \preceq \mathcal{H}_G$, there exists $g \in \mathcal{H}_{G_c}$ and $h \in \mathcal{H}_{G_s}$ such that

$$f = g + h$$

and

$$\|f\|^2_{\mathcal{H}_{K_c}} = \|f\|^2_{\mathcal{H}_K} = \|g+h\|^2_{\mathcal{H}_G} = \|g\|^2_{\mathcal{H}_{G_c}} + \|h\|^2_{\mathcal{H}_{G_s}}.$$

Therefore, to show that $\mathcal{H}_{K_c} \preceq \mathcal{H}_{G_c}$ it suffices to show that $h = 0$. Assume that $h \neq 0$. Note that $f - g \in \mathcal{H}_{K_c+G_c}$ (Pedrick, 1957), we get that

$$\mathcal{H}_{K_c+G_c} \cap \mathcal{H}_{G_s} \neq \{0\}. \tag{42}$$

However,

$$(K_c+G_c)(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}d(\mu_c+\nu_c)(t),\ \ x,y\in\mathbb{R}^d$$

and $\mu_c+\nu_c$ is absolutely continuous with respect to dx. Thus, Equation (42) contradicts Lemma 13. The contradiction proves that $\mathcal{H}_{K_c}\preceq\mathcal{H}_{G_c}$. Likewise, one can prove that $\mathcal{H}_{K_s}\preceq\mathcal{H}_{G_s}$. ■

By Proposition 15, we shall study $\mathcal{H}_{K_c}\preceq\mathcal{H}_{G_c}$ and $\mathcal{H}_{K_s}\preceq\mathcal{H}_{G_s}$ separately. We shall restrict ourselves to the case when the measures corresponding to K_c and G_c have the Radon-Nikodym property with respect to the Lebesgue measure and the measures corresponding to K_s and G_s are discrete. Specifically, the kernels to be considered are of the following special forms:

$$K_c(x,y):=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\varphi_1(t)dt,\ \ G_c(x,y):=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\varphi_2(t)dt,\ \ x,y\in\mathbb{R}^d \tag{43}$$

and

$$K_s(x,y):=\sum_{j\in\mathbb{J}_1} e^{i(x-y)\cdot t_j}A_j,\ \ G_s(x,y):=\sum_{k\in\mathbb{J}_2} e^{i(x-y)\cdot t_k}B_k,\ \ x,y\in\mathbb{R}^d.$$

Here, φ_1,φ_2 are two dx-Bochner integrable functions from $\mathbb{R}^d$ to $\mathcal{L}_+(\Lambda)$, $\{t_j:j\in\mathbb{J}_1\}$ and $\{t_k:k\in\mathbb{J}_2\}$ are countable sets of pairwise distinct points in $\mathbb{R}^d$, and A_j,B_j are nonzero operators in $\mathcal{L}_+(\Lambda)$ such that

$$\sum_{j\in\mathbb{J}_1}\|A_j\|_{\mathcal{L}(\Lambda)}<+\infty,\ \ \sum_{k\in\mathbb{J}_2}\|B_k\|_{\mathcal{L}(\Lambda)}<+\infty.$$

The following characterization is a direct consequence of Theorem 11.

Proposition 16 *Let K_c,G_c be given by (43). Then $\mathcal{H}_{K_c}\preceq\mathcal{H}_{G_c}$ if and only if $\varphi_1(t)\preceq\varphi_2(t)$ for almost every $t\in\mathbb{R}^d$ except for a subset in $\mathbb{R}^d$ of zero Lebesgue measure.*

Proof As φ_1,φ_2 are dx-Bochner integrable,

$$\int_{\mathbb{R}^d}\|\varphi_j(t)\|_{\mathcal{L}(\Lambda)}dt<+\infty,\ \ j=1,2.$$

Define a finite nonnegative Borel measure γ on $\mathbb{R}^d$ by setting for each Borel subset E in $\mathbb{R}^d$

$$\gamma(E):=\int_E\|\varphi_1(t)\|_{\mathcal{L}(\Lambda)}+\|\varphi_2(t)\|_{\mathcal{L}(\Lambda)}dt.$$

Evidently, K_c,G_c have the form

$$K_c(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\Gamma_1(t)d\gamma(t),\ \ G_c(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\Gamma_2(t)d\gamma(t),\ \ \ x,y\in\mathbb{R}^d,$$

where for $j=1,2$,

$$\Gamma_j(t):=\begin{cases}\dfrac{\varphi_j(t)}{\|\varphi_1(t)\|_{\mathcal{L}(\Lambda)}+\|\varphi_2(t)\|_{\mathcal{L}(\Lambda)}}, & \text{if } \|\varphi_1(t)\|_{\mathcal{L}(\Lambda)}+\|\varphi_2(t)\|_{\mathcal{L}(\Lambda)}>0,\\ 0, & \text{otherwise.}\end{cases}$$

It is also clear that $\operatorname{span}\{e^{ix\cdot t}:x\in\mathbb{R}^d\}$ is dense in $L^2(\mathbb{R}^d,d\gamma)$. By Theorem 11, $\mathcal{H}_{K_c}\preceq\mathcal{H}_{G_c}$ if and only if $\Gamma_1\preceq\Gamma_2\ \gamma-$a.e. Note that $\Gamma_1(t)\preceq\Gamma_2(t)$ if and only if $\varphi_1(t)\preceq\varphi_2(t)$. If $\varphi_1\preceq\varphi_2\ dx-$a.e. then

$\Gamma_1 \preceq \Gamma_2$ $\gamma-$a.e. as γ is absolutely continuous with respect to the Lebesgue measure. On the other hand, suppose that $\Gamma_1 \preceq \Gamma_2$ $\gamma-$a.e. Set

$$E := \{t \in \mathbb{R}^d : \|\varphi_1(t)\|_{\mathcal{L}(\Lambda)} + \|\varphi_2(t)\|_{\mathcal{L}(\Lambda)} > 0\}.$$

For $t \in E^c$, $\varphi_1(t) = \varphi_2(t) = 0$, and thus, $\varphi_1(t) \preceq \varphi_2(t)$. Assume that there exists a Borel subset $F \subseteq \mathbb{R}^d$ with a positive Lebesgue measure on which $\varphi_1(t) \not\preceq \varphi_2(t)$. Then $F \subseteq E$. We reach that $\gamma(F) > 0$ and $\Gamma_1(t) \not\preceq \Gamma_2(t)$ for $t \in F$, contradicting the fact that $\Gamma_1 \preceq \Gamma_2$ $\gamma-$a.e. ■

For K_s, G_s, we have the following result.

Proposition 17 *There holds $\mathcal{H}_{K_s} \preceq \mathcal{H}_{G_s}$ if and only if*

(1) $\{t_j : j \in \mathbb{J}_1\} \subseteq \{t_k : k \in \mathbb{J}_2\}$*;*

(2) *for each $j \in \mathbb{J}_1$, $A_j \preceq B_j$. Here, re-indexing by condition (1) if necessary, we may assume that $\mathbb{J}_1 \subseteq \mathbb{J}_2$.*

Proof Introduce a discrete scalar-valued Borel measure γ that is supported on $\{t_j : j \in \mathbb{J}_1\} \cup \{t_k : k \in \mathbb{J}_2\}$ by setting

$$\gamma(\{t_k\}) := \begin{cases} \|A_k\|_{\mathcal{L}(\Lambda)} + \|B_k\|_{\mathcal{L}(\Lambda)}, & k \in \mathbb{J}_1 \cap \mathbb{J}_2, \\ \|B_k\|_{\mathcal{L}(\Lambda)}, & k \in \mathbb{J}_2 \setminus \mathbb{J}_1, \\ \|A_k\|_{\mathcal{L}(\Lambda)}, & k \in \mathbb{J}_1 \setminus \mathbb{J}_2. \end{cases}$$

We also let

$$\Gamma_A(t_j) := \frac{A_j}{\gamma(\{t_j\})},\ \ j \in \mathbb{J}_1 \text{ and } \Gamma_A(t_k) := \frac{B_k}{\gamma(\{t_k\})},\ \ k \in \mathbb{J}_2.$$

They are discrete $\mathcal{L}(\Lambda)$-valued functions supported on $\{t_j : j \in \mathbb{J}_1\}$ and $\{t_k : k \in \mathbb{J}_2\}$, respectively. We reach the following integral representation:

$$K_s(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\Gamma_A(t)d\gamma(t) \text{ and } G_s(x,y) = \int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\Gamma_B(t)d\gamma(t),\ \ x,y \in \mathbb{R}^d.$$

By Theorem 11, $\mathcal{H}_{K_s} \preceq \mathcal{H}_{G_s}$ if and only if $\Gamma_A \preceq \Gamma_B$ $\gamma-$a.e. It is straightforward to verify that the latter is equivalent to conditions (1)-(2). ■

5.2 Hessian of Scalar-valued Reproducing Kernels

Propositions 16 and 17 were established based on Theorem 11. In this subsection, we shall consider special translation invariant reproducing kernels and establish the characterization of refinement using Theorem 7.

Let k be a continuously differentiable translation invariant reproducing kernel on $\mathbb{R}^d$. We consider the following matrix-valued functions

$$K(x,y) := \nabla^2_{xy}k(x,y) := \left[\frac{\partial^2 k}{\partial_{x_j}\partial_{y_k}}(x,y) : j,k \in \mathbb{N}_d\right],\ \ x,y \in \mathbb{R}^d. \tag{44}$$

To ensure that K is an $\mathcal{L}(\mathbb{C}^d)$-valued reproducing kernels on $\mathbb{R}^d$, we make use of the Bochner theorem to get some finite nonnegative Borel measure μ on $\mathbb{R}^d$ such that

$$k(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}d\mu(t),\ \ x,y\in\mathbb{R}^d \tag{45}$$

and impose the requirement that

$$\int_{\mathbb{R}^d} tt^T d\mu(t)<+\infty. \tag{46}$$

One sees by the Lebesgue dominated convergence theorem that

$$K(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}tt^T d\mu(t),\ \ x,y\in\mathbb{R}^d, \tag{47}$$

where we view $t\in\mathbb{R}^d$ as a $d\times 1$ vector and t^T denotes its transpose $[t_1,t_2,\ldots,t_d]$. By the general integral representation (17) of operator-valued reproducing kernels, K defined by (44) is an $\mathcal{L}(\mathbb{C}^d)$-valued reproducing kernel on $\mathbb{R}^d$. Matrix-valued translation invariant reproducing kernels of the form (44) are useful for the development of divergence-free kernel methods for solving some special partial differential equations (see, for example, Lowitzsh, 2003; Wendland, 2009, and the references therein). Another class of kernels constructed from the Hessian of a scalar-valued translation invariant reproducing kernel is widely applied to the learning of a multivariate function together with its gradient simultaneously (Mukherjee and Wu, 2006; Mukherjee and Zhou, 2006; Ying and Campbell, 2008). Such applications make use of kernels of the form

$$\overline{K}(x,y):=\begin{bmatrix} k(x,y) & (\nabla_y k(x,y))^* \\ \nabla_x k(x,y) & \nabla_{xy}^2 k(x,y)\end{bmatrix}. \tag{48}$$

One sees that under condition (46)

$$\overline{K}(x,y)=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}\rho(t)\rho(t)^* d\mu(t),\ \ x,y\in\mathbb{R}^d,$$

where

$$\rho(t)=[1,it_1,it_2,\ldots,it_d]^T,\ \ t\in\mathbb{R}^d.$$

We aim at refining matrix-valued reproducing kernels of the forms (44) and (48) in this subsection. Specifically, we let ν be another finite nonnegative Borel measure on $\mathbb{R}^d$ satisfying

$$\int_{\mathbb{R}^d} tt^T d\nu(t)<+\infty \tag{49}$$

and define for $x,y\in\mathbb{R}^d$

$$g(x,y):=\int_{\mathbb{R}^d} e^{i(x-y)\cdot t}d\nu(t),\ G(x,y):=\nabla_{xy}^2 g(x,y),\ \overline{G}(x,y):=\begin{bmatrix} g(x,y) & (\nabla_y g(x,y))^* \\ \nabla_x g(x,y) & \nabla_{xy}^2 g(x,y)\end{bmatrix}. \tag{50}$$

Our purpose is to characterize $\mathcal{H}_K\preceq\mathcal{H}_G$ and $\mathcal{H}_{\overline{K}}\preceq\mathcal{H}_{\overline{G}}$ in terms of k,g and μ,ν.

Theorem 18 *Let μ,ν be finite nonnegative Borel measures on $\mathbb{R}^d$ satisfying (46) and (49), and k,g defined by (45) and (50). Then $K,G,\overline{K},\overline{G}$ are matrix-valued translation invariant reproducing kernels on $\mathbb{R}^d$. The four relationships $\mathcal{H}_K\preceq\mathcal{H}_G$, $\mathcal{H}_{\overline{K}}\preceq\mathcal{H}_{\overline{G}}$, $\mathcal{H}_k\preceq\mathcal{H}_g$, and $\mu\preceq\nu$ are equivalent.*

Proof By Theorem 7 or a result in Xu and Zhang (2009), $\mathcal{H}_k \preceq \mathcal{H}_g$ if and only if $\mu \preceq \nu$. We shall show by Theorem 7 that $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mu \preceq \nu$. The equivalence of $\mathcal{H}_{\overline{K}} \preceq \mathcal{H}_{\overline{G}}$ and $\mu \preceq \nu$ can be proved similarly. Set

$$\phi(x,t) := e^{ix\cdot t} t^T, \;\; x,t \in \mathbb{R}^d.$$

Then for each $x,t \in \mathbb{R}^d$, $\phi(x,t)$ is a linear functional from $\mathbb{C}^d$ to $\mathbb{C}$. We observe by (47) that (17) holds true. So does (18). To apply Theorem 7, it remains to verify that $\operatorname{span}\{\phi(x,\cdot)\xi : x \in \mathbb{R}^d, \; \xi \in \mathbb{C}^d\}$ is dense in the Hilbert space $L^2(\mathbb{R}^d, d\mu)$, which is straightforward. The claim follows immediately from Theorem 7. ■

5.3 Transformation Reproducing Kernels

Let us consider a particular class of matrix-valued reproducing kernels whose universality was studied in Caponnetto et al. (2008). The kernels we shall construct are from an input space X to output space $\Lambda = \mathbb{C}^n$, where $n \in \mathbb{N}$. To this end, we let k, g be two scalar-valued reproducing kernels on another input space Y and T_p be mappings from X to Y, $p \in \mathbb{N}_n$. Set

$$K(x,y) := [k(T_p x, T_q y) : p,q \in \mathbb{N}_n], \;\; G(x,y) := [g(T_p x, T_q y) : p,q \in \mathbb{N}_n], \;\; x,y \in X. \tag{51}$$

It is known that K, G defined above are indeed $\mathcal{L}(\mathbb{C}^n)$-valued reproducing kernels (Caponnetto et al., 2008). This also becomes clear in the proof below. We are interested in the conditions for $\mathcal{H}_K \preceq \mathcal{H}_G$ to hold.

Proposition 19 *Let K, G be defined by (51). Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_{\bar{k}} \preceq \mathcal{H}_{\bar{g}}$, where $\bar{k}, \bar{g}$ are the restriction of k, g on $\cup_{p=1}^n T_p(X)$. In particular, if*

$$\bigcup_{p=1}^{n} T_p(X) = Y \tag{52}$$

then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_k \preceq \mathcal{H}_g$.

Proof It is legitimate to assume that (52) holds true as otherwise, we may replace Y by $\cup_{p=1}^n T_p(X)$, and k, g by $\bar{k}, \bar{g}$, respectively.

Choose arbitrary feature maps and feature spaces $\Phi_1 : Y \to \mathcal{W}_1$ for k and $\Phi_2 : Y \to \mathcal{W}_2$ for g such that

$$\overline{\operatorname{span}} \Phi_j(Y) = \mathcal{W}_j, \;\; j = 1,2. \tag{53}$$

By Proposition 6, $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$. We observe for all $x,y \in X$ and $\xi, \eta \in \mathbb{C}^n$ that

$$\begin{aligned}
\tilde{K}((x,\xi),(y,\eta)) &= (K(x,y)\xi,\eta)_{\mathbb{C}^n} = \sum_{p=1}^{n}\sum_{q=1}^{n} \xi_p \overline{\eta_q} k(T_p x, T_q y) \\
&= \sum_{p=1}^{n}\sum_{q=1}^{n} \xi_p \overline{\eta_q} (\Phi_1(T_p x), \Phi_1(T_q y))_{\mathcal{W}_1} \\
&= \left(\sum_{p=1}^{n} \xi_p \Phi_1(T_p x), \sum_{q=1}^{n} \eta_q \Phi_1(T_q y) \right)_{\mathcal{W}_1}.
\end{aligned}$$

Thus, $\tilde{\Phi}_1 : X \times \mathbb{C}^n \to \mathcal{W}_1$ defined by

$$\tilde{\Phi}_1(x,\xi) := \sum_{p=1}^{n} \xi_p \Phi_1(T_p x), \ \ x \in X, \ \xi \in \mathbb{C}^n$$

is a feature map for $\tilde{K}$. We next verify that $\operatorname{span}\{\tilde{\Phi}_1(x,\xi) : x \in X, \ \xi \in \mathbb{C}^n\}$ is dense in $\mathcal{W}_1$. Assume that $u \in \mathcal{W}_1$ is orthogonal to this linear span, that is,

$$\left(u, \sum_{p=1}^{n} \xi_p \Phi_1(T_p x)\right)_{\mathcal{W}_1} = 0 \text{ for all } x \in X, \ \xi \in \mathbb{C}^n.$$

Then we have $(u, \Phi_1(T_p x))_{\mathcal{W}_1} = 0$ for all $x \in X$ and $p \in \mathbb{N}_n$. It follows from (52) and (53) that $u = 0$. Similar facts hold for $\tilde{G}$.

By Lemma 2, $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$ if and only if for every $u \in \mathcal{W}_1$, there exists $v \in \mathcal{W}_2$ such that

$$\left(u, \sum_{p=1}^{n} \xi_p \Phi_1(T_p x)\right)_{\mathcal{W}_1} = \left(v, \sum_{p=1}^{n} \xi_p \Phi_2(T_p x)\right)_{\mathcal{W}_2} \text{ for all } x \in X \tag{54}$$

and

$$\|u\|_{\mathcal{W}_1} = \|v\|_{\mathcal{W}_2}. \tag{55}$$

Recall also that $\mathcal{H}_k \preceq \mathcal{H}_g$ if and only if for all $u \in \mathcal{W}_1$ there exists some $v \in \mathcal{W}_2$ satisfying (55) and

$$(u, \Phi_1(y))_{\mathcal{W}_1} = (v, \Phi_2(y))_{\mathcal{W}_2} \text{ for all } y \in Y. \tag{56}$$

Clearly, (56) implies (54). Conversely, if (54) holds true then we get that

$$(u, \Phi_1(T_p x))_{\mathcal{W}_1} = (v, \Phi_2(T_p x))_{\mathcal{W}_2} \text{ for all } x \in X \text{ and } p \in \mathbb{N}_n,$$

which together with (52) implies (56). We conclude that $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$ if and only if $\mathcal{H}_k \preceq \mathcal{H}_g$. ■

A more general case of refinement of transformation reproducing kernels is discussed below. It can be proved by arguments similar to those for the previous proposition.

Proposition 20 *Let T_p, S_p be mappings from X to Y and k, g be scalar-valued reproducing kernels on Y. Define*

$$K(x,y) := [k(T_p x, T_q y) : p,q \in \mathbb{N}_n], \ \ G(x,y) := [g(S_p x, S_q y) : p,q \in \mathbb{N}_n], \ \ x,y \in X.$$

Suppose that for all $p \in \mathbb{N}_n$, $\operatorname{span}\{k(T_p x, \cdot) : x \in X\}$ and $\operatorname{span}\{g(S_p x, \cdot) : x \in X\}$ are dense in $\mathcal{H}_k$ and $\mathcal{H}_g$, respectively. Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $\mathcal{H}_{k_p} \preceq \mathcal{H}_{g_p}$ for all $p \in \mathbb{N}_n$, where

$$k_p(x,y) := k(T_p x, T_p y), \ \ g_p(x,y) := g(S_p x, S_p y), \quad x,y \in X.$$

5.4 Finite Hilbert-Schmidt Reproducing Kernels

We consider refinement of finite Hilbert-Schmidt reproducing kernels in this subsection. Let B_j, C_j be invertible operators in $\mathcal{L}_+(\Lambda)$, $n \leq m \in \mathbb{N}$, and Ψ_j, $j \in \mathbb{N}_m$, be scalar-valued reproducing kernels on the input space X. Define

$$K(x,y) := \sum_{j=1}^{n} B_j \Psi_j(x,y), \;\; G(x,y) = \sum_{j=1}^{m} C_j \Psi_j(x,y), \quad x,y \in X. \tag{57}$$

By the general integral representation (20) and Proposition 8, K, G above are $\mathcal{L}(\Lambda)$-valued reproducing kernels on X. To ensure that representation (57) can not be further simplified, we shall work under the assumption that

$$\mathcal{H}_{\Psi_j} \cap \mathcal{H}_{\overline{\Psi}_j} = \{0\} \text{ for all } j \in \mathbb{N}_m, \tag{58}$$

where

$$\overline{\Psi}_j := \sum_{k \in \mathbb{N}_m \setminus \{j\}} \Psi_k.$$

Theorem 21 *Let K, G be defined by (57), where $B_j, C_j \in \mathcal{L}_+(\Lambda)$ are invertible and Ψ_j, $j \in \mathbb{N}_m$, are scalar-valued reproducing kernels on X satisfying (58). Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $B_j = C_j$, $j \in \mathbb{N}_n$.*

Proof We first find a feature map for $\tilde{K}$ and $\tilde{G}$. Let $\phi_j : X \to \mathcal{W}_j$ be an arbitrary feature map for Ψ_j such that $\operatorname{span}\phi_j(X)$ is dense in $\mathcal{W}_j$, and denote by $\Lambda \otimes \mathcal{W}_j$ the tensor product of Hilbert spaces Λ and $\mathcal{W}_j$, $j \in \mathbb{N}_m$. The space $\Lambda \otimes \mathcal{W}_j$ is a Hilbert space with the inner product

$$(\xi \otimes u, \eta \otimes v)_{\Lambda \otimes \mathcal{W}_j} := (\xi, \eta)_\Lambda (u, v)_{\mathcal{W}_j}, \;\; \xi, \eta \in \Lambda, \; u, v \in \mathcal{W}_j.$$

Set $\mathcal{W}$ the orthogonal direct sum of $\Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_n$, whose inner product is defined by

$$((\xi_j \otimes u_j : j \in \mathbb{N}_n), (\eta_j \otimes v_j : j \in \mathbb{N}_n))_{\mathcal{W}} := \sum_{j=1}^{n} (\xi_j, \eta_j)_\Lambda (u_j, v_j)_{\mathcal{W}_j}, \;\; \xi_j, \eta_j \in \Lambda, \; u_j, v_j \in \mathcal{W}_j, \;\; j \in \mathbb{N}_n.$$

We claim that $\Phi : X \times \Lambda \to \mathcal{W}$ defined by

$$\Phi(x, \xi) := (\sqrt{B_j}\xi \otimes \phi_j(x) : j \in \mathbb{N}_n), \;\; x \in X, \; \xi \in \Lambda$$

is a feature map for $\tilde{K}$. Here, $\sqrt{B_j}$, the square root of B_j, is the the unique operator A in $\mathcal{L}_+(\Lambda)$ such that $A^2 = B_j$. We verify for all $x, y \in X$ and $\xi, \eta \in \Lambda$ that

$$\begin{aligned}(\Phi(x,\xi), \Phi(y,\eta))_{\mathcal{W}} &= \sum_{j=1}^{n} (\sqrt{B_j}\xi, \sqrt{B_j}\eta)_\Lambda (\phi_j(x), \phi_j(y))_{\mathcal{W}_j} = \sum_{j=1}^{n} (B_j \xi, \eta)_\Lambda \Psi_j(x,y) \\ &= (K(x,y)\xi, \eta) = \tilde{K}((x,\xi),(y,\eta)).\end{aligned}$$

We next show that the denseness condition

$$\overline{\operatorname{span}}\{\Phi(x,\xi) : \; x \in X, \; \xi \in \Lambda\} = \mathcal{W} \tag{59}$$

is satisfied. To this end, suppose that we have $w_j \in \Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_n$ such that

$$((w_j : j \in \mathbb{N}_n), \Phi(x,\xi))_{\mathcal{W}} = \sum_{j=1}^{n} (w_j, \sqrt{B_j}\xi \otimes \phi_j(x))_{\Lambda\otimes\mathcal{W}_j} = 0 \text{ for all } x \in X \text{ and } \xi \in \Lambda. \tag{60}$$

Let $\{e_i : i \in \mathbb{I}\}$ and $\{f_k : k \in \mathbb{J}_j\}$ be an orthonormal basis for Λ and $\mathcal{W}_j$, respectively. Then $\{e_i \otimes f_k : i \in \mathbb{I}, k \in \mathbb{J}_j\}$ is an orthonormal basis for $\Lambda \otimes \mathcal{W}_j$. Note that although $\mathbb{I}$ or $\mathbb{J}_j$ might be uncountable, for each $\xi \in \Lambda$, $u \in \mathcal{W}_j$ and $w \in \Lambda \otimes \mathcal{W}_j$, the sets $\{i \in \mathbb{I} : (\xi, e_i)_\lambda \neq 0\}$, $\{k \in \mathbb{I}_j : (u, f_k)_{\mathcal{W}_j} \neq 0\}$ and $\{(i,j) \in \mathbb{I} \times \mathbb{J}_j : (w, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} \neq 0\}$ are all countable. By resorting to these orthonormal bases, we see that

$$(w_j, \sqrt{B_j}\xi \otimes \phi_j(x))_{\Lambda\otimes\mathcal{W}_j} = \sum_{k\in\mathbb{J}_j}\sum_{i\in\mathbb{I}} (w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} (e_i, \sqrt{B_j}\xi)_\Lambda (f_k, \phi_j(x))_{\mathcal{W}_j}.$$

One verifies by the Cauchy-Schwartz inequality that

$$\sum_{k\in\mathbb{J}_j}\sum_{i\in\mathbb{I}} (w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} (e_i, \sqrt{B_j}\xi)_\Lambda f_k$$

converges in $\mathcal{W}_j$. As a consequence, $(w_j, \sqrt{B_j}\xi \otimes \phi_j(\cdot))_{\Lambda\otimes\mathcal{W}_j} \in \mathcal{H}_{\Psi_j}$. This together with (60) implies by the assumption (58) that

$$(w_j, \sqrt{B_j}\xi \otimes \phi_j(x))_{\Lambda\otimes\mathcal{W}_j} = 0 \text{ for all } j \in \mathbb{N}_n,\ x \in X \text{ and } \xi \in \Lambda.$$

The above equation can be equivalently formulated as

$$\left(\sum_{k\in\mathbb{J}_j}\sum_{i\in\mathbb{I}} (w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} (e_i, \sqrt{B_j}\xi)_\Lambda f_k, \phi_j(x)\right)_{\mathcal{W}_j} = 0$$

By the denseness of $\phi_j(X)$ in $\mathcal{W}_j$,

$$\sum_{i\in\mathbb{I}} (w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} (e_i, \sqrt{B_j}\xi)_\Lambda = 0 \text{ for all } j \in \mathbb{N}_n,\ k \in \mathbb{J}_j \text{ and } \xi \in \Lambda.$$

We thus have for all $j \in \mathbb{N}_n$ and $k \in \mathbb{J}_j$ that $\sum_{i\in\mathbb{I}} (w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} e_i = 0$, which implies

$$(w_j, e_i \otimes f_k)_{\Lambda\otimes\mathcal{W}_j} = 0 \text{ for all } j \in \mathbb{N}_n,\ k \in \mathbb{J}_j,\ i \in \mathbb{I}.$$

Therefore, $w_j = 0$ for all $j \in \mathbb{N}_n$. Equation (59) hence holds true. Similar facts hold for $\tilde{G}$.

By Proposition 6, $\mathcal{H}_K \preceq \mathcal{H}_G$ is equivalent to $\mathcal{H}_{\tilde{K}} \preceq \mathcal{H}_{\tilde{G}}$, which by the above discussion and Lemma 2 holds true if and only if for all $w_j \in \Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_n$ there exist unique $\tilde{w}_j \in \Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_m$ such that

$$\sum_{j=1}^{n} (w_j, \sqrt{B_j}\xi \otimes \phi_j(x))_{\Lambda\otimes\mathcal{W}_j} = \sum_{j=1}^{m} (\tilde{w}_j, \sqrt{C_j}\xi \otimes \phi_j(x))_{\Lambda\otimes\mathcal{W}_j} \text{ for all } \xi \in \Lambda \text{ and } x \in X \tag{61}$$

and

$$\sum_{j=1}^{n} (w_j, w_j)_{\Lambda\otimes\mathcal{W}_j} = \sum_{j=1}^{m} (\tilde{w}_j, \tilde{w}_j)_{\Lambda\otimes\mathcal{W}_j}. \tag{62}$$

Let $w_j \in \Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_n$ be given. If $B_j = C_j$ for $j \in \mathbb{N}_n$ then we set $\tilde{w}_j := w_j$ for $j \in \mathbb{N}_n$, and $\tilde{w}_j = 0$ for $n+1 \leq j \leq m$. Clearly, such a choice satisfies Equations (61) and (62). Therefore, $\mathcal{H}_K \preceq \mathcal{H}_G$. Conversely, suppose that $\mathcal{H}_K \preceq \mathcal{H}_G$. Then for the special choice $w_j := \xi_j \otimes u_j$, $\xi_j \in \Lambda$, $u_j \in \mathcal{W}_j$, $j \in \mathbb{N}_n$, there exists $\tilde{w}_j \in \Lambda \otimes \mathcal{W}_j$, $j \in \mathbb{N}_m$ satisfying (61) and (62). As $\tilde{w}_j$ is unique by the denseness of the feature map for $\tilde{G}$, we must have $w_j = (\sqrt{C_j}^{\,-1}\sqrt{B_j}\xi_j) \otimes u_j$ for $j \in \mathbb{N}_n$, and $\tilde{w}_j = 0$ for $n+1 \leq j \leq m$. This together with (62) yields that

$$\sum_{j=1}^{n} (\xi_j, \xi_j)_\Lambda (u_j, u_j)_{\mathcal{W}_j} = \sum_{j=1}^{n} (\sqrt{B_j} C_j^{-1} \sqrt{B_j} \xi_j, \xi_j)_\Lambda (u_j, u_j)_{\mathcal{W}_j}.$$

By successively making $\xi_j \otimes u_j \neq 0$ and $\xi_k \otimes u_k = 0$ for $k \in \mathbb{N}_n \setminus \{j\}$, for $j \in \mathbb{N}_n$, we reach that

$$(\xi_j, \xi_j)_\Lambda = (\sqrt{B_j} C_j^{-1} \sqrt{B_j} \xi_j, \xi_j)_\Lambda \text{ for all } \xi_j \in \Lambda \text{ and } j \in \mathbb{N}_n.$$

As $\sqrt{B_j} C_j^{-1} \sqrt{B_j}$ is hermitian, it equals the identity operator on Λ. It follows that $B_j = C_j$ for all $j \in \mathbb{N}_n$. The proof is complete. ■

As a corollary of Theorem 21, we obtain an orthogonal decomposition of $\mathcal{H}_K$.

Corollary 22 *Let K be defined by (57), where B_j are invertible and Ψ_j, $j \in \mathbb{N}_n$ satisfy (58). Then*

$$\mathcal{H}_K = \bigoplus_{j=1}^{n} \mathcal{H}_{B_j \Psi_j}$$

and

$$\mathcal{H}_{\sum_{j=1}^{k} B_j \Psi_j} \preceq \mathcal{H}_{\sum_{j=1}^{k+1} B_j \Psi_j} \textit{ for } k \in \mathbb{N}_{n-1}.$$

A simplest case of (57) occurs when $\mathcal{H}_{\Psi_j}$ is of dimension 1 for $j \in \mathbb{N}_m$, which is covered below.

Corollary 23 *Let $B_j, C_k \in \mathcal{L}_+(\Lambda)$ be invertible for $j \in \mathbb{N}_n$ and $k \in \mathbb{N}_m$, and $\psi_k : X \to \mathbb{C}$, $k \in \mathbb{N}_m$, be linearly independent. Set*

$$K(x,y) := \sum_{j=1}^{n} B_j \psi_j(x) \overline{\psi_j(y)}, \;\; G(x,y) := \sum_{k=1}^{m} C_k \psi_k(x) \overline{\psi_k(y)}, \;\; x, y \in X.$$

Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if $B_j = C_j$ for all $j \in \mathbb{N}_n$.

More generally, we might consider K, G defined by two distinct classes of linearly independent functions from X to $\mathbb{C}$. The result below can be proved using arguments similar to those for Theorem 21.

Proposition 24 *Let $n \leq m \in \mathbb{N}_n$, $B_j, C_k \in \mathcal{L}_+(\Lambda)$ be invertible for $j \in \mathbb{N}_n$ and $k \in \mathbb{N}_m$, and $\{\psi_j : j \in \mathbb{N}_n\}$ and $\{\varphi_k : k \in \mathbb{N}_m\}$ be two classes of linearly independent functions from X to $\mathbb{C}$. Set*

$$K(x,y) := \sum_{j=1}^{n} B_j \psi_j(x) \overline{\psi_j(y)}, \;\; G(x,y) := \sum_{k=1}^{m} C_k \varphi_k(x) \overline{\varphi_k(y)}, \;\; x, y \in X.$$

Then $\mathcal{H}_K \preceq \mathcal{H}_G$ if and only if

(1) $\psi_j \in \operatorname{span}\{\varphi_k : k \in \mathbb{N}_m\}$ *for all* $j \in \mathbb{N}_n$*;*

(2) *the coefficients* $\lambda_{jl} \in \mathbb{C}$ *in the linear span*

$$\psi_j = \sum_{l=1}^{m} \lambda_{jl}\varphi_l, \ \ j \in \mathbb{N}_n$$

satisfy

$$\sum_{l=1}^{m} \lambda_{jl}\lambda_{kl}C_l^{-1} = \delta_{j,k}B_j^{-1} \textit{ for all } j,k \in \mathbb{N}_n.$$

We close this section with several concrete examples of finite Hilbert-Schmidt reproducing kernels of the form described in Corollary 23 and Proposition 24:

- polynomial kernels:

$$K(x,y) := \sum_{j=1}^{n} x^{\alpha_j} \cdot y^{\alpha_j} B_j, \ \ x,y \in \mathbb{R}^d$$

  where α_j are multi-indices and B_j are invertible operators in $\mathcal{L}_+(\Lambda)$, or

$$K(x,y) := \sum_{j=1}^{n} (x \cdot y)^{\beta_j} B_j, \ \ x,y \in \mathbb{R}^d$$

  where β_j are nonnegative integers.

- exponential kernels:

$$K(x,y) := \sum_{j=1}^{n} e^{i(x-y)\cdot t_j} B_j, \ \ x,y \in \mathbb{R}^d$$

  where $t_j \in \mathbb{R}^d$.

6. Existence

This section is devoted to the existence of nontrivial refinement of operator-valued reproducing kernels. Most of the results to be presented here are straightforward extensions of those in the scalar-valued case (Xu and Zhang, 2009).

Let X be the input space and Λ be a Hilbert space. The reproducing kernels under consideration are $\mathcal{L}(\Lambda)$-valued.

Proposition 25 *There does not exist a nontrivial refinement of an* $\mathcal{L}(\Lambda)$*-valued reproducing kernel* K *on* X *if and only if* $\mathcal{H}_K = \Lambda^X$*, the set of all the functions from* X *to* Λ*. If the cardinality of* X *is infinite then every* $\mathcal{L}(\Lambda)$*-valued reproducing kernel on* X *has a nontrivial refinement.*

Surprisingly, nontrivial results about the existence appear when X is of finite cardinality.

Proposition 26 *Let* X *consist of finitely many points* x_j*,* $j \in \mathbb{N}_n$ *for some* $n \in \mathbb{N}_n$*. A necessary condition for an* $\mathcal{L}(\Lambda)$*-valued reproducing kernel on* X *to have no nontrivial refinements is that*

$$\sum_{j=1}^{n}\sum_{k=1}^{n} (K(x_j,x_k)\xi_j,\xi_k)_\Lambda > 0 \textit{ for all } \xi_j \in \Lambda, j \in \mathbb{N}_n \textit{ with } \sum_{j=1}^{n} \|\xi_j\|_\Lambda > 0. \qquad (63)$$

A sufficient condition for K to have no nontrivial refinements is that

$$\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda \geq \lambda \sum_{j=1}^{n}\|\xi_j\|_\Lambda^2 \text{ for all } \xi_j \in \Lambda, j \in \mathbb{N}_n \tag{64}$$

for some constant $\lambda > 0$. Consequently, if Λ is finite-dimensional then K does not have a nontrivial refinement if and only if (63) holds true.

Proof Suppose that there exist $\xi_j \in \Lambda$, $j \in \mathbb{N}_n$, at least one of which is nonzero, such that

$$\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda = 0.$$

This implies that

$$\sum_{j=1}^{n} K(x_j,\cdot)\xi_j = 0.$$

We get by (3) that for all $f \in \mathcal{H}_K$

$$\sum_{j=1}^{n}(f(x_j),\xi_j)_\Lambda = \left(f, \sum_{j=1}^{n} K(x_j,\cdot)\xi_j\right)_{\mathcal{H}_K} = 0.$$

As a consequence, $\mathcal{H}_K$ does not contain the function $f : X \to \Lambda$ taking values $f(x_j) = \xi_j$ for $j \in \mathbb{N}_n$. By Proposition 25, there exist nontrivial refinements for K on X.

Suppose that (64) holds true for some positive constant λ. Assume that $\mathcal{H}_K$ is a proper subset of Λ^X. Then there exists some nonzero vector $(\xi_k : k \in \mathbb{N}_n) \in \Lambda^n$ orthogonal to $(f(x_k) : k \in \mathbb{N}_n)$ in Λ^n for all $f \in \mathcal{H}_K$. Letting $f = \sum_{j=1}^{n} K(x_j,\cdot)\xi_j$ yields that

$$\sum_{j=1}^{n}\sum_{k=1}^{n}(K(x_j,x_k)\xi_j,\xi_k)_\Lambda = \sum_{k=1}^{n}(f(x_k),\xi_k)_\Lambda = 0,$$

contradicting (64).

We complete the proof by pointing out that when Λ is finite-dimensional, (63) and (64) are equivalent. ■

It is worthwhile to note that when Λ is infinite-dimensional, condition (63) might not be sufficient for K to not have a nontrivial refinement. We give a concrete example to illustrate this.

Let X be a singleton $\{x\}$, $\Lambda := \ell^2(\mathbb{N})$ consisting of square-summable sequences indexed by $\mathbb{N}$, and $K(x_1,x_1)$ be the operator T on $\ell^2(\mathbb{N})$ defined by

$$Ta := \left(\frac{a_j}{j} : j \in \mathbb{N}\right), \quad a \in \ell^2(\mathbb{N}).$$

Apparently, $T \in \mathcal{L}_+(\ell^2(\mathbb{N}))$ and condition (63) is satisfied. Let $f \in \mathcal{H}_K$. Then there exist $a_n \in \ell^2(\mathbb{N})$, $n \in \mathbb{N}$ such that $K(x,\cdot)a_n$ converges to f in $\mathcal{H}_K$. Being a Cauchy sequence in $\mathcal{H}_K$, $\{K(x,\cdot)a_n : n \in \mathbb{N}\}$ satisfies

$$\lim_{n,m\to\infty}\|K(x,\cdot)a_n - K(x,\cdot)a_m\|_{\mathcal{H}_K}^2 = 0.$$

By (3),

$$\begin{aligned}\|K(x,\cdot)a_n-K(x,\cdot)a_m\|^2_{\mathcal{H}_K} &= (K(x,\cdot)(a_n-a_m),K(x,\cdot)(a_n-a_m))_{\mathcal{H}_K}\\ &= (K(x,x)(a_n-a_m),a_n-a_m)_{\ell^2(\mathbb{N})}=(T(a_n-a_m),a_n-a_m)_{\ell^2(\mathbb{N})}\\ &= \|\sqrt{T}a_n-\sqrt{T}a_m\|^2_{\ell^2(\mathbb{N})}.\end{aligned}$$

Combining the above two equations yields $\sqrt{T}a_n$ converges to some $b\in\ell^2(\mathbb{N}_n)$. We now have for each $c\in\ell^2(\mathbb{N})$ that

$$\begin{aligned}(f(x),c)_{\ell^2(\mathbb{N})} &= (f,K(x,\cdot)c)_{\mathcal{H}_K}=\lim_{n\to\infty}(K(x,\cdot)a_n,K(x,\cdot)c)_{\mathcal{H}_K}\\ &= \lim_{n\to\infty}(K(x,x)a_n,c)_{\ell^2(\mathbb{N})}=\lim_{n\to\infty}(Ta_n,c)_{\ell^2(\mathbb{N})}\\ &= \lim_{n\to\infty}(\sqrt{T}a_n,\sqrt{T}c)_{\ell^2(\mathbb{N})}=(b,\sqrt{T}c)_{\ell^2(\mathbb{N})}\\ &= (\sqrt{T}b,c)_{\ell^2(\mathbb{N})},\end{aligned}$$

which implies that $f(x)=\sqrt{T}b$. Since this is true for an arbitrary function $f\in\mathcal{H}_K$, the function $g:X\to\Lambda$ defined by

$$g(x):=\left(\frac{1}{j}:j\in\mathbb{N}\right)$$

is not in $\mathcal{H}_K$. Thus, K has a nontrivial refinement on X.

In the process of refining an operator-valued reproducing kernel, it is usually desirable to preserve favorable properties of the original kernel. We shall show that this is feasible as far as continuity and universality of operator-valued reproducing kernels are concerned. Let X be a metric space and K an $\mathcal{L}(\Lambda)$-valued reproducing kernel that is continuous from $X\times X$ to $\mathcal{L}(\Lambda)$ when the latter is equipped with the operator norm. Then one sees that $\mathcal{H}_K$ consists of continuous functions from X to Λ. For each compact subset $\mathcal{Z}\subseteq X$, denote by $\mathcal{C}(\mathcal{Z},\Lambda)$ the Banach space of all the continuous functions from $\mathcal{Z}$ to Λ with the norm

$$\|f\|_{\mathcal{C}(\mathcal{Z},\Lambda)}:=\max_{x\in\mathcal{Z}}\|f(x)\|_\Lambda,\ \ f\in\mathcal{C}(\mathcal{Z},\Lambda).$$

Following Micchelli et al. (2006) and Caponnetto et al. (2008), we call K a *universal kernel* on X if for all compact sets $\mathcal{Z}\subseteq X$ and all continuous functions $f:X\to\Lambda$ there exist

$$f_n\in\operatorname{span}\{K(x,\cdot)\xi:x\in\mathcal{Z},\ \xi\in\Lambda\},\ \ n\in\mathbb{N},$$

such that

$$\lim_{n\to\infty}\|f_n-f\|_{\mathcal{C}(\mathcal{Z},\Lambda)}=0.$$

In other words, K is universal if for all compact subsets $\mathcal{Z}\subseteq X$, the closure of $\operatorname{span}\{K(x,\cdot)\xi:x\in\mathcal{Z},\ \xi\in\Lambda\}$ in $\mathcal{C}(\mathcal{Z},\Lambda)$ equals the whose space $\mathcal{C}(\mathcal{Z},\Lambda)$.

For the preservation of continuity, we have the following affirmative result, whose proof is similar to the scalar-valued case (Xu and Zhang, 2009).

Proposition 27 *Let X be a metric space with infinite cardinality. Then every continuous $\mathcal{L}(\Lambda)$-valued reproducing kernel on X has a nontrivial continuous refinement.*

The following lemma about universality has been proved in Caponnetto et al. (2008), and in Micchelli et al. (2006) in the scalar-valued case. We provide a simplified proof here.

Lemma 28 *Let K be a continuous $\mathcal{L}(\Lambda)$-valued reproducing kernel on X with the feature map representation (5), where $\Phi : X \to \mathcal{L}(\Lambda, \mathcal{W})$ is continuous. Then for each compact subset $\mathcal{Z} \subseteq X$,*

$$\overline{\mathrm{span}}\{K(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\} = \overline{\{\Phi(\cdot)^* u : u \in \mathcal{W}\}},$$

where the closures are relative to the norm in $\mathcal{C}(\mathcal{Z}, \Lambda)$.

Proof All the closures to appear in the proof are relative to the norm in $\mathcal{C}(\mathcal{Z}, \Lambda)$. Let $K_{\mathcal{Z}}$ be the restriction of K on $\mathcal{Z}$. Then the restriction of Φ on $\mathcal{Z}$ remains a feature map for $K_{\mathcal{Z}}$. By Lemma 2,

$$\mathcal{H}_{K_{\mathcal{Z}}} = \{\Phi(\cdot)^* u : u \in \mathcal{W}\}. \tag{65}$$

It hence suffices to show that

$$\overline{\mathrm{span}}\{K(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\} = \overline{\mathrm{span}}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\} = \overline{\mathcal{H}_{K_{\mathcal{Z}}}}.$$

As $\mathrm{span}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\} \subseteq \mathcal{H}_{K_{\mathcal{Z}}}$,

$$\overline{\mathrm{span}}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\} \subseteq \overline{\mathcal{H}_{K_{\mathcal{Z}}}}. \tag{66}$$

On the other hand, for each $f \in \mathcal{H}_{K_{\mathcal{Z}}}$ there exist $f_n \in \mathrm{span}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\}$, $n \in \mathbb{N}$ that converges to f in the norm of $\mathcal{H}_{K_{\mathcal{Z}}}$. It follows that f_n converges to f in the norm of $\mathcal{C}(\mathcal{Z}, \Lambda)$. Therefore, $f \in \overline{\mathrm{span}}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\}$, implying that

$$\overline{\mathcal{H}_{K_{\mathcal{Z}}}} \subseteq \overline{\mathrm{span}}\{K_{\mathcal{Z}}(x,\cdot)\xi : x \in \mathcal{Z},\ \xi \in \Lambda\}. \tag{67}$$

Combining Equations (65), (66), and (67) proves the result. ■

The following positive result about universality can be proved by Lemma 28 and arguments similar to those used in Proposition 14 of Xu and Zhang (2009).

Proposition 29 *Let X be a metric space and K a continuous $\mathcal{L}(\Lambda)$-valued reproducing kernel on X. Then every continuous refinement of K on X remains universal.*

7. Numerical Experiments

We present in this final section three numerical experiments on the application of refinement of operator-valued reproducing kernels to multi-task learning. Suppose that f_0 is a function from the input space X to the output space Λ that we desire to learn from its finite sample data $\{(x_j, \xi_j) : j \in \mathbb{N}_m\} \subseteq X \times \Lambda$. Here m is the number of sampling points and

$$\xi_j = f_0(x_j) + \delta_j, \quad j \in \mathbb{N}_m$$

where $\delta_j \in \Lambda$ is the noise dominated by some unknown probability measure. To deal with the noise and have an acceptable generalization error, we use the following regularization network

$$\min_{f \in \mathcal{H}_K} \frac{1}{m} \sum_{j=1}^{m} \|f(x_j) - \xi_j\|_{\Lambda}^2 + \sigma \|f\|_{\mathcal{H}_K}^2, \tag{68}$$

where K is a chosen Λ-valued reproducing kernel on X. Our experiments will be designed so that underfitting and overfitting both have the chance to occur. To echo with the motivations in Section 2, when underfitting happens in the first experiment, we shall find a refinement G of K aiming at improving the performance of the minimizer of (68) in prediction. On the other hand, when overfitting appears in the second experiment, we shall then find a Λ-valued reproducing kernel L on X such that $\mathcal{H}_L \preceq \mathcal{H}_K$ with the same purpose.

Before moving on to the experiments, we make a remark on how (68) can be solved. The issue has been understood in the work by Micchelli and Pontil (2005). We say that K is *strictly positive-definite* if for all finite $y_j \in X$, $j \in \mathbb{N}_p$, and for all $\eta_j \in \Lambda$, $j \in \mathbb{N}_p$ all of which are not zero

$$\sum_{j=1}^{p}\sum_{k=1}^{p}(K(y_j,y_k)\eta_j,\eta_k)_\Lambda > 0.$$

If K is strictly positive-definite then the minimizer f_K of (68) has the form

$$f_K = \sum_{j=1}^{m} K(x_j,\cdot)\eta_j \tag{69}$$

where η_j's satisfy

$$\sum_{k=1}^{m} K(x_k,x_j)\eta_k + m\sigma\eta_j = \xi_j, \;\; j \in \mathbb{N}_m. \tag{70}$$

7.1 Experiment 1: Underfitting

The vector-valued function to be learned from finite examples is from the input space $X = [-1,1]$ to output space $\Lambda = \mathbb{R}^n$, where $n \in \mathbb{N}$. Specifically, it has the form

$$f_0(x) := \left[a_k|x-b_k| + c_k e^{-d_k x} : k \in \mathbb{N}_n\right], \;\; x \in [-1,1], \tag{71}$$

where a,b,c,d are constant vectors to be randomly generated. The $\mathcal{L}_+(\mathbb{R}^n)$-valued reproducing kernel that we shall use in the regularization network (68) is a Gaussian kernel

$$K(x,y) := S\exp\left(-\frac{(x-y)^2}{2}\right), \;\; x,y \in \mathbb{R},$$

where $S \in \mathcal{L}_+(\mathbb{R}^n)$ is strictly positive-definite. It can be identified by Lemma 2 that functions in $\mathcal{H}_K$ are of the form $\sqrt{S}v$, where v is an $\mathbb{R}^n$-valued function whose components come from the RKHS $\mathcal{H}_\mathcal{G}$ of the scalar-valued Gaussian kernel

$$\mathcal{G}(x,y) := \exp\left(-\frac{(x-y)^2}{2}\right), \;\; x,y \in \mathbb{R}. \tag{72}$$

Thus, each component of $\sqrt{S}v$ is from $\mathcal{H}_\mathcal{G}$. The function f_0 to be approximated is defined by (71). As $|x-b_k|$ is not even continuously differentiable, functions from the RKHS of the Gaussian kernel (72) with a fixed variance may not well approximate f_0. Underfitting is hence expected. If this is indeed observed then a remedy is to use the refinement of K given by

$$G(x,y) := S\exp\left(-\frac{(x-y)^2}{2}\right) + T(1+xy)^3, \;\; x,y \in \mathbb{R},$$

where $T \in \mathcal{L}_+(\mathbb{R}^n)$ is also strictly positive-definite. The RKHS of the scalar-valued polynomial kernel $(1+xy)^3$ clearly does not have a nontrivial intersection with the RKHS of the scalar-valued Gaussian kernel. Thus, by Corollary 22, $\mathcal{H}_K \preceq \mathcal{H}_G$, namely, G is a nontrivial refinement of K. Furthermore, as low order polynomials are added, the ability for functions in $\mathcal{H}_G$ to approximate the function $|x-b_k|$ is expected to be superior to those in $\mathcal{H}_K$. We perform extensive numerical simulations to confirm these conjectures.

The dimension n will be chosen from $\{2,4,8,16\}$. The number m of sampling points will be set to be 30. The sampling points x_j, $j \in \mathbb{N}_m$ will be randomly sampled from $[-1,1]$ by the uniform distribution and the outputs ξ_j are generated by

$$\xi_j = f_0(x_j) + \delta_j, \quad j \in \mathbb{N}_m, \tag{73}$$

where δ_j are vectors whose components will be randomly generated by the uniform distribution on $[-\delta,\delta]$ with δ being the noise level selected from $\{0.1,0.3,0.5\}$. For each dimension $n \in \{2,4,8,16\}$ and noise level $\delta \in \{0.1,0.3,0.5\}$, we run 50 simulations. In each of the simulations, we do the followings:

1. the components of the coefficient vectors a,b,c,d in the function f_0 given by (71) are randomly generated by the uniform distribution on $[1,3]$, $[-1,1]$, $[-2,2]$, and $[0,3]$, respectively;

2. the sampling points are randomly sampled from $[-1,1]$ by the uniform distribution and the outputs ξ_j are then generated by (73);

3. the matrices S and T are given by $S = A'A$ and $T = B'B$ where A,B are $n \times n$ real matrices whose components are randomly sampled from $[1,3]$ by the uniform distribution;

4. we then solve the minimizer f_K of (68) by (69) and (70);

5. for the refinement kernel G, we also obtain f_G as the minimizer of

$$\min_{f \in \mathcal{H}_G} \frac{1}{m} \sum_{j=1}^{m} \|f(x_j) - \xi_j\|_\Lambda^2 + \sigma \|f\|_{\mathcal{H}_G}^2, \tag{74}$$

6. the regularization parameters in (68) and (74) are optimally chosen so that the relative square approximation errors

$$\mathcal{E}_K := \frac{\int_{-1}^{1} \|f_K(t) - f_0(t)\|^2 dt}{\int_{-1}^{1} \|f_0(t)\|^2 dt}, \quad \mathcal{E}_G := \frac{\int_{-1}^{1} \|f_G(t) - f_0(t)\|^2 dt}{\int_{-1}^{1} \|f_0(t)\|^2 dt}. \tag{75}$$

are minimized, respectively.

We call $(\mathcal{E}_K, \mathcal{E}_G)$ obtained in each simulation an instance of approximation errors. Hence, we have 50 instances for each pair of (n,δ). They are said to form a group. There are 12 groups of instances of approximation errors. For each (n,δ), we shall calculate the mean and standard deviation of the difference $\mathcal{E}_K - \mathcal{E}_G$ in the corresponding group as a measurement of the difference in the performance of learning schemes (68) and (74). Before that, outliers of instances should be excluded. Although we do not know the distributions of $\mathcal{E}_K$ and $\mathcal{E}_G$, we shall use the three-sigma rule in statistics. In other words, we regard an instance $(\mathcal{E}_K, \mathcal{E}_G)$ as an outlier if the deviation of $\mathcal{E}_K$

	n=2	n=4	n=8	n=16
$\delta = 0.1$	(0.1024,0.0084) (0.0091,0.0081) (0.4128,0.0006) (0.6783,0.0025)	(0.0215,0.0182) (0.4095,0.0034)	(0.0230,0.0070) (0.0513,0.0091) (0.1554,0.0011) (0.1464,0.0026)	(0.0712,0.0015) (0.0364,0.0124)
$\delta = 0.3$	(0.0286,0.0228) (0.4811,0.0020)	(0.0663,0.0321) (0.1892,0.0041) (0.1674,0.0095)	(0.0407,0.0194) (0.1809,0.0023)	(0.1592,0.0018) (0.0309,0.0127) (0.0229,0.0099)
$\delta = 0.5$	(0.2053,0.0020) (0.1267,0.0034) (0.0669,0.0465)	(0.0377,0.0376) (0.3547,0.0033)	(0.2445,0.0028) (0.2762,0.0020) (0.0119,0.0264)	(0.1612,0.0043) (0.0541,0.0081)

Table 1: Outliers of instances of approximation errors $(\mathcal{E}_K, \mathcal{E}_G)$. An instance $(\mathcal{E}_K, \mathcal{E}_L)$ is considered to be an outlier if the deviation of one of its components to the respective mean in the group is more than three times the standard deviation of the group. Outliers are listed in an independent table because they should be excluded from the calculation of the mean and standard deviation of the approximation errors. Another reason is that adding them will make the plot of the approximation errors highly disproportional.

or $\mathcal{E}_G$ to their respective mean in the group exceeds three times their respective standard deviation. There are 32 outliers among the entire 600 instances, which are listed below in Table 1.

We make a few observations from Table 1. Firstly, $\mathcal{E}_G$ is smaller than $\mathcal{E}_K$ except for only one instance. For a large portion of the outliers, the approximation error $\mathcal{E}_K$ is considerably large (larger than 10%), a sign of underfitting of the kernel K. Those instances are of the greatest interest to us as we desire to see if the refinement kernel G can make a remedy when underfitting does happen. We see from Table 1 that for all of those outliers, the refinement kernel G always brings down the relative approximation error to be less than 1%. The improvement brought by G for other instances is also significant. The observations indicate that (74) performs significantly better in learning the function (71) from finite examples than (68). For further comparison, we compute the mean and standard deviation of the difference $\mathcal{E}_K - \mathcal{E}_G$ of the approximation errors after excluding the above outliers. The results are tabulated in Table 2 below. Note that a positive value of the mean implies that (74) performs better than (68). It is worthwhile to point out that among all the rest 568 instances excluding the outliers, there are only 33 where $\mathcal{E}_G$ is larger than $\mathcal{E}_K$. The largest value of $\mathcal{E}_G - \mathcal{E}_K$ is 0.0020. Therefore, we conclude that for all the (n, δ), (74) is superior to (68), and the larger the standard deviation in Table 2 is, the greater improvement the refinement kernel G brings.

We shall also plot the 12 groups of approximation errors $\mathcal{E}_K, \mathcal{E}_G$ for a visual comparison. To this end, we take out the instances for which $\mathcal{E}_K$ is too large to have an appropriate range in the vertical axes in the figures. Therefore, Figures 1 and 2 are not full embodiment of the improvement of (74) over (68). Nevertheless, one sees that the improvement brought by the refinement kernel G in these relatively well-behaved instances is still dramatic.

	n=2	n=4	n=8	n=16
$\delta=0.1$	0.0098 (0.0182)	0.0139 (0.0335)	0.0160 (0.0241)	0.0108 (0.0135)
$\delta=0.3$	0.0076 (0.0144)	0.0141 (0.0245)	0.0143 (0.0208)	0.0188 (0.0259)
$\delta=0.5$	0.0054 (0.0121)	0.0127 (0.0307)	0.0103 (0.0186)	0.0091 (0.0102)

Table 2: The mean and standard deviation (in parentheses) of $\mathcal{E}_K-\mathcal{E}_G$. The outliers of instances listed in Table 1 are not counted toward these calculations. If they were added, the improvement brought by the refinement kernel G would have been more dramatic.

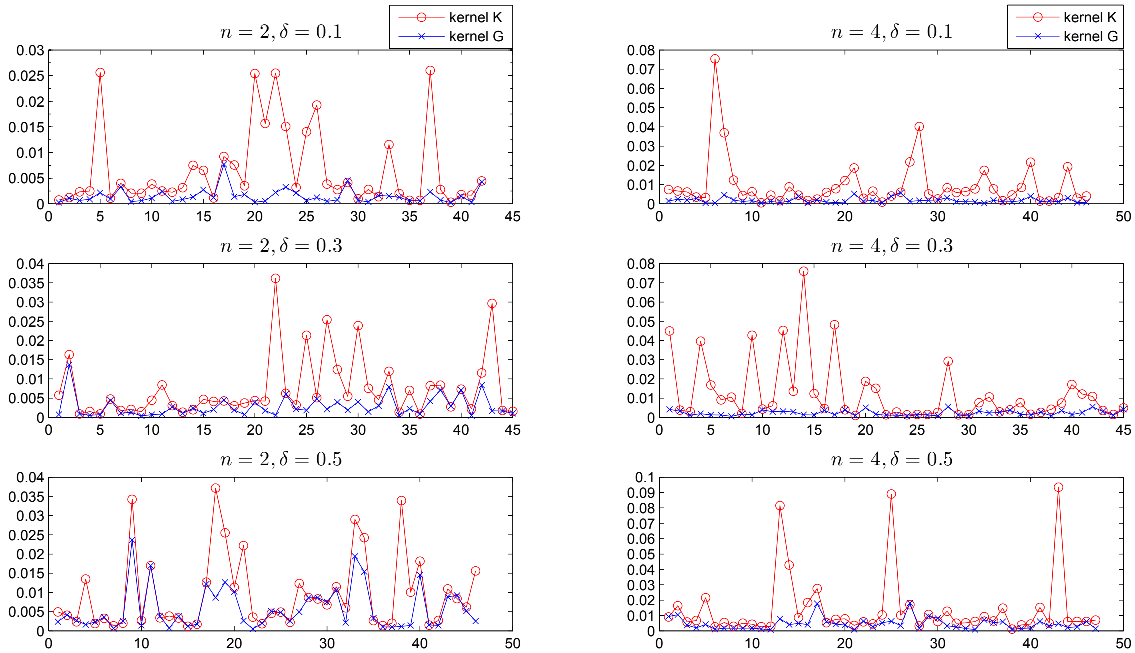


Figure 1: Relative approximation errors $\mathcal{E}_K,\mathcal{E}_G$ for $n=2,4$ and $\delta=0.1,0.3,0.5$. The outliers listed in Table 1 are not plotted here as they would make the figure highly disproportional.

7.2 Experiment 2: Overfitting

The target function we consider in the second experiment is

$$f_0(x)=\left[\frac{a_k}{1+25(x-b_k)^2}+c_ke^{-d_kx}:k\in\mathbb{N}_n\right],\quad x\in[-1,1],\tag{76}$$

where the components of the vectors $a,b,c,d\in\mathbb{R}^n$ will be randomly sampled by the uniform distribution from $[1,4]$, $[0,\frac{1}{2}]$, $[-2,2]$, and $[0,2]$ respectively in the numerical simulations. The dimension

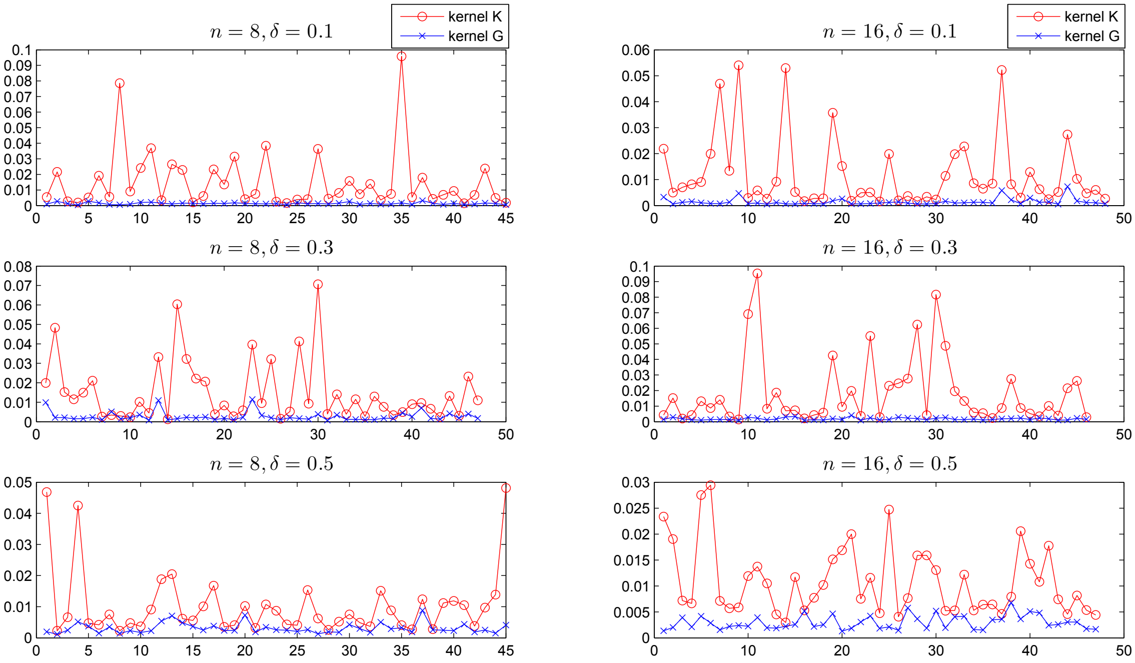


Figure 2: Relative approximation errors $\mathcal{E}_K, \mathcal{E}_G$ for $n = 8, 16$ and $\delta = 0.1, 0.3, 0.5$. The outliers listed in Table 1 are not plotted in the figure here.

n will be chosen from $\{2,4,8,16\}$. We fix $m := 20$ and shall sample the inputs x_j, $j \in \mathbb{N}_m$ randomly by the uniform distribution from $[-1,1]$. Similarly, the outputs $\xi_j \in \mathbb{R}^n$, $j \in \mathbb{N}_m$ will be generated by (73) where the noise level is to be selected from $\{0.1, 0.3, 0.5\}$.

In the first step, we substitute the sample data $\{(x_j, \xi_j) : j \in \mathbb{N}_m\}$ into the regularization network (68) with the following kernel

$$K(x,y) := S \exp\left(-\frac{(x-y)^2}{2}\right) + T(1+xy)^{18}, \quad x,y \in [-1,1], \tag{77}$$

where $S = A'A$ and $T = B'B$ with A, B being $n \times n$ real-matrices whose components will be randomly sampled by the uniform distribution from $[1,2]$. The target function (76) contains translations of the Runge function

$$\frac{1}{1+25x^2}, \quad x \in [-1,1].$$

It is well-known that approximating the Runge function by high order polynomial interpolations leads to overfitting. One sees by (70) that the regulation network (68) might be regarded as a regularized interpolation. Note also that the order of the polynomial kernel in (77) is 18, which is close to the number $m = 20$ of sampling points. Overfitting is hence expected. When this occurs, we propose to reduce the order of the polynomial kernel by considering

$$L(x,y) := S \exp\left(-\frac{(x-y)^2}{2}\right) + T \sum_{k=0}^{10} \binom{18}{k} (xy)^k, \quad x,y \in [-1,1].$$

	$\delta = 0.1$	$\delta = 0.3$	$\delta = 0.5$
n=2	(0.9000, 0.7843)	(2.9906,1.3509)	(1.8065, 0.8044),(1.1332, 0.3213) (19.6416, 7.6578)
n=4	(8.2450,5.8717) (1.6654,2.0466) (18.9615,12.0513) (0.9536, 1.0998)	(1.1760,0.1354) (0.4591, 0.7845)	(4.6316,7.0497),(2.0850,1.3204) (2.4657,1.1386) (5.7967,0.6122) (5.1196,2.6692)
n=8	(0.9102,1.3862) (1.2233,0.9489) (0.6711,0.2249)	(1.3517, 1.8339) (0.8450,0.2605) (0.3571, 0.7221) (2.2403, 2.0108) (5.6153,5.0954) (2.0763,1.3718) (2.2567,1.4024)	(0.6369, 0.3698),(0.6945,0.2878) (2.2371, 2.4008) (1.0738,0.4172) (1.0561,0.3067) (0.6791,1.0980) (3.6689,3.9566) (1.1238,0.2467)
n=16	(4.4905, 5.8886) (7.9187, 4.3445) (2.1619, 0.5061) (17.5145, 13.7894)	(26.0758,7.6125) (1.2255, 0.3181) (0.5140, 0.1817) (2.4289, 1.9022)	(73.0854,42.6904),(1.6070, 1.4224) (3.2674, 2.2622),(2.1632, 1.7059) (2.8067, 0.5791),(9.0120, 3.5443) (0.6064, 0.3365),(4.0484 , 0.4220) (1.0064, 0.8287)

Table 3: Outliers of instances of relative approximation errors $(\mathcal{E}_K, \mathcal{E}_L)$.

By Corollary 22, $\mathcal{H}_L \preceq \mathcal{H}_K$, namely, K is a refinement of L. We shall demonstrate by numerical simulations that

$$\min_{f\in\mathcal{H}_L} \frac{1}{m}\sum_{j=1}^{m} \|f(x_j)-\xi_j\|^2 + \sigma\|f\|^2_{\mathcal{H}_L} \tag{78}$$

outperforms (68) with the kernel (77). To this end, we shall conduct numerical experiments similar to those in the last subsection. Let f_K and f_L be the minimizer of (68) and (78), respectively. We shall measure the performance by the relative square approximation errors $\mathcal{E}_K$ and $\mathcal{E}_L$, which are defined in the same way as (75). For each pair of (n,δ), where $n \in \{2,4,8,16\}$ and $\delta \in \{0.1,0.3,0.5\}$, we run 20 numerical simulations where the regularization parameters σ are to be chosen so that $\mathcal{E}_K$ and $\mathcal{E}_L$ are minimized, respectively. As in the first experiment, we shall calculate the mean and standard deviation of $\mathcal{E}_K$ and $\mathcal{E}_L$ in each group after taking out some outliers. We shall also plot the relative errors for comparison. The results are shown below in the form of tables and figures.

We have more outliers compared to the first experiment. Using fewer sampling points and approximating the Runge function by polynomials both contributes to this. We observe that for the majority of these outliers, $\mathcal{E}_L$ is significantly smaller than $\mathcal{E}_K$, showing improvement of learning scheme (78) over (68). For further comparison, we shall compute the mean and variances of $\mathcal{E}_K - \mathcal{E}_L$ and plot the relative approximation errors $\mathcal{E}_K$ and $\mathcal{E}_L$ for the rest of instances.

A positive value of the mean in Table 4 implies that (78) performs better than (68). It is observed that kernel L brings improvement for all the choices of $n \in \{2,4,8,16\}$ and $\delta \in \{0.1,0.3,0.5\}$. We also remark that among all the 188 instances counted in Table 4, there are only 32 for which $\mathcal{E}_L > \mathcal{E}_K$. The mean and standard deviation of $\mathcal{E}_L - \mathcal{E}_K$ for these 32 instances are 0.0264 and

	n=2	n=4	n=8	n=16
$\delta = 0.1$	0.0289 (0.0846)	0.0511 (0.0587)	0.0173 (0.0779)	0.0157 (0.0146)
$\delta = 0.3$	0.0404 (0.0922)	0.0661 (0.0705)	0.0671 (0.0929)	0.0657 (0.0918)
$\delta = 0.5$	0.0629 (0.1098)	0.0130 (0.0233)	0.0484 (0.0758)	0.0625 (0.0821)

Table 4: The mean and standard deviation (in parentheses) of $\mathcal{E}_K - \mathcal{E}_L$. The outliers of instances listed in Table 3 are not counted toward these calculations. If they were added, the improvement brought by the refinement kernel G would have been more dramatic.

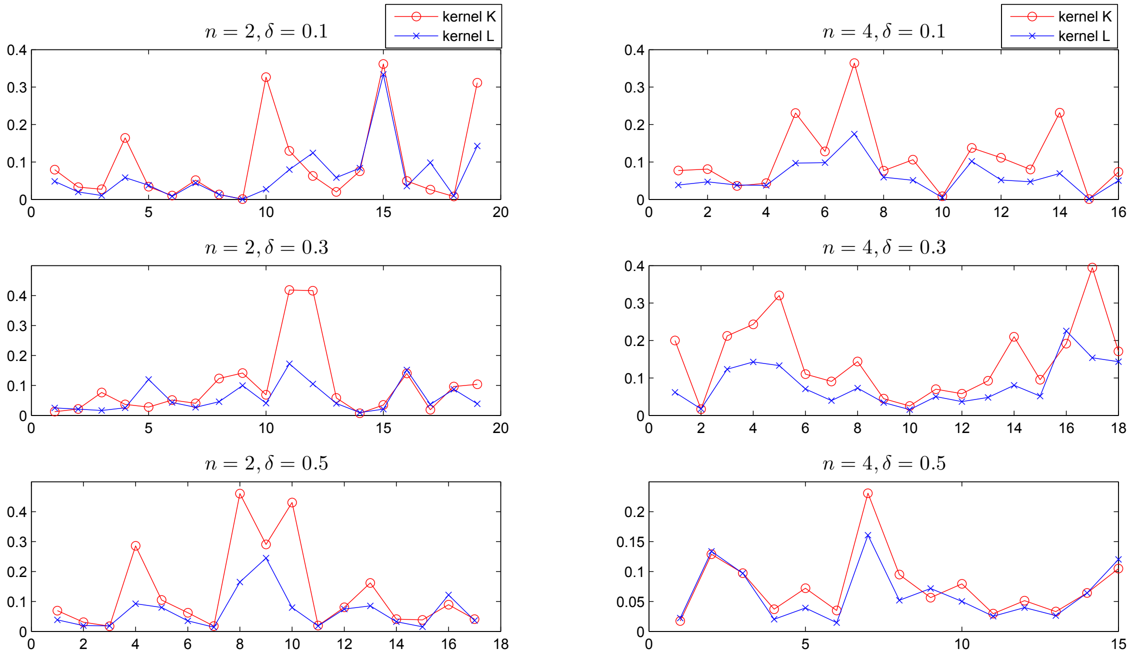


Figure 3: Relative approximation errors $\mathcal{E}_K, \mathcal{E}_L$ for $n = 2, 4$ and $\delta = 0.1, 0.3, 0.5$. The outliers listed in Table 3 are not plotted here as they will make the figure highly disproportional.

0.0306. We conclude that compared to (68), (78) improves the performance considerably in learning the function (76).

7.3 Experiment 3: Impact of Irrelevant Signals

Suggested by one of the anonymous reviewers, we shall examine the impact of irrelevant signals in the refinement kernel method. More specifically, we plan to apply the refinement kernel method

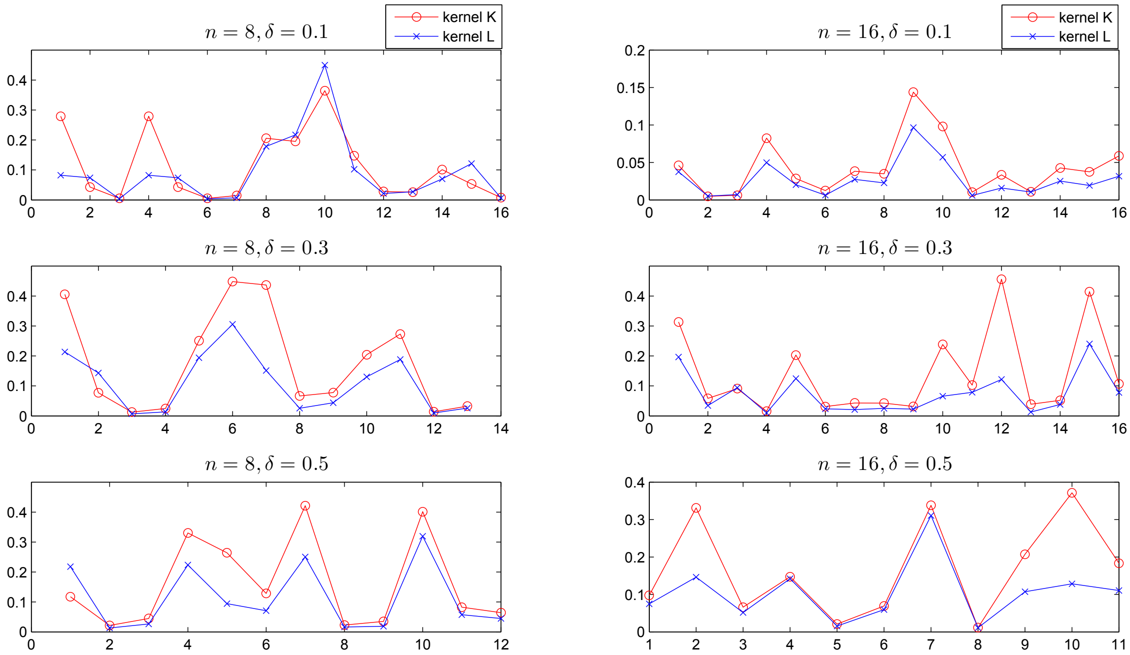


Figure 4: Relative approximation errors $\mathcal{E}_K, \mathcal{E}_L$ for $n = 8, 16$ and $\delta = 0.1, 0.3, 0.5$. The outliers listed in Table 3 are not plotted here.

	$\delta = 0.1$	$\delta = 0.3$	$\delta = 0.5$
$n = 4$	$(0.0164, 0.0083)$ $(0.2930, 0.0044)$ $(0.0074, 0.0076)$	$(0.1760,0.0074)$ $(0.0415, 0.0189)$ $(0.1464, 0.0254)$	$(0.1550, 0.0049)$ $(0.0837, 0.0302)$

Table 5: Outliers of instances of relative approximation errors $(\mathcal{E}_K, \mathcal{E}_G)$.

to the learning a vector-valued function whose components might be irrelevant. To avoid repetition and save space, we shall consider the underfitting case only and limit ourself to dimension $n = 4$. The instance investigated here is the function f_0 of the form (71), where we shall set $a_3 = a_4 = c_1 = c_2 = 0$. Thus, the first two components are irrelevant with the last two components of f_0. We then proceed with the same simulation procedures as those in experiment 1.

We obtain 3 groups of relative approximation error $(\mathcal{E}_K, \mathcal{E}_G)$ corresponding to the noise level $\delta = 0.1, 0.3, 0.5$. As in experiment 1, we first list all the outliers by the three-sigma rule in Table 5 below.

We observe from Table 5 that under the impact of irrelevant signals, among the above outliers, $\mathcal{E}_G$ is smaller than $\mathcal{E}_K$ except for only one instance $(0.0074, 0.0076)$. In 4 instances of the outliers, $\mathcal{E}_K$ is larger than 14%, while the refinement kernel G always brings down the relative approximation error to be less than 3%. In the overall 150 instances of relative approximation errors computed,

	$\delta = 0.1$	$\delta = 0.3$	$\delta = 0.5$
$n = 4$	0.0077 (0.0131)	0.0114 (0.0257)	0.0117 (0.0205)

Table 6: The mean and standard deviation (in parentheses) of $\mathcal{E}_K - \mathcal{E}_G$. The outliers of instances listed in Table 5 are not counted toward these calculations. If they were added, the improvement brought by the refinement kernel G would have been more dramatic.

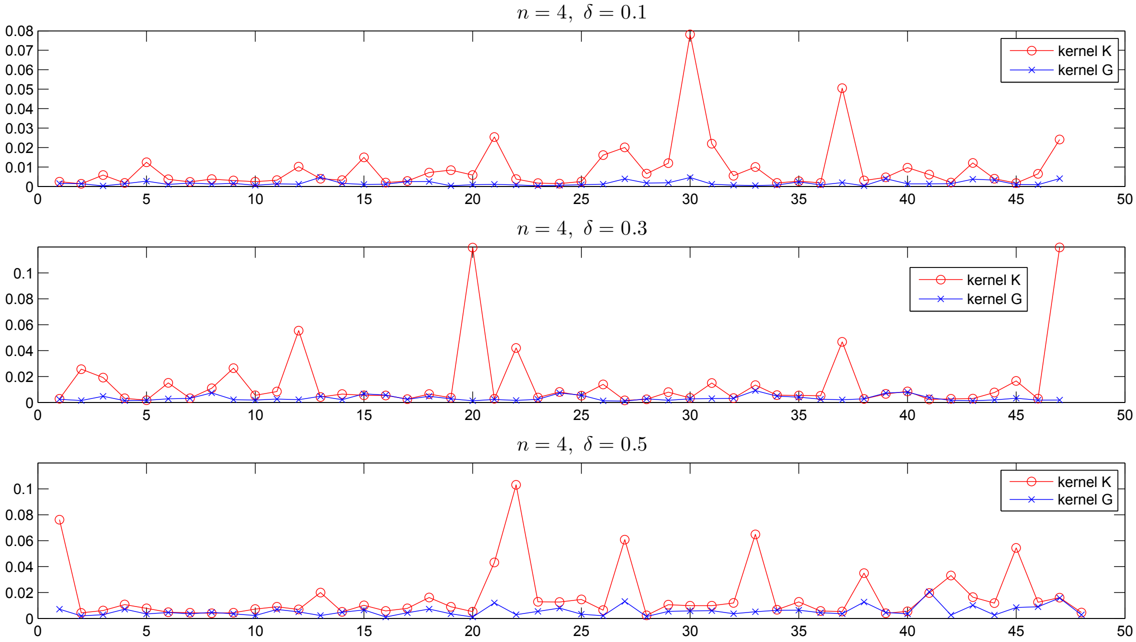


Figure 5: Relative approximation errors $\mathcal{E}_K, \mathcal{E}_G$ for $n = 4$ and $\delta = 0.1, 0.3, 0.5$. The outliers listed in Table 5 are not plotted here as they would make the figure highly disproportional.

there are only 13 instances where $\mathcal{E}_K$ is smaller than $\mathcal{E}_G$. For all these instances, $\mathcal{E}_G$ are of the same magnitude level with $\mathcal{E}_K$, showing competitive performance. For further comparison, we compute the mean and standard deviation of the difference $\mathcal{E}_K - \mathcal{E}_G$ after the above outliers are excluded. The results are shown in Table 6 below.

Finally, we plot the 3 groups of relative approximation errors $\mathcal{E}_K, \mathcal{E}_G$ for a visual comparison after the outliers in Table 5 are excluded.

We conclude from Tables 5, 6 and Figure 5 that for the learning problem considered in this subsection, the refinement kernel method works well under the impact of irrelevant signals.

8. Conclusion and Discussion

The refinement relationship between two operator-valued reproducing kernels provides a promising way of updating kernels for multi-task machine learning when overfitting or underfitting occurs. We establish several general characterizations of the refinement relationship. Particular attention has been paid to the case when the kernels under investigation have a vector-valued integral representation, the most general form of operator-valued reproducing kernels. By the characterizations, we present concrete examples of refining the translation invariant operator-valued reproducing kernels, Hessian of the scalar-valued Gaussian kernel, and finite Hilbert-Schmidt operator-valued reproducing kernels. Three numerical experiments confirm the potential usefulness of the proposed refinement method in updating kernels for multi-task learning. We plan to investigate the effect of the method by real application data in another occasion.

We discuss three issues that might deserve future research attention. The first one concerns about the computational saving brought by the refinement kernel method. Suppose a minimizer in an RKHS resulting from a particular learning algorithm is already computed but turns out to be unsatisfactory due to underfitting. When the kernel corresponding to the RKHS is refined, instead of running the algorithm from the scratch in the updated RKHS, we are wondering if the original minimizer can be made use of in order to reduce computational costs. In the scalar-valued case, it has been shown that this can be done for the classical regularization networks (Xu and Zhang, 2009). For the vector-valued case, one would need to carefully handle the complexity brought by the high dimension of the output space in order to establish a similar analysis. The second question is whether a multi-resolution analysis for vector-valued RKHS can be achieved by using the refinement kernel method. Our initial thinking and impression is that the approach in Xu and Zhang (2007) of using a bijective self-mapping of the input space can be carried over without much difficulty. Finally, we look at the requirement in the definition of refinement that the norm on the RKHS of the refinement kernel should coincide with that in the RKHS of the original kernel. As seen by the results in Section 5 and those in Xu and Zhang (2009), this strong condition poses a serious restriction in searching for refinement kernels. A remedy is to ask the two norms to be equivalent in the smaller space or to even just focus on the inclusion relation. Study along this direction has been done for scalar-valued kernels (Zhang and Zhao, 2011). It is shown there that this relaxation brings more freedom and choices in choosing kernels for refinement. Vector-valued counterparts are yet to be investigated. This approach also connects to a popular way of updating kernels pointed out by one of the reviewers, which is to tune a parameter (for example, the variance in the Gaussian kernel, the degree in a polynomial kernel, etc.) in the kernel. Although this practice seldom corresponds to a refinement, it does sometimes fall into the approach considered in Zhang and Zhao (2011). Examples include the exponential kernels, the inverse multiquadrics, the B-spline kernels, and the polynomial kernels (Zhang and Zhao, 2011).

Acknowledgments

The authors would like to express their appreciation to the anonymous reviewers for many useful comments and suggestions that help improve the paper. The project was supported by Guangdong Provincial Government of China through the “Computational Science Innovative Research Team” program. Haizhang Zhang was partially supported by Natural Science Foundation of China under grants 11101438 and 91130009, and by the One Hundred Distinguished Scholars Program of Sun

Yat-sen University. Yuesheng Xu was partially supported by the US National Science Foundation under grant DMS-0712827, by the Natural Science Foundation of China under grants 91130009 and 11071286, and by US Air Force Office of Scientific Research under grant FA9550-09-1-0511. Qinghui Zhang was partially supported by the Natural Science Foundation of China under grant 11001282, by Fundamental Research Funds for Central Universities of China, and by Guangdong Provincial Natural Science Foundation of China under grant S2011040003030.

References

N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

S. K. Berberian. *Notes on Spectral Theory*. Van Nostrand, New York, 1966.

M. S. Birman and M. Z. Solomjak. *Spectral Theory of Self-Adjoint Operators in Hilbert Space*. D. Reidel Publishing Company, Dordrecht, Holland, 1987.

S. Bochner. *Lectures on Fourier Integrals with an Author's Supplement on Monotonic Functions, Stieltjes Integrals, and Harmonic Analysis*. Annals of Mathematics Studies 42, Princeton University Press, New Jersey, 1959.

J. Burbea and P. Masani. *Banach and Hilbert Spaces of Vector-valued Functions*. Pitman Research Notes in Mathematics 90, Boston, MA, 1984.

A. Caponnetto, C. A. Micchelli, M. Pontil and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 9:1615–1646, 2008.

C. Carmeli, E. De Vito and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Anal. Appl.*, 4:377–408, 2006.

C. Carmeli, E. De Vito, A. Toigo and V. Umanita. Vector valued reproducing kernel Hilbert spaces and universality. *Anal. Appl.*, 8:19–61, 2010.

J. B. Conway. *A Course in Functional Analysis*. 2nd Edition, Springer-Verlag, New York, 1990.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39:1–49, 2002.

F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, Cambridge, 2007.

I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics 61, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

J. Diestel and J.J. Uhl, Jr. *Vector Measures*. American Mathematical Society, Providence, 1977.

T. Evgeniou, C. A. Micchelli and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

T. Evgeniou, M. Pontil and T. Poggio. Regularization networks and support vector machines. *Adv. Comput. Math.*, 13:1–50, 2000.

P. A. Fillmore. *Notes on Operator Theory*. Van Nostrand Company, New York, 1970.

R. A. Horn and C. B. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.

S. Lowitzsh. Approximation and interpolation employing divergence-free radial basis functions with applications. Ph.D. Thesis, Texas A&M University, College Station, Texas, 2003.

S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Comput.*, 17:177–204, 2005.

C. A. Micchelli, Y. Xu and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

S. Mukherjee and Q. Wu. Estimation of gradients and coordinate covariation in classification. *Journal of Machine Learning Research*, 7:2481-2514, 2006.

S. Mukherjee and D. X. Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7:519–549, 2006.

G. B. Pedrick. Theory of reproducing kernels for Hilbert spaces of vector valued functions. *Technical Report* **19**, University of Kansas, 1957.

W. Rudin. *Real and Complex Analysis*. 3rd Edition, McGraw-Hill, New York, 1987.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, Mass, 2002.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

H. Wendland. Divergence-free kernel methods for approximating the Stokes problem. *SIAM J. Numer. Anal.*, 47:3158–3179, 2009.

Y. Xu and H. Zhang. Refinable kernels. *Journal of Machine Learning Research*, 8:2083–2120, 2007.

Y. Xu and H. Zhang. Refinement of reproducing kernels. *Journal of Machine Learning Research*, 10:107–140, 2009.

Y. Ying and C. Campbell. Learning coordinate gradients with multi-task kernels. In *COLT*, 2008.

H. Zhang, Y. Xu, and J. Zhang. Reproducing kernel Banach spaces for machine learning. *Journal of Machine Learning Research*, 10:2741–2775, 2009.

H. Zhang and L. Zhao. On the inclusion relation of reproducing kernel Hilbert spaces. *Anal. Appl.*, accetped subject to minor revision, *arXiv:1106.4075*, 2011.

An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity

Nir Ailon[*] NAILON@CS.TECHNION.AC.IL
Department of Computer Science
Taub Building
Technion Haifa 32000, Israel

Editor: Sanjoy Dasgupta

Abstract

Given a set V of n elements we wish to linearly order them given pairwise preference labels which may be non-transitive (due to irrationality or arbitrary noise).

The goal is to linearly order the elements while disagreeing with as few pairwise preference labels as possible. Our performance is measured by two parameters: The number of disagreements (loss) and the query complexity (number of pairwise preference labels). Our algorithm adaptively queries at most $O(\varepsilon^{-6} n \log^5 n)$ preference labels for a regret of ε times the optimal loss. As a function of n, this is asymptotically better than standard (non-adaptive) learning bounds achievable for the same problem.

Our main result takes us a step closer toward settling an open problem posed by learning-to-rank (from pairwise information) theoreticians and practitioners: What is a provably correct way to sample preference labels? To further show the power and practicality of our solution, we analyze a typical test case in which a large margin linear relaxation is used for efficiently solving the simpler learning problems in our decomposition.

Keywords: statistical learning theory, active learning, ranking, pairwise ranking, preferences

1. Introduction

We study the problem of learning to rank from pairwise preferences, and solve a long-standing open problem that has led to development of many heuristics but no provable results.

The setting is as follows: We are given a set V of n elements from some universe, and we wish to linearly order them given pairwise preference labels. given two elements $u, v \in V$, a pairwise preference label is obtained as a response, typically from a human, to the question *which if preferred, u or v?* We assume no abstention, hence, either u is preferred to v (denoted $u \prec v$) or the other way around.

The goal is to linearly order the elements from the most preferred to the least preferred, while disagreeing with as few pairwise preference labels as possible. Our performance is measured by two parameters: The loss (number of pairwise preference labels we disagree with) and the query complexity (number of pairwise preference labels we obtain). This is a typical learning problem, with the exception that the sample space is finite, consisting of $\binom{n}{2}$ possibilities only.

The loss minimization problem given the entire $n \times n$ preference matrix is a well known NP-hard problem called MFAST (minimum feedback arc-set in tournaments) (Alon, 2006). Recently,

[*] Supported by a Marie Curie International Reintegration Grant PIRG07-GA-2010-268403

Kenyon-Mathieu and Schudy (2007) have devised a PTAS for it, namely, a polynomial (in n) -time algorithm computing a solution with loss at most $(1+\varepsilon)$ the optimal, for and $\varepsilon > 0$ (the degree of the polynomial may depend on ε). In our case each edge from the input graph is given for a unit cost. Our main algorithm is derived from Kenyon et al's algorithm. Our output, however, is not a solution to MFAST, but rather a reduction of the original learning problem to a different, simpler one. The reduced problem can be solved using any general ERM (empirical risk minimization) black-box. The sampling of preference labels from the original problem is adaptive, hence the combination of our algorithm and any ERM blackbox is an active learning one. We give examples with an SVM based ERM black-box toward the end.

1.1 Our Setting vs. The Usual "Learning to Rank" Problem

Our setting defers from much of the *learning to rank* (LTR) literature. Usually, the labels used in LTR problems are responses to individual elements, and not to pairs of elements. A typical example is the $1..5$ scale rating for restaurants, or $0,1$ rating (irrelevant/relevant) for candidate documents retrieved for a query (known as the *binary ranking* problem). The goal there is, as in ours, to order the elements while disagreeing with as little pairwise relations as possible, where a pairwise relation is derived from any two elements rated differently. Note that the underlying preference graph there is transitive, hence no combinatorial problem due to nontransitivity. In fact, some view the rating setting as an ordinal regression problem and not a ranking problem. Here the preference graph may contain cycles, and is hence agnostic with respect to the concept class we are allowed to output from, namely, permutations. We note that some LTR literature does consider the pairwise preference label approach, and there is much justification to it (see Carterette et al. 2008; Hüllermeier et al. 2008 and reference therein). As far as we know, our work provides a sound solution to a problem addressed by machine learning practitioners (e.g., Carterette et al. 2008) who use pairwise preferences as labels for the task of learning to rank items, but wish to avoid obtaining labels for the quadratically many preference pairs, without compromising low error bounds. We also show that the *problem of quadraticity* found in much work dealing with pairwise preference based learning to rank (e.g., from Crammer and Singer 2001 *the [pairwise] approach is time consuming since it requires increasing the sample size ... to* $O(n^2)$) can be alleviated in the light of new advances in combinatorial optimization (Ailon et al., 2008a; Kenyon-Mathieu and Schudy, 2007).

1.2 Using Kenyon and Schudy's PTAS as a Starting Point

As mentioned above, our main algorithm is derived from the PTAS of Kenyon-Mathieu and Schudy (2007), but it is important to note a significant difference between our work and theirs. A good way to explain this is to compare two learners, Larry and Linda. On the first day, Larry queries all $\binom{n}{2}$ pairwise preference labels and sends them to a perfect solver for MFAST. Linda uses our work to query only $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ preference labels and obtains a decomposition of the original input V into an *ordered list* of sub-problems $V_1, \ldots, V_k$ where each V_i is contained in V. Using the same perfect solver for the induced subproblems corresponding to each part and concatenating the individual output permutations, Linda will incur a loss of at most $(1+\varepsilon)$ that of Larry. If the decomposition is nontrivial, then Linda enjoys reduced query complexity for a small regret compared to Larry. The next day, both Larry and Linda realize that the perfect MFAST solver cannot deal with large inputs (the problem is NP Hard). They cannot use the PTAS of Kenyon-Mathieu and Schudy (2007) because they seek a multiplicative regret of $(1+\varepsilon)$ with respect to the optimal solution (we also say a *relative regret* of ε), and the sought ε makes this infeasible.[1] To remedy this, Larry takes advantage of the fact that the set V does not merely consist of abstract elements, but rather each $u \in V$ is endowed with a feature vector $\varphi(u)$ and hence each pair of points u, v is endowed with the combined feature vector $(\varphi(u), \varphi(v))$. As in typical learning, he posits that the order relation between u, v can be deduced from a linear function of $(\varphi(u), \varphi(v))$, and invokes an optimizer (e.g., SVM) on the relaxed problem, with all pairs as input. Note that Larry may try to sample pairs uniformly to reduce the query complexity (and, perhaps, the running time of the relaxed solver), but as we show below, he will be discouraged from doing so because in certain realistic cases a relative regret of ε may entail sampling the entire pairwise preference space. Linda uses the same relaxed optimizer, say, SVM. The labels she sends to the solver consist of a uniform sample of pairs from each block V_i, together with all pairs u, v residing in separate blocks from her aforementioned construction decomposition. From the former label type she would need only $O(n\,\text{poly}(\log n, \varepsilon^{-1}))$ many, because (per our decomposition design) within the blocks the cost of any solution is high, and hence a *relative* error is tantamount to an absolute error of similar magnitude, for which careful arguments allow low query complexity. From the latter label type, she would generate a label for all pairs u, v in distinct V_i, V_j, using a "made up" label corresponding to the order of V_i, V_j (recall that the decomposition is ordered).

As the above story suggests, we do not run the PTAS of Kenyon-Mathieu and Schudy (2007) verbatim, but use it only to obtain a certain decomposition of the input. Among other changes, a key change to their algorithm is required by replacing a highly sensitive greedy improvement step into a robust approximate one, by careful sampling. The main difficulty stems from the fact that after a single greedy improvement step, the sample becomes stale and requires refreshing. We show a query efficient refreshing technique that allows iterated approximate greedy improvement steps. Interestingly, the original analysis is amenable to this change. It is also interesting to note that the sampling scheme used for identifying greedy improvement steps for a current solution are similar to ideas used by Ailon et al. (2007, 2008b) and Halevy and Kushilevitz (2007) in the context of property testing and reconstruction, where elements are sampled from exponentially growing intervals in a linear order.

The 3-approximation algorithm for MFAST using QuickSort by Ailon et al. (2008a) is used in Kenyon-Mathieu and Schudy (2007) as well as here as an initialization step. Note that this is a sublinear algorithm. In fact, it samples only $O(n \log n)$ pairs from the $\binom{n}{2}$ possible, on expectation. Note also that the pairs from which we query the preference relation in QuickSort are chosen adaptively.

1.3 Our Work in the Context of Machine Learning Reductions

Our main algorithm reduces a given instance to smaller subproblems decomposing it. We compare the machine learning reduction approach to two other works, that of Balcan et al. (2008) and that of Ailon and Mohri (2010). The former also considers a reduction of the problem of learning to rank, but in the *bipartite ranking* (see Section 1.1) setting where, in the first place, it is assumed that individual elements are endowed with unknown labels on a scale (of size two). The output is a permutation, and the loss function is the number of pairwise inversions. Their work shows that the problem of minimizing the regrets of the underlying binary classification and ranking problems is, up to a constant, the same thing. Their work, in fact, questions the justification for the so-called

[1] The running time of the PTAS is exponential in ε^{-1}. We note here, for the sake of comparison, that our sampling scheme has complexity polynomial in ε^{-1}.

binary ranking problem. The latter (Ailon and Mohri, 2010) considers the same setting as here, and shows a query efficient algorithm that reduces the original instance, which may contain cycles, to a binary classification problem over an adaptively chosen set of $O(n \log n)$ pairs on expectation. The results there guarantee a total regret of at most twice that of the optimal.[2] Here we obtain at most $1+\varepsilon$ that of the optimal using $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ pairwise queries.

1.4 Our Work in the Context of Active Learning

Active learning is an important field of statistical learning theory and practice (El-Yaniv and Wiener, 2010; Balcan et al., 2010; Hanneke, 2007; Dasgupta, 2005; Culotta and McCallum, 2005; Roth and Small, 2006; Dasgupta et al., 2007; Atlas et al., 1994; Freund et al., 1997; Lindenbaum et al., 2004; Begleiter et al., 2008; Balcan et al., 2009; Angluin, 2004; Dasgupta et al., 2009; Fine et al., 2002; Baram et al., 2004; Atlas et al., 1994; Friedman, 2009; Atlas et al., 1990; Yu et al., 2006). In the most general setting, one wishes to improve on standard query complexity bounds (using, for example, VC or Rademacher complexity) by actively choosing which instances to obtain labels for. Many heuristics have been developed, while algorithms with provable bounds (especially in the agnostic case) are known for few problems. Balcan et al. (2010) show that any learning algorithm for a finite VC dimensional space admits an active learning algorithm which asymptotically beats, in query complexity, that of a passive learning algorithm. Their guarantees are, however, unverifiable in the sense that the learner does not know when to stop querying in order to achieve a certain error. Also, their scheme still requires a considerable amount of work in order to be applicable for individual problems. It is an interesting open question to apply it to the problem at hand and compare the results with our algorithms' guarantees. Also, Balcan et al. (2009) proposed an active learning algorithm called A2. A useful measure of complexity which was later defined by Hanneke (2007) is key in analysis of A2. He defined a disagreement coefficient for a concept space and showed how this measure could be used for active learning in certain cases. We show in Appendix B why this measure does not help here.

1.5 Our Work in the Context of Noisy Sorting

There is much literature in theoretical computer science on sorting noisy data. For example, Braverman and Mossel (2008) present an algorithm with an $O(n \log n)$ query complexity for exact order reconstruction when the input is Bayesian with certain natural priors. Feige et al. (2002) consider a scenario in which the input preference graph is transitive, but queries may result in noisy comparisons which may be inconsistent with previous information (hence, querying the same pair multiple times would result in difference independent responses). Ajtai et al. (2009) consider a setting in which each element has a latent value, and comparisons of two elements with similar value may result in errors. In this work the input is not Bayesian, query responses are fixed and elements do not have a latent value.

1.6 Paper Organization

In Section 2 we present basic definitions and lemmata, and in particular define what a good decomposition is and how it can be used in learning permutations from pairwise preferences. Section 3 presents our main active learning algorithm which is, in fact, an algorithm for producing a good

[2] Additionally, they consider the so called binary ranking, which is not the problem here.

decomposition query efficiently. The main result is presented in Theorem 7. Section 4 discusses our main results as a preconditioner for a standard SVM relaxation for the hard combinatorial problems underlying the problem of minimum feedback-arcset in sparse graphs.

2. Notation and Basic Lemmata

We start by introducing basic notations and definitions of the problem, together with results from statistical learning theory which we will later improve.

2.1 The Learning Theoretical Problem

Let V denote a finite set that we wish to rank. In a more general setting we are given a sequence $V^1, V^2, \dots$ of sets, but there is enough structure and interest in the single set case, which we focus on in this work. Denote by n the cardinality of V. We assume there is an underlying preference function W on pairs of elements in V, which is unknown to us. For any ordered pair $u, v \in V$, the preference value $W(u,v)$ takes the value of 1 if u is deemed preferred over v, and 0 otherwise. We enforce $W(u,v) + W(v,u) = 1$, hence, (V,W) is a tournament. We assume that W is *agnostic* in the sense that it does not necessarily encode a transitive preference function, and may contain errors and inconsistencies. For convenience, for any two real numbers a, b we will let $[a,b]$ denote the interval $\{x : a \leq x \leq b\}$ if $a \leq b$ and $\{x : b \leq x \leq a\}$ otherwise.

Assume now that we wish to predict W using a hypothesis h from some concept class $\mathcal{H}$. The hypothesis h will take an ordered pair $(u,v) \in V$ as input, and will output label of 1 to assert that u *precedes* v and 0 otherwise. We want $\mathcal{H}$ to contain only consistent hypotheses, satisfying transitivity (i.e., if $h(u,v) = h(v,w) = 1$ then $h(u,w) = 1$). A typical way to do this is using a linear score function: Each $u \in V$ is endowed with a feature vector $\varphi(u)$ in some RKHS H, a weight vector $w \in H$ is used for parametrizing each $h_w \in \mathcal{H}$, and the prediction is as follows:[3]

$$h_w(u,v) = \begin{cases} 1 & \langle w, \varphi(u) \rangle > \langle w, \varphi(v) \rangle \\ 0 & \langle w, \varphi(u) \rangle < \langle w, \varphi(v) \rangle \\ \mathbf{1}_{u<v} & \text{otherwise} \end{cases} .$$

Our work is relevant, however, to nonlinear hypothesis classes as well. We denote by $\Pi(V)$ the set permutations on the set V, hence we always assume $\mathcal{H} \subseteq \Pi(V)$. (Permutations π are naturally viewed as binary classifiers of pairs of elements via the preference predicate: The notation is, $\pi(u,v) = 1$ if and only if $u \prec_\pi v$, namely, if u precedes v in π. Slightly abusing notation, we also view permutations as injective functions from $[n]$ to V, so that the element $\pi(1) \in V$ is in the first, most preferred position and $\pi(n)$ is the least preferred one. We also define the function ρ_π inverse to π as the unique function satisfying $\pi(\rho_\pi(v)) = v$ for all $v \in V$. Hence, $u \prec_\pi v$ is equivalent to $\rho_\pi(u) < \rho_\pi(v)$.)

As in standard ERM setting, we assume a non-negative risk function $C_{u,v}$ penalizing the error of h with respect to the pair u, v, namely,

$$C_{u,v}(h, V, W) = \mathbf{1}_{h(u,v) \neq W(u,v)} .$$

[3] We assume that $V$is endowed with an arbitrary linear order relation, so we can formally write $u < v$ to arbitrarily yet consistently break ties.

The total loss, $C(h,V,W)$ is defined as $C_{u,v}$ summed over all unordered $u,v \in V$. Our goal is to devise an active learning algorithm for the purpose of minimizing this loss.

In this paper we find an almost optimal solution to the problem using important breakthroughs in combinatorial optimization of a related problem called *minimum feedback arc-set in tournaments* (MFAST). The relation between this NP-Hard problem and our learning problem has been noted before (Cohen et al., 1998), but no provable almost optimal active learning has been devised, as far as we know.

2.2 The Combinatorial Optimization Counterpart

MFAST is defined as follows: Assume we are given V and W and its entirety, in other words, we pay no price for reading W. The goal is to order the elements of V in a full linear order, while minimizing the total pairwise violation. More precisely, we wish to find a permutation π on the elements of V such that the total backward cost:

$$C(\pi,V,W) = \sum_{u \prec_\pi v} W(v,u) \tag{1}$$

is minimized. The expression in (1) will be referred to as the *MFAST cost* henceforth.

When W is given as input, this problem is known as the minimum feedback arc-set in tournaments (MFAST). A PTAS has been discovered for this NP-Hard very recently (Kenyon-Mathieu and Schudy, 2007). Though a major theoretical achievement from a combinatorial optimization point of view, the PTAS is not useful for the purpose of *learning to rank from pairwise preferences* because it is not query efficient. Indeed, it may require in some cases to read all quadratically many entries in W. In this work we fix this drawback, while using their main ideas for the purpose of machine learning to rank. We are not interested in MFAST per se, but use the algorithm by Kenyon-Mathieu and Schudy (2007) to obtain a certain useful decomposition of the input (V,W) from which our main active learning result easily follows.

Definition 1 *Given a set V of size n, an ordered decomposition is a list of pairwise disjoint subsets $V_1,\dots,V_k \subseteq V$ such that $\cup_{i=1}^k V_i = V$. For a given decomposition, we let $W|_{V_i}$ denote the restriction of W to $V_i \times V_i$ for $i = 1,\dots,k$. Similarly, for a permutation $\pi \in \Pi(v)$ we let $\pi|_{V_i}$ denote the restriction of the permutation to the elements of V_i (hence, $\pi|_{V_i} \in \Pi(V_i)$). We say that $\pi \in \Pi(V)$ respects $V_1,\dots,V_k$ if for all $u \in V_i, v \in V_j, i < j$, $u \prec_\pi v$. We denote the set of permutations $\pi \in \Pi(V)$ respecting the decomposition $V_1,\dots,V_k$ by $\Pi(V_1,\dots,V_k)$. We say that a subset U of V is* small *in V if $|U| \le \log n/\log\log n$, otherwise we say that U* is big *in V. A decomposition $V_1,\dots,V_k$ is* ε-good *with respect to W if:*[4]

- *Local chaos:*

$$\min_{\pi\in\Pi(V)} \sum_{i:V_i \text{ big in } V} C(\pi_{|V_i}, V_i, W_{|V_i}) \ge \varepsilon^2 \sum_{i:V_i \text{ big in } V} \binom{n_i}{2} . \tag{2}$$

- *Approximate optimality:*

$$\min_{\sigma\in\Pi(V_1,\dots,V_k)} C(\sigma,V,W) \le (1+\varepsilon) \min_{\pi\in\Pi(V)} C(\pi,V,W) . \tag{3}$$

[4] We will just say ε-good if W is clear from the context.

Intuitively, an ε-good decomposition identifies a block-ranking of the data that is difficult to rank in accordance with W internally on average among big blocks (*local chaos*), yet possible to rank almost optimally while respecting the decomposition (*approximate optimality*). We show how to take advantage of an ε-good decomposition for learning in Section 2.3. The ultimate goal will be to find an ε-good decomposition of the input set V using $O(\mathrm{poly}(\log n, \varepsilon^{-1}))$ queries into W.

2.3 Basic Results from Statistical Learning Theory

In statistical learning theory, one seeks to find a classifier minimizing an expected cost incurred on a random input by minimizing the empirical cost on a sample thereof. If we view pairs of elements in V as data points, then the MFAST cost can be cast, up to normalization, as an expected cost over a random draw of a data point. The distribution space is finite, hence we may view this as a transductive learning algorithm. Recall our notation of $\pi(u,v)$ denoting the indicator function for the predicate $u \prec_\pi v$. Thus π is viewed as a binary hypothesis function over $\binom{V}{2}$, and $\Pi(V)$ can be viewed as the concept class of all binary hypotheses satisfying transitivity: $\pi(u,v)+\pi(v,y) \geq \pi(u,y)$ for all u,v,y.

A sample E of unordered pairs gives rise to a *partial cost*, C_E defined as follows:

Definition 2 *Let (V,E) denote an undirected graph over V, which may contain parallel edges (E is a multi-set). The partial MFAST cost $C_E(\pi)$ is defined as*

$$C_E(\pi,V,W) = \binom{n}{2}|E|^{-1} \sum_{\substack{(u,v)\in E \\ u<_\pi v}} W(v,u) .$$

(The accounting of parallel edges in E is clear.) The function $C_E(\cdot,\cdot,\cdot)$ can be viewed as an *empirical unbiased estimator* of $C(\pi,V,W)$ if $E \subseteq \binom{V}{2}$ is chosen uniformly at random among all (multi)subsets of a given size.

The basic question in statistical learning theory is, how good is the minimizer π of C_E, in terms of C? The notion of VC dimension by Vapnik and Chervonenkis (1971) gives us a nontrivial bound which is, albeit suboptimal (as we shall soon see), a good start for our purpose.

Lemma 3 *The VC dimension of the set of permutations on V, viewed as binary classifiers on pairs of elements, is $n-1$.*

It is easy to show that the VC dimension is at most $O(n\log n)$. Indeed, the number of permutations is at most $n!$, and the VC dimension is always bounded by the log of the concept class cardinality. That the bound is linear was proven by Ailon and Radinsky (2011). We present the proof here in Appendix A for completeness. The implications of the VC bound are as follows.

Proposition 4 *Assume E is chosen uniformly at random (with repetitions) as a sample of m elements from $\binom{V}{2}$, where $m > n$. Then with probability at least $1-\delta$ over the sample, all permutations π satisfy:*

$$|C_E(\pi,V,W) - C(\pi,V,W)| = n^2 O\left(\sqrt{\frac{n\log m + \log(1/\delta)}{m}}\right) .$$

The consequence of Proposition 4 are as follows: If we want to minimize $C(\pi,V,W)$ over π to within an additive error of μn^2, and succeed in doing so with probability at least $1-\delta$, it is enough to choose a sample E of $O(\mu^{-2}(n\log n+\log\delta^{-1}))$ elements from $\binom{V}{2}$ uniformly at random (with repetitions), and optimize $C_E(\pi,V,W)$. Assume from now on that δ is at least e^{-n}, so that we get a more manageable sample bound of $O(\mu^{-2}n\log n)$. Before turning to optimizing $C_E(\pi,V,W)$, a hard problem in its own right (Karp, 1972; Dinur and Safra, 2002), we should first understand whether this bound is at all good for various scenarios. We need some basic notions of distance between permutations. For two permutations π,σ, the Kendall-Tau distance $d_\tau(\pi,\sigma)$ is defined as

$$d_\tau(\pi,\sigma)=\sum_{u\neq v}\mathbf{1}[(u\prec_\pi v)\wedge(v\prec_\sigma u)]\ .$$

The Spearman Footrule distance $d_{\text{foot}}(\pi,\sigma)$ is defined as

$$d_{\text{foot}}(\pi,\sigma)=\sum_u|\rho_\pi(u)-\rho_\sigma(u)|\ .$$

The following is a well known inequality due to Diaconis and Graham (1977) relating the two distance measures for all π,σ:

$$d_\tau(\pi,\sigma)\le d_{\text{foot}}(\pi,\sigma)\le 2d_\tau(\pi,\sigma)\ . \tag{4}$$

Clearly d_τ and d_{foot} are metrics. It is also clear that $C(\cdot,V,\cdot)$ is an extension of $d_\tau(\cdot,\cdot)$ to distances between permutations and binary tournaments, with the triangle inequality of the form $d_\tau(\pi,\sigma)\le C(\pi,V,W)+C(\sigma,V,W)$ satisfied for all W and $\pi,\sigma\in\Pi(V)$.

Assume now that we are able, using Proposition 4 and the ensuing comment, to find a solution π for MFAST, with an additive regret of $O(\mu n^2)$ with respect to an optimal solution π^* for some $\mu>0$. The triangle inequality implies that the distance $d_\tau(\pi,\pi^*)$ between our solution and the true optimal is $\Omega(\mu n^2)$. By (4), this means that $d_{\text{foot}}(\pi,\pi^*)=\Omega(\mu n^2)$. By the definition of d_{foot}, this means that the average element $v\in V$ is translated $\Omega(\mu n)$ positions away from its position in π^*. In a real life application (e.g., in information retrieval), one may want elements to be at most a constant γ positions away from their position in a correct permutation. This translates to a sought regret of $O(\gamma n)$ in $C(\pi,V,W)$, or, using the above notation, to $\mu=\gamma/n$. Clearly, Proposition 4 cannot guarantee less than a quadratic sample size for such a regret, which is tantamount to querying W in its entirety. We can do better: In this work, for any $\varepsilon>0$ we will achieve a regret of $O(\varepsilon C(\pi^*,V,W))$ using $O(\varepsilon^{-6}n\log^5 n)$ queries into W, regardless of how small the optimal cost $C(\pi^*,V,W)$ is. Hence, our regret is relative to the optimal loss. This is clearly not achievable using Proposition 4. Let us outline another practical case of interest. Assume a scenario in which a ground truth permutation $\pi\in\Pi(V)$ exists, and the noisy preference matrix W is generated by a human responder who errs on a pair u,v with probability $f(|\rho_\pi(u)-\rho_\pi(v)|)$, where f is some monotonically decreasing function. Intuitively, this scenario posits that people confuse the order of two elements the "closer" they are to each other. If, say, $f(x)=px^{-\nu}$ for some $\nu>0$ and $p>0$, then the cost of the optimal solution π would be $\Theta(pn^{2-\nu})$ with high probability.[5] Proposition 4 tells us that if we wanted to find a permutation with *relative* error of ε, namely, of absolute error $\Theta(\varepsilon pn^{2-\nu})$, then we would need $O(\varepsilon^{-2}p^{-2}n^{1+2\nu}\log n)$ queries. Our result achieves the same error with an almost linear dependence on n (albeit a worse dependence on ε).

One may argue that the VC bound measures the merits of uniform, non-adaptive sampling too pessimistically. This isn't the case. Consider the extreme case in which the optimal cost is zero. We

[5] We are assuming stochastic noise for the sake of the example, although this work deals with adversarial noise.

argue that a uniform sample of pairs *requires* $\Omega(n^2)$ query complexity. Indeed, if the optimal cost is zero then unless one queries all $n-1$ consecutive pairs in the unique optimal permutation, one cannot reveal it. It is now easy to see that sampling $o(n^2)$ pairs uniformly (either with or without repetition) would succeed in doing so with exponentially (in n) small probability. A relative ε approximation cannot be thus achieved. But we know that an adaptive sample of $O(n\log n)$ pairs on expectation (QuickSort) does better. It is folklore that $\Omega(n\log n)$ is also a lower bound in the perfect (zero cost) case. Hence, one cannot hope to get rid of the $\log n$ factor in our main result, Theorem 7 below.

Before continuing, we need need a slight generalization of Proposition 4.

Proposition 5 *Let $V_1,\dots,V_k$ be an ordered decomposition of V. Let $\mathcal{B}$ denote the set of indices $i \in [k]$ such that V_i is big in V. Assume E is chosen uniformly at random (with repetitions) as a sample of m elements from $\bigcup_{i\in\mathcal{B}}\binom{V_i}{2}$, where $m > n$. For each $i = 1,\dots,k$, let $E_i = E \cap \binom{V_i}{2}$. Define $C_E(\pi,\{V_1,\dots,V_k\},W)$ to be*

$$C_E(\pi,\{V_1,\dots,V_k\},W) = \left(\sum_{i\in\mathcal{B}}\binom{n_i}{2}\right)|E|^{-1}\sum_{i\in\mathcal{B}}\binom{n_i}{2}^{-1}|E_i|C_{E_i}(\pi_{|V_i},V_i,W_{|V_i}) \ . \tag{5}$$

(The normalization is defined so that the expression is an unbiased estimator of $\sum_{i\in\mathcal{B}}C(\pi_{|V_i},V_i,W_{|V_i})$. If $|E_i| = 0$ for some i, formally define $\binom{n_i}{2}^{-1}|E_i|C_{E_i}(\pi_{|V_i},V_i,W_{|V_i}) = 0$.) Then with probability at least $1-e^{-n}$ over the sample, all permutations $\pi \in \Pi(V)$ satisfy:

$$\left|C_E(\pi,\{V_1,\dots,V_k\},W) - \sum_{i\in\mathcal{B}}C(\pi|_{V_i},V_i,W|_{V_i})\right| = \sum_{i\in\mathcal{B}}\binom{n_i}{2}O\left(\sqrt{\frac{n\log m + \log(1/\delta)}{m}}\right) \ .$$

Proof Consider the set of binary functions $\prod_{i\in\mathcal{B}}\Pi(V_i)$ on the domain $\bigcup_{i\in\mathcal{B}}V_i\times V_i$, defined as follows: If $u,v \in V_j \times V_j$ for some $j \in \mathcal{B}$, then

$$((\pi_i)_{i\in\mathcal{B}})(u,v) = \pi_j(u,v) \ .$$

It is clear that the VC dimension of this function set is at most the sum of the VC dimensions of $\{\Pi(V_i)\}_{i\in\mathcal{B}}$, hence by Lemma 3 at most n. The result follows. ■

2.4 Using an ε-Good Partition

The following lemma explains why an ε-good partition is good for our purpose.

Lemma 6 *Fix $\varepsilon > 0$ and assume we have an ε-good partition (Definition 1) $V_1,\dots,V_k$ of V. Let $\mathcal{B}$ denote the set of $i \in [k]$ such that V_i is big in V, and let $\bar{\mathcal{B}} = [k]\setminus\mathcal{B}$. Let $n_i = |V_i|$ for $i = 1,\dots,n$, and let E denote a random sample of $O(\varepsilon^{-6}n\log n)$ elements from $\bigcup_{i\in\mathcal{B}}\binom{V_i}{2}$, each element chosen uniformly at random with repetitions. Let E_i denote $E\cap\binom{V_i}{2}$. Let $C_E(\pi,\{V_1,\dots,V_k\},W)$ be defined as in (5). For any $\pi \in \Pi(V_1,\dots,V_k)$ define:*

$$\tilde{C}(\pi) := C_E(\pi,\{V_1,\dots,V_k\},W) + \sum_{i\in\bar{\mathcal{B}}}C(\pi_{|V_i},V_i,W_{|V_i}) + \sum_{1\le i<j\le k}\sum_{(u,v)\in V_i\times V_j}\mathbf{1}_{v\prec_\pi u} \ .$$

Then the following event occurs with probability at least $1-e^{-n}$*: For all* $\sigma \in \Pi(V_1,\dots,V_k)$*,*

$$\left|\tilde{C}(\sigma) - C(\sigma,V,W)\right| \le \varepsilon \min_{\pi\in\Pi(V)} C(\pi,V,W)\ . \tag{6}$$

Also, if σ^* *is any minimizer of* $\tilde{C}(\cdot)$ *over* $\Pi(V_1,\dots,V_k)$*, then*

$$C(\sigma^*,V,W) \le (1+2\varepsilon) \min_{\pi\in\Pi(V)} C(\pi,V,W)\ . \tag{7}$$

Before we prove the lemma, let us discuss its consequences: Given an ε-good decomposition $V_1,\dots,V_k$ of V, the theorem implies that if we could optimize $\tilde{C}(\sigma)$ over $\sigma \in \Pi(V_1,\dots,V_k)$, we would obtain a permutation π with a *relative regret* of 2ε with respect to the optimizer of $C(\cdot,V,W)$ over $\Pi(V)$. Optimizing $\sum_{i\in\hat{\mathcal{B}}} C(\pi_{|V_i},V_i,W_{|V_i})$ is easy: Each V_i is of size at most $\log n/\log\log n$, hence exhaustively searching its corresponding permutation space can be done in polynomial time. In order to compute the cost of each permutation inside the small sets V_i, we would need to query $W_{|V_i}$ in its entirety. This incurs a query cost of at most $\sum_{i\in\bar{\mathcal{B}}}\binom{n_i}{2} = O(n\log n/\log\log n)$, which is dominated by the cost of obtaining the ε-good partition in the first place (see next section). Optimizing $C_E(\pi,\{V_1,\dots,V_k\},W)$ given E is a tougher nut to crack, is known as the minimum feedback arc-set (MFAS) problem and is computationally much harder than than MFAST (Karp, 1972; Dinur and Safra, 2002). For now we focus on query and not computational complexity, and notice that the size $|E| = O(\varepsilon^{-4} n\log n)$ of the sample set is all we need. In Section 4 we show a counterpart of Lemma 6 which provides similar guarantees for practitioners who choose to relax it using SVM, for which fast solvers exist.

Proof For any permutation $\sigma \in \Pi(V_1,\dots,V_k)$, it is clear that

$$\tilde{C}(\sigma) - C(\sigma,V,W) = C_E(\sigma,\{V_1,\dots,V_k\},W) - \sum_{i\in\mathcal{B}} C(\sigma_{|V_i},V_i,W_{|V_i})\ .$$

By Proposition 5, with probability at least $1-e^{-n}$ the absolute value of the RHS is bounded by $\varepsilon^3\sum_{i\in\mathcal{B}}\binom{n_i}{2}$, which is at most $\varepsilon\min_{\pi\in\Pi(V)} C(\pi,V,W)$ by (2). This establishes (6). Inequality (7) is obtained from (6) together with (3) and the triangle inequality. ■

3. A Query Efficient Algorithm for ε-Good Decomposing

The section is dedicated to proving the following:

Theorem 7 *Given a set* V *of size* n*, a preference oracle* W *and an error tolerance parameter* $0 < \varepsilon < 1$*, there exists a polynomial time algorithm which returns, with constant probability, an* ε*-good partition of* V*, querying at most* $O(\varepsilon^{-6} n\log^5 n)$ *locations in* W *on expectation. The running time of the algorithm (counting computations) is* $O(n\,\mathrm{poly}(\log n,\varepsilon^{-1}))$*.*

Before describing our algorithm, we need some definitions.

Definition 8 *Let* π *denote a permutation over* V*. Let* $v\in V$ *and* $i\in[n]$*. We define* $\pi_{v\to i}$ *to be the permutation obtained by moving the rank of* v *to* i *in* π*, and leaving the rest of the elements in the same order. For example, if* $V=\{x,y,z\}$ *and* $(\pi(1),\pi(2),\pi(3)) = (x,y,z)$*, then* $(\pi_{x\to3}(1),\pi_{x\to3}(2),\pi_{x\to3}(3)) = (y,z,x)$*.*

Definition 9 *Fix a permutation* π *over* V*, an element* $v \in V$ *and an integer* $i \in [n]$*. We define the number* $\mathrm{TestMove}(\pi, V, W, v, i)$ *as the decrease in the cost* $C(\cdot, V, W)$ *achieved by moving from* π *to* $\pi_{v \to i}$*. More precisely,* $\mathrm{TestMove}(\pi, V, W, v, i) = C(\pi, V, W) - C(\pi_{v\to i}, V, W)$ *. Equivalently, if* $i \geq \rho_\pi(v)$ *then*

$$\mathrm{TestMove}(\pi, V, W, v, i) = \sum_{u : \rho_\pi(u) \in [\rho_\pi(v)+1, i]} (W_{uv} - W_{vu}) \ .$$

A similar expression can be written for $i < \rho_\pi(v)$*.*

Now assume that we have a multi-set $E \subseteq \binom{V}{2}$*. We define* $\mathrm{TestMove}_E(\pi, V, W, v, i)$*, for* $i \geq \rho_\pi(v)$*, as*

$$\mathrm{TestMove}_E(\pi, V, W, v, i) \quad = \quad \frac{|i - \rho_\pi(v)|}{|\tilde{E}|} \sum_{u : (u,v) \in \tilde{E}} (W(u,v) - W(v,u)) \ ,$$

where the multiset $\tilde{E}$ *is defined as* $\{(u,v) \in E : \rho_\pi(u) \in [\rho_\pi(v)+1, i]\}$*. Similarly, for* $i < \rho_\pi(v)$ *we define*

$$\mathrm{TestMove}_E(\pi, V, W, v, i) \quad = \quad \frac{|i - \rho_\pi(v)|}{|\tilde{E}|} \sum_{u : (u,v) \in \tilde{E}} (W(v,u) - W(u,v)) \ , \tag{8}$$

where the multiset $\tilde{E}$ *is now defined as* $\{(u,v) \in E : \rho_\pi(u) \in [i, \rho_\pi(v) - 1]\}$*.*

Lemma 10 *Fix a permutation* π *over* V*, an element* $v \in V$*, an integer* $i \in [n]$ *and another integer* N*. Let* $E \subseteq \binom{V}{2}$ *be a random (multi)-set of size* N *with elements* $(v, u_1), \ldots, (v, u_N)$*, drawn so that for each* $j \in [N]$ *the element* u_j *is chosen uniformly at random from among the elements lying between* v *(exclusive) and position* i *(inclusive) in* π*.*
Then $\mathbf{E}[\mathrm{TestMove}_E(\pi, V, W, v, i)] = \mathrm{TestMove}(\pi, V, W, v, i)$*. Additionally, for any* $\delta > 0$*, except with probability of failure* δ*,*

$$|\,\mathrm{TestMove}_E(\pi, V, W, v, i) - \mathrm{TestMove}(\pi, V, W, v, i)| = O\left(|i - \rho_\pi(v)| \sqrt{\frac{\log \delta^{-1}}{N}}\right) \ .$$

The lemma is easily proven using, for example, Hoeffding tail bounds, using the fact that $|W(u,v)| \leq 1$ for all u, v.

3.1 The Decomposition Algorithm

Our decomposition algorithm SampleAndRank is detailed in Algorithm 1, with subroutines in Algorithms 2 and 3. It can be viewed as a query efficient improvement of the main algorithm of Kenyon-Mathieu and Schudy (2007). Another difference is that we are not interested in an approximation algorithm for MFAST: Whenever we reach a small block (line 3) or a big block with a probably approximately sufficiently high cost (line 8) in our recursion of Algorithm 2), we simply output it as a block in our partition. Denote the resulting outputted partition by $V_1, \ldots, V_k$. Denote by $\hat{\pi}$ the minimizer of $C(\cdot, V, W)$ over $\Pi(V_1, \ldots, V_k)$. Most of the analysis is dedicated to showing that $C(\hat{\pi}, V, W) \leq (1+\varepsilon) \min_{\pi \in \Pi(V)} C(\pi, V, W)$, thus establishing (3).

In order to achieve an efficient query complexity compared to that of Kenyon-Mathieu and Schudy (2007), we use procedure ApproxLocalImprove (Algorithm 3) to replace a greedy local improvement step there which is *not* query efficient. Aside from the aforementioned differences, we also raise here the reader's awareness to the query efficiency of QuickSort, which was established by Ailon and Mohri (2010).

SampleAndRank (Algorithm 1) takes the following arguments: The set V we want to rank, the preference matrix W and an accuracy argument ε. It is implicitly understood that the argument W passed to SampleAndRank is given as a query oracle, incurring a unit cost upon each access to a matrix element by the procedure and any nested calls.

The first step in SampleAndRank is to obtain an expected constant factor approximation π to MFAST on V, W, incurring an expected low query cost. More precisely, this step returns a random permutation π with an expected cost of $O(1)$ times that of the optimal solution to MFAST on V, W. The query complexity of this step is $O(n \log n)$ *on expectation* (Ailon and Mohri, 2010). Before continuing, we make the following assumption, which holds with constant probability using Markov probability bounds.

Assumption 11 *The cost $C(\pi, V, W)$ of the initial permutation π computed line 2 of* SampleAndRank *is at most $O(1)$ times that of the optimal solution π^* to MFAST on (V, W), and the query cost incurred in the computation is $O(n \log n)$.*

Following QuickSort, a recursive procedure SampleAndDecompose is called. It implements a divide-and-conquer algorithm. Before branching, it executes the following steps. Lines 5 to 9 are responsible for identifying local chaos, with sufficiently high probability. The following line 10 calls a procedure ApproxLocalImprove (Algorithm 3) which is responsible for performing query-efficient approximate greedy steps. We devote the next Sections 3.2-3.4 to describing this procedure. The establishment of the ε-goodness of SampleAndRank's output (establishing (3)) is deferred to Section 3.5.

3.2 Approximate Local Improvement Steps

The procedure ApproxLocalImprove takes as input a set V of size N, the preference oracle W, a permutation π on V, two numbers C_0, ε and an integer n. The number n is the size of the input in the root call to SampleAndDecompose, passed down in the recursion, and used for the purpose of controlling the success probability of each call to the procedure (there are a total of $O(n \log n)$ calls, and a union bound will be used to bound a failure probability, hence each call may fail with probability inversely polynomial in n). The goal of the procedure is to repeatedly identify, with high probability, single vertex moves that considerably decrease the cost. Note that in the PTAS of Kenyon-Mathieu and Schudy (2007), a crucial step in their algorithms entails identifying single vertex moves that decrease the cost by a magnitude which, given our sought query complexity, would not be detectable. Hence, our algorithm requires altering this crucial part in their algorithm.

The procedure starts by creating a *sample ensemble* $\mathcal{S} = \{E_{v,i} : v \in V, i \in [B, L]\}$, where $B = \log\lfloor\Theta(\varepsilon N/\log n)\rfloor$ and $L = \lceil \log N \rceil$. The size of each $E_{v,i} \in \mathcal{S}$ is $\Theta(\varepsilon^{-2} \log^2 n)$, and each element $(v, x) \in E_{v,i}$ was added (with possible multiplicity) by uniformly at random selecting, with repetitions, an element $x \in V$ positioned at distance at most 2^i from the position of v in π. Let $\mathcal{D}_\pi$ denote the distribution space from which $\mathcal{S}$ was drawn, and let $\Pr_{X \sim \mathcal{D}_\pi}[X = \mathcal{S}]$ denote the probability of obtaining a given sample ensemble $\mathcal{S}$.

We want $\mathcal{S}$ to enable us to approximate the improvement in cost obtained by moving a single element u to position j.

Definition 12 *Fix $u \in V$ and $j \in [n]$, and assume $\log|j-\rho_\pi(u)| \geq B$. Let $\ell = \lceil \log|j-\rho_\pi(u)| \rceil$. We say that $\mathcal{S}$ is successful at u, j if $|\{x : (u,x) \in E_{u,\ell}\} \cap \{x : \rho_\pi(x) \in [\rho_\pi(u), j]\}| = \Omega(\varepsilon^{-2}\log^2 n)$.*

In words, success of $\mathcal{S}$ at u, j means that sufficiently many samples $x \in V$ such that $\rho_\pi(x)$ is between $\rho_\pi(u)$ and j are represented in $E_{u,\ell}$. Conditioned on $\mathcal{S}$ being successful at u, j, note that the denominator of TestMove$_E$ (defined in (8)) does not vanish, and we can thereby define:

Definition 13 *$\mathcal{S}$ is a* good approximation *at u, j if*

$$\left|\text{TestMove}_{E_{u,\ell}}(\pi, V, W, u, j) - \text{TestMove}(\pi, V, W, u, j)\right| \leq \frac{1}{2}\varepsilon|j-\rho_\pi(u)|/\log n ,$$

where ℓ is as in Definition 12.

In words, $\mathcal{S}$ being a good approximation at u, j allows us to approximate a quantity of interest TestMove(π, V, W, u, j), and to detect whether it is sufficiently large, and more precisely, at least $\Omega(\varepsilon|j-\rho_\pi(u)|/\log n)$.

Definition 14 *We say that $\mathcal{S}$ is a good approximation if it is successful and a good approximation at all $u \in V$, $j \in [n]$ satisfying $\lceil \log|j-\rho_\pi(u)| \rceil \in [B, L]$.*

Using Chernoff bounds to ensure that $\mathcal{S}$ is successful $\forall u, j$ as in Definition 14, then using Hoeffding to ensure that $\mathcal{S}$ is a good approximation at all such u, j and finally union bounding we get

Lemma 15 *Except with probability $1 - O(n^{-4})$, $\mathcal{S}$ is a good approximation.*

Algorithm 1 SampleAndRank(V, W, ε)

```
$n \leftarrow |V|$
$\pi \leftarrow$ Expected $O(1)$-approx solution to MFAST using $O(n\log n)$ $W$-queries on expectation using QuickSort (Ailon et al., 2008a)
return SampleAndDecompose$(V, W, \varepsilon, n, \pi)$
```

3.3 Mutating the Pair Sample To Reflect a Single Element Move

Line 17 in ApproxLocalImprove requires elaboration. In lines 15-20, we check whether there exists an element u and position j, such that moving u to j (giving rise to $\pi_{u\to j}$) would considerably improve the MFAST cost of the procedure input, based on a high probability approximate calculation. The approximation is done using the sample ensemble $\mathcal{S}$. If such an element u exists, we execute the exchange $\pi \leftarrow \pi_{u\to j}$. With respect to the new value of the permutation π, the sample ensemble $\mathcal{S}$ becomes *stale*. By this we mean, that if $\mathcal{S}$ was a good approximation with respect to π, then it is no longer necessarily a good approximation with respect to $\pi_{u\to j}$. We must refresh it. Before the next iteration of the while loop, we perform in line 17 a transformation $\varphi_{u\to j}$ to $\mathcal{S}$, so that the resulting sample ensemble $\varphi_{u\to j}(\mathcal{S})$ is distributed according to $\mathcal{D}_{\pi_{u\to j}}$. More precisely, we will define a transformation φ such that

$$\varphi_{u\to j}(\mathcal{D}_\pi) = D_{\pi_{u\to j}} , \tag{9}$$

where the left hand side denotes the distribution obtained by drawing from $\mathcal{D}_\pi$ and applying $\varphi_{u\to j}$ to the result. The transformation $\varphi_{u\to j}$ is performed as follows. Denoting $\varphi_{u\to j}(\mathcal{S}) = \mathcal{S}' = \{E'_{v,i} : v \in V, i \in [B, L]\}$, we need to define each $E'_{v,i}$.

Algorithm 2 SampleAndDecompose(V,W,ε,n,π)

```
$N \leftarrow |V|$
if $N \leq \log n/\log\log n$ then
    return trivial partition $\{V\}$
end if
$E \leftarrow$ random subset of $O(\varepsilon^{-4}\log n)$ elements from $\binom{V}{2}$ (with repetitions)
$C \leftarrow C_E(\pi,V,W)$        ($C$ is an additive $O(\varepsilon^2 N^2)$ approximation of $C$ w.p. $\geq 1-n^{-4}$)
if $C = \Omega(\varepsilon^2 N^2)$ then
    return trivial partition $\{V\}$
end if
$\pi_1 \leftarrow$ ApproxLocalImprove$(V,W,\pi,\varepsilon,n)$
$k \leftarrow$ random integer in the range $[N/3, 2N/3]$
$V_L \leftarrow \{v \in V : \rho_\pi(v) \leq k\}$, $\pi_L \leftarrow$ restriction of $\pi_1$ to $V_L$
$V_R \leftarrow V \setminus V_L$,                 $\pi_R \leftarrow$ restriction of $\pi_1$ to $V_R$
return concatenation of decomposition SampleAndDecompose$(V_L,W,\varepsilon,n,\pi_L)$ and decomposition SampleAndDecompose$(V_R,W,\varepsilon,n,\pi_R)$
```

Algorithm 3 ApproxLocalImprove(V,W,π,ε,n) (Note: π used as both input and output)

```
$N \leftarrow |V|$, $B \leftarrow \lceil\log(\Theta(\varepsilon N/\log n)\rceil$, $L \leftarrow \lceil\log N\rceil$
if $N = O(\varepsilon^{-3}\log^3 n)$ then
    return
end if
for $v \in V$ do
    $r \leftarrow \rho_\pi(v)$
    for $i = B \ldots L$ do
        $E_{v,i} \leftarrow \emptyset$
        for $m = 1..\Theta(\varepsilon^{-2}\log^2 n)$ do
            $j \leftarrow$ integer uniformly at random chosen from $[\max\{1, r-2^i\}, \min\{n, r+2^i\}]$
            $E_{v,i} \leftarrow E_{v,i} \cup \{(v,\pi(j))\}$
        end for
    end for
end for
while $\exists u \in V$ and $j \in [n]$ s.t. (setting $\ell := \lceil\log|j-\rho_\pi(u)|\rceil$):

            $\ell \in [B,L]$ and TestMove$_{E_{u,\ell}}(\pi,V,W,u,j) > \varepsilon|j-\rho_\pi(u)|/\log n$

 do
    for $v \in V$ and $i \in [B,L]$ do
        refresh sample $E_{v,i}$ with respect to the move $u \to j$ (see Section 3.3)
    end for
    $\pi \leftarrow \pi_{u \to j}$
end while
```

Definition 16 *We say that $E_{v,i}$ is* interesting *in the context of π and $\pi_{u\to j}$ if the two sets T_1, T_2 defined as*

$$\begin{aligned} T_1 &= \{x \in V : |\rho_\pi(x) - \rho_\pi(v)| \le 2^i\} \\ T_2 &= \{x \in V : |\rho_{\pi_{u\to j}}(x) - \rho_{\pi_{u\to j}}(v)| \le 2^i\} \end{aligned}$$

differ.

We set $E'_{v,i} = E_{v,i}$ for all v, i for which $E_{v,i}$ is *not* interesting.

Observation 17 *There are at most $O(|\rho_\pi(u) - j|\log n)$ interesting choices of v, i. Additionally, if $v \ne u$, then for T_1, T_2 as in Definition 16, $|T_1 \Delta T_2| = O(1)$, where Δ denotes symmetric difference.*

Fix one interesting choice v, i. Let T_1, T_2 be as in Definition 16. By the last observation, each of T_1 and T_2 contains $O(1)$ elements that are not contained in the other. Assume $|T_1| = |T_2|$, let $X_1 = T_1 \setminus T_2$, and $X_2 = T_2 \setminus T_1$. Fix any injection $\alpha : X_1 \to X_2$, and extend $\alpha : T_1 \to T_2$ so that $\alpha(x) = x$ for all $x \in T_1 \cap T_2$. Finally, define

$$E'_{v,i} = \{(v, \alpha(x)) : (v, x) \in E_{v,i}\} . \tag{10}$$

(The case $|T_1| \ne |T_2|$ may occur due to the clipping of the ranges $[\rho_\pi(v) - 2^i, \rho_\pi(v) + 2^i]$ and $[\rho_{\pi_{u\to j}}(v) - 2^i, \rho_{\pi_{u\to j}}(v) + 2^i]$ to a smaller range. This is a simple technicality which may be taken care of by formally extending the set V by N additional elements $\tilde{v}_1^L, \dots, \tilde{v}_N^L$, extending the definition of ρ_π for all permutation π on V so that $\rho_\pi(\tilde{v}_a^L) = -a + 1$ for all a and similarly $N = |V|$ additional elements $\tilde{v}_1^R, \dots, \tilde{v}_N^R$ such that $\rho_\pi(\tilde{v}_a^R) = N + a$. Formally extend W so that $W(v, \tilde{v}_a^L) = W(\tilde{v}_a^L, v) = W(v, \tilde{v}_a^R) = W(\tilde{v}_a^R, v) = 0$ for all $v \in V$ and a. This eliminates the need for clipping ranges in line 10 in ApproxLocalImprove.)

Finally, for $v = u$ we create $E'_{v,i}$ from scratch by repeating the loop in line 7 for that v.

It is easy to see that (9) holds. We need, however, something stronger that (9). Since our analysis assumes that $\mathcal{S} \sim \mathcal{D}_\pi$ is successful, we must be able to measure the distance (in total variation) between the random variable $(\mathcal{D}_\pi |$ success) defined by the process of drawing from $\mathcal{D}_\pi$ and conditioning on the result's success, and $\mathcal{D}_{\pi_{u\to j}}$. By Lemma 15, the total variation distance between $(\mathcal{D}_\pi |$ success) and $\mathcal{D}_{\pi_{u\to j}}$ is $O(n^{-4})$. Using a simple chain rule argument, we conclude the following:

Lemma 18 *Fix π^0 on V of size N, and fix $u_1, \dots, u_k \in V$ and $j_1, \dots, j_k \in [n]$. Consider the following process. We draw $\mathcal{S}^0$ from $\mathcal{D}_{\pi^0}$, and define*

$$\begin{gathered} \mathcal{S}^1 = \varphi_{u_1\to j_1}(\mathcal{S}^0), \mathcal{S}^2 = \varphi_{u_2\to j_2}(\mathcal{S}^1), \quad \cdots \quad , \mathcal{S}^k = \varphi_{u_k\to j_k}(\mathcal{S}^{k-1}) \\ \pi^1 = \pi^0_{u_1\to j_1}, \pi^2 = \pi^1_{u_2\to j_2}, \quad \cdots \quad , \pi^k = \pi^{k-1}_{u_k\to j_k} . \end{gathered}$$

Consider the random variable S^k conditioned on $S^0, S^1, \dots, S^{k-1}$ being successful for $\pi_0, \dots, \pi^{k-1}$, respectively. Then the total variation distance between the distribution of S^k and the distribution $\mathcal{D}_{\pi^k}$ is at most $O(kn^{-4})$.

3.4 Bounding the Query Complexity of Computing $\varphi_{u\to j}(\mathcal{S})$

We now need a notion of distance between $\mathcal{S}$ and $\mathcal{S}'$, measuring how many extra pairs were introduced ino the new sample family. These pairs may incur the cost of querying W. We denote this measure as $\mathrm{dist}(\mathcal{S},\mathcal{S}')$, and define it as $\mathrm{dist}(\mathcal{S},\mathcal{S}') := \left|\bigcup_{v,i} E_{v,i} \Delta E'_{v,i}\right|$.

Lemma 19 *Assume $\mathcal{S} \sim \mathcal{D}_\pi$ for some permutation π, and $\mathcal{S}' = \varphi_{u\to j}$. Then* $\mathbf{E}[\mathrm{dist}(\mathcal{S},\mathcal{S}')] = O(\varepsilon^{-3}\log^3 n)$.

Proof Denote $\mathcal{S} = \{E_{v,i}\}$ and $\mathcal{S}' = \{E'_{v,i}\}$. Fix some $v \neq u$. By construction, the sets $E_{v,i}$ for which $E_{v,i} \neq E'_{v,i}$ must be interesting, and there are at most $O(|\rho_\pi(u) - j|\log n)$ such, using Observation 17. Fix such a choice of v, i. By (10), $E_{v,i}$ will indeed differ from $E'_{v,i}$ only if it contains an element (v,x) for some $x \in T_1 \setminus T_2$. But the probability of that is at most

$$1 - (1 - O(2^{-i}))^{\Theta(\varepsilon^{-2}\log^2 n)} \leq 1 - e^{-\Theta(\varepsilon^{-2}2^{-i}\log^2 n)} = O(\varepsilon^{-2}2^{-i}\log^2 n)$$

(We used the fact that $i \geq B$, where B is as defined in line 1 of ApproxLocalImprove, and $N = \Omega(\varepsilon^{-3}\log^3 n)$ as guaranteed in line 3 of ApproxLocalImprove.) Therefore, the expected size of $E'_{v,i}\Delta E_{v,i}$ (counted with multiplicities) is $O(\varepsilon^{-2}2^{-i}\log^2 n)$.

Now consider all the interesting sets $E_{v_1.i_1}, \dots, E_{v_P,i_P}$. For each possible value i it is easy to see that there are at most $2|\rho_\pi(u) - j|$ p's for which $i_p = i$. Therefore,

$$\mathbf{E}\left[\sum_{p=1}^{P} |E'_{v_p,i_p}\Delta E_{v_p,i_p}|\right] = O\left(\varepsilon^{-2}|\rho_\pi(u) - j|\log^2 n \sum_{i=B}^{L} 2^{-i}\right) ,$$

where B, L are defined in line 1 in ApproxLocalImprove. Summing over $i \in [B, L]$, we get at most $O(\varepsilon^{-3}|\rho_\pi(u) - j|\log^3 n/N)$. For $v = u$, the set $\{E_{v,i}\}$ is drawn from scratch, clearly contributing $O(\varepsilon^{-2}\log^3 n)$ to $\mathrm{dist}(\mathcal{S},\mathcal{S}')$. The claim follows. ■

3.5 Analysis of SampleAndDecompose

Throughout the execution of the algorithm, various *high probability* events must occur in order for the algorithm guarantees to hold. Let $\mathcal{S}_1, \mathcal{S}_2, \dots$ denote the sample families that are given rise to through the executions of ApproxLocalImprove, either between lines 5 and 14, or as a mutation done between lines 15 and 20. We will need the first $\Theta(n^4)$ to be good approximations, based on Definition 14. Denote this favorable event $\mathcal{E}_1$. By Lemma 18, and using a union bound, with constant probability (say, 0.99) this happens. We also need the cost approximation C obtained in line 5 to be successful. Denote this favorable event $\mathcal{E}_2$. By Hoeffding tail bounds, this happens with probability $1 - O(n^{-4})$ for each execution of the line. This line is obviously executed at most $O(n\log n)$ times, and hence we can lower bound the probability of success of all executions by 0.99.

From now throughout, we make the following assumption, which is true by the above with probability at least 0.97.

Assumption 20 *Events $\mathcal{E}_1$ and $\mathcal{E}_2$ hold true.*

Note that by conditioning the remainder of our analysis on this assumption may bias some expectation upper bounds derived earlier and in what follows. This bias can multiply the estimates by at most $1/0.97$, which can be absorbed in the O-notation of these bounds.

Let π^* denote the optimal permutation for the root call to SampleAndDecompose with V,W,ε. The permutation π is, by Assumption 11, a constant factor approximation for MFAST on V,W. Using the triangle inequality, we conclude that $d_\tau(\pi,\pi^*) \leq C(\pi,V,W)+C(\pi^*,V,W)$. Hence, $E[d_\tau(\pi,\pi^*)] = O(C(\pi^*,V,W))$. From this we conclude, using (4), that

$$E[d_{\text{foot}}(\pi,\pi^*)] = O(C(\pi^*,V,W)) \ .$$

Now consider the recursion tree $\mathcal{T}$ of SampleAndDecompose. Denote I the set of internal nodes, and by $\mathcal{L}$ the set of leaves (i.e., executions exiting from line 8). For a call SampleAndDecompose corresponding to a node X in the recursion tree, denote the input arguments by $(V_X,W,\varepsilon,n,\pi_X)$. Let $L[X],R[X]$ denote the left and right children of X respectively. Let k_X denote the integer k in 11 in the context of $X \in I$. Hence, by our definitions, $V_{L[X]},V_{R[X]},\pi_{L[X]}$ and $\pi_{R[X]}$ are precisely V_L,V_R,π_L,π_R from lines 12-13 in the context of node X.

Take, as in line $1, N_X = |V_X|$. Let π^*_X denote the optimal MFAST solution for instance $(V_X,W_{|V_X})$. By $\mathcal{E}_1$ we conclude that the first $\Theta(n^4)$ times in which we iterate through the while loop in ApproxLocalImprove (counted over all calls to ApproxLocalImprove), the cost of $\pi_{X\,u\to j}$ is an actual improvement compared to π_X (for the current value of π_X,u and j in iteration), and the improvement in cost is of magnitude at least $\Omega(\varepsilon|\rho_{\pi_X}(u)-j|/\log n)$, which is $\Omega(\varepsilon^2 N_X/\log^2 n)$ due to the use of B defined in line 1. But this means that the number of iterations of the while loop in line 15 of ApproxLocalImprove is

$$O(\varepsilon^{-2} C(\pi_X,V_X,W_{|V_X})\log^2 n/N_X) \ .$$

Indeed, otherwise the true cost of the running solution would go below 0. Since $C(\pi_X,V_X,W_{|V_X})$ is at most $\binom{N_X}{2}$, the number of iterations is hence at most $O(\varepsilon^{-2}N_X\log^2 n)$. By Lemma 19 the expected query complexity incurred by the call to ApproxLocalImprove is therefore $O(\varepsilon^{-5}N_X\log^5 n)$. Summing over the recursion tree, the total query complexity incurred by calls to ApproxLocalImprove is, on expectation, at most $O(\varepsilon^{-5}n\log^6 n)$.

Now consider the moment at which the while loop of ApproxLocalImprove terminates. Let π_{1X} denote the permutation obtained at that point, returned to SampleAndDecompose in line 10. We classify the elements $v \in V_X$ to two families: V_X^{short} denotes all $u \in V_X$ s.t. $|\rho_{\pi_{1X}}(u)-\rho_{\pi^*_X}(u)| = O(\varepsilon N_X/\log n)$, and V_X^{long} denotes $V_X \setminus V_X^{\text{short}}$. We know, by assumption, that the last sample ensemble $\mathcal{S}$ used in ApproxLocalImprove was a good approximation, hence for all $u \in V_X^{\text{long}}$,

$$\text{TestMove}(\pi_{1X},V_X,W_{|V_X},u,\rho_{\pi^*_X}(u)) = O(\varepsilon|\rho_{\pi_{1X}}(u)-\rho_{\pi^*_X}(u)|/\log n). \tag{11}$$

Definition 21 *(Kenyon-Mathieu and Schudy, 2007) For $u \in V_X$, we say that u crosses k_X if the interval $[\rho_{\pi_{1X}}(u),\rho_{\pi^*_X}(u)]$ contains the integer k_X.*

Let V_X^{cross} denote the (random) set of elements $u \in V_X$ that cross k_X as chosen in line 11. We define a key quantity T_X as in Kenyon-Mathieu and Schudy (2007) as follows:

$$T_X = \sum_{u\in V_X^{\text{cross}}} \text{TestMove}(\pi_{1X},V_X,W_{|V_X},u,\rho_{\pi^*_X}(u)) \ .$$

Following (11), the elements $u \in V_X^{\text{long}}$ can contribute at most

$$O\left(\varepsilon \sum_{u\in V_X^{\text{long}}} |\rho_{\pi_{1X}}(u)-\rho_{\pi^*_X}(u)|/\log n\right)$$

to T_X. Hence the total contribution from such elements is, by definition,

$$O(\varepsilon d_{\text{foot}}(\pi_{1X}, \pi^*_X)/\log n)$$

which is, using (4) at most $O(\varepsilon d_\tau(\pi_{1X}, \pi^*_X)/\log n)$. Using the triangle inequality and the definition of π^*_X, the last expression, in turn, is at most $O(\varepsilon C(\pi_{1X}, V_X, W_{|V_X})/\log n)$.

We now bound the contribution of the elements $u \in V_X^{\text{short}}$ to T_X. The probability of each such element to cross k is $O(|\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)|/N_X)$. Hence, the total expected contribution of these elements to T_X is

$$O\left(\sum_{u \in V_X^{\text{short}}} |\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)|^2/N_X\right) . \tag{12}$$

Under the constraints $\sum_{u \in V_X^{\text{short}}} |\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)| \leq d_{\text{foot}}(\pi_{1X}, \pi^*_X)$ and $|\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)| = O(\varepsilon N_X/\log n)$, the maximal value of (12) is

$$O(d_{\text{foot}}(\pi_{1X}, \pi^*_X)\varepsilon N_X/(N_X \log n)) = O(d_{\text{foot}}(\pi_{1X}, \pi^*_X)\varepsilon/\log n) .$$

Again using (4) and the triangle inequality, the last expression is

$$O(\varepsilon C(\pi_{1X}, V_X, W_{|V_X})/\log n) .$$

Combining the accounting for V^{long} and V^{short}, we conclude

$$E_{k_X}[T_X] = O(\varepsilon C(\pi^*_X, V_X, W_{|V_X})/\log n) , \tag{13}$$

where the expectation is over the choice of k_X in line 11 of SampleAndDecompose.

We are now in a position to use a key Lemma by Kenyon-Mathieu and Schudy (2007). First we need a definition: Consider the optimal solution π'_X respecting $V_{L[X]}, V_R[X]$ in lines 12 and 13. By this we mean that π'_X must rank all of the elements in V_{XL} before (to the left of) V_{RX}. For the sake of brevity, let C^*_X be shorthand for $C(\pi^*_X, V_X, W_{|V_X})$ and C'_X for $C(\pi^{;}_X, V_X, W_{|V_X})$.

Lemma 22 *(Kenyon-Mathieu and Schudy, 2007) With respect to the distribution of the number k_X in line 11 of* SampleAndDecompose*,*

$$E[C'_X] \leq O\left(\frac{d_{\text{foot}}(\pi_{1X}, \pi^*_X)^{3/2}}{N_X}\right) + E[T_X] + C^*_X . \tag{14}$$

Using (4), we can replace $d_{\text{foot}}(\pi_{1X}, \pi^*_X)$ with $d_\tau(\pi_{1X}, \pi^*_X)$ in (14). Using the triangle inequality, we can then, in turn, replace $d_\tau(\pi_{1X}, \pi^*_X)$ with $C(\pi_{1X}, V_X, W_{|V_X})$.

3.6 Summing Over the Recursion Tree

Let us study the implication of (14) for our purpose. Recall that $\{V_1, \dots, V_k\}$ is the decomposition returned by SampleAndRank, where each V_i corresponds to a leaf in the recursion tree. Also recall that $\hat{\pi}$ denotes the minimizer of $C(\cdot, V, W)$ over all permutations in $\Pi(V_1, \dots, V_k)$ respecting the decomposition. Given Assumption 20 it suffices, for our purposes, to show that $\hat{\pi}$ is a (relative) small approximation for MFAST on V,W. Our analysis of this account is basically that of Kenyon-Mathieu and Schudy (2007), with slight changes stemming from bounds we derive on $E[T_X]$. We present the proof in full detail for the sake of completeness. Let RT denote the root node.

For $X \in I$, let β_X denote the contribution of the split $L[X], R[X]$ to the LHS of (3). More precisely,

$$\beta_X = \sum_{u \in L[X], v \in R[X]} \mathbf{1}_{W(v,u)=1} \ ,$$

so we get $\sum_{1 \le i < j \le k} \sum_{(u,v) \in V_i \times V_j} \mathbf{1}_{W(v,u)=1} = \sum_{X \in I} \beta_X$.

For any $X \in I$, note also that by our definitions $\beta_X = C'_X - C^*_{L[X]} - C^*_{R[X]}$. Hence, using Lemma 22 and the ensuing comment,

$$E[\beta_X] \le O\left(E\left[\frac{C(\pi_{1X}, V_X, W_{|V_X})^{3/2}}{N_X}\right]\right) + E[T_X] + E[C^*_X] - E[C^*_{L[X]}] - E[C^*_{R[X]}] \ ,$$

where the expectations are over the entire space of random decisions made by the algorithm execution. Summing the last inequality over $X \in I$, we get (minding the cancellations):

$$\begin{aligned} E\left[\sum_{X \in I} \beta_X\right] \ \le \ & O\left(\sum_{X \in I} E\left[\frac{C(\pi_{1X}, V_X, W_{|V_X})^{3/2}}{N_X}\right]\right) + E\left[\sum_{X \in I} T_X\right] + \\ & C^*_{\mathrm{RT}} - \sum_{X \in L} E[C^*_X] \ . \end{aligned} \tag{15}$$

The expression $E[\sum_{X \in I} T_X]$ is bounded by $O\left(E\left[\sum_{X \in I} \varepsilon \sum C^*_X / \log n\right]\right)$ using (13) (which depends on Assumption 20). Clearly the sum of C^*_X for X ranging over nodes $X \in I$ in a particular level is at most $C(\pi_{\mathrm{RT}}, V, W)$ (again using Assumption 20 to assert that the cost of π_{1X} is less than the cost of π_X at each node X). By taking Assumption 11 into account, $C(\pi_{\mathrm{RT}}, V, W)$ is $O(C^*_{\mathrm{RT}})$. Hence, summing over all $O(\log n)$ levels,

$$E\left[\sum_{X \in I} T_X\right] = O(\varepsilon C^*_{\mathrm{RT}}) \ .$$

Let $C_{1X} = C(\pi_{1X}, V_X, W_{|V_X})$ for all $x \in I$. Denote by F the expression in the O-notation of the first summand in the RHS of (15), more precisely:

$$F = \sum_{X \in I} E\left[\frac{C_{1X}^{3/2}}{N_X}\right] \ , \tag{16}$$

where we remind the reader that $N_X = |V_X|$. It will suffice to show that under Assumption 20, the following inequality holds with probability 1:

$$G((C_{1X})_{X \in I}, (N_X)_{X \in I}) := \sum_{X \in I} C_{1X}^{3/2} / N_X \le c_3 \varepsilon C_{1\mathrm{RT}} \ , \tag{17}$$

where $c_3 > 0$ is some global constant. This turns out to require a bit of elementary calculus. A complete proof of this assertion is not included in the extended abstract of Kenyon-Mathieu and Schudy (2007). We present a version of the proof here for the sake of completeness.

Under assumption 20, the following two constraints hold uniformly for all $X \in I$ with probability 1: Letting $C_X = C(\pi_X, V_X, W_{|V_X})$,

$(A1)$ If X is other than RT, let Y be its sibling and P their parent. In case $Y \in I$:

$$C_{1X} + C_{1Y} \le C_{1P} \ . \tag{18}$$

(In case $Y \in \mathcal{L}$, we simply have that $C_{1X} \le C_{1P}$.[6]) To see this, notice that $C_{1X} \le C_X$, and similarly, in case $Y \in I, C_{1Y} \le C_Y$. Clearly $C_X + C_Y \le C_{1P}$, because π_X, π_Y are simply restrictions of π_{1P} to disjoint blocks of V_P. The required inequality (18) is proven.

$(A2)$ $C_{1X} \le c_2 \varepsilon^2 N_X^2$ for some global $c_2 > 0$.

In order to show (17), we may increase the values C_{1X} for $X \ne \text{RT}$ in the following manner: Start with the root node. If it has no children, there is nothing to do because then $G = 0$. If it has only one child $X \in I$, continuously increase C_{1X} until either $C_{1X} = C_{1\text{RT}}$ (making $(A1)$ tight) or $C_{1X} = c_2\varepsilon^2 N_X^2$ (making $(A2)$ above tight). Then recurse on the subtree rooted by X. In case RT has two children $X, Y \in I$ (say, X on left), continuously increase C_{1X} until either $C_{1X} + C_{1Y} = C_{1\text{RT}}$ ($(A1)$ tight) or until $C_{1X} = c_2\varepsilon^2 N_X^2$ ($(A2)$ tight) . Then do the same for C_{1Y}, namely, increase it until $(A1)$ is tight or until $C_{1Y} = c_2\varepsilon^2 N_Y^2$ ($(A2)$ tight). Recursively perform the same procedure for the subtrees rooted by X, Y.

After performing the above procedure, let I_1 denote the set of internal nodes X for which $(A1)$ is tight, namely, either the sibling Y of X is a leaf and $C_{1X} = C_{1P}$ (where P is X's parent) or the sibling $Y \in I$ and $C_{1X} + C_{1Y} = C_{1P}$ (in which case also $Y \in I_1$). Let $I_2 = I \setminus I_1$. By our construction, for all $X \in I_2, C_{1X} = c_2\varepsilon^2 N_X^2$.

Note that if $X \in I_2$ then its children (more precisely, those in I) cannot be in I_1. Indeed, this would violate $(A2)$ for at least one child, in virtue of the fact that N_Y lies in the range $[N_X/3, 2N_X/3]$ for any child Y of X. Hence, the set $I_1 \cup \{\text{RT}\}$ forms a connected subtree which we denote by $\mathcal{T}_1$. Let $P \in \mathcal{T}_1$ be an internal node in $\mathcal{T}_1$. Assume it has one child in $\mathcal{T}_1$, call it X. Then $C_{1X} = C_{1P}$ and in virtue of $N_X \le 2N_P/3$ we have $C_{1P}^{3/2}/N_P \le (2/3)^{3/2} C_{1X}^{3/2}/N_X$. Now assume P has two children $X, Y \in \mathcal{T}_1$. Then $C_{1X} + C_{1Y} = C_{1P}$. Using elementary calculus, we also have that $C_{1P}^{3/2}/N_P \le (C_{1X}^{3/2}/N_X + C_{1Y}^{3/2}/N_Y)/\sqrt{2}$ (indeed, the extreme case occurs for $N_X = N_Y = N_P/2$ and $C_{1X} = C_{1Y} = C_{1P}/2$). We conclude that for any P internal in $\mathcal{T}_1$, the corresponding contribution $C_{1P}^{3/2}/N_P$ to G is geometrically dominated by that of its children in I_1. Hence the entire sum $G_1 = \sum_{X \in I_1 \cup \{\text{RT}\}} C_{1X}^{3/2}/N_X$ is bounded by $c_4 \sum_{X \in \mathcal{L}_1} C_{1X}^{3/2}/N_X$ for some constant c_4, where $\mathcal{L}_1$ is the set of leaves of $\mathcal{T}_1$. For each such leaf $X \in \mathcal{L}_1$, we have that $C_{1X}^{3/2}/N_X \le c_2^{3/2}\varepsilon C_{1X}$ (using $(A2)$), hence $\sum_{X \in \mathcal{L}_1} C_{1X}^{3/2}/N_X \le \sum_{X \in \mathcal{L}_1} c_2^{3/2}\varepsilon C_{1X} \le c_2^{3/2}\varepsilon C_{1R}$ (the rightmost inequality in the chain follows from $\{V_X\}_{X \in \mathcal{L}_1}$ forming a disjoint cover of $V = V_{\text{RT}}$, together with (A_1)). We conclude that $G_1 \le c_4 c_2^{3/2} \varepsilon C_{1R}$.

To conclude (17), it remains to show that $G_2 = G - G_1 = \sum_{X \in I_2}$. For $P \in G_2$, clearly $C_{1P}^{3/2}/N_P = c_2^{3/2}\varepsilon^3 N_P^3$. Hence, if $X, Y \in G_2$ are children of P in I_2 then $C_{1P}^{3/2}/N_P \ge c_5 C_{1X}^{3/2}/N_X + C_{1Y}^{3/2}/N_Y$ and if X is the unique child of P in I_2, then $C_{1P}^{3/2}/N_P \ge c_5 C_{1X}^{3/2}/N_X$, for some global $c_5 > 1$. In other words, the contribution to G_2 corresponding to P geometrically dominates the sum of the corresponding contributions of its children. We conclude that G_2 is at most some constant c_6 times $\sum_{X \in \text{root}(I_2)} C_{1X}^{3/2}/N_X$, where $\text{root}(I_2)$ is the set of roots of the forest induced by I_2. As before, it is clear that $\{V_X\}_{X \in \text{root}(I_2)}$ is a disjoint collection, hence as before we conclude that $G_2 \le c_7 \varepsilon C_{1R}$ for some global $c_7 > 0$. The assertion (17) follows, and hence (16).

[6] We can say something stronger in this case, but we won't need it here.

Plugging our bounds in (15), we conclude that

$$E\left[\sum_{X\in I}\beta_X\right] \le C^*_{\mathrm{RT}}(1+O(\varepsilon)) - \sum_{X\in\mathcal{L}} E[C^*_X]\ .$$

Clearly $C(\hat{\pi},V,W) = \sum_{X\in I}\beta_X + \sum_{X\in\mathcal{L}} C^*_X$. Hence

$$E[C(\hat{\pi},V,W)] = (1+O(\varepsilon))C^*_{\mathrm{RT}} = (1+O(\varepsilon))C^*\ .$$

We conclude the desired assertion on expectation. Assumption 20, together with a simple counting of accesses to W gives our main result, Theorem 7, as a simple corollary. A simple counting of accesses to W proves Theorem 7.

4. Using Our Decomposition as a Preconditioner for SVM

We consider the following practical scenario, which is can be viewed as an improvement over a version of the well known SVMrank (Joachims, 2002; Herbrich et al., 2000) for the preference label scenario.

Consider the setting developed in Section 2.1, where each element u in V is endowed with a feature vector $\varphi(u) \in \mathbb{R}^d$ for some d (we can also use infinite dimensional spaces via kernels, but the effective dimension is never more than $n = |V|$). Assume, additionally, that $\|\phi(u)\|_2 \le 1$ for all $u \in V$ (otherwise, normalize). Our hypothesis class $\mathcal{H}$ is parametrized by a weight vector $w \in \mathbb{R}^d$, and each associated permutation π_w is obtained by sorting the elements of V in decreasing order of a score given by $\mathrm{score}_w(u) = \langle\varphi(u), w\rangle$. In other words, $u \prec_{\pi_w} v$ if $\mathrm{score}_w(u) > \mathrm{score}_w(v)$ (in case of ties, assume any arbitrary tie breaking scheme).

The following SVM formulation is a convex relaxation for the problem of optimizing $C(h,V,W)$ over our chosen concept class $\mathcal{H}$:

$$\begin{array}{lrl}
\text{(SVM1)} & \text{minimize} & F_1(w,\xi) = \sum_{u,v}\xi_{u,v} \\
 & \text{s.t. } \forall u,v : W(u,v) = 1 & \mathrm{score}_w(u) - \mathrm{score}_w(v) \ge 1 - \xi_{u,v} \\
 & \forall u,v & \xi_{u,v} \ge 0 \\
 & & \|w\| \le c
\end{array}$$

Instead of optimizing (SVM1) directly, we make the following observation. An ε-good decomposition $V_1,\ldots,V_k$ gives rise to a surrogate learning problem over $\Pi(V_1,\ldots,V_k) \subseteq \Pi(V)$, such that optimizing over the restricted set does not compromise optimality over $\Pi(V)$ by more than a relative regret of ε (property (3)). In turn, optimizing over $\Pi(V_1,\ldots,V_k)$ can be done separately for each block V_i. A natural underlying SVM corresponding to this idea is captured as follows:

$$\begin{array}{lrl}
\text{(SVM2)} & \text{minimize} & F_2(w,\xi) = \sum_{u,v\in\Delta_1\cup\Delta_2}\xi_{u,v} \\
 & \text{s.t. } \forall (u,v) \in \Delta_1\cup\Delta_2 & \mathrm{score}_w(u) - \mathrm{score}_w(v) \ge 1 - \xi_{u,v} \\
 & \forall u,v & \xi_{u,v} \ge 0 \\
 & & \|w\| \le c\ ,
\end{array}$$

where $\Delta_1 = \bigcup_{1\le i<j\le k} V_i \times V_j$ and $\Delta_2 = \bigcup_{i=1}^{k}\{(u,v) : u,v \in V_i \wedge W(u,v) = 1\}$.

Abusing notation, for $w \in \mathbb{R}^d$ s.t. $\|w\| \le c$, let $F_1(w)$ denote $\min F_1(w,\xi)$, where the minimum is taken over all ξ that satisfy the constraints of SVM1. Observe that $F_1(w)$ is simply $F_1(w,\xi)$, where ξ is taken as:

$$\xi_{u,v} = \begin{cases} \max\{0, 1-\mathrm{score}_w(u)+\mathrm{score}_w(v)\} & W(u,v)=1 \\ 0 & \text{otherwise} \end{cases} .$$

Similarly define $F_2(w)$ as the minimizer of $F_2(w,\xi)$, which is obtained by setting:

$$\xi_{u,v} = \begin{cases} \max\{0, 1-\mathrm{score}_w(u)+\mathrm{score}_w(v)\} & (u,v)\in\Delta_1\cup\Delta_2 \\ 0 & \text{otherwise} \end{cases} . \tag{19}$$

Let π^* denote the optimal solution to MFAST on V, W.

We do not know how to directly relate the optimal solution to SVM1 and that of SVM2. However, we can can replace SVM2 with a careful sampling of constraints thereof, such that (i) the solution to the subsampled SVM is optimal to within a relative error of ε as a solution to SVM2, and (ii) the sampling is such that only $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ queries to W are necessary in order to construct it. This result, which we quantify in what follows, strongly relies on the local chaos property of the ε-good decomposition (2) and some combinatorics on permutations.

Our subsampled SVM which we denote by SVM3, is obtained as follows. For ease of notation we assume that all blocks $V_1,\ldots,V_k$ are big in V, otherwise a simple accounting of small blocks needs to be taken care of, adding notational clutter. Let Δ_3 be a subsample of size M (chosen shortly) of Δ_2, each element chosen uniformly at random from Δ_2 (with repetitions - hence Δ_3 is a multi-set). Define:

$$\begin{array}{lrl} \text{(SVM3)} & \text{minimize} & F_3(w,\xi) = \sum_{u,v\in\Delta_1}\xi_{u,v} + \frac{\sum_{i=1}^{k}\binom{n_i}{2}}{M}\sum_{u,v\in\Delta_3}\xi_{u,v} \\ & \text{s.t. } \forall (u,v)\in\Delta_1\cup\Delta_3 & \mathrm{score}_w(u)-\mathrm{score}_w(v) \ge 1-\xi_{u,v} \\ & \forall u,v & \xi_{u,v}\ge 0 \\ & & \|w\|\le c \end{array}$$

As before, define $F_3(w)$ to be $F_3(w,\xi)$, where $\xi = \xi(w)$ is the minimizer of $F_3(w,\cdot)$ and is taken as

$$\xi_{u,v} = \begin{cases} \max\{0, 1-\mathrm{score}_w(u)+\mathrm{score}_w(v)\} & (u,v)\in\Delta_1\cup\Delta_3 \\ 0 & \text{otherwise} \end{cases} .$$

Our ultimate goal is to show that for quite small M, SVM3 is a good approximation of SVM2. To that end we first need another lemma.

Lemma 23 *Any feasible solution* (w,ξ) *for SVM1 satisfies* $\sum_{u,v}\xi_{u,v} \ge C(\pi^*, V, W)$.

Proof The following has been proven by Ailon et al. (2008a): Consider *non-transitive* triangles induced by W: These are triplets (u,v,y) of elements in V such that $W(u,v) = W(v,y) = W(y,u) = 1$. Note that any permutation must disagree with at least one pair of elements contained in a non-transitive triangle. Let T denote the set of non-transitive triangles. Now consider an assignment of non-negative weights β_t for each $t \in T$. We say that the weight system $\{\beta_t\}_{t\in T}$ *packs T* if for all $u,v \in V$ such that $W(u,v) = 1$, the sum $\sum_{(u,v) \text{ in } t} \beta_t$ is at most 1. (By *u,v in t* we mean that u,v are two of the three elements inducing t.) Let $\{\beta^*_t\}_{t\in T}$ be a weight system packing T with the maximum possible value of the sum of weights. Then

$$\sum_{t\in T} \beta^*_t \geq C(\pi^*,V,W)/3 \ . \tag{20}$$

Now consider one non-transitive triangle $t = (u,v,y) \in T$. We lower bound $\xi_{u,v} + \xi_{v,y} + \xi_{y,u}$ for any ξ such that w,ξ is a feasible solution to SVM1. Letting $a = \text{score}_w(u) - \text{score}_w(v), b = \text{score}_w(v) - \text{score}_w(y), c = \text{score}_w(y) - \text{score}_w(u)$, we get from the constraints in SVM1 that $\xi_{u,v} \geq 1-a, \xi_{v,y} \geq 1-b, \xi_{y,u} \geq 1-c$. But clearly $a+b+c = 0$, hence

$$\xi_{u,v} + \xi_{v,y} + \xi_{y,u} \geq 3 \ . \tag{21}$$

Now notice that the objective function of SVM1 can be bounded from below as follows:

$$\begin{aligned} \sum_{u,v} \xi_{u,v} &\geq \sum_{t=(u,v,y)\in T} \beta^*_t(\xi_{u,v} + \xi_{v,y} + \xi_{y,u}) \\ &\geq \sum_{t=(u,v,y)\in T} \beta^*_t \cdot 3 \\ &\geq C(\pi^*,V,W) \ . \end{aligned}$$

(The first inequality was due to the fact that $\{\beta^*_t\}_{t\in T}$ is a packing of the non-transitive triangles, hence the total weight corresponding to each pair u,v is at most 1. The second inequality is from (21) and the third is from (20).) This concludes the proof. ■

Theorem 24 *Let $\varepsilon \in (0,1)$ and $M = O(\varepsilon^{-6}(1+2c)^2 d \log(1/\varepsilon))$. Then with high constant probability, for all w such that $\|w\| \leq c$,*

$$|F_3(w) - F_2(w)| = O(\varepsilon F_2(w)) \ .$$

Proof Let $B_d(c) = \{z \in \mathbb{R}^d : \|z\| \leq c\}$. Fix a vector $w \in B_d(c)$. Over the random choice of Δ_3, it is clear that $E[F_3(w)] = F_2(w)$. We need a strong concentration bound. From the observation that $|\xi_{u,v}| \leq 1+2c$ for all u,v, we conclude (using Hoeffding bound) that for all $\mu > 0$,

$$\Pr[|F_3(w) - F_2(w)| \geq \mu] \leq \exp\left\{ \frac{-\mu^2 M}{\left(\sum_{i=1}^k \binom{n_i}{2}(1+2c)\right)^2} \right\} \ . \tag{22}$$

Let $\eta = \varepsilon^3$ and consider an η-net of vectors w in the ball $B_d(c)$. By this we mean a subset $\Gamma \subseteq B_d(c)$ such that for all $z \in B_d(c)$ there exists $w \in \Gamma$ s.t. $\|z - w\| \leq \eta$. Standard volumetric arguments imply that there exists such a set Γ of cardinality at most $(c/\eta)^d$.

Let $z \in \Gamma$ and $w \in B_d(c)$ such that $\|w - z\| \leq \eta$. From the definition of F_2, F_3, it is clear that

$$|F_2(w) - F_2(z)| \leq \sum_{i=1}^k \binom{n_i}{2} \varepsilon^3, \quad |F_3(w) - F_3(z)| \leq \sum_{i=1}^k \binom{n_i}{2} \varepsilon^3 \ . \tag{23}$$

Using (22), we conclude that for any $\mu > 0$, by taking $M = O(\mu^{-2}(\sum\binom{n_i}{2}))^2(1+2c)^2 d\log(c\eta^{-1}))$, with constant probability over the choice of Δ_3, uniformly for all $z \in \Gamma$:

$$|F_3(z) - F_2(z)| \leq \mu \ .$$

Take $\mu = \varepsilon^3 \sum_{i=1}^{k}\binom{n_i}{2}$. We conclude (plugging in our choice of μ and the definition of η) that by choosing

$$M = O(\varepsilon^{-6}(1+2c)^2 d\log(c/\varepsilon)) \ ,$$

with constant probability, uniformly for all $z \in \Gamma$:

$$|F_3(z) - F_2(z)| \leq \varepsilon^3 \sum_{i=1}^{k}\binom{n_i}{2} \ .$$

Using (23) and the triangle inequality, we conclude that for all $w \in B_d(c)$,

$$|F_3(w) - F_2(w)| \leq 3\varepsilon^3 \sum_{i=1}^{k}\binom{n_i}{2} \ . \tag{24}$$

By property (2) of the ε-goodness definition, (24) implies

$$|F_3(w) - F_2(w)| \leq 3\varepsilon \min_{\pi\in\Pi(V)} \sum_{i=1}^{k} C(\pi_{|V_i}, V_i, W_{|V_i}) = 3\varepsilon \sum_{i=1}^{k} \min_{\sigma\in\Pi(V_i)} C(\sigma, V_i, W_{|V_i}) \ .$$

By Lemma 23 applied separately in each block V_i, this implies

$$|F_3(w) - F_2(w)| \leq 3\varepsilon \sum_{i=1}^{k} \sum_{u,v\in V_i} \xi_{u,v} = 3\varepsilon F_2(w),$$

(where $\xi = \xi(w)$ is as defined in (19).) This concludes the proof. ■

5. Limitations and Future Work

Optimality. The exponent of ε^{-6} in Theorem 7 seems rather high, and it would be interesting to improve it. A better dependence of ε^{-4} has been recently claimed by Ailon et al. (2011). It would be interesting to find the correct bound.

Practicality. Our bounds are asymptotic, and our work calls for experimentation in order to determine in which cases our sampling technique beats uniform sampling.

Searching in natural permutation subspaces. Algorithm 1, which leads to our main result Theorem 7, is heavily based on dividing and conquering. This is also the main limitation of this work. To understand this limitation, consider the scenario of Section 4. There, the practitioner searches in the limited space of *linearly induced permutations*, namely, permutations induced by a linear functional applied to the features endowing the elements in V. It is not hard to conceive a scenario in which our divide and conquer step constrains the algorithm to search in a region of permutations that does not intersect this restricted search space. This, in fact, was the reason for our inability to relate between SVM1 and SVM2 (and its subsampled counterpart, SVM3). There is nothing special about linearly induced permutations, for this matter. In a recently studied scenario (Jamieson and Nowak, 2011), for example, one searches in the space of permutations induced by score functions computed as the distance from a fixed point from some metric space in which V is embedded. The same problem exists there as well: our sampling algorithm cannot be used to find almost optimal solutions within any restricted permutation subspace. Interestingly, the main result of Ailon et al. (2011), achieved while this work has been under review, has alleviated this problem using new techniques.

Acknowledgments

The author gratefully acknowledges the help of Warren Schudy with derivation of some of the bounds in this work. Special thanks to Ron Begleiter as well as to anonymous reviewers for helpful comments.

Appendix A. Linear VC Bound of Permutation Set

To see why the VC dimension of the set of permutations viewed as binary function over the set of all possible $\binom{n}{2}$ preferences, it is enough to show that any collection of n pairs of elements cannot be *shattered* by the set of permutation. (Refer to the definition of VC dimension by Vapnik and Chervonenkis (1971) for a definition of shattering). Indeed, any such collection must contain a cycle, and the set of permutations cannot direct a cycle cyclically.

Appendix B. Why The Disagreement Coefficient Does Not Help Here

We now show why a straightforward application of the disagreement coefficient (Hanneke, 2007) is not useful in our setting. The key idea of Hanneke (2007) is a definition of a measure of distance between concepts, equalling the volume of data points on which they disagree on. Using this measure, one then defines a ball $B_r(\pi)$ of radius r around a concept (a permutation) π, in an obvious way. The disagreement coefficient Θ is then defined as the smallest possible number bounding as a linear function Θr the volume of points on which the hypotheses in B_r are not unanimous on. Adopting this idea here, the underlying distance between hypotheses (permutations) is simply the Kendall-tau distance $d_\tau(\pi,\sigma)$ divided by $\binom{n}{2}$. We need to normalize this distance because Hanneke's work, as does most statistical machine learning work, assumes a probability measure on the space of instances (pairs of elements), while we used the counting measure for various reasons of simplicity. We define the normalized distance function as $\hat{d}_\tau(\pi,\sigma) = \binom{n}{2}^{-1} d_\tau(\pi,\sigma)$.

If we consider a ball $B_r(\pi)$ of radius $r > 2/n$ around some permutation π on V, then it is easy to see that there does not exist a pair of elements $u, v \in V$ on which $B_r(\pi)$ is unanimous on. Indeed, a simple swap of any two elements results in a permutation π' satisfying $\hat{d}_\tau(\pi,\pi') \leq 2/n$. This means that the disagreement coefficient, by definition, is it least $\Omega(n)$. Recall that the VC dimension of the space of permutations, viewed as $\binom{n}{2}$-dimensional binary preference vectors, is at most n. Plugging these bounds into the analysis of Hanneke (2007) of the famous A2 algorithm using the disagreement coefficient results in a sample complexity which is $\Omega(n^3)$ for any desired error rate. Clearly this is suboptimal because the number of pairs is only $O(n^2)$.

References

Nir Ailon and Mehryar Mohri. Preference based learning to rank. *Machine Learning*, 80:189–212, 2010.

Nir Ailon and Kira Radinsky. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Struct. Algorithms*, 31(3):371–383, 2007.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008a.

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, 2008b.

Nir Ailon, Ron Begleiter, and Esther Ezra. A new active learning scheme with applications to learning to rank from pairwise preferences. *arxiv:1110.2136*, 2011.

Miklos Ajtai, Vitaly Feldman, Avinatan Hassidim, and Jelani Nelson. Sorting and selection with imprecise comparisons. In *Automata, Languages and Programming*, volume 5555 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin / Heidelberg, 2009.

Noga Alon. Ranking tournaments. *SIAM J. Discret. Math.*, 20(1):137–142, 2006.

Dana Angluin. Queries revisited. *Theor. Comput. Sci.*, 313(2):175–194, 2004.

Les Atlas, David Cohn, Richard Ladner, Mohamed A. El-Sharkawi, and Robert J. Marks. Training connectionist networks with queries and selective sampling. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 566–573, 1990.

Les Atlas, David Cohn, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72 (1-2):139–153, 2008.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1):78–89, 2009.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Vaughan. The true sample complexity of active learning. *Machine Learning*, 80:111–139, 2010.

Yoram Baram, Ran El-Yaniv, and Kobi Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5:255–291, 2004.

Ron Begleiter, Ran El-Yaniv, and Dmitry Pechyony. Repairing self-confident active-transductive learners using systematic exploration. *Pattern Recognition Letters*, 29(9):1245–1251, 2008.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 268–276, Philadelphia, PA, USA, 2008.

Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Here or there: Preference judgments for relevance. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR)*, pages 16–27, 2008.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *Proceedings of the 10th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 451–457, 1998.

Koby Crammer and Yoram Singer. Pranking with ranking. In *Proceedings of the 14th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 641–647, 2001.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th Conference on Artificial Intelligence (AAAI)*, pages 746–751, 2005.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Proceedings of the 18th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 235–242, 2005.

Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Proceedings of the 21st Conference on Advances in Neural Information Processing Systems (NIPS)*, 2007.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

Persi Diaconis and Ronald L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):pp. 262–268, 1977.

Irit Dinur and Shmuel Safra. On the importance of being biased. In *Proceedings of the 34th Annual Symposium on the Theory of Computing (STOC)*, pages 33–42, 2002.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *J. Machine Learning Research*, 11:1605–1641, 2010.

Uri Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with unreliable information. In *Proceedings of the 22nd Annual Symposium on the Theory of Computing (STOC)*, pages 128–137, 2002.

Shai Fine, Ran Gilad-Bachrach, and Eli Shamir. Query by committee, linear separation and random walks. *Theoretical Computer Science*, 284(1):25–51, 2002.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

Eric J. Friedman. Active learning for smooth problems. *In Proceedings of the* 22^{nd} *Annual Conference on Learning Theory (COLT)*, 2009.

Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4): 1107–1138, 2007.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 353–360, 2007.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Advances in Large Margin Classifiers*, chapter 7 (Large Margin Rank Boundaries for Ordinal Regression), pages 115–132. MIT Press, 2000.

Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916, 2008.

Kevin G. Jamieson and Robert D. Nowak. Active ranking using pairwise comparisons. In *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2011.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the 39th Annual Symposium on the Theory of Computing (STOC)*, pages 95–103, 2007.

Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54:125–152, 2004.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 413–424, 2006.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 1081–1088, 2006.

Journal of Machine Learning Research 13 (2012) 165-202 Submitted 11/10, Revised 10/11; Published 1/12

Optimal Distributed Online Prediction Using Mini-Batches

Ofer Dekel OFERD@MICROSOFT.COM
Ran Gilad-Bachrach RANG@MICROSOFT.COM
Microsoft Research
1 Microsoft Way
Redmond, WA 98052, USA

Ohad Shamir OHADSH@MICROSOFT.COM
Microsoft Research
1 Memorial Drive
Cambridge, MA 02142, USA

Lin Xiao LIN.XIAO@MICROSOFT.COM
Microsoft Research
1 Microsoft Way
Redmond, WA 98052, USA

Editor: Tong Zhang

Abstract

Online prediction methods are typically presented as serial algorithms running on a single processor. However, in the age of web-scale prediction problems, it is increasingly common to encounter situations where a single processor cannot keep up with the high rate at which inputs arrive. In this work, we present the *distributed mini-batch* algorithm, a method of converting many serial gradient-based online prediction algorithms into distributed algorithms. We prove a regret bound for this method that is asymptotically optimal for smooth convex loss functions and stochastic inputs. Moreover, our analysis explicitly takes into account communication latencies between nodes in the distributed environment. We show how our method can be used to solve the closely-related distributed stochastic optimization problem, achieving an asymptotically linear speed-up over multiple processors. Finally, we demonstrate the merits of our approach on a web-scale online prediction problem.

Keywords: distributed computing, online learning, stochastic optimization, regret bounds, convex optimization

1. Introduction

Many natural prediction problems can be cast as stochastic online prediction problems. These are often discussed in the serial setting, where the computation takes place on a single processor. However, when the inputs arrive at a high rate and have to be processed in real time, there may be no choice but to distribute the computation across multiple cores or multiple cluster nodes. For example, modern search engines process thousands of queries a second, and indeed they are implemented as distributed algorithms that run in massive data-centers. In this paper, we focus on such *large-scale* and *high-rate* online prediction problems, where parallel and distributed computing is critical to providing a real-time service.

First, we begin by defining the stochastic online prediction problem. Suppose that we observe a stream of inputs $z_1, z_2, \ldots$, where each z_i is sampled independently from a fixed unknown distribution over a sample space $\mathcal{Z}$. Before observing each z_i, we predict a point w_i from a set W. After making the prediction w_i, we observe z_i and suffer the loss $f(w_i, z_i)$, where f is a predefined loss function. Then we use z_i to improve our prediction mechanism for the future (e.g., using a stochastic gradient method). The goal is to accumulate the smallest possible loss as we process the sequence of inputs. More specifically, we measure the quality of our predictions using the notion of *regret*, defined as

$$R(m) \;=\; \sum_{i=1}^{m} \left(f(w_i, z_i) - f(w^\star, z_i)\right) \;,$$

where $w^\star = \arg\min_{w \in W} \mathbb{E}_z[f(w,z)]$. Regret measures the difference between the cumulative loss of our predictions and the cumulative loss of the fixed predictor $w^\star$, which is optimal with respect to the underlying distribution. Since regret relies on the stochastic inputs z_i, it is a random variable. For simplicity, we focus on bounding the expected regret $\mathbb{E}[R(m)]$, and later use these results to obtain high-probability bounds on the actual regret. In this paper, we restrict our discussion to convex prediction problems, where the loss function $f(w,z)$ is convex in w for every $z \in \mathcal{Z}$, and W is a closed convex subset of $\mathbb{R}^n$.

Before continuing, we note that the stochastic online *prediction* problem is closely related, but not identical, to the stochastic *optimization* problem (see, e.g., Wets, 1989; Birge and Louveaux, 1997; Nemirovski et al., 2009). The main difference between the two is in their goals: in stochastic optimization, the goal is to generate a sequence $w_1, w_2, \ldots$ that quickly converges to the minimizer of the function $F(\cdot) = \mathbb{E}_z[f(\cdot, z)]$. The motivating application is usually a static (batch) problem, and not an online process that occurs over time. Large-scale static optimization problems can always be solved using a serial approach, at the cost of a longer running time. In online prediction, the goal is to generate a sequence of predictions that accumulates a small loss along the way, as measured by regret. The relevant motivating application here is providing a real-time service to users, so our algorithm must keep up with the inputs as they arrive, and we cannot choose to slow down. In this sense, distributed computing is critical for large-scale online prediction problems. Despite these important differences, our techniques and results can be readily adapted to the stochastic online optimization setting.

We model our distributed computing system as a set of k *nodes*, each of which is an independent processor, and a *network* that enables the nodes to communicate with each other. Each node receives an incoming stream of examples from an outside source, such as a load balancer/splitter. As in the real world, we assume that the network has a limited bandwidth, so the nodes cannot simply share all of their information, and that messages sent over the network incur a non-negligible latency. However, we assume that network operations are *non-blocking*, meaning that each node can continue processing incoming traffic while network operations complete in the background.

How well can we perform in such a distributed environment? At one extreme, an ideal (but unrealistic) solution to our problem is to run a serial algorithm on a single "super" processor that is k times faster than a standard node. This solution is optimal, simply because any distributed algorithm can be simulated on a fast-enough single processor. It is well-known that the optimal regret bound that can be achieved by a gradient-based serial algorithm on an arbitrary convex loss is $O(\sqrt{m})$ (e.g., Nemirovski and Yudin, 1983; Cesa-Bianchi and Lugosi, 2006; Abernethy et al., 2009). At the other extreme, a trivial solution to our problem is to have each node operate in isolation of the other $k-1$ nodes, running an independent copy of a serial algorithm, without any communication over the network. We call this the *no-communication* solution. The main disadvantage of this solution is that the performance guarantee, as measured by regret, scales poorly with the network size k. More specifically, assuming that each node processes m/k inputs, the expected regret per node is $O(\sqrt{m/k})$. Therefore, the total regret across all k nodes is $O(\sqrt{km})$ - namely, a factor of $\sqrt{k}$ worse than the ideal solution. The first sanity-check that any distributed online prediction algorithm must pass is that it outperforms the naïve no-communication solution.

In this paper, we present the *distributed mini-batch* (DMB) algorithm, a method of converting any serial gradient-based online prediction algorithm into a parallel or distributed algorithm. This method has two important properties:

- It can use any gradient-based update rule for serial online prediction as a black box, and convert it into a parallel or distributed online prediction algorithm.
- If the loss function $f(w,z)$ is smooth in w (see the precise definition in Equation (5)), then our method attains an asymptotically optimal regret bound of $O(\sqrt{m})$. Moreover, the coefficient of the dominant term $\sqrt{m}$ is the same as in the serial bound, and *independent* of k and of the network topology.

The idea of using mini-batches in stochastic and online learning is not new, and has been previously explored in both the serial and parallel settings (see, e.g., Shalev-Shwartz et al., 2007; Gimpel et al., 2010). However, to the best of our knowledge, our work is the first to use this idea to obtain such strong results in a parallel and distributed learning setting (see Section 7 for a comparison to related work).

Our results build on the fact that the optimal regret bound for serial stochastic gradient-based prediction algorithms can be refined if the loss function is smooth. In particular, it can be shown that the hidden coefficient in the $O(\sqrt{m})$ notation is proportional to the standard deviation of the stochastic gradients evaluated at each predictor w_i (Juditsky et al., 2011; Lan, 2009; Xiao, 2010). We make the key observation that this coefficient can be effectively reduced by averaging a mini-batch of stochastic gradients computed at the same predictor, and this can be done in parallel with simple network communication. However, the non-negligible communication latencies prevent a straightforward parallel implementation from obtaining the optimal serial regret bound.[1] In order to close the gap, we show that by letting the mini-batch size grow slowly with m, we can attain the optimal $O(\sqrt{m})$ regret bound, where the dominant term of order $\sqrt{m}$ is *independent* of the number of nodes k and of the latencies introduced by the network.

The paper is organized as follows. In Section 2, we present a template for stochastic gradient-based serial prediction algorithms, and state refined variance-based regret bounds for smooth loss functions. In Section 3, we analyze the effect of using mini-batches in the serial setting, and show that it does not significantly affect the regret bounds. In Section 4, we present the DMB algorithm, and show that it achieves an asymptotically optimal serial regret bound for smooth loss functions. In Section 5, we show that the DMB algorithm attains the optimal rate of convergence for stochastic optimization, with an asymptotically linear speed-up. In Section 6, we complement our theoretical results with an experimental study on a realistic web-scale online prediction problem. While substantiating the effectiveness of our approach, our empirical results also demonstrate some interesting

[1] For example, if the network communication operates over a minimum-depth spanning tree and the diameter of the network scales as $\log(k)$, then we can show that a straightforward implementation of the idea of parallel variance reduction leads to an $O\big(\sqrt{m\log(k)}\big)$ regret bound. See Section 4 for details.

Algorithm 1: Template for a serial first-order stochastic online prediction algorithm.

```
for $j = 1, 2, \ldots$ do
    predict $w_j$
    receive input $z_j$ sampled i.i.d. from unknown distribution
    suffer loss $f(w_j, z_j)$
    define $g_j = \nabla_w f(w_j, z_j)$
    compute $(w_{j+1}, a_{j+1}) = \phi(a_j, g_j, \alpha_j)$
end
```

properties of mini-batching that are not reflected in our theory. We conclude with a comparison of our methods to previous work in Section 7, and a discussion of potential extensions and future research in Section 8. The main topics presented in this paper are summarized in Dekel et al. (2011). Dekel et al. (2011) also present robust variants of our approach, which are resilient to failures and node heterogeneity in an asynchronous distributed environment.

2. Variance Bounds for Serial Algorithms

Before discussing distributed algorithms, we must fully understand the serial algorithms on which they are based. We focus on gradient-based optimization algorithms that follow the template outlined in Algorithm 1. In this template, each prediction is made by an unspecified *update rule*:

$$(w_{j+1}, a_{j+1}) = \phi(a_j, g_j, \alpha_j). \tag{1}$$

The update rule ϕ takes three arguments: an auxiliary state vector a_j that summarizes all of the necessary information about the past, a gradient g_j of the loss function $f(\cdot, z_j)$ evaluated at w_j, and an iteration-dependent parameter α_j such as a stepsize. The update rule outputs the next predictor $w_{j+1} \in W$ and a new auxiliary state vector a_{j+1}. Plugging in different update rules results in different online prediction algorithms. For simplicity, we assume for now that the update rules are deterministic functions of their inputs.

As concrete examples, we present two well-known update rules that fit the above template. The first is the *projected gradient descent* update rule,

$$w_{j+1} = \pi_W \left(w_j - \frac{1}{\alpha_j} g_j \right), \tag{2}$$

where π_W denotes the Euclidean projection onto the set W. Here $1/\alpha_j$ is a decaying learning rate, with α_j typically set to be $\Theta(\sqrt{j})$. This fits the template in Algorithm 1 by defining a_j to simply be w_j, and defining ϕ to correspond to the update rule specified in Equation (2). We note that the projected gradient method is a special case of the more general class of *mirror descent* algorithms (e.g., Nemirovski et al., 2009; Lan, 2009), which all fit in the template of Equation (1).

Another family of update rules that fit in our setting is the *dual averaging* method (Nesterov, 2009; Xiao, 2010). A dual averaging update rule takes the form

$$w_{j+1} = \arg\min_{w \in W} \left\{ \left\langle \sum_{i=1}^{j} g_i, w \right\rangle + \alpha_j h(w) \right\} , \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the vector inner product, $h : W \to \mathbb{R}$ is a strongly convex auxiliary function, and α_j is a monotonically increasing sequence of positive numbers, usually set to be $\Theta(\sqrt{j})$. The dual averaging update rule fits the template in Algorithm 1 by defining a_j to be $\sum_{i=1}^{j} g_i$. In the special case where $h(w) = (1/2)\|w\|_2^2$, the minimization problem in Equation (3) has the closed-form solution

$$w_{j+1} = \pi_W \left(-\frac{1}{\alpha_j} \sum_{i=1}^{j} g_j \right) . \tag{4}$$

For stochastic online prediction problems with convex loss functions, both of these update rules have expected regret bound of $O(\sqrt{m})$. In general, the coefficient of the dominant $\sqrt{m}$ term is proportional to an upper bound on the expected norm of the stochastic gradient (e.g., Zinkevich, 2003). Next we present refined bounds for smooth convex loss functions, which enable us to develop optimal distributed algorithms.

2.1 Optimal Regret Bounds for Smooth Loss Functions

As stated in the introduction, we assume that the loss function $f(w,z)$ is convex in w for each $z \in \mathcal{Z}$ and that W is a closed convex set. We use $\|\cdot\|$ to denote the Euclidean norm in $\mathbb{R}^n$. For convenience, we use the notation $F(w) = \mathbb{E}_z[f(w,z)]$ and assume $w^\star = \arg\min_{w \in W} F(w)$ always exists. Our main results require a couple of additional assumptions:

- *Smoothness* - we assume that f is L-smooth in its first argument, which means that for any $z \in \mathcal{Z}$, the function $f(\cdot,z)$ has L-Lipschitz continuous gradients. Formally,

$$\forall z \in \mathcal{Z}, \quad \forall w, w' \in W, \qquad \|\nabla_w f(w,z) - \nabla_w f(w',z)\| \le L\|w - w'\| . \tag{5}$$

- *Bounded Gradient Variance* - we assume that $\nabla_w f(w,z)$ has a σ^2-bounded variance for any fixed w, when z is sampled from the underlying distribution. In other words, we assume that there exists a constant $\sigma \ge 0$ such that

$$\forall w \in W, \qquad \mathbb{E}_z \left[\left\| \nabla_w f(w,z) - \nabla F(w)] \right\|^2 \right] \le \sigma^2 .$$

Using these assumptions, regret bounds that explicitly depend on the gradient variance can be established (Juditsky et al., 2011; Lan, 2009; Xiao, 2010). In particular, for the projected stochastic gradient method defined in Equation (2), we have the following result:

Theorem 1 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$ and assume that the stochastic gradient $\nabla_w f(w,z)$ has σ^2-bounded variance for all $w \in W$. In addition, assume that W is convex and bounded, and let $D = \sqrt{\max_{u,v \in W} \|u - v\|^2/2}$. Then using $\alpha_j = L + (\sigma/D)\sqrt{j}$ in Equation (2) gives*

$$\mathbb{E}[R(m)] \;\le\; \big(F(w_1) - F(w^\star)\big) + D^2 L + 2D\sigma\sqrt{m}.$$

In the above theorem, the assumption that W is a bounded set does not play a critical role. Even if the learning problem has no constraints on w, we could always confine the search to a bounded set (say, a Euclidean ball of some radius) and Theorem 1 guarantees an $O(\sqrt{m})$ regret compared to the optimum within that set.

Similarly, for the dual averaging method defined in Equation (3), we have:

Theorem 2 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$, assume that the stochastic gradient $\nabla_w f(w,z)$ has σ^2-bounded variance for all $w \in W$, and let $D = \sqrt{h(w^\star) - \min_{w\in W} h(w)}$. Then, by setting $w_1 = \arg\min_{w\in W} h(w)$ and $\alpha_j = L + (\sigma/D)\sqrt{j}$ in the dual averaging method we have*

$$\mathbb{E}[R(m)] \;\le\; \big(F(w_1) - F(w^\star)\big) + D^2 L + 2D\sigma\sqrt{m}.$$

For both of the above theorems, if $\nabla F(w^\star) = 0$ (which is certainly the case if $W = \mathbb{R}^n$), then the expected regret bounds can be simplified to

$$\mathbb{E}[R(m)] \;\le\; 2D^2 L + 2D\sigma\sqrt{m}\,. \tag{6}$$

Proofs for these two theorems, as well as the above simplification, are given in Appendix A. Although we focus on expected regret bounds here, our results can equally be stated as high-probability bounds on the actual regret (see Appendix B for details).

In both Theorem 1 and Theorem 2, the parameters α_j are functions of σ. It may be difficult to obtain precise estimates of the gradient variance in many concrete applications. However, note that any upper bound on the variance suffices for the theoretical results to hold, and identifying such a bound is often easier than precisely estimating the actual variance. A loose bound on the variance will increase the constants in our regret bounds, but will not change its qualitative $O(\sqrt{m})$ rate.

Euclidean gradient descent and dual averaging are not the only update rules that can be plugged into Algorithm 1. The analysis in Appendix A (and Appendix B) actually applies to a much larger class of update rules, which includes the family of mirror descent updates (Nemirovski et al., 2009; Lan, 2009) and the family of (non-Euclidean) dual averaging updates (Nesterov, 2009; Xiao, 2010). For each of these update rules, we get an expected regret bound that closely resembles the bound in Equation (6).

Similar results can also be established for loss functions of the form $f(w,z) + \Psi(w)$, where $\Psi(w)$ is a simple convex regularization term that is not necessarily smooth. For example, setting $\Psi(w) = \lambda\|w\|_1$ with $\lambda > 0$ promotes sparsity in the predictor w. To extend the dual averaging method, we can use the following update rule in Xiao (2010):

$$w_{j+1} = \arg\min_{w\in W} \left\{ \left\langle \frac{1}{j}\sum_{i=1}^{j} g_i,\, w \right\rangle + \Psi(w) + \frac{\alpha_j}{j} h(w) \right\}.$$

Similar extensions to the mirror descent method can be found in, for example, Duchi and Singer (2009). Using these composite forms of the algorithms, the same regret bounds as in Theorem 1 and Theorem 2 can be achieved even if $\Psi(w)$ is nonsmooth. The analysis is almost identical to Appendix A by using the general framework of Tseng (2008).

Asymptotically, the bounds we presented in this section are only controlled by the variance σ^2 and the number of iterations m. Therefore, we can think of any of the bounds mentioned above as an abstract function $\psi(\sigma^2, m)$, which we assume to be monotonically increasing in its arguments.

2.2 Analyzing the No-Communication Parallel Solution

Using the abstract notation $\psi(\sigma^2, m)$ for the expected regret bound simplifies our presentation significantly. As an example, we can easily give an analysis of the no-communication parallel solution described in the introduction.

Algorithm 2: Template for a serial mini-batch algorithm.

```
for $j = 1,2,\ldots$ do
    initialize $\bar{g}_j := 0$
    for $s = 1,\ldots,b$ do
        define $i := (j-1)b+s$
        predict $w_j$
        receive input $z_i$ sampled i.i.d. from unknown distribution
        suffer loss $f(w_j,z_i)$
        $g_i := \nabla_w f(w_j,z_i)$
        $\bar{g}_j := \bar{g}_j + (1/b)g_i$
    end
    set $(w_{j+1},a_{j+1}) = \phi\big(a_j,\bar{g}_j,\alpha_j\big)$
end
```

In the naïve no-communication solution, each of the k nodes in the parallel system applies the same serial update rule to its own substream of the high-rate inputs, and no communication takes place between them. If the total number of examples processed by the k nodes is m, then each node processes at most $\lceil m/k \rceil$ inputs. The examples received by each node are i.i.d. from the original distribution, with the same variance bound σ^2 for the stochastic gradients. Therefore, each node suffers an expected regret of at most $\psi(\sigma^2, \lceil m/k \rceil)$ on its portion of the input stream, and the total regret bound is obtain by simply summing over the k nodes, that is,

$$\mathbb{E}[R(m)] \ \le \ k\psi\left(\sigma^2, \left\lceil \frac{m}{k} \right\rceil\right).$$

If $\psi(\sigma^2,m) = 2D^2L + 2D\sigma\sqrt{m}$, as in Equation (6), then the expected total regret is

$$\mathbb{E}[R(m)] \ \le \ 2kD^2L + 2D\sigma k\sqrt{\left\lceil \frac{m}{k} \right\rceil}.$$

Comparing this bound to $2D^2L + 2D\sigma\sqrt{m}$ in the ideal serial solution, we see that it is approximately $\sqrt{k}$ times worse in its leading term. This is the price one pays for the lack of communication in the distributed system. In Section 4, we show how this $\sqrt{k}$ factor can be avoided by our DMB approach.

3. Serial Online Prediction using Mini-Batches

The expected regret bounds presented in the previous section depend on the variance of the stochastic gradients. The explicit dependency on the variance naturally suggests the idea of using averaged gradients over mini-batches to reduce the variance. Before we present the distributed mini-batch algorithm in the next section, we first analyze a *serial* mini-batch algorithm.

In the setting described in Algorithm 1, the update rule is applied after each input is received. We deviate from this setting and apply the update only periodically. Letting b be a user-defined *batch size* (a positive integer), and considering every b consecutive inputs as a *batch*. We define the *serial mini-batch algorithm* as follows: Our prediction remains constant for the duration of each batch, and is updated only when a batch ends. While processing the b inputs in batch j, the algorithm calculates and accumulates gradients and defines the average gradient

$$\bar{g}_j = \frac{1}{b}\sum_{s=1}^{b} \nabla_w f(w_j, z_{(j-1)b+s}) \; .$$

Hence, each batch of b inputs generates a single average gradient. Once a batch ends, the serial mini-batch algorithm feeds $\bar{g}_j$ to the update rule ϕ as the j^{th} gradient and obtains the new prediction for the next batch and the new state. See Algorithm 2 for a formal definition of the serial mini-batch algorithm. The appeal of the serial mini-batch setting is that the update rule is used less frequently, which may have computational benefits.

Theorem 3 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$ and assume that the stochastic gradient $\nabla_w f(w,z_i)$ has σ^2-bounded variance for all w. If the update rule ϕ has the serial regret bound $\psi(\sigma^2,m)$, then the expected regret of Algorithm 2 over m inputs is at most*

$$b\psi\left(\frac{\sigma^2}{b}, \left\lceil\frac{m}{b}\right\rceil\right).$$

If $\psi(\sigma^2,m) = 2D^2L + 2D\sigma\sqrt{m}$, then the expected regret is bounded by

$$2bD^2L + 2D\sigma\sqrt{m+b}.$$

Proof Assume without loss of generality that b divides m, and that the serial mini-batch algorithm processes exactly m/b complete batches.[2] Let $\mathcal{Z}^b$ denote the set of all sequences of b elements from $\mathcal{Z}$, and assume that a sequence is sampled from $\mathcal{Z}^b$ by sampling each element i.i.d. from $\mathcal{Z}$. Let $\bar{f} : W \times \mathcal{Z}^b \mapsto \mathbb{R}$ be defined as

$$\bar{f}(w,(z_1,\ldots,z_b)) \;=\; \frac{1}{b}\sum_{s=1}^{b} f(w,z_s) \; .$$

In other words, $\bar{f}$ averages the loss function f across b inputs from $\mathcal{Z}$, while keeping the prediction constant. It is straightforward to show that $\mathbb{E}_{\bar{z}\in\mathcal{Z}^b}\bar{f}(w,\bar{z}) = \mathbb{E}_{z\in\mathcal{Z}} f(w,z) = F(w)$.

Using the linearity of the gradient operator, we have

$$\nabla_w \bar{f}(w,(z_1,\ldots,z_b)) = \frac{1}{b}\sum_{s=1}^{b} \nabla_w f(w,z_s) \; .$$

Let $\bar{z}_j$ denote the sequence $(z_{(j-1)b+1},\ldots,z_{jb})$, namely, the sequence of b inputs in batch j. The vector $\bar{g}_j$ in Algorithm 2 is precisely the gradient of $\bar{f}(\cdot,\bar{z}_j)$ evaluated at w_j. Therefore the serial mini-batch algorithm is equivalent to using the update rule ϕ with the loss function $\bar{f}$.

Next we check the properties of $\bar{f}(w,\bar{z})$ against the two assumptions in Section 2.1. First, if f is L-smooth then $\bar{f}$ is L-smooth as well due to the triangle inequality. Then we analyze the variance of the stochastic gradient. Using the properties of the Euclidean norm, we can write

$$\begin{aligned} \left\|\nabla_w \bar{f}(w,\bar{z}) - \nabla F(w)\right\|^2 &= \left\|\frac{1}{b}\sum_{s=1}^{b}\left(\nabla_w f(w,z_s) - \nabla F(w)\right)\right\|^2 \\ &= \frac{1}{b^2}\sum_{s=1}^{b}\sum_{s'=1}^{b}\Big\langle \nabla_w f(w,z_s) - \nabla F(w), \nabla_w f(w,z_{s'}) - \nabla F(w)\Big\rangle. \end{aligned}$$

[2] We can make this assumption since if b does not divide m then we can pad the input sequence with additional inputs until $m/b = \lceil m/b \rceil$, and the expected regret can only increase.

Notice that z_s and $z_{s'}$ are independent whenever $s \neq s'$, and in such cases,

$$\begin{aligned}&\mathbb{E}\Big\langle \nabla_w f(w,z_s) - \nabla F(w), \nabla_w f(w,z_{s'}) - \nabla F(w)\Big\rangle \\ = \quad &\Big\langle \mathbb{E}\big[\nabla_w f(w,z_s) - \nabla F(w)\big],\, \mathbb{E}\big[\nabla_w f(w,z_{s'}) - \nabla F(w)\big]\Big\rangle \;=\; 0.\end{aligned}$$

Therefore, we have for every $w \in W$,

$$\mathbb{E}\left\|\nabla_w \bar{f}(w,\bar{z}) - \nabla F(w)\right\|^2 \;=\; \frac{1}{b^2}\sum_{s=1}^{b} \mathbb{E}\left\|\left(\nabla_w f(w,z_s) - \nabla F(w)\right)\right\|^2 \;\leq\; \frac{\sigma^2}{b}. \tag{7}$$

So we conclude that $\nabla_w \bar{f}(w,\bar{z}_j)$ has a (σ^2/b)-bounded variance for each j and each $w \in W$. If the update rule ϕ has a regret bound $\psi(\sigma^2,m)$ for the loss function f over m inputs, then its regret for $\bar{f}$ over m/b batches is bounded as

$$\mathbb{E}\left[\sum_{j=1}^{m/b}\left(\bar{f}(w_j,\bar{z}_j) - \bar{f}(w^\star,\bar{z}_j)\right)\right] \;\leq\; \psi\left(\frac{\sigma^2}{b},\frac{m}{b}\right).$$

By replacing $\bar{f}$ above with its definition, and multiplying both sides of the above inequality by b, we have

$$\mathbb{E}\left[\sum_{j=1}^{m/b}\sum_{i=(j-1)b+1}^{jb}\left(f(w_j,z_i) - f(w^\star,z_i)\right)\right] \;\leq\; b\psi\left(\frac{\sigma^2}{b},\frac{m}{b}\right).$$

If $\psi(\sigma^2,m) = 2D^2L + 2D\sigma\sqrt{m}$, then simply plugging in the general bound $b\psi({\sigma^2}/{b}, \lceil {m}/{b}\rceil)$ and using $\lceil {m}/{b}\rceil \leq {m}/{b} + 1$ gives the desired result. However, we note that the optimal algorithmic parameters, as specified in Theorem 1 and Theorem 2, must be changed to $\alpha_j = L + ({\sigma}/{\sqrt{b}D})\sqrt{j}$ to reflect the reduced variance σ^2/b in the mini-batch setting. ■

The bound in Theorem 3 is asymptotically equivalent to the $2D^2L + 2D\sigma\sqrt{m}$ regret bound for the basic serial algorithms presented in Section 2. In other words, performing the mini-batch update in the serial setting does not significantly hurt the performance of the update rule. On the other hand, it is also not surprising that using mini-batches in the serial setting does not improve the regret bound. After all, it is still a serial algorithm, and the bounds we presented in Section 2.1 are optimal. Nevertheless, our experiments demonstrate that in real-world scenarios, mini-batching can in fact have a very substantial positive effect on the transient performance of the online prediction algorithm, even in the serial setting (see Section 6 for details). Such positive effects are not captured by our asymptotic, worst-case analysis.

4. Distributed Mini-Batch for Stochastic Online Prediction

In this section, we show that in a distributed setting, the mini-batch idea can be exploited to obtain nearly optimal regret bounds. To make our setting as realistic as possible, we assume that any communication over the network incurs a latency. More specifically, we view the network as an undirected graph $\mathcal{G}$ over the set of nodes, where each edge represents a bi-directional network link. If nodes u and v are not connected by a link, then any communication between them must be relayed through other nodes. The latency incurred between u and v is therefore proportional to the graph distance between them, and the longest possible latency is thus proportional to the diameter of $\mathcal{G}$.

In addition to latency, we assume that the network has limited bandwidth. However, we would like to avoid the tedious discussion of data representation, compression schemes, error correcting, packet sizes, etc. Therefore, we do not explicitly quantify the bandwidth of the network. Instead, we require that the communication load at each node remains constant, and does not grow with the number of nodes k or with the rate at which the incoming functions arrive.

Although we are free to use any communication model that respects the constraints of our network, we assume only the availability of a distributed vector-sum operation. This is a standard[3] synchronized network operation. Each vector-sum operation begins with each node holding a vector v_j, and ends with each node holding the sum $\sum_{j=1}^{k} v_j$. This operation transmits messages along a rooted minimum-depth spanning-tree of $\mathcal{G}$, which we denote by $\mathcal{T}$: first the leaves of $\mathcal{T}$ send their vectors to their parents; each parent sums the vectors received from his children and adds his own vector; the parent then sends the result to his own parent, and so forth; ultimately the sum of all vectors reaches the tree root; finally, the root broadcasts the overall sum down the tree to all of the nodes.

An elegant property of the vector-sum operation is that it uses each up-link and each down-link in $\mathcal{T}$ exactly once. This allows us to start vector-sum operations back-to-back. These vector-sum operations will run concurrently without creating network congestion on any edge of $\mathcal{T}$. Furthermore, we assume that the network operations are *non-blocking*, meaning that each node can continue processing incoming inputs while the vector-sum operation takes place in the background. This is a key property that allows us to efficiently deal with network latency. To formalize how latency affects the performance of our algorithm, let μ denote the number of inputs that are processed by the entire system during the period of time it takes to complete a vector-sum operation across the entire network. Usually μ scales linearly with the diameter of the network, or (for appropriate network architectures) logarithmically in the number of nodes k.

4.1 The DMB Algorithm

We are now ready to present a general technique for applying a deterministic update rule ϕ in a distributed environment. This technique resembles the serial mini-batch technique described earlier, and is therefore called the *distributed mini-batch* algorithm, or DMB for short.

Algorithm 3 describes a template of the DMB algorithm that runs in parallel on each node in the network, and Figure 1 illustrates the overall algorithm work-flow. Again, let b be a batch size, which we will specify later on, and for simplicity assume that k divides b and μ. The DMB algorithm processes the input stream in batches $j = 1, 2, \ldots$, where each batch contains $b + \mu$ consecutive inputs. During each batch j, all of the nodes use a common predictor w_j. While observing the first b inputs in a batch, the nodes calculate and accumulate the stochastic gradients of the loss function f at w_j. Once the nodes have accumulated b gradients altogether, they start a distributed vector-sum operation to calculate the sum of these b gradients. While the vector-sum operation completes in the background, μ additional inputs arrive (roughly μ/k per node) and the system keeps processing them using the same predictor w_j. The gradients of these additional μ inputs are discarded (to this end, they do not need to be computed). Although this may seem wasteful, we show that this waste can be made negligible by choosing b appropriately.

[3] For example, all-reduce with the sum operation is a standard operation in MPI.

Algorithm 3: Distributed mini-batch (DMB) algorithm (running on each node).

```
for $j = 1, 2, \ldots$ do
    initialize $\hat{g}_j := 0$
    for $s = 1, \ldots, b/k$ do
        predict $w_j$
        receive input $z$ sampled i.i.d. from unknown distribution
        suffer loss $f(w_j, z)$
        compute $g := \nabla_w f(w_j, z)$
        $\hat{g}_j := \hat{g}_j + g$
    end
    call the distributed vector-sum to compute the sum of $\hat{g}_j$ across all nodes
    receive $\mu/k$ additional inputs and continue predicting using $w_j$
    finish vector-sum and compute average gradient $\bar{g}_j$ by dividing the sum by $b$
    set $(w_{j+1}, a_{j+1}) = \phi(a_j, \bar{g}_j, \alpha_j)$
end
```

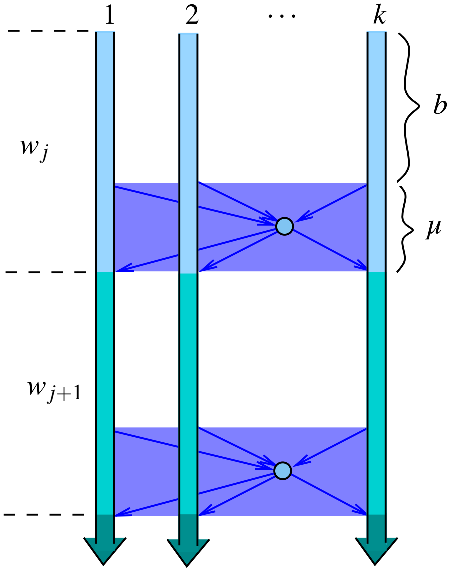


Figure 1: Work flow of the DMB algorithm. Within each batch $j = 1, 2, \ldots$, each node accumulates the stochastic gradients of the first b/k inputs. Then a vector-sum operation across the network is used to compute the average across all nodes. While the vector-sum operation completes in the background, a total of μ inputs are processed by the processors using the same predictor w_j, but their gradients are not collected. Once all of the nodes have the overall average $\bar{g}_j$, each node updates the predictor using the same deterministic serial algorithm.

Once the vector-sum operation completes, each node holds the sum of the b gradients collected during batch j. Each node divides this sum by b and obtains the average gradient, which we denote by $\bar{g}_j$. Each node feeds this average gradient to the update rule ϕ, which returns a new synchronized prediction w_{j+1}. In summary, during batch j each node processes $(b+\mu)/k$ inputs using the same predictor w_j, but only the first b/k gradients are used to compute the next predictor. Nevertheless, all $b+\mu$ inputs are counted in our regret calculation.

If the network operations are conducted over a spanning tree, then an obvious variants of the DMB algorithm is to let the root apply the update rule to get the next predictor, and then broadcast it to all other nodes. This saves repeated executions of the update rule at each node (but requires interruption or modification of the standard vector-sum operations in the network communication model). Moreover, this guarantees all the nodes having the same predictor even with update rules that depends on some random bits.

Theorem 4 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$ and assume that the stochastic gradient $\nabla_w f(w,z_i)$ has σ^2-bounded variance for all $w \in W$. If the update rule ϕ has the serial regret bound $\psi(\sigma^2,m)$, then the expected regret of Algorithm 3 over m samples is at most*

$$(b+\mu)\,\psi\left(\frac{\sigma^2}{b},\left\lceil\frac{m}{b+\mu}\right\rceil\right) .$$

Specifically, if $\psi(\sigma^2,m) = 2D^2L + 2D\sigma\sqrt{m}$, then setting the batch size $b = m^{1/3}$ gives the expected regret bound

$$2D\sigma\sqrt{m} + 2Dm^{1/3}\left(LD+\sigma\sqrt{\mu}\right) + 2D\sigma m^{1/6} + 2D\sigma\mu m^{-1/6} + 2\mu D^2L. \tag{8}$$

In fact, if $b = m^\rho$ for any $\rho \in (0,1/2)$, the expected regret bound is $2D\sigma\sqrt{m} + o(\sqrt{m})$.

To appreciate the power of this result, we compare the specific bound in Equation (8) with the ideal serial solution and the naïve no-communication solution discussed in the introduction. It is clear that our bound is asymptotically equivalent to the ideal serial bound $\psi(\sigma^2,m)$—even the constants in the dominant terms are identical. Our bound scales nicely with the network latency and the cluster size k, because μ (which usually scales logarithmically with k) does not appear in the dominant $\sqrt{m}$ term. On the other hand, the naïve no-communication solution has regret bounded by $k\psi\left(\sigma^2, {}^{m}\!/_{k}\right) = 2kD^2L + 2D\sigma\sqrt{km}$ (see Section 2.2). If $1 \ll k \ll m$, this bound is worse than the bound in Theorem 4 by a factor of $\sqrt{k}$.

Finally, we note that choosing b as m^ρ for an appropriate ρ requires knowledge of m in advance. However, this requirement can be relaxed by applying a standard doubling trick (Cesa-Bianchi and Lugosi, 2006). This gives a single algorithm that does not take m as input, with asymptotically similar regret. If we use a fixed b regardless of m, the dominant term of the regret bound becomes $2D\sigma\sqrt{\log(k)m/b}$; see the following proof for details.

Proof Similar to the proof of Theorem 3, we assume without loss of generality that k divides $b+\mu$, we define the function $\bar{f} : W \times \mathcal{Z}^b \mapsto \mathbb{R}$ as

$$\bar{f}(w,(z_1,\ldots,z_b)) \;=\; \frac{1}{b}\sum_{s=1}^{b} f(w,z_s) \ ,$$

and we use $\bar{z}_j$ to denote the *first b inputs* in batch j. By construction, the function $\bar{f}$ is L-smooth and its gradients have σ^2/b-bounded variance. The average gradient $\bar{g}_j$ computed by the DMB algorithm is the gradient of $\bar{f}(\cdot, \bar{z}_j)$ evaluated at the point w_j. Therefore,

$$\mathbb{E}\left[\sum_{j=1}^{m/(b+\mu)} \big(\bar{f}(w_j, \bar{z}_j) - \bar{f}(w^\star, \bar{z}_j)\big)\right] \le \psi\left(\frac{\sigma^2}{b}, \frac{m}{b+\mu}\right). \tag{9}$$

This inequality only involve the additional μ examples in counting the number of batches as $m/b+\mu$. In order to count them in the total regret, we notice that

$$\forall j, \quad \mathbb{E}\left[\bar{f}(w_j, \bar{z}_j) \,|\, w_j\right] = \mathbb{E}\left[\frac{1}{b+\mu} \sum_{i=(j-1)(b+\mu)+1}^{j(b+\mu)} f(w_j, z_i) \,\middle|\, w_j\right],$$

and a similar equality holds for $\bar{f}(w^\star, z_i)$. Substituting these equalities in the left-hand-side of Equation (9) and multiplying both sides by $b+\mu$ yields

$$\mathbb{E}\left[\sum_{j=1}^{m/(b+\mu)} \sum_{i=(j-1)(b+\mu)+1}^{j(b+\mu)} \big(f(w_j, z_i) - f(w^\star, z_i)\big)\right] \le (b+\mu)\psi\left(\frac{\sigma^2}{b}, \frac{m}{b+\mu}\right).$$

Again, if $(b+\mu)$ divides m, then the left-hand side above is exactly the expected regret of the DMB algorithm over m examples. Otherwise, the expected regret can only be smaller.

For the concrete case of $\psi(\sigma^2, m) = 2D^2L + 2D\sigma\sqrt{m}$, plugging in the new values for σ^2 and m results in a bound of the form

$$\begin{aligned} (b+\mu)\psi\left(\frac{\sigma^2}{b}, \left\lceil \frac{m}{b+\mu} \right\rceil\right) &\le (b+\mu)\psi\left(\frac{\sigma^2}{b}, \frac{m}{b+\mu} + 1\right) \\ &\le 2(b+\mu)D^2L + 2D\sigma\sqrt{m + \frac{\mu}{b}m + \frac{(b+\mu)^2}{b}}. \end{aligned}$$

Using the inequality $\sqrt{x+y+z} \le \sqrt{x} + \sqrt{y} + \sqrt{z}$, which holds for any nonnegative numbers x, y and z, we bound the expression above by

$$2(b+\mu)D^2L + 2D\sigma\sqrt{m} + 2D\sigma\sqrt{\frac{\mu m}{b}} + 2D\sigma\frac{b+\mu}{\sqrt{b}}.$$

It is clear that with $b = Cm^\rho$ for any $\rho \in (0, 1/2)$ and any constant $C > 0$, this bound can be written as $2D\sigma\sqrt{m} + o(\sqrt{m})$. Letting $b = m^{1/3}$ gives the smallest exponents in the $o(\sqrt{m})$ terms. ■

In the proofs of Theorem 3 and Theorem 4, decreasing the variance by a factor of b, as given in Equation (7), relies on properties of the Euclidean norm. For serial gradient-type algorithms that are specified with different norms (see the general framework in Appendix A), the variance does not typically decrease as much. For example, in the dual averaging method specified in Equation (3), if we use $h(w) = 1/(2(p-1))\|w\|_p^2$ for some $p \in (1, 2]$, then the "variance" bounds for the stochastic gradients must be expressed in the dual norm, that is, $\mathbb{E}\|\nabla_w f(w,z) - \nabla F(w)\|_q^2 \le \sigma^2$, where $q = p/(p-1) \in [2, \infty)$. In this case, the variance bound for the averaged function becomes

$$\mathbb{E}\left\|\nabla_w \bar{f}(w, \bar{z}) - \nabla F(w)\right\|_q^2 \le C(n,q)\frac{\sigma^2}{b},$$

where $C(n,q) = \min\{q-1, O(\log(n))\}$ is a space-dependent constant.[4] Nevertheless, we can still obtain a linear reduction in b even for such non-Euclidean norms. The net effect is that the regret bound for the DMB algorithm becomes $2D\sqrt{C(n,q)}\sigma\sqrt{m} + o(\sqrt{m})$.

4.2 Improving Performance on Short Input Streams

Theorem 4 presents an optimal way of choosing the batch size b, which results in an asymptotically optimal regret bound. However, our asymptotic approach hides a potential shortcoming that occurs when m is small. Say that we know, ahead of time, that the sequence length is $m = 15,000$. Moreover, say that the latency is $\mu = 100$, and that $\sigma = 1$ and $L = 1$. In this case, Theorem 4 determines that the optimal batch size is $b \sim 25$. In other words, for every 25 inputs that participate in the update, 100 inputs are discarded. This waste becomes negligible as b grows with m and does not affect our asymptotic analysis. However, if m is known to be small, we can take steps to improve the situation.

Assume for simplicity that b divides μ. Now, instead of running a single distributed mini-batch algorithm, we run $c = 1 + \mu/b$ independent interlaced instances of the distributed mini-batch algorithm on each node. At any given moment, $c-1$ instances are asleep and one instance is active. Once the active instance collects b/k gradients on each node, it starts a vector-sum network operation, awakens the next instance, and puts itself to sleep. Note that each instance awakens after $(c-1)b = \mu$ inputs, which is just in time for its vector-sum operation to complete.

In the setting described above, c different vector-sum operations propagate concurrently through the network. The distributed vector sum operation is typically designed such that each network link is used at most once in each direction, so concurrent sum operations that begin at different times should not compete for network resources. The batch size should indeed be set such that the generated traffic does not exceed the network bandwidth limit, but the latency of each sum operation should not be affected by the fact that multiple sum operations take place at once.

Simply interlacing c independent copies of our algorithm does not resolve the aforementioned problem, since each prediction is still defined by $1/c$ of the observed inputs. Therefore, instead of using the predictions prescribed by the individual online predictors, we use their average. Namely, we take the most recent prediction generated by each instance, average these predictions, and use this average in place of the original prediction.

The advantages of this modification are not apparent from our theoretical analysis. Each instance of the algorithm handles m/c inputs and suffers a regret of at most

$$b\psi\left(\frac{\sigma^2}{b}, 1 + \frac{m}{bc}\right) \quad ,$$

and, using Jensen's inequality, the overall regret using the average prediction is upper bounded by

$$bc\psi\left(\frac{\sigma^2}{b}, 1 + \frac{m}{bc}\right) \quad .$$

The bound above is precisely the same as the bound in Theorem 4. Despite this fact, we conjecture that this method will indeed improve empirical results when the batch size b is small compared to the latency term μ.

[4] For further details of algorithms using p-norm, see Xiao (2010, Section 7.2) and Shalev-Shwartz and Tewari (2011). For the derivation of $C(n,q)$ see for instance Lemma B.2 in Cotter et al. (2011).

5. Stochastic Optimization

As we discussed in the introduction, the *stochastic optimization* problem is closely related, but not identical, to the stochastic online prediction problem. In both cases, there is a loss function $f(w,z)$ to be minimized. The difference is in the way success is measured. In online prediction, success is measured by regret, which is the difference between the cumulative loss suffered by the prediction algorithm and the cumulative loss of the best fixed predictor. The goal of stochastic optimization is to find an approximate solution to the problem

$$\underset{w \in W}{\text{minimize}} \quad F(w) \triangleq \mathbb{E}_z[f(w,z)] ,$$

and success is measured by the difference between the expected loss of the final output of the optimization algorithm and the expected loss of the true minimizer $w^\star$. As before, we assume that the loss function $f(w,z)$ is convex in w for any $z \in \mathcal{Z}$, and that W is a closed convex set.

We consider the same *stochastic approximation* type of algorithms presented in Algorithm 1, and define the final output of the algorithm, after processing m i.i.d. samples, to be

$$\bar{w}_m = \frac{1}{m} \sum_{j=1}^{m} w_j .$$

In this case, the appropriate measure of success is the optimality gap

$$G(m) \ = \ F(\bar{w}_m) - F(w^\star) .$$

Notice that the optimality gap $G(m)$ is also a random variable, because $\bar{w}_m$ depends on the random samples $z_1, \ldots, z_m$. It can be shown (see, e.g., Xiao, 2010, Theorem 3) that for convex loss functions and i.i.d. inputs, we always have

$$\mathbb{E}[G(m)] \ \leq \ \frac{1}{m} \mathbb{E}[R(m)] .$$

Therefore, a bound on the expected optimality gap can be readily obtained from a bound on the expected regret of the same algorithm. In particular, if f is an L-smooth convex loss function and $\nabla_w f(w,z)$ has σ^2-bounded variance, and our algorithm has a regret bound of $\psi(\sigma^2, m)$, then it also has an expected optimality gap of at most

$$\bar{\psi}(\sigma^2, m) = \frac{1}{m} \psi(\sigma^2, m) .$$

For the specific regret bound $\psi(\sigma^2, m) = 2D^2L + 2D\sigma\sqrt{m}$, which holds for the serial algorithms presented in Section 2, we have

$$\mathbb{E}[G(m)] \ \leq \ \bar{\psi}(\sigma^2, m) \ = \ \frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}} .$$

5.1 Stochastic Optimization using Distributed Mini-Batches

Our template of a DMB algorithm for stochastic optimization (see Algorithm 4) is very similar to the one presented for the online prediction setting. The main difference is that we do not have to process inputs while waiting for the vector-sum network operation to complete. Again let b be the batch size, and the number of batches $r = \lfloor m/b \rfloor$. For simplicity of discussion, we assume that b divides m.

Algorithm 4: Template of DMB algorithm for stochastic optimization.

```
$r \leftarrow \lfloor \frac{m}{b} \rfloor$
for $j = 1, 2, \ldots, r$ do
    reset $\hat{g}_j = 0$
    for $s = 1, \ldots, b/k$ do
        receive input $z_s$ sampled i.i.d. from unknown distribution
        calculate $g_s = \nabla_w f(w_j, z_s)$
        calculate $\hat{g}_j \leftarrow \hat{g}_j + g_i$
    end
    start distributed vector sum to compute the sum of $\hat{g}_j$ across all nodes
    finish distributed vector sum and compute average gradient $\bar{g}_j$
    set $(w_{j+1}, a_{j+1}) = \phi(a_j, \bar{g}_j, j)$
end
Output: $\frac{1}{r}\sum_{j=1}^{r} w_j$
```

Theorem 5 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$ and assume that the stochastic gradient $\nabla_w f(w,z)$ has σ^2-bounded variance for all $w \in W$. If the update rule ϕ used in a serial setting has an expected optimality gap bounded by $\bar{\psi}(\sigma^2, m)$, then the expected optimality gap of Algorithm 4 after processing m samples is at most*

$$\bar{\psi}\left(\frac{\sigma^2}{b}, \frac{m}{b}\right) .$$

If $\bar{\psi}(\sigma^2, m) = \frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}}$, then the expected optimality gap is bounded by

$$\frac{2bD^2L}{m} + \frac{2D\sigma}{\sqrt{m}} .$$

The proof of the theorem follows along the lines of Theorem 3, and is omitted.

We comment that the accelerated stochastic gradient methods of Lan (2009), Hu et al. (2009) and Xiao (2010) can also fit in our template for the DMB algorithm, but with more sophisticated updating rules. These accelerated methods have an expected optimality bound of $\bar{\psi}(\sigma^2, m) = 4D^2L/m^2 + 4D\sigma/\sqrt{m}$, which translates into the following bound for the DMB algorithm:

$$\bar{\psi}\left(\frac{\sigma^2}{b}, \frac{m}{b}\right) = \frac{4b^2D^2L}{m^2} + \frac{4D\sigma}{\sqrt{m}} .$$

Most recently, Ghadimi and Lan (2010) developed accelerated stochastic gradient methods for strongly convex functions that have the convergence rate $\bar{\psi}(\sigma^2, m) = O(1)\left(L/m^2 + \sigma^2/\nu m\right)$, where ν is the strong convexity parameter of the loss function. The corresponding DMB algorithm has a convergence rate

$$\bar{\psi}\left(\frac{\sigma^2}{b}, \frac{m}{b}\right) = O(1)\left(\frac{b^2L}{m^2} + \frac{\sigma^2}{\nu m}\right).$$

Apparently, this also fits in the DMB algorithm nicely.

The significance of our result is that the dominating factor in the convergence rate is not affected by the batch size. Therefore, depending on the value of m, we can use large batch sizes without affecting the convergence rate in a significant way. Since we can run the workload associated with a single batch in parallel, this theorem shows that the mini-batch technique is capable of turning many serial optimization algorithms into parallel ones. To this end, it is important to analyze the speed-up of the parallel algorithms in terms of the running time (wall-clock time).

5.2 Parallel Speed-Up

Recall that k is the number of parallel computing nodes and m is the total number of i.i.d. samples to be processed. Let $b(m)$ be the batch size that depends on m. We define a *time-unit* to be the time it takes a single node to process one sample (including computing the gradient and updating the predictor). For convenience, let δ be the latency of the vector-sum operation in the network (measured in number of time-units).[5] Then the parallel speed-up of the DMB algorithm is

$$S(m) = \frac{m}{\frac{m}{b(m)}\left(\frac{b(m)}{k}+\delta\right)} = \frac{k}{1+\frac{\delta}{b(m)}k} ,$$

where $m/b(m)$ is the number of batches, and $b(m)/k+\delta$ is the wall-clock time by k processors to finish one batch in the DMB algorithm. If $b(m)$ increases at a fast enough rate, then we have $S(m) \to k$ as $m \to \infty$. Therefore, we obtain an asymptotically linear speed-up, which is the ideal result that one would hope for in parallelizing the optimization process (see Gustafson, 1988).

In the context of stochastic optimization, it is more appropriate to measure the speed-up with respect to the same optimality gap, not the same amount of samples processed. Let ε be a given target for the expected optimality gap. Let $m_{\mathrm{srl}}(\varepsilon)$ be the number of samples that the serial algorithm needs to reach this target and let $m_{\mathrm{DMB}}(\varepsilon)$ be the number of samples needed by the DMB algorithm. Slightly overloading our notation, we define the parallel speed-up with respect to the expected optimality gap ε as

$$S(\varepsilon) = \frac{m_{\mathrm{srl}}(\varepsilon)}{\frac{m_{\mathrm{DMB}}(\varepsilon)}{b}\left(\frac{b}{k}+\delta\right)} . \tag{10}$$

In the above definition, we intentionally leave the dependence of b on m unspecified. Indeed, once we fix the function $b(m)$, we can substitute it into the equation $\bar{\psi}({}^{\sigma^2}\!/_b, {}^m\!/_b) = \varepsilon$ to solve for the exact form of $m_{\mathrm{DMB}}(\varepsilon)$. As a result, b is also a function of ε.

Since both $m_{\mathrm{srl}}(\varepsilon)$ and $m_{\mathrm{DMB}}(\varepsilon)$ are upper bounds for the actual running times to reach ε-optimality, their ratio $S(\varepsilon)$ may not be a precise measure of the speed-up. However, it is difficult in practice to measure the actual running times of the algorithms in terms of reaching ε-optimality. So we only hope $S(\varepsilon)$ gives a conceptual guide in comparing the actual performance of the algorithms. The following result shows that if the batch size b is chosen to be of order m^{ρ} for any $\rho \in (0,1/2)$, then we still have asymptotic linear speed-up.

Theorem 6 *Let $f(w,z)$ be an L-smooth convex loss function in w for each $z \in \mathcal{Z}$ and assume that the stochastic gradient $\nabla_w f(w,z)$ has σ^2-bounded variance for all $w \in W$. Suppose the update rule ϕ used in the serial setting has an expected optimality gap bounded by $\bar{\psi}(\sigma^2,m) = \frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}}$. If the*

[5] The relationship between δ and μ defined in the online setting (see Section 4) is roughly $\mu = k\delta$.

batch size in the DMB algorithm is chosen as $b(m) = \Theta(m^\rho)$*, where* $\rho \in (0, 1/2)$*, then we have*

$$\lim_{\varepsilon \to 0} S(\varepsilon) = k.$$

Proof By solving the equation

$$\frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}} = \varepsilon\ ,$$

we see that the following number of samples is sufficient for the serial algorithm to reach ε-optimality:

$$m_{\rm srl}(\varepsilon) \ = \ \frac{D^2\sigma^2}{\varepsilon^2}\left(1+\sqrt{1+\frac{2L\varepsilon}{\sigma^2}}\right)^2 \ .$$

For the DMB algorithm, we use the batch size $b(m) = ({}^{\theta\sigma}\!/_{DL})m^\rho$, with some $\theta > 0$, to obtain the equation

$$\frac{2b(m)D^2L}{m} + \frac{2D\sigma}{\sqrt{m}} \ = \ \frac{2D\sigma}{m^{1/2}}\left(1+\frac{\theta}{m^{1/2-\rho}}\right) \ = \ \varepsilon. \qquad (11)$$

We use $m_{\rm DMB}(\varepsilon)$ to denote the solution of the above equation. Apparently $m_{\rm DMB}(\varepsilon)$ is a monotone function of ε and $\lim_{\varepsilon\to 0} m_{\rm DMB}(\varepsilon) = \infty$. For convenience (with some abuse of notation), let $b(\varepsilon)$ to denote $b(m_{\rm DMB}(\varepsilon))$, which is also monotone in ε and satisfies $\lim_{\varepsilon\to 0} b(\varepsilon) = \infty$. Moreover, for any batch size $b > 1$, we have $m_{\rm DMB}(\varepsilon) \geq m_{\rm srl}(\varepsilon)$. Therefore, from Equation (10) we get

$$\limsup_{\varepsilon\to 0} S(\varepsilon) \leq \lim_{\varepsilon\to 0} \frac{k}{1+\frac{\delta}{b(\varepsilon)}k} = k.$$

Next we show $\liminf_{\varepsilon\to 0} S(\varepsilon) \geq k$. For any $\eta > 0$, let

$$m_\eta(\varepsilon) = \frac{4D^2\sigma^2(1+\eta)^2}{\varepsilon^2}.$$

which is monotone decreasing in ε, and can be seen as the solution to the equation

$$\frac{2D\sigma}{m^{1/2}}(1+\eta) \ = \ \varepsilon.$$

Comparing this equation with Equation (11), we see that, for any $\eta > 0$, there exists an ε' such that for all $0 < \varepsilon \leq \varepsilon'$, we have $m_{\rm DMB}(\varepsilon) \leq m_\eta(\varepsilon)$. Therefore,

$$\liminf_{\varepsilon\to 0} S(\varepsilon) \ \geq \ \lim_{\varepsilon\to 0} \frac{m_{\rm srl}(\varepsilon)}{m_\eta(\varepsilon)} \frac{k}{1+\frac{\delta}{b(\varepsilon)}k} \ = \ \lim_{\varepsilon\to 0} \frac{\left(1+\sqrt{1+\frac{2L\varepsilon}{\sigma^2}}\right)^2}{4(1+\eta)^2} \frac{k}{1+\frac{\delta}{b(\varepsilon)}k} \ = \ \frac{1}{(1+\eta)^2}k.$$

Since the above inequality holds for any $\eta > 0$, we can take $\eta \to 0$ and conclude that $\liminf_{\varepsilon\to 0} S(\varepsilon) \geq k$. This finishes the proof. ■

For accelerated stochastic gradient methods whose convergence rates have a similar dependence on the gradient variance (Lan, 2009; Hu et al., 2009; Xiao, 2010; Ghadimi and Lan, 2010), the batch size b has a even smaller effect on the convergence rate (see discussions after Theorem 5), which implies a better parallel speed-up.

6. Experiments

We conducted experiments with a large-scale online binary classification problem. First, we obtained a log of one billion queries issued to the Internet search engine Bing. Each entry in the log specifies a time stamp, a query text, and the id of the user who issued the query (using a temporary browser cookie). A query is said to be *highly monetizable* if, in the past, users who issued this query tended to then click on online advertisements. Given a predefined list of one million highly monetizable queries, we observe the queries in the log one-by-one and attempt to predict whether the next query will be highly monetizable or not. A clever search engine could use this prediction to optimize the way it presents search results to the user. A prediction algorithm for this task must keep up with the stream of queries received by the search engine, which calls for a distributed solution.

The predictions are made based on the recent query-history of the current user. For example, the predictor may learn that users who recently issued the queries “island weather” and “sunscreen reviews” (both not highly monetizable in our data) are likely to issue a subsequent query which is highly monetizable (say, a query like “Hawaii vacation”). In the next section, we formally define how each input, z_t, is constructed.

First, let n denote the number of distinct queries that appear in the log and assume that we have enumerated these queries, $q_1, \ldots, q_n$. Now define $x_t \in \{0,1\}^n$ as follows

$$x_{t,j} = \begin{cases} 1 & \text{if query } q_j \text{ was issued by the current user during the last two hours,} \\ 0 & \text{otherwise.} \end{cases}$$

Let y_t be a binary variable, defined as

$$y_t = \begin{cases} +1 & \text{if the current query is highly monetizable,} \\ -1 & \text{otherwise.} \end{cases}$$

In other words, y_t is the binary label that we are trying to predict. Before observing x_t or y_t, our algorithm chooses a vector $w_t \in \mathbb{R}^n$. Then x_t is observed and the resulting binary prediction is the sign of their inner product $\langle w_t, x_t \rangle$. Next, the correct label y_t is revealed and our binary prediction is incorrect if $y_t \langle w_t, x_t \rangle \leq 0$. We can re-state this prediction problem in an equivalent way by defining $z_t = y_t x_t$, and saying that an incorrect prediction occurs when $\langle w_t, z_t \rangle \leq 0$.

We adopt the logistic loss function as a smooth convex proxy to the error indicator function. Formally, define f as

$$f(w,z) = \log_2 \left(1 + \exp(-\langle w, z \rangle)\right) .$$

Additionally, we introduced the convex regularization constraint $\|w_t\| \leq C$, where C is a predefined regularization parameter.

We ran the synchronous version of our distributed algorithm using the Euclidean dual averaging update rule (4) in a cluster simulation. The simulation allowed us to easily investigate the effects of modifying the number of nodes in the cluster and the latencies in the network.

We wanted to specify a realistic latency in our simulation, which faithfully mimics the behavior of a real network in a search engine datacenter. To this end, we assumed that the nodes are connected via a standard 1Gbs Ethernet network. Moreover, we assumed that the nodes are arranged in a precomputed logical binary-tree communication structure, and that all communication is done along the edges in this tree. We conservatively estimated the round-trip latency between proximal nodes in the tree to be 0.5ms. Therefore, the total time to complete each vector-sum network operation

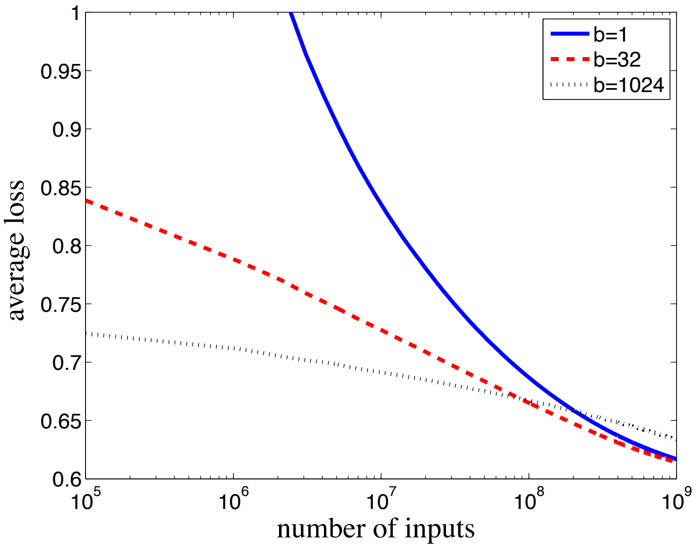


Figure 2: The effects of of the batch size when serial mini-batching on average loss. The mini-batches algorithm was applied with different batch sizes. The x-axis presents the number of instances observed, and the y-axis presents the average loss. Note that the case $b = 1$ is the standard serial dual-averaging algorithm.

is $\log_2(k)$ ms, where k is the number of nodes in the cluster. We assumed that our search engine receives 4 queries per ms (which adds up to ten billion queries a month). Overall, the number of queries discarded between mini-batches is $\mu = 4\log_2(k)$.

In all of our experiments, we use the algorithmic parameter $\alpha_j = L + \gamma\sqrt{j}$ (see Theorem 2). We set the smoothness parameter L to a constant, and the parameter γ to a constant divided by $\sqrt{b}$. This is because L depends only on the loss function f, which does not change in DMB, while γ is proportional to σ, the standard deviation of the gradient-averages. We chose the constants by manually exploring the parameter space on a separate held-out set of 500 million queries.

We report all of our results in terms of the average loss suffered by the online algorithm. This is simply defined as $(1/t)\sum_{i=1}^{t} f(w_i, z_i)$. We cannot plot regret, as we do not know the offline risk minimizer $w^\star$.

6.1 Serial Mini-Batching

As a warm-up, we investigated the effects of modifying the mini-batch size b in a standard serial Euclidean dual averaging algorithm. This is equivalent to running the distributed simulation with a cluster size of $k = 1$, with varying mini-batch size. We ran the experiment with $b = 1, 2, 4, \ldots, 1024$. Figure 2 shows the results for three representative mini-batch sizes. The experiments tell an interesting story, which is more refined than our theoretical upper bounds. While the asymptotic worst-case theory implies that batch-size should have no significant effect, we actually observe that mini-batching accelerates the learning process on the first 10^8 inputs. On the other hand, after 10^8 inputs, a large mini-batch size begins to hurt us and the smaller mini-batch sizes gain the lead. This behavior is not an artifact of our choice of the parameters γ and L, as we observed a similar behavior

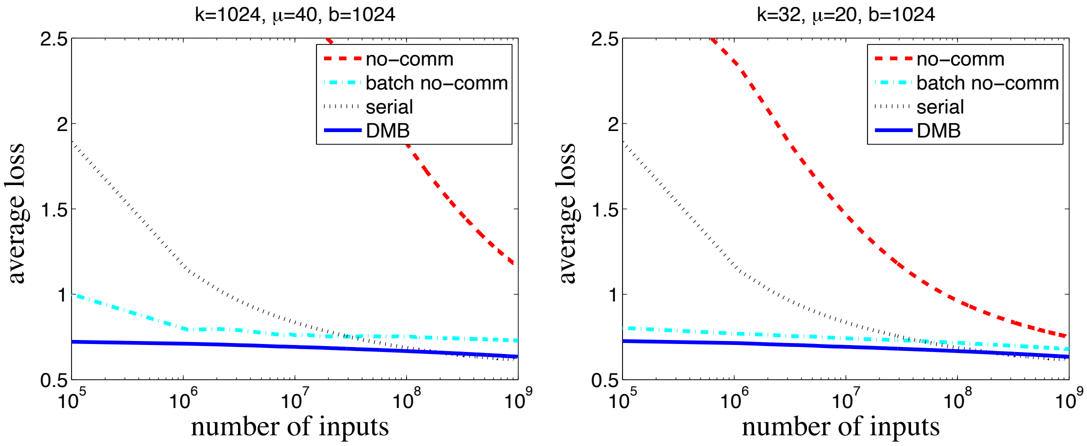


Figure 3: Comparing DBM with the serial algorithm and the no-communication distributed algorithm. Results for a large cluster of $k = 1024$ machines are presented on the left. Results for a small cluster of $k = 32$ machines are presented on the right.

for many different parameter setting, during the initial stage when we tuned the parameters on a held-out set.

Similar transient behaviors also exist for multi-step stochastic gradient methods (see, e.g., Polyak, 1987, Section 4.3.2), where the multi-step interpolation of the gradients also gives the smoothing effects as using averaged gradients. Typically such methods converge faster in the early iterations when the iterates are far from the optimal solution and the relative value of the stochastic noise is small, but become less effective asymptotically.

6.2 Evaluating DBM

Next, we compared the average loss of the DBM algorithm with the average loss of the serial algorithm and the no-communication algorithm (where each cluster node works independently). We tried two versions of the no-communication solution. The first version simply runs k independent copies of the serial prediction algorithm. The second version runs k independent copies of the serial mini-batch algorithm, with a mini-batch size of 128. We included the second version of the no-communication algorithm after observing that mini-batching has significant advantages even in the serial setting. We experimented with various cluster sizes and various mini-batch sizes. As mentioned above, we set the latency of the DBM algorithm to $\mu = 4\log_2(k)$. Taking a cue from our theoretical analysis, we set the batch size to $b = m^{1/3} \simeq 1024$. We repeated the experiment for various cluster sizes and the results were very consistent. Figure 3 presents the average loss of the three algorithms for clusters of sizes $k = 1024$ and $k = 32$. Clearly, the simple no-communication algorithm performs very poorly compared to the others. The no-communication algorithm that uses mini-batch updates on each node does surprisingly well, but is still outperformed quite significantly by the DMB solution.

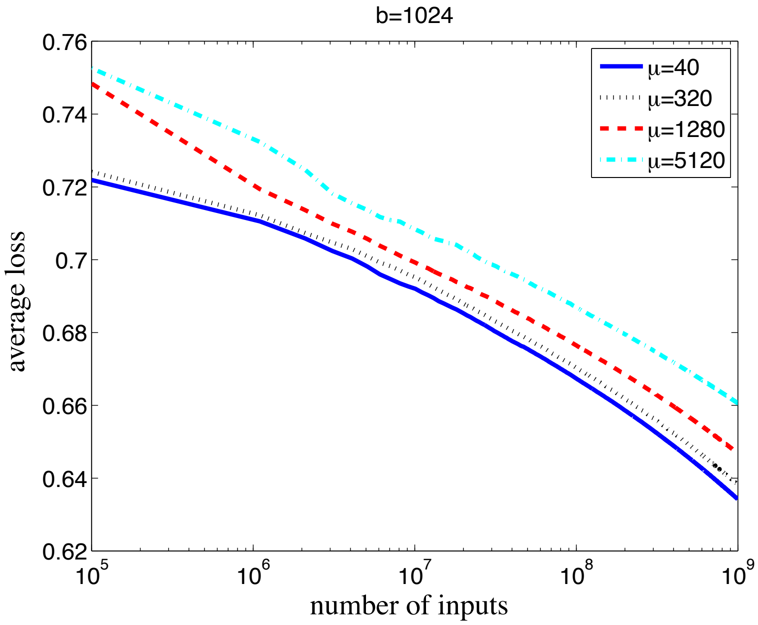


Figure 4: The effects of increased network latency. The loss of the DMB algorithm is reported with different latencies as measured by μ. In all cases, the batch size is fixed at $b = 1024$.

6.3 The Effects of Latency

Network latency results in the DMB discarding gradients, and slows down the algorithm's progress. The theoretical analysis shows that this waste is negligible in the asymptotic worst-case sense. However, latency will obviously have some negative effect on any finite prefix of the input stream. We examined what would happen if the single-link latency were much larger than our 0.5ms estimate (e.g., if the network is very congested or if the cluster nodes are scattered across multiple datacenters). Concretely, we set the cluster size to $k = 1024$ nodes, the batch size to $b = 1024$, and the single-link latency to $0.5, 1, 2, \ldots, 512$ ms. That is, 0.5ms mimics a realistic 1Gbs Ethernet link, while 512ms mimics a network whose latency between any two machines is 1024 times greater, namely, each vector-sum operation takes a full second to complete. Note that μ is still computed as before, namely, for latency $0.5 \cdot 2^p$, $\mu = 2^p 4 \log_2(k) = 2^p \cdot 40$. Figure 4 shows how the average loss curve reacts to four representative latencies. As expected, convergence rate degrades monotonically with latency. When latency is set to be 8 times greater than our realistic estimate for 1Gbs Ethernet, the effect is minor. When the latency is increased by a factor of 1024, the effect becomes more noticeable, but still quite small.

6.4 Optimal Mini-Batch Size

For our final experiment, we set out to find the optimal batch size for our problem on a given cluster size. Our theoretical analysis is too crude to provide a sufficient answer to this question. The theory basically says that setting $b = \Theta(m^\rho)$ is asymptotically optimal for any $\rho \in (0, 1/2)$, and

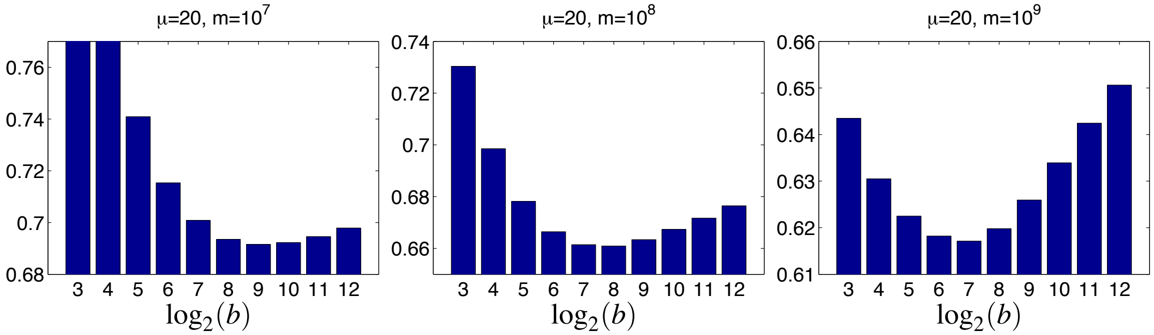


Figure 5: The effect of different mini-batch sizes (b) on the DBM algorithm. The DMB algorithm was applied with different batch sizes $b = 8, \ldots, 4096$. The loss is reported after 10^7 instances (left), 10^8 instances (middle) and 10^9 instances (right).

that $b = \Theta(m^{1/3})$ is a pretty good concrete choice. We have already seen that larger batch sizes accelerate the initial learning phase, even in a serial setting. We set the cluster size to $k = 32$ and set batch size to $8, 16, \ldots, 4096$. Note that $b = 32$ is the case where each node processes a single example before engaging in a vector-sum network operation. Figure 5 depicts the average loss after $10^7, 10^8$, and 10^9 inputs. As noted in the serial case, larger batch sizes ($b = 512$) are beneficial at first ($m = 10^7$), while smaller batch sizes ($b = 128$) are better in the end ($m = 10^9$).

6.5 Discussion

We presented an empirical evaluation of the serial mini-batch algorithm and its distributed version, the DMB algorithm, on a realistic web-scale online prediction problem. As expected, the DMB algorithm outperforms the näive no-communication algorithm. An interesting and somewhat unexpected observation is the fact that the use of large batches improves performance even in the serial setting. Moreover, the optimal batch size seems to generally decrease with time.

We also demonstrated the effect of network latency on the performance of the DMB algorithm. Even for relatively large values of μ, the degradation in performance was modest. This is an encouraging indicator of the efficiency and robustness of the DMB algorithm, even when implemented in a high-latency environment, such as a grid.

7. Related Work

In recent years there has been a growing interest in distributed online learning and distributed optimization.

Langford et al. (2009) address the distributed online learning problem, with a similar motivation to ours: trying to address the scalability problem of online learning algorithms which are inherently sequential. The main observation Langford et al. (2009) make is that in many cases, computing the gradient takes much longer than computing the update according to the online prediction algorithm. Therefore, they present a pipeline computational model. Each worker alternates between computing the gradient and computing the update rule. The different workers are synchronized such that no two workers perform an update simultaneously.

Similar to results presented in this paper, Langford et al. (2009) attempted to show that it is possible to achieve a cumulative regret of $O(\sqrt{m})$ with k parallel workers, compared to the $O\left(\sqrt{km}\right)$ of the naïve solution. However their work suffers from a few limitations. First, their proofs only hold for unconstrained convex optimization where no projection is needed. Second, since they work in a model where one node at a time updates a shared predictor, while the other nodes compute gradients, the scalability of their proposed method is limited by the ratio between the time it takes to compute a gradient to the time it takes to run the update rule of the serial online learning algorithm.

In another related work, Duchi et al. (2010) present a distributed dual averaging method for optimization over networks. They assume the loss functions are Lipschitz continuous, but their gradients may not be. Their method does not need synchronization to average gradients computed at the same point. Instead, they employ a distributed consensus algorithm on all the gradients generated by different processors at different points. When applied to the stochastic online prediction setting, even for the most favorable class of communication graphs, with constant spectral gaps (e.g., expander graphs), their best regret bound is $O\left(\sqrt{km}\log(m)\right)$. This bound is no better than one would get by running k parallel machines without communication (see Section 2.2).

In another recent work, Zinkevich et al. (2010) study a method where each node in the network runs the classic stochastic gradient method, using random subsets of the overall data set, and only aggregate their solutions in the end (by averaging their final weight vectors). In terms of online regret, it is obviously the same as running k machines independently without communication. So a more suitable measure is the optimality gap (defined in Section 5) of the final averaged predictor. Even with respect to this measure, their expected optimality gap does not show advantage over running k machines independently. A similar approach was also considered by Nesterov and Vial (2008) and an experimental study of such a method was reported in Harrington et al. (2003).

A key difference between our DMB framework and many related work is that DMB does not consider distributed comuting as a constraint to overcome. Instead, our novel use of the variance-based regret bounds can exploit parallel/distributed computing to obtain the asymptotic optimal regret bound. Beyond the asymptotic optimality of our bounds, our work has other features that set it apart from previous work. As far as we know, we are the first to propose a general principled framework for distributing many gradient-based update rule, with a concrete regret analysis for the large family of mirror descent and dual averaging update rules. Additionally, our work is the first to explicitly include network latency in our regret analysis, and to theoretically guarantee that a large latency can be overcome by setting parameters appropriately.

8. Conclusions and Further Research

The increase in serial computing power of modern computers is out-paced by the growth rate of web-scale prediction problems and data sets. Therefore, it is necessary to adopt techniques that can harness the power of parallel and distributed computers.

In this work we studied the problems of distributed stochastic online prediction and distributed stochastic optimization. We presented a family of distributed online algorithms with asymptotically optimal regret and optimality gap guarantees. Our algorithms use the distributed computing infrastructure to reduce the variance of stochastic gradients, which essentially reduces the noise in the algorithm's updates. Our analysis shows that asymptotically, a distributed computing system can perform as well as a hypothetical fast serial computer. This result is far from trivial, and much of the prior art in the field did not show any provable gain by using distributed computers.

While the focus of this work is the theoretical analysis of a distributed online prediction algorithm, we also presented experiments on a large-scale real-world problem. Our experiments showed that indeed the DMB algorithm outperforms other simple solutions. They also suggested that improvements can be made by optimizing the batch size and adjusting the learning rate based on empirical measures.

Our formal analysis hinges on the fact that the regret bounds of many stochastic online update rules scale with the variance of the stochastic gradients when the loss function is smooth. It is unclear if smoothness is a necessary condition, or if it can be replaced with a weaker assumption. In principle, our results apply in a broader setting. For any serial update rule ϕ with a regret bound of $\psi(\sigma^2,m) = C\sigma\sqrt{m} + o\left(\sqrt{m}\right)$, the DMB algorithm and its variants have the optimal regret bound of $C\sigma\sqrt{m} + o\left(\sqrt{m}\right)$, provided that the bound $\psi(\sigma^2,m)$ applies equally to the function f and to the function

$$\bar{f}\left(w,(z_1,\ldots,z_b)\right) \;=\; \frac{1}{b}\sum_{s=1}^{b} f(w,z_s)\;.$$

Note that this result holds independently of the network size k and the network latency μ. Extending our results to non-smooth functions is an interesting open problem. A more ambitious challenge is to extend our results to the non-stochastic case, where inputs may be chosen by an adversary.

An important future direction is to develop distributed learning algorithms that perform robustly and efficiently on heterogeneous clusters and in asynchronous distributed environments. This direction has been further explored in Dekel et al. (2011). For example, one can use the following simple reformulation of the DMB algorithm in a master-workers setting: each worker process inputs at its own pace and periodically sends the accumulated gradients to the master; the master applies the update rule whenever the number of accumulated gradients reaches a certain threshold and broadcasts the new predictor back to the workers. In a dynamic environment, where the network can be partitioned and reconnected and where nodes can be added and removed, a new master (or masters) can be chosen as needed by a standard leader election algorithm. We refer the reader to Dekel et al. (2011) for more details.

A central property of our method is that all of the gradients in a batch must be taken at the same prediction point. In an asynchronous distributed computing environment (see, e.g., Tsitsiklis et al., 1986; Bertsekas and Tsitsiklis, 1989), this can be quite wasteful. In order to reduce the waste generated by the need for global synchronization, we may need to allow different nodes to accumulate gradients at different yet close points. Such a modification is likely to work since the smoothness assumption precisely states that gradients of nearby points are similar. There have been extensive studies on distributed optimization with inaccurate or delayed subgradient information, but mostly without the smoothness assumption (e.g., Nedić et al., 2001; Nedić and Ozdaglar, 2009). We believe that our main results under the smoothness assumption can be extended to asynchronous and distributed environments as well.

Acknowledgments

The authors are grateful to Karthik Sridharan for spotting the fact that for non-Euclidean spaces, our variance-reduction argument had to be modified with a space-dependent constant (see the discussion at the end of Section 4.1). We also thank the reviewers for their helpful comments.

Appendix A. Smooth Stochastic Online Prediction in the Serial Setting

In this appendix, we prove expected regret bounds for stochastic dual averaging and stochastic mirror descent applied to smooth loss functions. In the main body of the paper, we discussed only the Euclidean special case of these algorithms, while here we present the algorithms and regret bounds in their full generality. In particular, Theorem 1 is a special case of Theorem 9, and Theorem 2 is a special case of Theorem 7.

Recall that we observe a stochastic sequence of inputs $z_1, z_2, \ldots$, where each $z_i \in \mathcal{Z}$. Before observing each z_i we predict $w_i \in W$, and suffer a loss $f(w_i, z_i)$. We assume W is a closed convex subset of a finite dimensional vector space $\mathcal{V}$ with endowed norm $\|\cdot\|$. We assume that $f(w,z)$ is convex and differentiable in w, and we use $\nabla_w f(w,z)$ to denote the gradient of f with respect to its first argument. $\nabla_w f(w,z)$ is a vector in the dual space $\mathcal{V}^*$, with endowed norm $\|\cdot\|_*$.

We assume that $f(\cdot,z)$ is L-smooth for any realization of z. Namely, we assume that $f(\cdot,z)$ is differentiable and that

$$\forall z \in \mathcal{Z}, \quad \forall w, w' \in W, \qquad \|\nabla_w f(w,z) - \nabla_w f(w',z)\|_* \leq L\|w - w'\| \ .$$

We define $F(w) = \mathbb{E}_z[f(w,z)]$ and note that $\nabla_w F(w) = \mathbb{E}_z[\nabla_w f(w,z)]$ (see Rockafellar and Wets, 1982). This implies that

$$\forall w, w' \in W, \qquad \|\nabla_w F(w) - \nabla_w F(w')\|_* \leq L\|w - w'\| \ .$$

In addition, we assume that there exists a constant $\sigma \geq 0$ such that

$$\forall w \in W, \qquad \mathbb{E}_z[\|\nabla_w f(w,z) - \nabla_w E_z[f(w,z)]\|_*^2] \leq \sigma^2 \ .$$

We assume that $w^\star = \arg\min_{w \in W} F(w)$ exists, and we abbreviate $F^\star = F(w^\star)$.

Under the above assumptions, we are concerned with bounding the expected regret $\mathbb{E}[R(m)]$, where regret is defined as

$$R(m) \ = \ \sum_{i=1}^{m} \left(f(w_i, z_i) - f(w^\star, z_i)\right) \ .$$

In order to present the algorithms in their full generality, we first recall the concepts of strongly convex function and Bregman divergence.

A function $h : W \to \mathbb{R} \cup \{+\infty\}$ is said to be *μ-strongly convex* with respect to $\|\cdot\|$ if

$$\forall \alpha \in [0,1], \quad \forall u, v \in W, \quad h(\alpha u + (1-\alpha)v) \leq \alpha h(u) + (1-\alpha)h(v) - \frac{\mu}{2}\alpha(1-\alpha)\|u - v\|^2 \ .$$

If h is μ-strongly convex then for any $u \in \operatorname{dom} h$, and $v \in \operatorname{dom} h$ that is sub-differentiable, then

$$\forall s \in \partial h(v), \quad h(u) \geq h(v) + \langle s, u - v\rangle + \frac{\mu}{2}\|u - v\|^2 \ .$$

(See, e.g., Goebel and Rockafellar, 2008.) If a function h is strictly convex and differentiable (on an open set contained in $\operatorname{dom} h$), then we can defined the Bregman divergence generated by h as

$$d_h(u,v) = h(u) - h(v) - \langle \nabla h(v), u - v \rangle \ .$$

We often drop the subscript h in d_h when it is obvious from the context. Some key properties of the Bregman divergence are:

- $d(u,v) \geq 0$, and the equality holds if and only if $u = v$.
- In general $d(u,v) \neq d(v,u)$, and d may not satisfy the triangle inequality.
- The following *three-point identity* follows directly from the definition:

$$d(u,w) = d(u,v) + d(v,w) + \langle \nabla h(v) - \nabla h(w), u - v \rangle \ .$$

The following inequality is a direct consequence of the μ-strong convexity of h:

$$d(u,v) \geq \frac{\mu}{2} \|u - v\|^2 \ . \tag{12}$$

A.1 Stochastic Dual Averaging

The proof techniques for the stochastic dual averaging method are adapted from those for the accelerated algorithms presented in Tseng (2008) and Xiao (2010).

Let $h : W \to \mathbb{R}$ be a 1-strongly convex function. Without loss of generality, we can assume that $\min_{w \in W} h(w) = 0$. In the stochastic dual averaging method, we predict each w_i by

$$w_{i+1} = \arg\min_{w \in W} \left\{ \left\langle \sum_{j=1}^{i} g_j, w \right\rangle + (L + \beta_{i+1}) h(w) \right\} \ , \tag{13}$$

where g_j denotes the stochastic gradient $\nabla_w f(w_j, z_j)$, and $(\beta_i)_{i \geq 1}$ is a sequence of positive and nondecreasing parameters (i.e., $\beta_{i+1} \geq \beta_i$). As a special case of the above, we initialize w_1 to

$$w_1 = \arg\min_{w \in W} h(w) \ . \tag{14}$$

We are now ready to state a bound on the expected regret of the dual averaging method, in the smooth stochastic case.

Theorem 7 *The expected regret of the stochastic dual averaging method is bounded as*

$$\forall m, \quad \mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + (L + \beta_m) h(w^\star) + \frac{\sigma^2}{2} \sum_{i=1}^{m-1} \frac{1}{\beta_i}.$$

The optimal choice of β_i is exactly of order $\sqrt{i}$. More specifically, let $\beta_i = \gamma\sqrt{i}$, where γ is a positive parameter. Then Theorem 7 implies that

$$\mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + L h(w^\star) + \left(\gamma h(w^\star) + \frac{\sigma^2}{\gamma} \right) \sqrt{m}.$$

Choosing $\gamma = \sigma/\sqrt{h(w^\star)}$ gives

$$\mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + Lh(w^\star) + \left(2\sigma\sqrt{h(w^\star)}\right)\sqrt{m}.$$

If $\nabla F(w^\star) = 0$ (this is certainly the case if W is the whole space), then we have

$$F(w_1) - F(w^\star) \leq \frac{L}{2}\|w_1 - w^\star\|^2 \leq Lh(w^\star).$$

Then the expected regret bound can be simplified as

$$\mathbb{E}[R(m)] \leq 2Lh(w^\star) + \left(2\sigma\sqrt{h(w^\star)}\right)\sqrt{m}.$$

To prove Theorem 7 we require the following fundamental lemma, which can be found, for example, in Nesterov (2005), Tseng (2008) and Xiao (2010).

Lemma 8 *Let W be a closed convex set, φ be a convex function on W, and h be μ-strongly convex on W with respect to $\|\cdot\|$. If*

$$w^+ = \arg\min_{w\in W}\left\{\varphi(w) + h(w)\right\},$$

then

$$\forall w \in W, \qquad \varphi(w) + h(w) \geq \varphi(w^+) + h(w^+) + \frac{\mu}{2}\|w - w^+\|^2.$$

With Lemma 8, we are now ready to prove Theorem 7.

Proof First, we define the linear functions

$$\ell_i(w) = F(w_i) + \langle \nabla F(w_i), w - w_i\rangle, \qquad \forall i \geq 1,$$

and (using the notation $g_i = \nabla f(w_i, z_i)$)

$$\hat{\ell}_i(w) = F(w_i) + \langle g_i, w - w_i\rangle = \ell_i(w) + \langle q_i, w - w_i\rangle,$$

where

$$q_i = g_i - \nabla F(w_i).$$

Therefore, the stochastic dual averaging method specified in Equation (13) is equivalent to

$$w_i = \arg\min_{w\in W}\left\{\sum_{j=1}^{i-1}\hat{\ell}_j(w) + (L+\beta_i)h(w)\right\}.$$

Using the smoothness assumption, we have (e.g., Nesterov 2004, Lemma 1.2.3)

$$\begin{aligned}
F(w_{i+1}) &\leq \ell_i(w_{i+1}) + \frac{L}{2}\|w_{i+1} - w_i\|^2 \\
&= \hat{\ell}_i(w_{i+1}) + \frac{L+\beta_i}{2}\|w_{i+1} - w_i\|^2 - \langle q_i, w_{i+1} - w_i\rangle - \frac{\beta_i}{2}\|w_{i+1} - w_i\|^2 \\
&\leq \hat{\ell}_i(w_{i+1}) + \frac{L+\beta_i}{2}\|w_{i+1} - w_i\|^2 + \|q_i\|_*\|w_{i+1} - w_i\| - \frac{\beta_i}{2}\|w_{i+1} - w_i\|^2 \\
&= \hat{\ell}_i(w_{i+1}) + \frac{L+\beta_i}{2}\|w_{i+1} - w_i\|^2 - \left(\frac{1}{\sqrt{2\beta_i}}\|q_i\|_* - \sqrt{\frac{\beta_i}{2}}\|w_{i+1} - w_i\|\right)^2 + \frac{\|q_i\|_*^2}{2\beta_i} \\
&\leq \hat{\ell}_i(w_{i+1}) + \frac{L+\beta_i}{2}\|w_{i+1} - w_i\|^2 + \frac{\|q_i\|_*^2}{2\beta_i}. \qquad (15)
\end{aligned}$$

Next we use Lemma 8 with $\varphi(w) = \sum_{j=1}^{i-1} \hat{\ell}_j(w)$ and $\mu = (L+\beta_i)$,

$$\sum_{j=1}^{i-1} \hat{\ell}_j(w_{i+1}) + (L+\beta_i)h(w_{i+1}) \geq \sum_{j=1}^{i-1} \hat{\ell}_j(w_i) + (L+\beta_i)h(w_i) + \frac{L+\beta_i}{2}\|w_{i+1}-w_i\|^2,$$

Combining the above inequality with Equation (15), we have

$$\begin{aligned} F(w_{i+1}) &\leq \hat{\ell}_i(w_{i+1}) + \sum_{j=1}^{i-1} \hat{\ell}_j(w_{i+1}) + (L+\beta_i)h(w_{i+1}) - \sum_{j=1}^{i-1} \hat{\ell}_j(w_i) - (L+\beta_i)h(w_i) + \frac{\|q_i\|_*^2}{2\beta_i} \\ &\leq \sum_{j=1}^{i} \hat{\ell}_j(w_{i+1}) + (L+\beta_{i+1})h(w_{i+1}) - \sum_{j=1}^{i-1} \hat{\ell}_j(w_i) - (L+\beta_i)h(w_i) + \frac{\|q_i\|_*^2}{2\beta_i}, \end{aligned}$$

where in the last inequality, we used the assumptions $\beta_{i+1} > \beta_i > 0$ and $h(w_{i+1}) \geq 0$. Summing the above inequality from $i=1$ to $i=m-1$, we have

$$\begin{aligned} \sum_{i=2}^{m} F(w_i) &\leq \sum_{i=1}^{m-1} \hat{\ell}_i(w_m) + (L+\beta_m)h(w_m) + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} \\ &\leq \sum_{i=1}^{m-1} \hat{\ell}_i(w^\star) + (L+\beta_m)h(w^\star) + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} \\ &\leq \sum_{i=1}^{m-1} \ell_i(w^\star) + (L+\beta_m)h(w^\star) + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} + \sum_{i=1}^{m-1} \langle q_i, w^\star - w_i \rangle \\ &\leq (m-1)F(w^\star) + (L+\beta_i)h(w^\star) + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} + \sum_{i=1}^{m-1} \langle q_i, w^\star - w_i \rangle. \end{aligned}$$

Therefore,

$$\sum_{i=2}^{m} \big(F(w_i) - F(w^\star)\big) \leq (L+\beta_m)h(w^\star) + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} + \sum_{i=1}^{m-1} \langle q_i, w^\star - w_i \rangle. \tag{16}$$

Notice that each w_i is a deterministic function of $z_1, \ldots, z_{i-1}$, so

$$\mathbb{E}_{z_i}\big(\langle q_i, w^\star - w_i \rangle \,|\, z_1, \ldots, z_{i-1}\big) = 0$$

by recalling the definition $q_i = \nabla f(w_i, z_i) - \nabla F(w_i)$. Taking expectation of both sides of Equation (16) with respect to $z_1, \ldots, z_m$, and adding the term $F(w_1) - F(w^\star)$, we have

$$\mathbb{E} \sum_{i=1}^{m} \big(F(w_i) - F(w^\star)\big) \leq F(w_1) - F(w^\star) + (L+\beta_m)h(w^\star) + \sum_{i=1}^{m-1} \frac{\sigma^2}{2\beta_i}.$$

Theorem 7 is proved by further noticing

$$\mathbb{E} f(w_i, z_i) = \mathbb{E} F(w_i), \qquad \mathbb{E} f(w^\star, z_i) = F(w^\star), \qquad \forall i \geq 1,$$

which are due to the fact that w_i is a deterministic function of $z_0, \ldots, z_{i-1}$. ■

A.2 Stochastic Mirror Descent

Variance-based convergence rates for the stochastic Mirror Descent methods are due to Juditsky et al. (2011), and were extended to an accelerated stochastic Mirror Descent method by Lan (2009). For completeness, we adapt their proofs to the context of regret for online prediction problems.

Again let $h : W \to \mathbb{R}$ be a differentiable 1-strongly convex function with $\min_{w \in W} h(w) = 0$. Also let d be the Bregman divergence generated by h. In the stochastic mirror descent method, we use the same initialization as in the dual averaging method (see Equation (14)) and then we set

$$w_{i+1} = \arg\min_{w\in W} \Big\{ \langle g_i, w\rangle + (L+\beta_i) d(w, w_i) \Big\}, \qquad i \geq 1.$$

As in the dual averaging method, we assume that the sequence $(\beta_i)_{i\geq 1}$ to be positive and nondecreasing.

Theorem 9 *Assume that the convex set W is closed and bounded. In addition assume $d(u,v)$ is bounded on W and let*

$$D^2 = \max_{u,v\in W} d(u,v).$$

Then the expected regret of the stochastic mirror descent method is bounded as

$$\mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + (L+\beta_m) D^2 + \frac{\sigma^2}{2} \sum_{i=1}^{m-1} \frac{1}{\beta_i}.$$

Similar to the dual averaging case, using the sequence of parameters $\beta_i = (\sigma/D)\sqrt{i}$ gives the expected regret bound

$$\mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + LD^2 + (2\sigma D)\sqrt{m}.$$

Again, if $\nabla F(w^\star) = 0$, we have $F(w_1) - F(w^\star) \leq (L/2)\|w_1 - w^\star\|^2 \leq Lh(w^\star) \leq LD^2$, thus the simplified bound

$$\mathbb{E}[R(m)] \leq 2LD^2 + (2\sigma D)\sqrt{m}.$$

We note that here we have stronger assumptions than in the dual averaging case. These assumptions are certainly satisfied by using the standard Euclidean distance $d(u,v) = (1/2)\|u-v\|_2^2$ on a compact convex set W. However, it excludes the case of using the KL-divergence $d(u,v) = \sum_{i=1}^n u_i \log(u_i/v_i)$ on the simplex, because the KL-divergence is unbounded on the simplex. Nevertheless, it is possible to remove such restrictions by considering other variants of the stochastic mirror descent method. For example, if we use a constant β_i that depends on the prior knowledge of the number of total steps to be performed, then we can weaken the assumption and replace D in the above bounds by $\sqrt{h(w^\star)}$. More precisely, we have

Theorem 10 *Suppose we know the total number of steps m to be performed by the stochastic mirror descent method ahead of time. Then by using the initialization in Equation (14) and the constant parameter*

$$\beta_i = \frac{\sigma}{\sqrt{2h(w^\star)}}\sqrt{m},$$

we have the expected regret bound

$$\mathbb{E}[R(m)] \leq (F(w_1) - F(w^\star)) + Lh(w^\star) + \sigma\sqrt{2h(w^\star)}\sqrt{m}.$$

Theorem 10 is essentially the same as a result in Lan (2009), who also developed an accelerated versions of the stochastic mirror descent method. To prove Theorem 9 and Theorem 10 we need the following standard Lemma, which can be found in Chen and Teboulle (1993), Lan et al. (2011) and Tseng (2008).

Lemma 11 *Let W be a closed convex set, φ be a convex function on W, and h be a differentiable, strongly convex function on W. Let d be the Bregman divergence generated by h. Given $u \in W$, if*

$$w^+ = \operatorname*{arg\,min}_{w \in W} \left\{ \varphi(w) + d(w,u) \right\},$$

then

$$\varphi(w) + d(w,u) \geq \varphi(w^+) + d(w^+,u) + d(w,w^+).$$

We are ready to prove Theorem 9 and Theorem 10.

Proof We start with the inequality in Equation (15). Using Equation (12) with $\mu = 1$ gives

$$F(w_{i+1}) \leq \hat{\ell}_i(w_{i+1}) + (L+\beta_i)d(w_{i+1},w_i) + \frac{\|q_i\|_*^2}{2\beta_i}. \tag{17}$$

Now using Lemma 11 with $\varphi(w) = \hat{\ell}_i(w)$ yields

$$\hat{\ell}_i(w_{i+1}) + (L+\beta_i)d(w_{i+1},w_i) \leq \hat{\ell}_i(w^\star) + (L+\beta_i)d(w^\star,w_i) - (L+\beta_i)d(w^\star,w_{i+1}).$$

Combining with Equation (17) gives

$$\begin{aligned}
F(w_{i+1}) \;&\leq\; \hat{\ell}_i(w^\star) + (L+\beta_i)d(w^\star,w_i) - (L+\beta_i)d(w^\star,w_{i+1}) + \frac{\|q_i\|_*^2}{2\beta_i} \\
&= \ell_i(w^\star) + (L+\beta_i)d(w^\star,w_i) - (L+\beta_{i+1})d(w^\star,w_{i+1}) + (\beta_{i+1}-\beta_i)d(w^\star,w_{i+1}) \\
&\quad + \frac{\|q_i\|_*^2}{2\beta_i} + \langle q_i, w^\star - w_i \rangle \\
&\leq F(w^\star) + (L+\beta_i)d(w^\star,w_i) - (L+\beta_{i+1})d(w^\star,w_{i+1}) + (\beta_{i+1}-\beta_i)D^2 \\
&\quad + \frac{\|q_i\|_*^2}{2\beta_i} + \langle q_i, w^\star - w_i \rangle,
\end{aligned}$$

where in the last inequality, we used the definition of D^2 and the assumption that $\beta_{i+1} \geq \beta_i$. Summing the above inequality from $i = 1$ to $i = m-1$, we have

$$\begin{aligned}
\sum_{i=2}^{m} F(w_i) \;\;&\leq\;\; (m-1)F(w^\star) + (L+\beta_1)d(w^\star,w_1) - (L+\beta_m)d(w^\star,w_m) + (\beta_m-\beta_1)D^2 \\
&\qquad + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} + \sum_{i=1}^{m-1} \langle q_i, w^\star - w_i \rangle.
\end{aligned}$$

Notice that $d(w^\star,w_i) \geq 0$ and $d(w^\star,w_1) \leq D^2$, so we have

$$\sum_{i=2}^{m} F(w_i) \leq (m-1)F(w^\star) + (L+\beta_m)D^2 + \sum_{i=1}^{m-1} \frac{\|q_i\|_*^2}{2\beta_i} + \sum_{i=1}^{m-1} \langle q_i, w^\star - w_i \rangle.$$

The rest of the proof for Theorem 9 is similar to that for the dual averaging method (see arguments following Equation (16)).

Finally we prove Theorem 10. From the proof of Theorem 9 above, we see that if $\beta_i = \beta_m$ is a constant for all $i = 1,\dots,m$, then we have

$$\mathbb{E}\sum_{i=2}^{m}(F(w_i) - F(w^\star)) \le (L+\beta_m)d(w^\star, w_1) + \frac{\sigma^2}{2}\sum_{i=1}^{m-1}\frac{1}{\beta_i}.$$

Notice that for the above result, we do not need to assume boundedness of W, nor boundedness of the Bregman divergence $d(u,v)$. Since we use $w_1 = \arg\min_{w\in W} h(w)$ and assume $h(w_1) = 0$ (without loss of generality), it follows $d(w^\star, w_1) \le h(w^\star)$. Plugging in $\beta_m = (\sigma/\sqrt{2h(w^\star)})\sqrt{m}$ gives the desired result. ■

Appendix B. High-Probability Bounds

For simplicity, the theorems stated throughout the paper involved bounds on the expected regret, $\mathbb{E}[R(m)]$. A stronger type of result is a high-probability bound, where $R(m)$ itself is bounded with arbitrarily high probability $1-\delta$, and the bound having only logarithmic dependence on δ. Here, we demonstrate how our theorems can be extended to such high-probability bounds.

First, we need to justify that the expected regret bounds for the online prediction rules discussed in Appendix A have high-probability versions. For simplicity, we will focus on a high-probability version of the regret bound for dual averaging (Theorem 7), but exactly the same technique will work for stochastic mirror descent (Theorem 9 and Theorem 10). With these results in hand, we will show how our main theorem for distributed learning using the DMB algorithm (Theorem 4) can be extended to a high-probability version. Identical techniques will work for the other theorems presented in the paper.

Before we begin, we will need to make a few additional mild assumptions. First, we assume that there are positive constants B, G such that $|f(w,z)| \le B$ and $\|\nabla_w f(w,z)\| \le G$ for all $w \in W$ and $z \in \mathcal{Z}$. Second, we assume that there is a positive constant $\hat{\sigma}$ such that $\mathrm{Var}_z(f(w,z) - f(w^\star,z)) \le \hat{\sigma}^2$ for all $w \in W$ (note that $\hat{\sigma}^2 \le 4B^2$ always holds). Third, that W has a bounded diameter D, namely $\|w - w'\| \le D$ for all $w, w' \in W$.

Under these assumptions, we can show the following high-probability version of Theorem 7.

Theorem 12 *For any m and any $\delta \in (0,1]$, the regret of the stochastic dual averaging method is bounded with probability at least $1-\delta$ over the sampling of $z_1,\dots,z_m$ by*

$$\begin{aligned}R(m) \le\ & (F(w_1) - F(w^\star)) + (L+\beta_m)h(w^\star) + \frac{\sigma^2}{2}\sum_{i=1}^{m-1}\frac{1}{\beta_i}\\ &+ 2\log(2/\delta)\left(DG + \frac{2G^2}{\beta_1}\right)\sqrt{1 + 36\frac{G^2\sigma^2\sum_{i=1}^{m}\frac{1}{\beta_i^2} + D^2\sigma^2 m}{\log(2/\delta)}}\\ &+ 4\log(2/\delta)B\sqrt{1 + \frac{18m\hat{\sigma}^2}{\log(2/\delta)}}.\end{aligned}$$

Proof The proof of the theorem is identical to the one of Theorem 7, up to Equation (16):

$$\sum_{i=2}^{m}\big(F(w_i)-F(w^\star)\big) \le (L+\beta_m)h(w^\star)+\sum_{i=1}^{m-1}\frac{\|q_i\|^2}{2\beta_i}+\sum_{i=1}^{m-1}\langle q_i, w^\star - w_i\rangle. \tag{18}$$

In the proof of Theorem 7, we proceeded by taking expectations of both sides with respect to the sequence $z_1,\ldots,z_m$. Here, we will do things a bit differently.

The main technical tool we use is a well-known Bernstein-type inequality for martingales (e.g., Cesa-Bianchi and Lugosi, 2006, Lemma A.8), an immediate corollary of which can be stated as follows: suppose $x_1,\ldots,x_m$ is a martingale difference sequence with respect to the sequence $z_1,\ldots,z_m$, such that $|x_i| \le b$, and let

$$v=\sum_{i=1}^{m}\mathrm{Var}(x_i|z_1,\ldots,z_{i-1}).$$

Then for any $\delta\in(0,1)$, it holds with probability at least $1-\delta$ that

$$\sum_{i=1}^{m}x_i \le b\log(1/\delta)\sqrt{1+\frac{18v}{\log(1/\delta)}}. \tag{19}$$

Recall the definition $q_i=\nabla f(w_i,z_i)-\nabla F(w_i)$, and let $\sigma_i^2=\mathbb{E}[\|q_i\|^2]$. Note that $\sigma_i^2\le\sigma^2$. We will first use this result for the sequence

$$x_i=\frac{\|q_i\|^2-\sigma_i^2}{2\beta_i}+\langle q_i, w^\star-w_i\rangle.$$

It is easily seen that $\mathbb{E}_{z_i}[x_i|z_1,\ldots,z_{i-1}]=0$, so it is indeed a martingale difference sequence w.r.t. $z_1,\ldots,z_m$. Moreover, $|\langle q_i,w^\star-w_i\rangle|\le D\|q_i\|\le 2DG$, $\|q_i\|^2\le 4G^2$. In terms of the variances, let Var_{z_i} and $\mathbb{E}_{z_i}$ be shorthand for the variance (resp. expectation) over z_i conditioned over $z_1,\ldots,z_{i-1}$. Then

$$\begin{aligned}\mathrm{Var}_{z_i}(x_i) &\le 2\mathrm{Var}_{z_i}\left(\frac{\|q_i\|^2-\sigma_i^2}{2\beta_i}\right)+2\mathrm{Var}_{z_i}\left(\langle q_i,w^\star-w_i\rangle\right)\\ &\le \frac{1}{2}\mathbb{E}_{z_i}\left(\frac{\|q_i\|^4}{\beta_i^2}\right)+2\mathbb{E}_{z_i}[(\langle q_i,w^\star-w_i\rangle)^2]\\ &\le 2G^2\mathbb{E}_{z_i}\left(\frac{\|q_i\|^2}{\beta_i^2}\right)+2\|w^\star-w_i\|^2\mathbb{E}_{z_i}[\|q_i\|^2]\\ &\le 2G^2\frac{\sigma_i^2}{\beta_i^2}+2D^2\sigma_i^2 \;\le\; 2G^2\frac{\sigma^2}{\beta_i^2}+2D^2\sigma^2.\end{aligned}$$

Combining these observations with Equation (19), we get that with probability at least $1-\delta$,

$$\sum_{i=1}^{m-1}\frac{\|q_i\|^2-\sigma^2}{\beta_i}+\langle q_i,w^\star-w_i\rangle \le \left(2DG+\frac{4G^2}{\beta_1}\right)\log(1/\delta)\sqrt{1+36\frac{G^2\sigma^2\sum_{i=1}^{m}\frac{1}{\beta_i^2}+D^2\sigma^2 m}{\log(1/\delta)}}. \tag{20}$$

A similar type of bound can be derived for the sequence $x_i=(f(w_i,z_i)-f(w^\star,z_i))-(F(w_i)-F(w^\star))$. It is easily verified to be a martingale difference sequence w.r.t. $z_1,\ldots,z_m$, since

$$\mathbb{E}\left[(f(w_i,z_i)-f(w^\star,z_i))-(F(w_i)-F(w^\star))\,|z_1,\ldots,z_{i-1}\right]=0.$$

Also,

$$|(f(w_i,z_i)-f(w^\star,z_i))-(F(w_i)-F(w^\star))|\le 4B,$$

and

$$\begin{aligned}\mathrm{Var}_{z_i}\left(\big(f(w_i,z_i)-f(w^\star,z_i)\big)-\big(F(w_i)-F(w^\star)\big)\right) &= \mathrm{Var}_{z_i}\big(f(w_i,z_i)-f(w^\star,z_i)\big)\\ &\le \hat{\sigma}^2\ .\end{aligned}$$

So again using Equation (19), we have that with probability at least $1-\delta$ that

$$\sum_{i=1}^{m}(f(w_i,z_i)-f(w^\star,z_i))-(F(w_i)-F(w^\star))\le 4B\log(1/\delta)\sqrt{1+\frac{18m\hat{\sigma}^2}{\log(1/\delta)}}\ . \qquad (21)$$

Finally, adding $F(w_1)-F(w^\star)$ to both sides of Equation (18), and combining Equation (20) and Equation (21) with a union bound, the result follows. ■

Comparing the theorem to Theorem 7, and assuming that $\beta_i=\Theta(\sqrt{i})$, we see that the bound has additional $O(\sqrt{m})$ terms. However, the bound retains the important property of having the dominant terms multiplied by the variances $\sigma^2,\hat{\sigma}^2$. Both variances become smaller in the mini-batch setting, where the update rules are applied over averages of b such functions and their gradients. As we did earlier in the paper, let us think of this bound as an abstract function $\psi(\sigma^2,\hat{\sigma}^2,\delta,m)$. Notice that now, the regret bound also depends on the function variance $\hat{\sigma}^2$, and the confidence parameter δ.

Theorem 13 *Let f is an L-smooth convex loss function. Assume that the stochastic gradient $\nabla_w f(w,z_i)$ is bounded by a constant and has σ^2-bounded variance for all i and all w, and that $f(w,z_i)$ is bounded by a constant and has $\hat{\sigma}^2$-bounded variance for all i and for all w. If the update rule ϕ has a serial high-probability regret bound $\psi(\sigma^2,\hat{\sigma}^2,\delta,m)$. then with probability at least $1-\delta$, the total regret of Algorithm 3 over m examples is at most*

$$(b+\mu)\psi\left(\frac{\sigma^2}{b},\frac{\hat{\sigma}^2}{b},\delta,1+\frac{m}{b+\mu}\right)+O\left(\hat{\sigma}\sqrt{\left(1+\frac{\mu}{b}\right)\log(1/\delta)m}\right)\ .$$

Comparing the obtained bound to the one in Theorem 4, we note that we pay an additional $O(\sqrt{m})$ factor.

Proof The proof closely resembles the one of Theorem 4. We let $\bar{z}_j$ denote the first b inputs on batch j, and define $\bar{f}$ as the average loss on these inputs. Note that for any w, the variance of $\bar{f}(w,\bar{z}_j)$ is at most $\hat{\sigma}^2/b$, and the variance of $\nabla_w\bar{f}(w,z)$ is at most σ^2/b. Therefore, with probability at least $1-\delta$, it holds that

$$\sum_{j=1}^{\bar{m}}\big(\bar{f}(w_j,\bar{z}_j)-\bar{f}(w^\star,\bar{z}_j)\big)\le\psi\left(\frac{\sigma^2}{b},\frac{\hat{\sigma}^2}{b},\delta,\bar{m}\right)\ . \qquad (22)$$

where $\bar{m}$ is the number of inputs given to the update rule ϕ. Let Z_j denote the set of all examples received between the commencement of batch j and the commencement of batch $j+1$, including the vector-sum phase in between ($b+\mu$ examples overall). In the proof of Theorem 4, we had that

$$\mathbb{E}\left[\big(\bar{f}(w_j,\bar{z}_j)-\bar{f}(w^\star,\bar{z}_j)\big)\,|\,w_j\right]\ =\ \mathbb{E}\left[\frac{1}{b+\mu}\sum_{z\in Z_j}(f(w_j,z_i)-f(w^\star,z_i))\ \Big|\ w_j\right]\ ,$$

and thus the *expected value* of the left-hand side of Equation (22) equals the total regret, divided by $b+\mu$. Here, we need to work a bit harder. To do so, note that the sequence of random variables

$$\left(\frac{1}{b}\sum_{z\in\bar{z}_j}\big(f(w_j,z)-f(w^\star,z)\big)\right)-\left(\frac{1}{b+\mu}\sum_{z\in Z_j}\big(f(w_j,z)-f(w^\star,z)\big)\right),$$

indexed by j, is a martingale difference sequence with respect to $Z_1,Z_2,\ldots$. Moreover, conditioned on $Z_1,\ldots,Z_{j-1}$, the variance of each such random variable is at most $4\hat{\sigma}^2/b$. To see why, note that the first sum has conditional variance $\hat{\sigma}^2/b$, since the summands are independent and each has variance $\hat{\sigma}^2$. Similarly, the second sum has conditional variance $\hat{\sigma}^2/(b+\mu)\leq\hat{\sigma}^2/b$. Applying the Bernstein-type inequality for martingales discussed in the proof of Theorem 12, we get that with probability at least $1-\delta$,

$$\sum_{j=1}^{\bar{m}}\frac{1}{b+\mu}\sum_{z\in Z_j}\big(f(w_j,z)-f(w^\star,z)\big)\leq\sum_{j=1}^{\bar{m}}\frac{1}{b}\sum_{z\in\bar{z}_j}\big(f(w_j,z)-f(w^\star,z)\big)+O\left(\hat{\sigma}\sqrt{\frac{\bar{m}\log(1/\delta)}{b}}\right),$$

where the O-notation hides only a (linear) dependence on the absolute bound over $|f(w,z)|$ for all w,z, that we assume to hold.

Combining this and Equation (22) with a union bound, we get that with probability at least $1-\delta$,

$$\sum_{j=1}^{\bar{m}}\sum_{z\in Z_j}\big(f(w_j,z)-f(w^\star,z)\big)\leq(b+\mu)\psi\left(\frac{\sigma^2}{b},\frac{\hat{\sigma}^2}{b},\delta,\frac{m}{b+\mu}\right)+O\left((b+\mu)\hat{\sigma}\sqrt{\frac{\bar{m}\log(1/\delta)}{b}}\right).$$

If $b+\mu$ divides m, then $\bar{m}=m/(b+\mu)$, and we get a bound of the form

$$(b+\mu)\psi\left(\frac{\sigma^2}{b},\frac{\hat{\sigma}^2}{b},\delta,\frac{m}{b+\mu}\right)+O\left(\hat{\sigma}\sqrt{\left(1+\frac{\mu}{b}\right)\log(1/\delta)m}\right).$$

Otherwise, we repeat the ideas of Theorem 3 to get the regret bound. ■

References

J. Abernethy, A. Agarwal, A. Rakhlin, and P. L. Bartlett. A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, August 1993.

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*, 2011.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 713–720, 2011.

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2873–2898, 2009.

J. Duchi, A. Agarwal, and M. Wainwright. Distributed dual averaging in networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 550–558, 2010.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. Technical report, Department of Industrial and System Engineering, University of Florida, Gainesville, FL, 2010.

K. Gimpel, D. Das, and N. A. Smith. Distributed asynchronous online learning for natural language processing. In *Proceedings of the Fourth Conference on Computational Natural Language Learning*, pages 213–222, 2010.

R. Goebel and R. T. Rockafellar. Local strong convexity and local Lipschitz continuity of the gradient of convex functions. *Journal of Convex Analysis*, 15(2):263–270, 2008.

J. L. Gustafson. Reevaluating Amdahl's Law. *Communications of the ACM*, 31(5):532–533, 1988.

E. Harrington, R. Herbrich, J. Kivinen, J. Platt, and R. C. Williamson. Online Bayes point machines. In *Proceedings of the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 241–252, 2003.

C. Hu, J. T. Kwok, and W. Pan. Accelerated gradient methods for stochastic optimization and online learning. In Y. Bengio, D. Schuurmans, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 781–789, 2009.

A. Juditsky, A. Nemirovski, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1:1–42, 2011.

G. Lan. An optimal method for stochastic composite optimization. Technical report, Georgia Institute of Technology, 2009.

G. Lan, Z. Lu, and R. D. C. Monteiro. Primal-dual first-order methods with $O(1/\varepsilon)$ iteration-complexity for cone programming. *Mathematical Programming*, 126:1–29, 2011.

J. Langford, A. J. Smola, and M. Zinkevich. Slow learners are fast. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 2331–2339, 2009.

A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

A. Nedić, D. P. Bertsekas, and V. S. Borkar. Distributed asynchronous incremental subgradient methods. In D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, Studies in Computational Mathematics, pages 381–407. Elsevier, 2001.

A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Series in Discrete Mathematics. Wiley-Interscience, 1983.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

Y. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103: 127–152, 2005.

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, August 2009.

Y. Nesterov and J.-Ph. Vial. Confidence level solutions for stochastic programming. *Automatica*, 44(6):1559–1568, 2008.

B. T. Polyak. *Introduction to Optimization*. Translations Series in Mathematics and Engineering. Optimization Software, Inc., New York, 1987.

R. T. Rockafellar and R. J-B Wets. On the interchange of subdifferentiation and conditional expectation for convex functionals. *Stochastics: An International Journal of Probability and Stochastic Processes*, 7(3):173–182, 1982.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 807–814, 2007.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Submitted to *SIAM Journal on Optimization*, 2008.

J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

R. J-B Wets. Stochastic programming. In G. Nemhauser and A. Rinnnooy Kan, editors, *Handbook for Operations Research and Management Sciences*, volume 1, pages 573–629. Elsevier Science Publishers, Amsterdam, The Netherlands, 1989.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 928–936, Washington DC, 2003.

M. Zinkevich, M. Weimer, A. Smola, and L. Li. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2595–2603, 2010.

Journal of Machine Learning Research 13 (2012) 203-225 Submitted 9/10; Revised 12/11; Published 1/12

Active Clustering of Biological Sequences*

Konstantin Voevodski KVODSKI@GOOGLE.COM
Google†
76 Ninth Avenue, 10th Floor
New York, NY 10011, USA

Maria-Florina Balcan NINAMF@CC.GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Heiko Röglin HEIKO@ROEGLIN.ORG
Department of Computer Science
University of Bonn
Bonn, Germany

Shang-Hua Teng SHANGHUA@USC.EDU
Computer Science Department
University of Southern California
Los Angeles, CA 90089, USA

Yu Xia YUXIA@BU.EDU
Bioinformatics Program and Department of Chemistry
Boston University
Boston, MA 02215, USA

Editor: Rocco Servedio

Abstract

Given a point set S and an unknown metric d on S, we study the problem of efficiently partitioning S into k clusters while querying few distances between the points. In our model we assume that we have access to *one versus all* queries that given a point $s \in S$ return the distances between s and all other points. We show that given a natural assumption about the structure of the instance, we can efficiently find an accurate clustering using only $O(k)$ distance queries. Our algorithm uses an *active* selection strategy to choose a small set of points that we call landmarks, and considers only the distances between landmarks and other points to produce a clustering. We use our procedure to cluster proteins by sequence similarity. This setting nicely fits our model because we can use a fast sequence database search program to query a sequence against an entire data set. We conduct an empirical study that shows that even though we query a small fraction of the distances between the points, we produce clusterings that are close to a desired clustering given by manual classification.

Keywords: clustering, active clustering, k-median, approximation algorithms, approximation stability, clustering accuracy, protein sequences

$*$. A preliminary version of this article appeared under the title *Efficient Clustering with Limited Distance Information* in the Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, AUAI Press, Corvallis, Oregon, 632-641.

†. Most of this work was completed at Boston University.

1. Introduction

Clustering from pairwise distance information is an important problem in the analysis and exploration of data. It has many variants and formulations and it has been extensively studied in many different communities, and many different clustering algorithms have been proposed.

Many application domains ranging from computer vision to biology have recently faced an explosion of data, presenting several challenges to traditional clustering techniques. In particular, computing the distances between all pairs of points, as required by traditional clustering algorithms, has become infeasible in many application domains. As a consequence it has become increasingly important to develop effective clustering algorithms that can operate with limited distance information.

In this work we initiate a study of clustering with limited distance information; in particular we consider clustering with a small number of *one versus all* queries. We can imagine at least two different ways to query distances between points. One way is to ask for distances between pairs of points, and the other is to ask for distances between one point and all other points. Clearly, a one versus all query can be implemented as n pairwise queries, where n is the size of the point set, but we draw a distinction between the two because the former is often significantly faster in practice if the query is implemented as a database search.

Our main motivating example for considering one versus all distance queries is sequence similarity search in biology. A program such as BLAST (Altschul et al., 1990) (Basic Local Alignment Search Tool) is optimized to search a single sequence against an entire database of sequences. On the other hand, performing n pairwise sequence alignments takes several orders of magnitude more time, even if the pairwise alignment is very fast. The disparity in runtime is due to the hashing that BLAST uses to identify regions of similarity between the input sequence and sequences in the database. The program maintains a hash table of all *words* in the database (substrings of a certain length), linking each word to its locations. When a query is performed, BLAST considers each word in the input sequence, and runs a local sequence alignment in each of its locations in the database. Therefore the program only performs a limited number of local sequence alignments, rather than aligning the input sequence to each sequence in the database. Of course, the downside is that we never consider alignments between sequences that do not share a word. However, in this case an alignment may not be relevant anyway, and we can assign a distance of infinity to the two sequences. Even though the search performed by BLAST is heuristic, it has been shown that protein sequence similarity identified by BLAST is meaningful (Brenner et al., 1998).

Motivated by such scenarios, in this paper we consider the problem of clustering a data set with an unknown distance function, given only the capability to ask one versus all distance queries. We design an efficient algorithm for clustering accurately with a small number of such queries. To formally analyze the correctness of our algorithm we assume that the distance function is a metric, and that our clustering problem satisfies a natural approximation stability property with respect to the k-median objective function for clustering. In particular, our analysis assumes the (c,ε) approximation stability property of Balcan et al. (2009). For an objective function Φ (such as k-median), the (c,ε)-property assumes that any clustering that is a c-approximation of Φ is structurally close to some "target" clustering C_T (has error of at most ε with respect to C_T). Given this assumption, our goal is to find a clustering that is structurally close to the target (has error of at most ε), which is what we call an *accurate* clustering.

Our first main contribution is designing an algorithm that given the $(1+\alpha,\varepsilon)$-property for the k-median objective finds an accurate clustering with probability at least $1-\delta$ by using only $O(k+\ln\frac{1}{\delta})$ one versus all queries. Our analysis requires that the clusters of the target clustering have size at least $O(\varepsilon n/\alpha)$. In particular, we use the same assumption as Balcan et al. (2009), and we obtain effectively the same performance guarantees as Balcan et al. but by only using a very small number of one versus all queries. In addition to handling this more difficult scenario, we also provide a much faster algorithm. The algorithm of Balcan et al. (2009) can be implemented in $O(n^3)$ time, where n is the size of the point set, while the one proposed here runs in time $O((k+\ln\frac{1}{\delta})n\log n)$.

Our algorithm uses an *active* selection strategy to choose a small set of landmark points. In each iteration our *Landmark-Selection* procedure chooses one of the farthest points from the ones chosen already, where distance from a point s to a set X is given by $\min_{x\in X} d(s,x)$. This procedure is motivated by the observation that if we select points that are far from all the points chosen already, we can quickly cover all the dense regions of the data set. At the same time, our procedure uses some randomness to avoid choosing outliers. After selecting a small set of landmarks, we use a robust single-linkage clustering procedure that we call *Expand-Landmarks*, which constructs a clustering linking only the landmarks that have $s_{\min}$ points in an r-ball around them, for an appropriate choice of $s_{\min}$ and increasing estimates of r. After our initial work a similar robust single-linkage clustering algorithm has been used in Chaudhuri and Dasgupta (2010), which is a generalization of a procedure presented in Wishart (1969). Our algorithm uses only the distances between landmarks and other points to compute a clustering. Therefore the number of one versus all distance queries required is equivalent to the number of landmarks.

The runtime of our algorithm is $O(|L|n\log n)$, where L is the set of landmarks that have been selected. Our adaptive selection procedure significantly reduces the time and query complexity of the algorithm. We show that using our adaptive procedure it suffices to choose only $O(k+\ln\frac{1}{\delta})$ landmarks to compute an accurate clustering with probability at least $1-\delta$. If we use a non-adaptive selection strategy and simply choose landmarks uniformly at random, we must sample a point from each ground truth cluster and therefore need at least $O(k\ln\frac{k}{\delta})$ landmarks to find an accurate clustering. More exactly, the non-adaptive selection strategy requires $O(\frac{n}{\Delta}\ln\frac{k}{\delta})$ landmarks, where Δ is the size of the smallest ground truth cluster. Therefore if we simply choose landmarks uniformly at random, performance can degrade significantly if some clusters are much smaller than the average cluster size. Our theoretic assumption does require that the ground truth clusters are large, but $O(\varepsilon n/\alpha)$ can still be much smaller than the average cluster size, in which case our adaptive selection procedure gives a more significant improvement in runtime and query complexity of the algorithm.

We use our algorithm to cluster proteins by sequence similarity, and compare our results to gold standard manual classifications given in the Pfam (Finn et al., 2010) and SCOP (Murzin et al., 1995) databases. These classification databases are used ubiquitously in biology to observe evolutionary relationships between proteins and to find close relatives of particular proteins. We find that for one of these sources we obtain clusterings that usually closely match the given classification, and for the other the performance of our algorithm is comparable to that of the best known algorithms using the full distance matrix. Both of these classification databases have limited coverage, so a completely automated method such as ours can be useful in clustering proteins that have yet to be classified. Moreover, our method can cluster very large data sets because it is efficient and does not require the full distance matrix as input, which may be infeasible to obtain for a very large data set.

1.1 Related Work

A theoretical assumption that is related to the (c,ϵ)-property is ϵ-separability, which is used in Ostrovsky et al. (2006). This property is also referred to as irreducibility in Badoiu et al. (2002) and Kumar et al. (2005). A clustering instance is ϵ-separated if the cost of the optimal k-clustering is at most ϵ^2 times the cost of the optimal clustering using $k-1$ clusters. The ϵ-separability and (c,ϵ) properties are related: in the case when the clusters are large the Ostrovsky et al. (2006) condition implies the Balcan et al. (2009) condition (see Balcan et al., 2009).

Ostrovsky et al. also present a sampling method for choosing initial centers, which when followed by a single Lloyd-type descent step gives a constant factor approximation of the k-means objective if the instance is ϵ-separated. However, their sampling method needs information about the full distance matrix because the probability of picking two points as the first two cluster centers is proportional to their squared distance. A very similar (independently proposed) strategy is used by Arthur and Vassilvitskii (2007) to obtain an $O(\log k)$-approximation of the k-means objective on arbitrary instances. Their work was further extended by Ailon et al. (2009) to give a constant factor approximation using $O(k \log k)$ centers. The latter two algorithms can be implemented with k and $O(k \log k)$ one versus all distance queries, respectively.

Awasthi et al. (2010) have since improved the approximation guarantee of Ostrovsky et al. (2006) and some of the results of Balcan et al. (2009). In particular, they show a way to arbitrarily closely approximate the k-median and k-means objective when the Balcan et al. (2009) condition is satisfied and all the target clusters are large. In their analysis they use a property called weak deletion-stability, which is implied by the Ostrovsky et al. (2006) condition and the Balcan et al. (2009) condition when the target clusters are large. However, in order to find a c-approximation (and given our assumption a clustering that is ϵ-close to the target) the runtime of their algorithm is $n^{O(1/(c-1)^2)}k^{O(1/(c-1))}$. On the other hand, the runtime of our algorithm is completely independent of c, so it remains efficient even when the (c,ϵ)-property holds only for some very small constant c.

Approximate clustering using sampling has been studied extensively in recent years (see Mishra et al., 2001; Ben-David, 2007; Czumaj and Sohler, 2007). The methods proposed in these papers yield constant factor approximations to the k-median objective using at least $O(k)$ one versus all distance queries. However, as the constant factor of these approximations is at least 2, the proposed sampling methods do not necessarily yield clusterings close to the target clustering C_T if the (c,ϵ)-property holds only for some small constant $c < 2$, which is the interesting case in our setting.

Clustering using coresets is another approach that can be effective in the limited information setting. A coreset is a small representative sample D of the original point set P, which has the property that computational problems on P can be reduced to problems on D. In particular, Feldman and Langberg (2011) give ways to construct coresets such that a $(1+\alpha)$-approximation to the k-median problem on D gives a $(1+\alpha)$-approximation on the full data set. Therefore by using these coresets we can find a $(1+\alpha)$-approximation to the k-median problem using a number of one-versus-all distance queries that is equal to the size of the coreset. However, the size of the coreset of Feldman and Langberg (2011) is $O(k \log(1/\alpha)/\alpha^3)$, so this approach may require significantly more queries to find an accurate clustering in our model if the (c,ϵ)-property holds only for some very small constant c.

Our landmark selection strategy is related to the *farthest first traversal* used by Dasgupta (2002). In each iteration this traversal selects the point that is farthest from the ones chosen so far, where as in our algorithm the distance from a point s to a set X is given by $\min_{x \in X} d(s,x)$. This traversal was originally used by Gonzalez (1985) to give a 2-approximation to the k-center problem. The same procedure is also used in the FastMap algorithm in Faloutsos and Lin (1995) as a heuristic to find a pair of distant objects. Farthest first traversal is used in Dasgupta (2002) to produce a hierarchical clustering where for each k the induced k-clustering is a constant factor approximation of the optimal k-center clustering. Our selection strategy is somewhat different from this traversal because in each iteration we uniformly at random choose one of the farthest points from the ones selected so far. In addition, the theoretical guarantees we provide are quite different from those of Gonzales and Dasgupta.

To our knowledge, our work is the first to provide theoretical guarantees for active clustering (clustering by adaptively using only some of the distances between the objects) under a natural condition on the input data. Following the initial publication of this work, Eriksson et al. (2011) have provided another active clustering procedure with theoretical guarantees for hierarchical clustering under a different condition on the input data.

2. Preliminaries

Given a metric space $M = (X, d)$ with point set X, an unknown distance function d satisfying the triangle inequality, and a set of points $S \subseteq X$ with cardinality n, we would like to find a k-clustering C that partitions the points in S into k sets $C_1, \ldots, C_k$ by using *one versus all* distance queries.

In our analysis we assume that S satisfies the (c, ε)-property of Balcan et al. (2009) for the k-median objective function. The k-median objective is to minimize $\Phi(C) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)$, where c_i is the median of cluster C_i, which is the point $y \in C_i$ that minimizes $\sum_{x \in C_i} d(x, y)$. Let $\mathrm{OPT}_\Phi = \min_C \Phi(C)$, where the minimum is over all k-clusterings of S, and denote by $C^* = \{C_1^*, \ldots, C_k^*\}$ a clustering achieving this value.

To formalize the (c, ε)-property we need to define a notion of distance between two k-clusterings $C = \{C_1, \ldots, C_k\}$ and $C' = \{C'_1, \ldots, C'_k\}$. As in Balcan et al. (2009), we define the distance between C and C' as the fraction of points on which they disagree under the optimal matching of clusters in C to clusters in C':

$$\mathrm{dist}(C, C') = \min_{f \in F_k} \frac{1}{n} \sum_{i=1}^{k} |C_i - C'_{f(i)}|,$$

where F_k is the set of bijections $f \colon \{1, \ldots, k\} \to \{1, \ldots, k\}$. Two clusterings C and C' are ε-*close* if $\mathrm{dist}(C, C') < \varepsilon$.

We assume that there exists some unknown relevant "target" clustering C_T and given a proposed clustering C we define the error of C with respect to C_T as $\mathrm{dist}(C, C_T)$. Our goal is to find a clustering of low error.

The (c, ε)-property is defined as follows.

Definition 1 *We say that the instance (S, d) satisfies the (c, ε)-property for the k-median objective function with respect to the target clustering C_T if any clustering of S that approximates* OPT_Φ *within a factor of c is ε-close to C_T, that is, $\Phi(C) \leq c \cdot \mathrm{OPT}_\Phi \Rightarrow \mathrm{dist}(C, C_T) < \varepsilon$.*

In the analysis of the next section we denote by c_i^* the center point of C_i^*, and use OPT to refer to the value of C^* using the k-median objective, that is, $\mathrm{OPT} = \Phi(C^*)$. We define the *weight* of point x to be the contribution of x to the k-median objective in C^*: $w(x) = \min_i d(x, c_i^*)$. Similarly, we use $w_2(x)$ to denote x's distance to the second-closest cluster center among $\{c_1^*, c_2^*, \ldots, c_k^*\}$. In addition, let w be the average weight of the points: $w = \frac{1}{n} \sum_{x \in S} w(x) = \frac{\mathrm{OPT}}{n}$.

3. Clustering With Limited Distance Information

Algorithm 1 Landmark-Clustering($S, \alpha, \varepsilon, \delta, k$)

```
$b = (1 + 17/\alpha)\varepsilon n$;
$q = 2b$;
iter $= 4k + 16 \ln \frac{1}{\delta}$;
$s_{\min} = b + 1$;
$n' = n - b$;
$L$ = **Landmark-Selection**($q$, iter, $S$);
$C'$ = **Expand-Landmarks** ($k, s_{\min}, n', L, S$);
Choose some working landmark $l_i$ from each cluster $C'_i$;
**for** each $x \in S$ **do**
  Insert $x$ into the cluster $C''_j$ for $j$ = argmin$_i d(x, l_i)$;
**end for**
**return** $C''$;
```

In this section we present a new algorithm that accurately clusters a set of points assuming that the clustering instance satisfies the (c, ε)-property for $c = 1 + \alpha$, and the clusters in the target clustering C_T are not too small. The algorithm presented here is much faster than the one given by Balcan et al., and does not require all pairwise distances as input. Instead, we only require $O(k + \ln \frac{1}{\delta})$ one versus all distance queries to achieve the same performance guarantee as in Balcan et al. (2009) with probability at least $1 - \delta$.

Our clustering method is described in Algorithm 1. We start by using the *Landmark-Selection* procedure to *adaptively* select a small set of landmarks. This procedure repeatedly chooses uniformly at random one of the q farthest points from the ones selected so far (for an appropriate q), where the distance from a point s to a set X is given by $\min_{x \in X} d(s, x)$. We use $d_{\min}(s)$ to refer to the minimum distance between s and any point selected so far. Each time we select a new landmark l, we use a one versus all distance query to get the distances between l and all other points in the data set, and update $d_{\min}(s)$ for each point $s \in S$. To select a new landmark in each iteration, we choose a random number $i \in \{n - q + 1, \dots, n\}$ and use a linear time selection algorithm to select the ith farthest point. The complete description of this procedure is given in Algorithm 2. We note that our algorithm uses only the distances between landmarks and other points to produce a clustering.

Expand-Landmarks then expands a ball B_l around each landmark $l \in L$. We use the variable r to denote the radius of all the balls: for each landmark $l \in L$, $B_l = \{s \in S \mid d(s, l) \leq r\}$. For each ball there are at most n relevant values of r, each adding at least one more point to it, which results in at most $|L|n$ values of r to try in total. We call a landmark l *working* if B_l contains *at least* $s_{\min}$ points. The algorithm maintains a graph $G_B = (V_B, E_B)$, where $v_l \in V_B$ represents the ball B_l around working landmark l, and two vertices are connected by an (undirected) edge if the corresponding balls overlap on any point: $(v_{l_1}, v_{l_2}) \in E_B$ iff $B_{l_1} \cap B_{l_2} \neq \emptyset$. We emphasize that this graph considers only the balls that have *at least* $s_{\min}$ points in them. In addition, we maintain the set of points in these balls Clustered $= \{s \in S \mid \exists l \colon s \in B_l \text{ and } v_l \in V_B\}$, and a list of the connected components of G_B, which we refer to as Components$(G_B) = \{\text{Comp}_1, ..., \text{Comp}_m\}$.

In each iteration we expand one of the balls by a point, and update G_B, Components(G_B), and Clustered. If G_B has exactly k components, and $|\text{Clustered}| \geq n'$, we terminate and report a clustering

Algorithm 2 Landmark-Selection(q, iter, S)

```
Choose $l \in S$ uniformly at random;
$L = \{l\}$;
for each $d(l,s) \in$ QUERY-ONE-VS-ALL$(l,S)$ do
    $d_{\min}(s) = d(l,s)$;
end for
for $i$ = 1 to iter $- 1$ do
    Let $s_1, ..., s_n$ be an ordering of points in $S$ such that $d_{\min}(s_j) \leq d_{\min}(s_{j+1})$ for $j \in \{1, \ldots, n-1\}$;
    Choose $l \in \{s_{n-q+1}, \ldots, s_n\}$ uniformly at random;
    $L = L \cup \{l\}$;
    for each $d(l,s) \in$ QUERY-ONE-VS-ALL$(l,S)$ do
        if $d(l,s) < d_{\min}(s)$ then
            $d_{\min}(s) = d(l,s)$;
        end if
    end for
end for
return $L$;
```

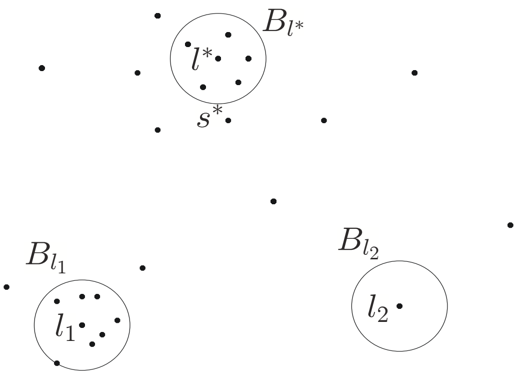


Figure 1: Balls around landmarks are displayed, with the next point to be added to a ball labeled as s^*.

that has a cluster C_i for each component Comp$_i$, where each C_i contains points in balls in Comp$_i$. If this condition is never satisfied, we report **no-cluster**. A sketch of this algorithm is given in Algorithm 3. In our description Expand-Ball() is an abstraction for expanding one of the balls by a single point, which is performed by finding the next closest landmark-point pair (l^*, s^*), and adding s^* to B_{l^*} (see Figure 1). In Section 4 we give a full description of an efficient implementation of *Expand-Landmarks*.

The last step of our algorithm takes the clustering C' returned by *Expand-Landmarks* and improves it. We compute a set L' that contains exactly one working landmark from each cluster $C'_i \in C'$ (any working landmark is sufficient), and assign each point $x \in S$ to the cluster corresponding to the closest landmark in L'.

We now present our main theoretical guarantee for Algorithm 1.

Algorithm 3 Expand-Landmarks($k, s_{\min}, n', L, S$)

```
r = 0;
while (($l^*, s^*$) = Expand-Ball()) != null do
    $r = d(l^*, s^*)$;
    update $G_B$, Components($G_B$), and Clustered;
    if |Components($G_B$)| $= k$ and |Clustered| $\geq n'$ then
        return $C = \{C_1, ..., C_k\}$ where $C_i = \{s \in S \mid \exists l\colon s \in B_l$ and $v_l \in \text{Comp}_i\}$;
    end if
end while
return no-cluster;
```

Theorem 2 *Given a metric space $M = (X, d)$, where d is unknown, and a set of points $S \subseteq X$, if the instance (S, d) satisfies the $(1+\alpha, \varepsilon)$-property for the k-median objective function and if each cluster in the target clustering C_T has size at least $(4 + 51/\alpha)\varepsilon n$, then Landmark-Clustering outputs a clustering that is ε-close to C_T with probability at least $1-\delta$ in time $O((k + \ln\frac{1}{\delta})|S|\log|S|)$ using $O(k + \ln\frac{1}{\delta})$* one versus all *distance queries.*

Before we prove the theorem, we will introduce some notation and use an analysis similar to the one in Balcan et al. (2009) to argue about the structure of the clustering instance that follows from our assumptions. Let $\varepsilon^* = \text{dist}(C_T, C^*)$. By our assumption that the k-median clustering of S satisfies the $(1+\alpha, \varepsilon)$-property we have $\varepsilon^* < \varepsilon$. Because each cluster in the target clustering has at least $(4 + 51/\alpha)\varepsilon n$ points, and the *optimal k-median clustering* C^* differs from the target clustering by $\varepsilon^* n \leq \varepsilon n$ points, each cluster in C^* must have at least $(3 + 51/\alpha)\varepsilon n$ points.

Let us define the *critical distance* $d_{\text{crit}} = \frac{\alpha w}{17\varepsilon}$. We call a point x *good* if both $w(x) < d_{\text{crit}}$ and $w_2(x) - w(x) \geq 17 d_{\text{crit}}$, else x is called *bad*. In other words, the *good* points are those points that are close to their own cluster center and far from any other cluster center. In addition, we will break up the *good* points into *good sets* X_i, where X_i is the set of the *good* points in the optimal cluster C_i^*. So each set X_i is the "core" of the optimal cluster C_i^*.

Note that the distance between two points $x, y \in X_i$ satisfies $d(x,y) \leq d(x, c_i^*) + d(c_i^*, y) = w(x) + w(y) < 2d_{\text{crit}}$. In addition, the distance between any two points in different good sets is greater than $16 d_{\text{crit}}$. To see this, consider a pair of points $x \in X_i$ and $y \in X_{j \neq i}$. The distance from x to y's cluster center c_j^* is at least $17 d_{\text{crit}}$. By the triangle inequality, $d(x,y) \geq d(x, c_j^*) - d(y, c_j^*) > 17 d_{\text{crit}} - d_{\text{crit}} = 16 d_{\text{crit}}$.

If the k-median instance (M, S) satisfies the $(1+\alpha, \varepsilon)$-property with respect to C_T, and each cluster in C_T has size at least $2\varepsilon n$, then

1. less than $(\varepsilon - \varepsilon^*)n$ points $x \in S$ on which C_T and C^* agree have $w_2(x) - w(x) < \frac{\alpha w}{\varepsilon}$.

2. at most $17\varepsilon n/\alpha$ points $x \in S$ have $w(x) \geq \frac{\alpha w}{17\varepsilon}$.

The first part is proved by Balcan et al. (2009). The intuition is that if too many points on which C_T and C^* agree are close enough to the second-closest center among $\{c_1^*, c_2^*, \ldots, c_k^*\}$, then we can move them to the clusters corresponding to those centers, producing a clustering that is structurally far from C_T, but whose objective value is close to OPT, violating the $(1+\alpha, \varepsilon)$-property. The second part follows from the fact that $\sum_{x \in S} w(x) = OPT = wn$.

Then using these facts and the definition of ε^* it follows that at most $\varepsilon^* n + (\varepsilon - \varepsilon^*)n + 17\varepsilon n/\alpha = \varepsilon n + 17\varepsilon n/\alpha = (1+17/\alpha)\varepsilon n = b$ points are bad. Hence each $|X_i| = |C_i^* \backslash B| \geq (2+34/\alpha)\varepsilon n = 2b$.

In the remainder of this section we prove that given this structure of the clustering instance, *Landmark-Clustering* finds an accurate clustering. We first show that almost surely the set of landmarks returned by *Landmark-Selection* has the property that each of the cluster cores has a landmark near it, which we refer to as the *landmark spread* property. We then argue that given a set of such landmarks, *Expand-Landmarks* finds a partition C' that clusters most of the points in each core correctly. We conclude with the proof of the theorem, which argues that the clustering returned by the last step of our procedure is a further improved clustering that is very close to C^* and C_T.

The *Landmark-Clustering* algorithm first uses *Landmark-Selection*(q, iter, S) to choose a set of landmarks. We say that the *landmark spread* property holds if there is a landmark closer than $2d_{\text{crit}}$ to some point in each good set. The following lemma proves that for $q = 2b$ after selecting only $\text{iter} = O(k + \ln\frac{1}{\delta})$ points the chosen landmarks will have this property with probability at least $1-\delta$.

Lemma 3 *Given a set of landmarks L = Landmark-Selection $(2b, 4k+16\ln\frac{1}{\delta}, S)$, the landmark spread property is satisfied with probability at least* $1-\delta$.

Proof Because there are at most b bad points and in each iteration we uniformly at random choose one of $2b$ points, the probability that a good point is added to L is at least 1/2 in each iteration. Using a Chernoff bound, we show in Lemma 4 that the probability that fewer than k good points have been added to L after $t > 2k$ iterations is less than $e^{-t(1-\frac{2k}{t})^2/4}$. For $t = 4k + 16\ln\frac{1}{\delta}$

$$e^{-t(1-\frac{2k}{t})^2/4} < e^{-(4k+16\ln\frac{1}{\delta})0.5^2/4} < e^{-16\ln\frac{1}{\delta}/16} = \delta.$$

Therefore after $t = 4k + 16\ln\frac{1}{\delta}$ iterations this probability is smaller than δ.

We argue that once we select k good points using our procedure, one of them must be closer than $2d_{\text{crit}}$ to some point in each good set. As in Algorithm 2, we use $d_{\min}(s)$ to denote the minimum distance from point s to any landmark that has been selected so far: $d_{\min}(s) = \min_{l \in L'} d(l,s)$, where L' is the set of landmarks that have been selected so far.

There are two possibilities regarding the first k good points added to L: we select them from distinct good sets, or at least two points are selected from the same good set. If the former is true, the *landmark spread* property trivially holds. If the latter is true, consider the first time that a second point is chosen from the same good set X_i. Let us call these two points x and y (they may be the same point), and assume that y is chosen after x. The distance between x and y must be less than $2d_{\text{crit}}$ because they are in the same good set. Therefore when y is chosen, $d_{\min}(y) \leq d(x,y) < 2d_{\text{crit}}$. Moreover, y is chosen from $\{s_{n-2b+1}, ..., s_n\}$, where $d_{\min}(s_j) \leq d_{\min}(s_{j+1})$. Therefore when y is chosen, at least $n-2b+1$ points $s \in S$ (including y) satisfy $d_{\min}(s) \leq d_{\min}(y) < 2d_{\text{crit}}$. Because each good set satisfies $|X_i| \geq 2b$, it follows that there must be a landmark closer than $2d_{\text{crit}}$ to some point in each good set.

■

Lemma 4 *The probability that fewer than k good points have been chosen as landmarks after t* $> 2k$ *iterations of Landmark-Selection is less than* $e^{-t(1-\frac{2k}{t})^2/4}$.

Proof Let X_i be an indicator random variable defined as follows: $X_i = 1$ if the point chosen in iteration i is a good point, and 0 otherwise. Let $X = \sum_{i=1}^{t} X_i$, and μ be the expectation of X. In other words, X is the number of good points chosen after t iterations of the algorithm, and μ is its expected value.

Because in each round we uniformly at random choose one of $2b$ points and there are at most b bad points in total, $\mathrm{E}[X_i] \geq 1/2$ and hence $\mu \geq t/2$. By the Chernoff bound, for any $\delta > 0$, $\Pr[X < (1-\delta)\mu] < e^{-\mu\delta^2/2}$.

If we set $\delta = 1 - \frac{2k}{t}$, we have $(1-\delta)\mu = (1-(1-\frac{2k}{t}))\mu \geq (1-(1-\frac{2k}{t}))t/2 = k$. Assuming that $t \geq 2k$, it follows that $\Pr[X < k] \leq \Pr[X < (1-\delta)\mu] < e^{-\mu\delta^2/2} = e^{-\mu(1-\frac{2k}{t})^2/2} \leq e^{-t/2(1-\frac{2k}{t})^2/2}$. ■

The algorithm then uses the *Expand-Landmarks* procedure to find a k-clustering C'. The following lemma states that C' is an accurate clustering, and has an additional property that is relevant for the last part of the algorithm.

Lemma 5 *Given a set of landmarks L that satisfy the landmark spread property, Expand-Landmarks with parameters* $s_{\min} = b+1$*, and* $n' = n-b$ *returns a k-clustering* $C' = \{C'_1, C'_2, \ldots C'_k\}$ *in which each cluster contains points from a single distinct good set* X_i*. If we let* σ *be a bijection mapping each good set* X_i *to the cluster* $C'_{\sigma(i)}$ *containing points from* X_i*, the distance between* c_i^* *and any working landmark l in* $C'_{\sigma(i)}$ *satisfies* $d(c_i^*, l) < 5d_{\text{crit}}$.

Proof Lemma 6 argues that because the good sets X_i are well-separated, for $r < 4d_{\text{crit}}$ no ball of radius r can overlap (intersect) more than one X_i, and two balls that overlap different X_i cannot share any points. Lemma 7 argues that because there is a landmark near each good set, there is a value of $r^* < 4d_{\text{crit}}$ such that each X_i is contained in some ball of radius r^*. Moreover, because we only consider balls that have more than b points in them, and the number of bad points is at most b, each ball in G_B must overlap some good set. We can use these facts to argue for the correctness of the algorithm. We refer to the clustering computed in line 6 of *Expand-Landmarks* as the clustering *induced* by G_B, in which each cluster contains points in balls that are in the same component in G_B.

First we observe that for $r = r^*$, G_B has exactly k components and each good set X_i is contained within a distinct component of the induced clustering. Each ball in G_B overlaps with some X_i, and because $r^* < 4d_{\text{crit}}$, we know that each ball in G_B overlaps with exactly one X_i. We also know that balls that overlap different X_i cannot share any points and are thus not connected in G_B. Therefore balls that overlap different X_i will be in different components in G_B. Moreover, each X_i is contained in some ball of radius r^*. For each good set X_i let us designate by B_i a ball that contains all the points in X_i (Figure 2), which is in G_B because the size of each good set satisfies $|X_i| > b$. Any ball in G_B that overlaps X_i will be connected to B_i, and will thus be in the same component as B_i. Therefore for $r = r^*$, G_B has exactly k components, one for each good set X_i, with the corresponding cluster containing all the points in X_i.

There are at least $n-b$ points that are in some X_i, therefore for $r = r^*$ the number of clustered points is at least $n-b$. Hence for $r = r^*$ the condition in line 5 of *Expand-Landmarks* will be satisfied and in line 6 the algorithm will return a k-clustering in which each cluster contains points from a single distinct good set X_i.

Now let us suppose that we start with $r = 0$. Consider the first value of $r = r'$ for which the condition in line 5 is satisfied. At this point G_B has exactly k components and the number of points that are not clustered is at most b. It must be the case that $r' \leq r^* < 4d_{\text{crit}}$ because we know that

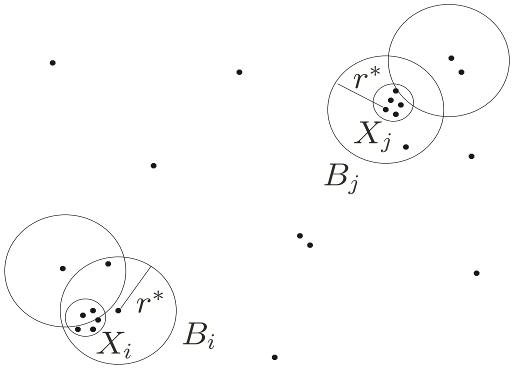


Figure 2: Balls B_i and B_j of radius r^* are shown, which contain good sets X_i and X_j, respectively. The radius of the balls is small in comparison to the distance between the good sets.

the condition is satisfied for $r = r^*$, and we are considering all relevant values of r in ascending order. As before, each ball in G_B must overlap some good set X_i. Again using Lemma 6 we argue that because $r < 4d_{\text{crit}}$, no ball can overlap more than one X_i and two balls that overlap different X_i cannot share any points. It follows that each cluster induced by G_B contains points from a single X_i (so we cannot merge the good sets). Moreover, because the size of each good set satisfies $|X_i| > b$, and there are at most b points that are not clustered, each induced cluster must contain points from a distinct X_i (so we cannot split the good sets). Thus we will return a k-clustering in which each cluster contains points from a single distinct good set X_i.

To prove the second part of the statement, let σ be a bijection matching each good set X_i to the cluster $C'_{\sigma(i)}$ containing points from X_i. Clearly, for any working landmark l in $C'_{\sigma(i)}$ it must be the case that B_l overlaps X_i. Let s^* denote any point on which B_l and X_i overlap. By the triangle inequality, the distance between c_i^* and l satisfies $d(c_i^*, l) \leq d(c_i^*, s^*) + d(s^*, l) < d_{\text{crit}} + r < 5d_{\text{crit}}$. Therefore the distance between c_i^* and any working landmark $l \in C'_{\sigma(i)}$ satisfies $d(c_i^*, l) < 5d_{\text{crit}}$. ■

Lemma 6 *A ball of radius $r < 4d_{\text{crit}}$ cannot contain points from more than one good set X_i, and two balls of radius $r < 4d_{\text{crit}}$ that overlap (intersect) different X_i cannot share any points.*

Proof To prove the first part, consider a ball B_l of radius $r < 4d_{\text{crit}}$ around landmark l. In other words, $B_l = \{s \in S \mid d(s,l) \leq r\}$. If B_l overlaps more than one good set, then it must have at least two points from different good sets $x \in X_i$ and $y \in X_j$. By the triangle inequality it follows that $d(x,y) \leq d(x,l) + d(l,y) \leq 2r < 8d_{\text{crit}}$. However, we know that $d(x,y) > 16d_{\text{crit}}$, giving a contradiction.

To prove the second part, consider two balls B_{l_1} and B_{l_2} of radius $r < 4d_{\text{crit}}$ around landmarks l_1 and l_2. In other words, $B_{l_1} = \{s \in S \mid d(s,l_1) \leq r\}$, and $B_{l_2} = \{s \in S \mid d(s,l_2) \leq r\}$. Assume that they overlap with different good sets X_i and X_j: $B_{l_1} \cap X_i \neq \emptyset$ and $B_{l_2} \cap X_j \neq \emptyset$. For the purpose of contradiction, let's assume that B_{l_1} and B_{l_2} share at least one point: $B_{l_1} \cap B_{l_2} \neq \emptyset$, and use s^* to refer to this point. By the triangle inequality, it follows that the distance between any point $x \in B_{l_1}$ and $y \in B_{l_2}$ satisfies $d(x,y) \leq d(x,s^*) + d(s^*,y) \leq [d(x,l_1) + d(l_1,s^*)] + [d(s^*,l_2) + d(l_2,y)] \leq 4r < 16d_{\text{crit}}$.

Because B_{l_1} overlaps with X_i and B_{l_2} overlaps with X_j, it follows that there is a pair of points $x \in X_i$ and $y \in X_j$ such that $d(x,y) < 16d_{\text{crit}}$, a contradiction. Therefore if B_{l_1} and B_{l_2} overlap different good sets, $B_{l_1} \cap B_{l_2} = \emptyset$. ■

Lemma 7 *Given a set of landmarks L that satisfy the landmark spread property, there is some value of $r^* < 4d_{\text{crit}}$ such that each X_i is contained in some ball B_l around landmark $l \in L$ of radius r^*.*

Proof For each good set X_i choose a point $s_i \in X_i$ and a landmark $l_i \in L$ that satisfy $d(s_i, l_i) < 2d_{\text{crit}}$. The distance between l_i and each point $x \in X_i$ satisfies $d(l_i, x) \leq d(l_i, s_i) + d(s_i, x) < 2d_{\text{crit}} + 2d_{\text{crit}} = 4d_{\text{crit}}$. Consider $r^* = \max_{l_i} \max_{x \in X_i} d(l_i, x)$. Clearly, each X_i is contained in a ball B_{l_i} of radius r^* and $r^* < 4d_{\text{crit}}$. ■

Lemma 8 *Suppose the distance between c_i^* and any working landmark l in $C'_{\sigma(i)}$ satisfies $d(c_i^*, l) < 5d_{\text{crit}}$. Then given a point $x \in C_i^*$ that satisfies $w_2(x) - w(x) \geq 17d_{\text{crit}}$, for any working landmark $l_1 \in C'_{\sigma(i)}$ and any working landmark $l_2 \in C'_{\sigma(j \neq i)}$ it must be the case that $d(x, l_1) < d(x, l_2)$.*

Proof We will show that $d(x, l_1) < w(x) + 5d_{\text{crit}}$, and $d(x, l_2) > w(x) + 12d_{\text{crit}}$. This implies that $d(x, l_1) < d(x, l_2)$.

To prove the former, by the triangle inequality $d(x, l_1) \leq d(x, c_i^*) + d(c_i^*, l_1) = w(x) + d(c_i^*, l_1) < w(x) + 5d_{\text{crit}}$.

To prove the latter, by the triangle inequality $d(x, l_2) \geq d(x, c_j^*) - d(l_2, c_j^*)$. Because $d(x, c_j^*) \geq w_2(x)$ and $d(l_2, c_j^*) < 5d_{\text{crit}}$, we have

$$d(x, l_2) > w_2(x) - 5d_{\text{crit}}. \tag{1}$$

Moreover, because $w_2(x) - w(x) \geq 17d_{\text{crit}}$, we have

$$w_2(x) \geq 17d_{\text{crit}} + w(x). \tag{2}$$

Combining Equations 1 and 2 it follows that $d(x, l_2) > 17d_{\text{crit}} + w(x) - 5d_{\text{crit}} = w(x) + 12d_{\text{crit}}$. ■

Proof [Theorem 2] After using Landmark-Selection to choose $O(k + \ln \frac{1}{\delta})$ points, with probability at least $1 - \delta$ there is a landmark closer than $2d_{\text{crit}}$ to some point in each good set. Given a set of landmarks with this property, each cluster in the clustering $C' = \{C'_1, C'_2, \ldots C'_k\}$ output by *Expand-Landmarks* contains points from a single distinct good set X_i. This clustering can exclude up to b points, all of which may be good. Nonetheless, this means that C' may disagree with C^* on only the bad points and at most b good points. The number of points that C' and C^* disagree on is therefore at most $2b = O(\varepsilon n/\alpha)$. Thus, C' is at least $O(\varepsilon/\alpha)$-close to C^*, and at least $O(\varepsilon/\alpha + \varepsilon)$-close to C_T.

Moreover, C' has an additional property that allows us to find a clustering that is ε-close to C_T. If we use σ to denote a bijection mapping each good set X_i to the cluster $C'_{\sigma(i)}$ containing points from X_i, any working landmark $l \in C'_{\sigma(i)}$ is closer than $5d_{\text{crit}}$ to c_i^*. We can use this observation to find all points that satisfy one of the properties of the good points: points x such that $w_2(x) - w(x) \geq 17d_{\text{crit}}$. Let us call these points the *detectable* points. To clarify, the detectable points are those points that are much closer to their own cluster center than to any other cluster center in C^*, and the *good* points are a subset of the detectable points that are also very close to their own cluster center.

To find the detectable points using C', we choose some working landmark l_i from each C'_i. For each point $x \in S$, we then insert x into the cluster C''_j for $j = \operatorname{argmin}_i d(x, l_i)$. Lemma 8 argues that each detectable point in C^*_i is closer to every working landmark in $C'_{\sigma(i)}$ than to any working landmark in $C'_{\sigma(j \neq i)}$. It follows that C'' and C^* agree on all the detectable points. Because there are fewer than $(\varepsilon - \varepsilon^*)n$ points on which C_T and C^* agree that are not detectable, it follows that $\text{dist}(C'', C_T) < (\varepsilon - \varepsilon^*) + \text{dist}(C_T, C^*) = (\varepsilon - \varepsilon^*) + \varepsilon^* = \varepsilon$.

Therefore using $O(k + \ln \frac{1}{\delta})$ landmarks we compute an accurate clustering with probability at least $1 - \delta$. The runtime of *Landmark-Selection* is $O(|L|n)$ if we use a linear time selection algorithm to select the next point in each iteration, where $|L|$ is the number of landmarks. Using a min-heap to store all landmark-point pairs and a disjoint-set data structure to keep track of the connected components of G_B, *Expand-Landmarks* can be implemented in $O(|L|n \log n)$ time. A detailed description of this implementation is given in Section 4. The last part of our procedure takes $O(kn)$ time, so the overall runtime of our algorithm is $O(|L|n \log n)$. Therefore to compute an accurate clustering with probability at least $1 - \delta$ the runtime of our algorithm is $O((k + \ln \frac{1}{\delta})n \log n)$. Moreover, we only consider the distances between the landmarks and other points, so we only use $O(k + \ln \frac{1}{\delta})$ one versus all distance queries. ■

4. Implementation of Expand-Landmarks

In order to efficiently expand balls around landmarks, we build a min-heap H of landmark-point pairs (l, s), where the key of each pair is the distance between l and s. In each iteration we find $(l^*, s^*) = H$.deleteMin(), and then add s^* to items(l^*), which stores the points in B_{l^*}. We store points that have been clustered (points in balls of size at least $s_{\min}$) in the set Clustered.

Our implementation assigns each clustered point s to a "representative" landmark, denoted by $lm(s)$. The representative landmark of s is the landmark l of the first large ball B_l that contains s. To efficiently update the components of G_B, we maintain a disjoint-set data structure U that contains sets corresponding to the connected components of G_B, where each ball B_l is represented by landmark l. In other words, U contains a set $\{l_1, l_2, l_3\}$ iff $B_{l_1}, B_{l_2}, B_{l_3}$ form a connected component in G_B.

For each large ball B_l our algorithm will consider all points $s \in B_l$ and perform Update-Components(l, s), which works as follows. If s does not have a representative landmark we assign it to l, otherwise s must already be in $B_{lm(s)}$, and we assign B_l to the same component as $B_{lm(s)}$. If none of the points in B_l are assigned to other landmarks, it will be in its own component. A detailed description of the algorithm is given in Algorithm 4.

During the execution of the algorithm the connected components of G_B must correspond to the sets of U (where each ball B_l is represented by landmark l). Lemma 9 argues that if B_{l_1} and B_{l_2} are *directly* connected in G_B, l_1 and l_2 must be in the same set in U. It follows that whenever B_{l_1} and B_{l_2} are in the same connected component in G_B, l_1 and l_2 will be in the same set in U. Moreover, if B_{l_1} and B_{l_2} are not in the same component in G_B, then l_1 and l_2 cannot be in the same set in U because both start in distinct sets (line 22), and it is not possible for a set containing l_1 to be merged with a set containing l_2.

Algorithm 4 Expand-Landmarks($k, s_{\min}, n', L, S$)

```
A = ();
for each $s \in S$ do
    $lm(s)$ = null;
    for each $l \in L$ do
        A.add($((l,s), d(l,s))$);
    end for
end for
$H$ = build-heap($A$);
for each $l \in L$ do
    items($l$) = ();
end for
Set Clustered = ();
U = ();
while $H$.hasNext() do
    $(l^*, s^*)$ = $H$.deleteMin();
    items($l^*$).add($s^*$);
    if items($l^*$).size() $> s_{\min}$ then
        Update-Components($l^*, s^*$);
        Clustered.add($s*$);
    end if
    if items($l^*$).size() == $s_{\min}$ then
        $U$.MakeSet($l^*$);
        for each $s \in$ items($l^*$) do
            Update-Components($l^*, s$);
            Clustered.add($s$);
        end for
    end if
    if Clustered.size() $\geq n'$ and $U$.size() == $k$ then
        return  Format-Clustering();
    end if
end while
return  no-cluster;
```

Algorithm 5 Update-Components(l, s)

```
if $lm(s)$ == null then
    $lm(s)$ = $l$;
else
    $c_1$ = $U$.find($l$);
    $c_2$ = $U$.find($lm(s)$);
    $U$.union($c_1, c_2$);
end if
```

Algorithm 6 Format-Clustering()

```
C = ();
for each Set $L'$ in $U$ do
    Set Cluster = ();
    for each $l \in L'$ do
        for each $s \in$ items($l$) do
            Cluster.add($s$);
        end for
    end for
    $C$.add(Cluster);
end for
return $C$;
```

Lemma 9 *If balls B_{l_1} and B_{l_2} are directly connected in G_B, then landmarks l_1 and l_2 must be in the same set in U.*

Proof If B_{l_1} and B_{l_2} are directly connected in G_B, then B_{l_1} and B_{l_2} must overlap on some point s. Without loss of generality, suppose s is added to B_{l_1} before it is added to B_{l_2}. When s is added to B_{l_1}, $lm(s) = l_1$ if s does not yet have a representative landmark (lines 1-2 of Update-Components), or $lm(s) = l'$ and both l_1 and l' are put in the same set (lines 4-6 of Update-Components). When s is added to B_{l_2}, if $lm(s) = l_1$, then l_1 and l_2 will be put in the same set in U. If $lm(s) = l'$, l' and l_2 will be put in the same set in U, which also contains l_1. ■

It takes $O(|L|n)$ time to build H (linear in the size of the heap). Each deleteMin() operation takes $O(\log(|L|n))$ (logarithmic in the size of the heap), which is equivalent to $O(\log(n))$ because $|L| \leq n$. If U is implemented by a union-find algorithm, Update-Components takes amortized time of $O(\alpha(|L|)$, where α denotes the inverse Ackermann function. Moreover, Update-Components may be called at most once for each iteration of the while loop in Expand-Landmarks (for a pair (l^*, s^*) it is either called on line 18 if B_{l^*} is large enough, or it is called on line 24 when B_{l^*} grows large enough). All other operations also take time proportional to the number of landmark-point pairs. So the runtime of this algorithm is $O(|L|n)$ + iter $\cdot\, O(\log n + \alpha(|L|))$, where iter is the number of iterations of the while loop. As the number of iterations is bounded by $|L|n$, and $\alpha(|L|)$ is effectively constant, this gives a worst-case running time of $O(|L|n \log n)$.

5. Empirical Study

We use our *Landmark Clustering* algorithm to cluster proteins using sequence similarity. As mentioned in the Introduction, one versus all distance queries are particularly relevant in this setting because of sequence database search programs such as BLAST (Altschul et al., 1990) (Basic Local Alignment Search Tool). BLAST aligns a queried sequence to sequences in the database, and produces a "bit score" for each alignment, which is a measure of its quality (we invert the bit score to make it a distance). However, BLAST does not consider alignments with some of the sequences in the database, in which case we assign distances of infinity to the corresponding sequences. We observe that if we define distances in this manner they almost form a metric in practice: when we draw triplets of sequences at random and check the distances between them the triangle inequality is almost always satisfied. Moreover, BLAST is very successful at detecting sequence homology in large sequence databases, therefore it is plausible that k-median clustering using these distances is approximately stable with respect to a relevant target clustering C_T, which groups together sequences with shared evolutionary ancestry.

We perform experiments on data sets obtained from two classification databases: Pfam (Finn et al., 2010), version 24.0, October 2009; and SCOP (Murzin et al., 1995), version 1.75, June 2009. Both of these sources classify proteins by their evolutionary relatedness, therefore we can use their classifications as a ground truth to evaluate the clusterings produced by our algorithm and other methods.

Pfam classifies proteins using hidden Markov models (HMMs) that represent multiple sequence alignments. There are two levels in the Pfam classification hierarchy: family and clan. In our clustering experiments we compare with a classification at the family level because the relationships at the clan level are less likely to be discerned with sequence alignment. In each experiment we randomly select several large families (of size between 1000 and 10000) from Pfam-A (the manually curated part of the classification), retrieve the sequences of the proteins in these families, and use our *Landmark-Clustering* algorithm to cluster the data set.

SCOP groups proteins on the basis of their 3D structures, so it only classifies proteins whose structure is known. Thus the data sets from SCOP are much smaller in size. The SCOP classification is also hierarchical: proteins are grouped by class, fold, superfamily, and family. We consider the classification at the superfamily level because this seems most appropriate given that we are only using sequence information. As with the Pfam data, in each experiment we create a data set by randomly choosing several superfamilies (of size between 20 and 200), retrieve the sequences of the corresponding proteins, and use our *Landmark-Clustering* algorithm to cluster the data set.

Once we cluster a particular data set, we compare the clustering to the manual classification using the distance measure from the theoretical part of our work. To find the fraction of misclassified points under the optimal matching of clusters in C to clusters in C' we solve a minimum weight bipartite matching problem where the cost of matching C_i to $C'_{f(i)}$ is $|C_i - C'_{f(i)}|/n$. In addition, we compare clusterings to manual classifications using the F-measure, which is used in another study that clusters protein sequences (Paccanaro et al., 2006). The F-measure is a similarity score between 0 and 1, where 1 indicates an exact match between the two clusterings (see Appendix A). This measure has also been used in other studies (see Cheng et al., 2006), and is related to our notion of clustering distance (see Lemma 10 in Appendix A). Surprisingly, the F-measure is not symmetric; in our experiments we compute the similarity of a clustering C to the manual classification C_M as $F(C_M, C)$.

5.1 Choice of Parameters

To run *Landmark-Clustering*, we set k using the number of clusters in the ground truth clustering. For each Pfam data set we use $5k$ landmarks/queries, and for each SCOP data set we use $10k$ landmarks/queries. In addition, our algorithm uses three parameters $(q, s_{\min}, n')$ whose value is set in the proof based on α and ε, assuming that the clustering instance satisfies the $(1+\alpha, \varepsilon)$-property. In practice we must choose some value for each parameter. In our experiments we set q as a function of the average size of the ground truth clusters (ave-size), $s_{\min}$ as a function of the size of the smallest ground truth cluster (min-size), and n' as a function of the number of points in the data set. For the

Pfam data sets we set $q =$ ave-size, $s_{\min} = 0.25 \cdot$ min-size, and $n' = 0.7n$. Because the selection of landmarks is randomized, for each data set we compute several clusterings, compare each to the ground truth, and report the median quality.

Landmark-Clustering is most sensitive to the $s_{\min}$ parameter, and will not report a clustering if $s_{\min}$ is too small or too large. We recommend trying several values of this parameter, in increasing or decreasing order, until one gets a clustering and none of the clusters are too large. If the user gets a clustering where one of the clusters is very large, this likely means that several ground truth clusters have been merged. This may happen because $s_{\min}$ is too small causing balls of outliers to connect different cluster cores, or $s_{\min}$ is too large causing balls intersecting different cluster cores to overlap.

In our SCOP experiments we have to use the above-mentioned heuristic to set the $s_{\min}$ parameter. We start with $s_{\min} =$ min-size, and decrement it until we get exactly k clusters and none of the clusters are too large (larger than twice the size of the largest ground truth cluster). For the SCOP data sets we set q = ave-size, and $n' = 0.5n$. As before, for each data set we compute several clusterings, compare each to the ground truth, and report the median quality.

Our algorithm is less sensitive to the n' parameter. However, if the user sets n' too large some ground truth clusters may be merged, so we recommend using a smaller value ($0.5n \leq n' \leq 0.7n$) because all of the points are still clustered during the last step. Again, for some values of n' the algorithm may not output a clustering, or output a clustering where some of the clusters are too large.

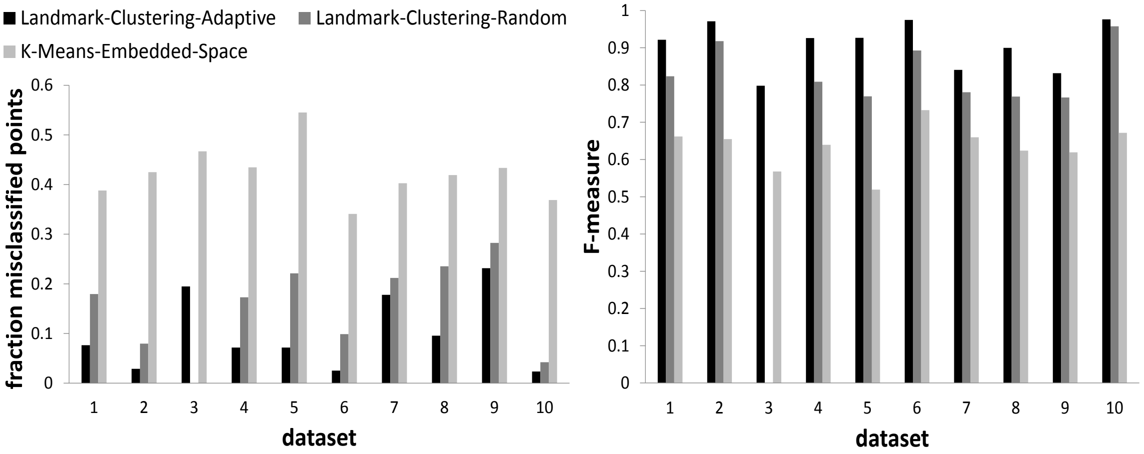
It is important to not choose an extreme value for the q parameter. The value of q must be large enough to avoid repeatedly choosing outliers (if $q = 1$ we are likely to choose an outlier in each iteration), but small enough to quickly find a landmark near each cluster core. If we set $q = n$, the algorithm selects landmarks uniformly at random, and we may need significantly more landmarks to choose one from each cluster core by chance.

In our experiments we compare the algorithm that uses the adaptive selection strategy with the alternative that chooses landmarks uniformly at random. The alternative algorithm uses exactly the same number of landmarks, and other parameters stay the same as well. When the data has the structure that follows from our assumptions, the non-adaptive selection strategy may require significantly more landmarks to cover all cluster cores (especially if the sizes of the ground truth clusters are not well-balanced). Therefore when the data has the right structure and we cannot afford to use many landmarks, we expect to find more accurate clusterings with the adaptive selection strategy.

5.2 Results

Figure 3 shows the results of our experiments on the Pfam data sets. As discussed earlier, to test our adaptive landmark selection strategy we compare our algorithm, which is labeled *Landmark-Clustering-Adaptive*, with the same algorithm that chooses landmarks uniformly at random, which we refer to as *Landmark-Clustering-Random*. We can see that for a lot of the data sets *Landmark-Clustering-Adaptive* finds a clustering that is quite close to the ground truth. The alternative algorithm does not perform as well, and for data set 3 fails to find a clustering altogether.

The Pfam data sets are very large, so as a benchmark for comparison we can only consider algorithms that use a comparable amount of distance information (because we do not have the full distance matrix). A natural choice is the following algorithm: uniformly at random choose a set of



(a) Comparison using fraction of misclassified points

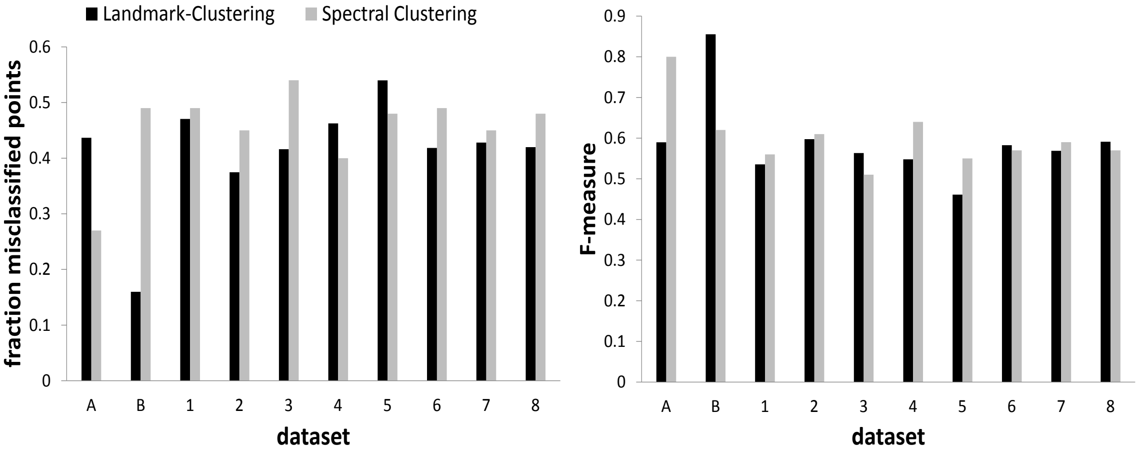
(b) Comparison using the F-measure

Figure 3: Comparing the performance of *Landmark-Clustering-Adaptive*, *Landmark-Clustering-Random*, and k-means in the embedded space on 10 data sets from Pfam. Data sets *1-10* are created by randomly choosing 8 families from Pfam of size s, $1000 \leq s \leq 10000$. *(a)* Comparison using the distance measure from the theoretical part of our work. *(b)* Comparison using the F-measure.

landmarks L, $|L| = d$; embed each point in a d-dimensional space using distances to L; use k-means clustering in this space (with distances given by the Euclidean norm). This embedding scheme is a Lipschitz embedding with singleton subsets (see Tang and Crovella, 2003), which gives distances with low distortion for points near each other in a metric space.

Notice that this procedure uses exactly d one versus all distance queries, so we can set d equal to the number of queries used by our algorithm. We expect this algorithm to work well, and if we look at Figure 3 we can see that it finds reasonable clusterings. Still, the clusterings reported by this algorithm do not closely match the Pfam classification, showing that our results are indeed significant.

Figure 4 shows the results of our experiments on the SCOP data sets. For these data sets we find less accurate clusterings, which is likely because the SCOP classification is based on biochemical and structural evidence in addition to sequence evidence. By contrast, the Pfam classification is based entirely on sequence information. Still, because the SCOP data sets are much smaller, we can compare our algorithm with methods that require distances between all the points. In particular, Paccanaro et al. (2006) show that spectral clustering using sequence similarity data works well when applied to the proteins in SCOP. Thus we use the exact method described by Paccanaro et al. (2006) as a benchmark for comparison on the SCOP data sets. Moreover, other than clustering randomly generated data sets from SCOP, we also consider the two main examples from Paccanaro et al., which are labeled A and B in the figure. From Figure 4 we can see that the performance of *Landmark-Clustering* is comparable to that of the spectral method, which is very good considering that the spectral clustering algorithm significantly outperforms other clustering algorithms on this



(a) Comparison using fraction of misclassified points

(b) Comparison using the F-measure

Figure 4: Comparing the performance of *Landmark-Clustering* and spectral clustering on 10 data sets from SCOP. Data sets *A* and *B* are the two main examples from Paccanaro et al. (2006), the other data sets (*1-8*) are created by randomly choosing 8 superfamilies from SCOP of size s, $20 \leq s \leq 200$. *(a)* Comparison using the distance measure from the theoretical part of our work. *(b)* Comparison using the F-measure.

data (Paccanaro et al., 2006). Moreover, the spectral clustering algorithm requires the full distance matrix as input, and takes much longer to run.

For the SCOP data sets we do not see any significant difference in performance when we compare the adaptive and non-adaptive landmark selection strategies. This is likely because we are using a lot of landmarks (10 times the number of clusters), and selecting landmarks uniformly at random is sufficient to cover the dense groups of points. Unfortunately for these data the algorithm has little success if we use fewer than $10k$ landmarks (it usually cannot find a clustering altogether), so we cannot test how the two selection strategies perform when we use fewer landmarks.

5.3 Testing the (c, ε)-property

To see whether approximation stability of the k-median objective function is a reasonable assumption for our data, we look at whether our data sets resemble the structure that is implied by our assumption. We do this by measuring the separation of the ground truth clusters in our data sets. For each data set in our study, we sample some points from each ground truth cluster. We then look at whether the sampled points are more similar to points in the same cluster than to points in other clusters. More specifically, for each point we record the median within-cluster similarity, and the maximum between-cluster similarity. If our data sets indeed have well-separated cluster cores, as implied by our assumption, then for a lot of the points the median within-cluster similarity should be significantly larger than the maximum between-cluster similarity. We can see that this is indeed the case for the Pfam data sets. However, this is not typically the case for the SCOP data sets, where most points have little similarity to the majority of the points in their ground truth cluster. These observations explain our results on the two sets of data: we are able to accurately cluster the Pfam data sets, and our algorithm is much less accurate on the SCOP data sets. The complete results of these experiments can be found at `http://xialab.bu.edu/resources/ac`.

Testing whether the (c,ε)-property holds for the k-median objective is an NP-complete problem (Schalekamp et al., 2010). Moreover, in our experiments when we set the parameters of the algorithm we don't preserve the relationships between them as in Algorithm 1. In particular, in our experiments when we set n' to $n - s_{\min} + 1$ as in Algorithm 1, the algorithm usually fails to report a clustering no matter what value of $s_{\min}$ we try. This means that these data sets in fact do not satisfy our exact theoretic assumptions. Still, when we only slightly break the dependence between the parameters, we are able to find accurate clusterings for the Pfam data sets. For the SCOP data sets we have to further break the dependence between the parameters, and use an additional heuristic to estimate $s_{\min}$, which is not surprising because these data do not have the structure that the algorithm exploits.

6. Conclusion and Open Questions

In this work we presented a new algorithm for clustering large data sets with limited distance information. As opposed to previous settings, our goal was not to approximate some objective function like the k-median objective, but to find clusterings close to the ground truth. We proved that our algorithm yields accurate clusterings with only a small number of one versus all distance queries, given a natural assumption about the structure of the clustering instance. This assumption has been previously analyzed by Balcan et al. (2009), but in the full distance information setting. By contrast, our algorithm uses only a small number of queries, it is much faster, and it has effectively the same formal performance guarantees as the one introduced by Balcan et al. (2009).

To demonstrate the practical use of our algorithm, we clustered protein sequences using a sequence database search program as the one versus all query. We compared our results to gold standard manual classifications of protein evolutionary relatedness given in Pfam (Finn et al., 2010) and SCOP (Murzin et al., 1995). We find that our clusterings are quite accurate when we compare with the classification given in Pfam. For SCOP our clusterings are as accurate as state of the art methods, which take longer to run and require the full distance matrix as input.

Our main theoretical guarantee assumes large target clusters. It would be interesting to design a provably correct algorithm for the case of small clusters as well. It would also be interesting to study other objective functions for clustering under similar approximation stability assumptions. In particular, Voevodski et al. (2011) study the implications of the (c,ε)-property for the *min-sum* objective function. However, the algorithm presented there is not as efficient and is less accurate in clustering protein sequences.

Acknowledgments

Konstantin Voevodski was supported by an IGERT Fellowship through NSF grant DGE-0221680 awarded to the ACES Training Program at BU Center for Computational Science. Maria Florina Balcan was supported in part by NSF grant CCF-0953192, by ONR grant N00014-09-1-0751, AFOSR grant FA9550-09-1-0538, and a Google Research Award. Heiko Röglin was supported by a Veni grant from the Netherlands Organisation for Scientific Research. Shang-Hua Teng was supported in part by NSF grant CCR-0635102.

Appendix A.

In this section we give the definition of F-measure, which is another way to compare two clusterings. We also show a relationship between our measure of distance and the F-measure.

A.1 F-measure

The F-measure compares two clusterings C and C' by matching each cluster in C to a cluster in C' using a harmonic mean of Precision and Recall, and then computing a "per-point" average. If we match C_i to C'_j, Precision is defined as $P(C_i,C'_j) = \frac{|C_i \cap C'_j|}{|C_i|}$. Recall is defined as $R(C_i,C'_j) = \frac{|C_i \cap C'_j|}{|C_j|}$. For C_i and C'_j the harmonic mean of Precision and Recall is then equivalent to $\frac{2 \cdot |C_i \cap C'_j|}{|C_i|+|C'_j|}$, which we denote by $\mathrm{pr}(C_i,C'_j)$ to simplify notation. The F-measure is then defined as

$$F(C,C') = \frac{1}{n} \sum_{C_i \in C} |C_i| \max_{C'_j \in C'} \mathrm{pr}(C_i,C'_j).$$

Note that this quantity is between 0 and 1, where 1 corresponds to an exact match between the two clusterings.

Lemma 10 *Given two clusterings C and C', if* $\mathrm{dist}(C,C') = d$ *then $F(C,C') \geq 1 - 3d/2$.*

Proof Denote by σ the optimal matching of clusters in C to clusters in C', which achieves a misclassification of dn points. We show that just considering $\mathrm{pr}(C_i,C'_{\sigma(i)})$ for each $C_i \in C$ achieves an F-measure of at least $1-3d/2$:

$$F(C,C') \geq \frac{1}{n} \sum_{C_i \in C} |C_i| \mathrm{pr}(C_i,C'_{\sigma(i)}) \geq 1 - 3d/2.$$

To see this, for a match of C_i to $C'_{\sigma(i)}$ we denote by m_i^1 the number of points that are in C_i but not in $C'_{\sigma(i)}$, and by m_i^2 the number of points that are in $C'_{\sigma(i)}$ but not in C_i: $m_i^1 = |C_i - C'_{\sigma(i)}|$, $m_i^2 = |C'_{\sigma(i)} - C_i|$. Because the total number of misclassified points is dn it follows that

$$\sum_{C_i \in C} m_i^1 = \sum_{C_i \in C} m_i^2 = dn.$$

By definition, $|C_i \cap C'_{\sigma(i)}| = |C_i| - m_i^1$. Moreover, $|C'_{\sigma(i)}| = |C'_{\sigma(i)} \cap C_i| + m_i^2 \leq |C_i| + m_i^2$. It follows that

$$\mathrm{pr}(C_i,C'_{\sigma(i)}) = \frac{2(|C_i| - m_i^1)}{|C_i| + |C'_{\sigma(i)}|} \geq \frac{2(|C_i| - m_i^1)}{2|C_i| + m_i^2} = \frac{2|C_i| + m_i^2}{2|C_i| + m_i^2} - \frac{m_i^2 + 2m_i^1}{2|C_i| + m_i^2} \geq 1 - \frac{m_i^2 + 2m_i^1}{2|C_i|}.$$

We can now see that

$$\frac{1}{n} \sum_{C_i \in C} |C_i| \mathrm{pr}(C_i,C'_{\sigma(i)}) \geq \frac{1}{n} \sum_{C_i \in C} |C_i| (1 - \frac{m_i^2 + 2m_i^1}{2|C_i|}) = \frac{1}{n} \sum_{C_i \in C} |C_i| - \frac{1}{2n} \sum_{C_i \in C} m_i^2 + 2m_i^1 = 1 - \frac{3dn}{2n}.$$

■

References

N. Ailon, R. Jaiswal, and C. Monteleoni. Streaming k-means approximation. In *Advances in Neural Information Processing Systems*, 2009.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, 2007.

P. Awasthi, A. Blum, and O. Sheffet. Stability yields a PTAS for k-median and k-means clustering. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, 2010.

M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via coresets. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002.

M. F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, 2009.

S. Ben-David. A framework for statistical clustering with constant time approximation algorithms for *k*-median and *k*-means clustering. *Machine Learning*, 66(2-3):243–257, 2007.

S. E. Brenner, C. Chothia, and T. J. Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences USA*, 95(11):6073–6078, 1998.

K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, 2010.

D. Cheng, R. Kannan, S. Vempala, and G. Wang. A divide-and-merge methodology for clustering. *ACM Transactions on Database Systems*, 31(4):1499–1525, 2006.

A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Structures and Algorithms*, 30(1-2):226–256, 2007.

S. Dasgupta. Performance guarantees for hierarchical clustering. In Jyrki Kivinen and Robert Sloan, editors, *Computational Learning Theory*, volume 2375 of *Lecture Notes in Computer Science*, pages 235–254. Springer Berlin / Heidelberg, 2002.

B. Eriksson, G. Dasarathy, A. Singh, and R. Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, 1995.

D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, 2011.

R. D. Finn, J. Mistry, J. Tate, P. Coggill, A. Heger, J. E. Pollington, O. L. Gavin, P. Gunesekaran, G. Ceric, K. Forslund, L. Holm, E. L. Sonnhammer, S. R. Eddy, and A. Bateman. The Pfam protein families database. *Nucleic Acids Research*, 38:D211–222, 2010.

T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

A. Kumar, Y. Sabharwal, and S. Sen. Linear time algorithms for clustering problems in any dimensions. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, 2005.

N. Mishra, D. Oblinger, and L Pitt. Sublinear time approximate clustering. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, 2001.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. Scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science*, 2006.

A. Paccanaro, J. A. Casbon, and M. A. S. Saqi. Spectral clustering of protein sequences. *Nucleic Acids Research*, 34(5):1571–1580, 2006.

F. Schalekamp, M. Yu, and A. van Zuylen. Clustering with or without the approximation. In *Proceedings of the 16th Annual International Computing and Combinatorics Conference*, 2010.

L. Tang and M. Crovella. Virtual landmarks for the internet. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, 2003.

K. Voevodski, M. F. Balcan, H. Röglin, S. Teng, and Y. Xia. Min-sum clustering of protein sequences with limited distance information. In *Proceedings of the 1st International Workshop on Similarity-Based Pattern Analysis and Recognition*, 2011.

D. Wishart. Mode analysis: A generalization of nearest neighbor which reduces chaining effects. In *Proceedings of the Colloquium on Numerical Taxonomy held in the University of St. Andrews*, 1969.

Journal of Machine Learning Research 13 (2012) 227-253 Submitted 3/11; Revised 9/11; Published 2/12

Multi Kernel Learning with Online-Batch Optimization[*]

Francesco Orabona[†] FRANCESCO@ORABONA.COM
Toyota Technological Institute at Chicago
6045 South Kenwood Avenue
60637 Chicago, IL, USA

Luo Jie[†] LUOJIE@YAHOO-INC.COM
Yahoo! Labs
701 First Avenue
94089 Sunnyvale, CA, USA

Barbara Caputo BCAPUTO@IDIAP.CH
Idiap Research Institute
Centre du Parc, Rue Marconi 19
1920 Martigny, Switzerland

Editor: Francis Bach

Abstract

In recent years there has been a lot of interest in designing principled classification algorithms over multiple cues, based on the intuitive notion that using more features should lead to better performance. In the domain of kernel methods, a principled way to use multiple features is the Multi Kernel Learning (MKL) approach.

Here we present a MKL optimization algorithm based on stochastic gradient descent that has a guaranteed convergence rate. We directly solve the MKL problem in the primal formulation. By having a p-norm formulation of MKL, we introduce a parameter that controls the level of sparsity of the solution, while leading to an easier optimization problem. We prove theoretically and experimentally that 1) our algorithm has a faster convergence rate as the number of kernels grows; 2) the training complexity is linear in the number of training examples; 3) very few iterations are sufficient to reach good solutions. Experiments on standard benchmark databases support our claims.

Keywords: multiple kernel learning, learning kernels, online optimization, stochastic subgradient descent, convergence bounds, large scale

1. Introduction

In recent years there has been a lot of interest in designing principled classification algorithms over multiple cues, based on the intuitive notion that using more features should lead to better performance. Moreover, besides the purpose of decreasing the generalization error, practitioners are often interested in more flexible algorithms which can perform feature selection while training. This is for instance the case when a lot of features are available but among them noisy ones are hidden. Selecting the features also improves the interpretability of the decision function.

This has been translated into various algorithms, that dates back to the ’90s (Wolpert, 1992), based on a two-layers structure. There a classifier is trained for each cue and then their outputs

[*] A preliminary version of this paper appeared in Orabona et al. (2010).
[†] Work done mainly while at Idiap Research Institute.

are combined by another classifier. This approach has been re-invented many times, with different flavors (Nilsback and Caputo, 2004; Sanderson and Paliwal, 2004; Jie et al., 2010a; Gehler and Nowozin, 2009b; Jin et al., 2010). In general, the two layers approaches use Cross-Validation (CV) methods to create the training set for the second layer (Wolpert, 1992; Gehler and Nowozin, 2009b). If the second layer is a linear classifier, these methods are equivalent to a linear combination of the single classifiers.

Focusing on the domain of the Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000), the use of multiple cues corresponds to the use of multiple kernels. Hence, instead of combining kernel classifiers, the focus of research has moved on how to build an optimal new kernel as a weighted combination of kernels.

A recent approach in this field is to use a two-stage procedure, in which the first stage finds the optimal weights to combine the kernels, using an improved definition of the kernel alignment (Cristianini et al., 2002) as a proxy of the generalization error, and a standard SVM as second stage (Cortes et al., 2010). This approach builds on the previous works on the maximization of the kernel alignment to combine kernels (Lanckriet et al., 2004). However in this approach, even if theoretically principled, the global optimality is not guaranteed, because the optimization process split in two phases.

A different approach with a joint optimization process is Multi Kernel Learning (MKL) (Lanckriet et al., 2004; Bach et al., 2004; Sonnenburg et al., 2006; Zien and Ong, 2007; Rakotomamonjy et al., 2008; Varma and Babu, 2009; Kloft et al., 2009). In MKL one solves a joint optimization problem while also learning the optimal weights for combining the kernels. MKL methods are theoretically founded, because they are based on the minimization of an upper bound of the generalization error (Kakade et al., 2009; Cortes et al., 2010), like in standard SVM. In most of these approaches the objective function is designed to impose sparsity on the weights of the combination using an l_1-norm constraint (Bach et al., 2004; Sonnenburg et al., 2006; Zien and Ong, 2007; Rakotomamonjy et al., 2008; Varma and Babu, 2009). However solving it is far more complex than training a single SVM classifier. In fact, the l_1 norm is not smooth, so it slows down the optimization process. The original MKL problem by Lanckriet et al. (2004) was cast as a semidefinite programming (SDP). SDP are known to have poor scalability, hence much of the subsequent research focused on devising more efficient optimization procedures. The first step towards practical MKL algorithms was to restrict the weights coefficients to be non-negative. In this way, it was possible to recast the problem as a much more efficient semi-infinite linear programming (SILP) (Sonnenburg et al., 2005; Rubinstein, 2005). This has allowed to solve the MKL problem with alternating optimization approaches (Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Xu et al., 2008; Nath et al., 2009), first optimizing over the kernel combination weights, with the current SVM solution fixed, then finding the SVM solution, given the current weights. One advantage of the alternating optimization approach is that it is possible to use existing efficient SVM solvers, such as Joachims (1999) and Chang and Lin (2001), for the SVM optimization step. On the other hand, for these algorithms, it is usually not possible to prove a bound on the maximum number of iterations needed, even if they are known to converge. In fact, to the best of our knowledge, none of the existing MKL algorithms provides theoretical guarantees on the convergence rate. For the same reason it is not possible to know the asymptotic computational complexity of these algorithms, and often these dependencies are estimated numerically for the specific implementation at hand. For example, multiclass MKL SILP algorithm (Sonnenburg et al., 2006; Zien and Ong, 2007) seems to depend polynomially on the number of training examples and number of classes with an exponent of ~ 2.4 and ~ 1.7 respectively. For the other algorithms these dependencies are not clear. Another disadvantage is that they need to solve the inner SVM problem till optimality. In fact, to guarantee convergence, the solution needs to be of a high enough precision so that the kernel weight gradient computation is accurate. On the other hand the learning process is usually stopped early, before reaching the optimal solution, based on the common assumption that it is enough to have an approximate solution of the optimization function. Considering the fact that the current MKL algorithms are solved based on their dual representation, this might mean being stopped far from the optimal solution (Hush et al., 2006), with unknown effects on the convergence.

An important point is that, very often, these approaches fail to improve much over the naive baseline of just summing all the kernels (Kloft et al., 2009). Recently, researchers start to realize that when the optimal Bayes classifier is not sparse, brutally imposing sparsity will hurt the generalization performance. Motivated by this, the l_p-norm constraint has been proposed (Kloft et al., 2009; Orabona et al., 2010; Vishwanathan et al., 2010), instead of l_1-norm constrain, to be able to tune the level of sparsity of the solution and to obtain an easier problem too. In particular Vishwanathan et al. (2010) derived the dual of a variation of the l_p MKL problem for $p > 1$, suited to be optimized with the popular Sequential Minimal Optimization algorithm (Platt, 1999). However even for their algorithm it is not clear how the convergence rate depends on p and how to generalize the algorithm to generic loss functions, such as the structured losses (Tsochantaridis et al., 2004). This limitation on the use of particular loss functions is common to all the recent MKL optimization algorithms. An alternative way to be able to tune the sparsity of the MKL solution, inspired by the elastic-net regularization, has been proposed by Tomioka and Suzuki (2010).

The main contribution of this paper is a new optimization algorithm to solve efficiently the l_p-MKL problem, with a guaranteed convergence rate to the optimal solution. We minimize it with a two-stage algorithm. The first one is an online initialization procedure that determines quickly the region of the space where the optimal solution lives. The second stage refines the solution found by the first stage, using a stochastic gradient descent algorithm. Bounds on the convergence rate are proved for the overall process. Notably different from the other methods, our algorithm solves the optimization problem directly in the primal formulation, in both stages. This allows us to use *any* convex loss function, as the multiclass loss proposed by Crammer and Singer (2002) or general structured losses (Tsochantaridis et al., 2004), without any change to the core of the algorithm. Using recent approaches in optimization theory, the algorithm takes advantage of the abundance of information to reduce the training time (Shalev-Shwartz and Srebro, 2008). In fact, we show that the presence of a large number of kernels helps the optimization process instead of hindering it, obtaining, theoretically and practically, a faster convergence rate with more kernels. Our algorithm has a training time that depends linearly on the number of training examples, with a convergence rate sub-linear in the number of features/kernels used, when a sparse solution is favored. At the same time, it achieves state-of-the-art performance on standard benchmark databases. We call this algorithm OBSCURE, Online-Batch Strongly Convex mUlti keRnel lEarning.

The rest of the paper presents the theory and the experimental results supporting our claims. Section 2 revises the basic definitions of Multi Kernel Learning and generalizes it to the l_p-norm formulation. Section 3 presents the OBSCURE algorithm and Section 4 shows our theoretical guarantees, while Section 5 reports experiments on categorization tasks. We conclude the paper with discussions and future works.

2. p-norm Multi Kernel Learning

In this section we first introduce formally the MKL framework and its notation, then its p-norm generalization.

2.1 Definitions

In the following we define some notations and we also introduce some concepts of convex analysis. For a more thorough introduction see, for example, Boyd and Vandenberghe (2004).

2.1.1 NOTATION

We indicate matrices and vectors with bold letters. We also introduce two notations that will help us to synthesize the following formulas. We indicate by $[w^j]_1^F := \left[w^1, w^2, \cdots, w^F\right]$, and with a bar, for example, $\bar{w}$, the vector formed by the concatenation of the F vectors w^j, hence $\bar{w} = [w^1, w^2, \cdots, w^F] = [w^j]_1^F$.

We consider closed convex functions $f : S \to \mathbb{R}$, where in the following S will always denote a proper subset of $\mathbb{X}$, an Euclidean vector space.[1] We will indicate the inner product between two vectors of $\mathbb{X}$, w and w', as $w \cdot w'$. Given a convex function $f : S \to \mathbb{R}$, its Fenchel conjugate $f^* : \mathbb{X} \to \mathbb{R}$ is defined as $f^*(u) = \sup_{v \in S}(v \cdot u - f(v))$. A generic norm of a vector $w \in \mathbb{X}$ is indicated by $\|w\|$, and its dual $\|\cdot\|_*$ is the norm defined as $\|y\|_* = \sup\{x \cdot y : \|x\| \leq 1\}$. A vector x is a subgradient of a function f at v, if $\forall u \in S, f(u) - f(v) \geq (u - v) \cdot x$. The differential set of f at v, indicated with $\partial f(v)$, is the set of all the subgradients of f at v. If f is convex and differentiable at v then $\partial f(v)$ consists of a single vector which is the gradient of f at v and is denoted by $\nabla f(v)$. A function $f : S \to \mathbb{R}$ is said to be λ-strongly convex with respect to a convex and differentiable function h iff for any $u, v \in S$ and any subgradient $\partial f(u)$, $f(v) \geq f(u) + \partial f(u) \cdot (v - u) + \lambda(h(v) - h(u) - (v - u) \cdot \nabla h(v))$, where the terms in parenthesis form the Bregman divergence between v and u of h.

2.1.2 BINARY AND MULTI-CLASS CLASSIFIERS

Let $\{x_i, y_i\}_{i=1}^N$, with $N \in \mathbb{N}, x_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$, be the training set. Consider a function $\phi(x) : \mathbb{X} \to \mathbb{H}$ that maps the samples into a high, possibly infinite, dimensional space. In the binary case $\mathbb{Y} = \{-1, 1\}$, and we use the standard setting to learn with kernels,[2] in which the prediction on a sample x is a function of the scalar product between an hyperplane w and the transformed sample $\phi(x)$. With multiple kernels, we will have F corresponding functions $\phi^j(\cdot)$, $i = 1, \cdots, F$, and F corresponding kernels $K^j(x, x')$ defined as $\phi^j(x) \cdot \phi^j(x')$.

For multiclass and structured classification $\mathbb{Y} = \{1, \ldots, M\}$, and we follow the common approach to use joint feature maps $\phi(x, y) : \mathbb{X} \times \mathbb{Y} \to \mathbb{H}$ (Tsochantaridis et al., 2004). Again, we will have F functions $\phi^j(\cdot, \cdot), i = 1, \cdots, F$, and F kernels $K^j((x, y), (x', y'))$ as $\phi^j(x, y) \cdot \phi^j(x', y')$. This definition includes the case of training M different hyperplanes, one for each class. Indeed $\phi^j(x, y)$ can be defined as

$$\phi^j(x, y) = [0, \cdots, 0, \underbrace{\phi'^j(x)}_{y}, 0, \cdots, 0],$$

[1] We allow the functions to assume infinite values, as a way to restrict their domains to proper subsets of $\mathbb{X}$. However in the following the convex functions of interest will always be evaluated on vectors that belong to their domains.

[2] For simplicity we will not use the bias here, but it can be easily added modifying the kernel definition.

where $\phi'^j(\cdot)$ is a transformation that depends only on the data. Similarly w will be composed by M blocks, $[w^1, \cdots, w^M]$. Hence, by construction, $w \cdot \phi^j(x,r) = w^r \cdot \phi'^j(x)$. According to the defined notation, $\bar{\phi}(x,y) = [\phi^1(x,y), \cdots, \phi^F(x,y)]$. These definitions allow us to have a general notation for the binary and multiclass setting.

2.1.3 LOSS FUNCTION

In the following we consider convex Lipschitz loss functions. The most commonly used loss in binary classification is the hinge loss (Cristianini and Shawe-Taylor, 2000).

$$\ell^{HL}(w,x,y) = |1 - y\bar{w} \cdot \bar{\phi}(x)|_+,$$

where $|t|_+$ is defined as $\max(t,0)$. We also consider the following multi-class loss function (Crammer and Singer, 2002; Tsochantaridis et al., 2004), that will be used to specialize our results.

$$\ell^{MC}(w,x,y) = \max_{y' \neq y} |1 - \bar{w} \cdot (\bar{\phi}(x,y) - \bar{\phi}(x,y'))|_+ \ . \tag{1}$$

This loss function is convex and it upper bounds the multi-class misclassification loss.

2.1.4 NORMS AND DUAL NORMS

For $w \in \mathbb{R}^d$ and $p \geq 1$, we denote by $\|w\|_p$ the p-norm of w, that is, $\|w\|_p = (\sum_{i=1}^d |w_i|^p)^{1/p}$.

The dual norm of $\|\cdot\|_p$ is $\|\cdot\|_q$, where p and q satisfy $1/p + 1/q = 1$. In the following p and q will always satisfy this relation.

2.1.5 GROUP NORM

It is possible to define a $(2,p)$ *group norm* $\|\bar{w}\|_{2,p}^2$ on $\bar{w}$ as

$$\|\bar{w}\|_{2,p} := \left\| \left[\|w^1\|_2, \|w^2\|_2, \cdots, \|w^F\|_2 \right] \right\|_p,$$

that is the p-norm of the vector of F elements, formed by 2-norms of the vectors w^j. The dual norm of $\|\cdot\|_{2,p}$ is $\|\cdot\|_{2,q}$ (Kakade et al., 2009). These kind of norms have been used as *block regularization* in the LASSO literature (Yuan and Lin, 2006).

2.2 Multi Kernel Learning

The MKL optimization problem was first proposed by Bach et al. (2004) and extended to multiclass by Zien and Ong (2007). It can be written as

$$\begin{aligned} &\min_{w_j} \frac{\lambda}{2} \left(\sum_{j=1}^F \|w^j\|_2 \right)^2 + \frac{1}{N} \sum_{i=1}^N \xi_i \\ &\text{s.t.} \ \ \bar{w} \cdot (\bar{\phi}(x_i,y_i) - \bar{\phi}(x_i,y)) \geq 1 - \xi_i, \forall i, y \neq y_i \ . \end{aligned} \tag{2}$$

An equivalent formulation can be derived from the first one through a variational argument (Bach et al., 2004)

$$\begin{aligned}
&\min_{w_j,\alpha_j\geq 0} \frac{\lambda}{2}\left(\sum_{j=1}^{F}\frac{\|w^j\|_2}{\alpha_j}\right)^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i \\
&\text{s.t. } \bar{w}\cdot(\bar{\phi}(x_i,y_i)-\bar{\phi}(x_i,y)) \geq 1-\xi_i,\ \forall i, y\neq y_i \\
&\|\alpha\|_1^2 \leq 1 \ .
\end{aligned} \tag{3}$$

This formulation has been used by Bach et al. (2004) and Sonnenburg et al. (2006), while the formulation proposed by Rakotomamonjy et al. (2008) is slightly different, although it can be proved to be equivalent. The reason to introduce this variational formulation is to use an alternating optimization strategy to efficiently solve the constrained minimization problem. However in the following we will show that it is possible to efficiently minimize directly the formulation in (2), or at least one variation of it.

We first rewrite (2) with group norms. Using the notation defined above, we have

$$\min_{\bar{w}} \ \frac{\lambda}{2}\|\bar{w}\|_{2,1}^2 + \frac{1}{N}\sum_{i=1}^{N}\ell^{MC}\left(\bar{w},x_i,y_i\right), \tag{4}$$

where $\bar{w}=[w_1,w_2,\cdots,w_F]$. The $(2,1)$ group norm is used to induce sparsity in the domain of the kernels. This means that the solution of the optimization problem will select a subset of the F kernels. However, even if sparsity can be desirable for specific applications, it could lead to a decrease in performance. Moreover the problem in (4) is not strongly convex (Kakade et al., 2009), so its optimization algorithm is rather complex and its rate of convergence is usually slow (Bach et al., 2004; Sonnenburg et al., 2006).

We generalize the optimization problem (4), using a generic group norm and a generic convex loss function

$$\min_{\bar{w}} \ \frac{\lambda}{2}\|\bar{w}\|_{2,p}^2 + \frac{1}{N}\sum_{i=1}^{N}\ell\left(\bar{w},x_i,y_i\right), \tag{5}$$

where $1<p\leq 2$. We also define $f(\bar{w}) := \frac{\lambda}{2}\|\bar{w}\|_{2,p}^2 + \frac{1}{N}\sum_{i=1}^{N}\ell\left(\bar{w},x_i,y_i\right)$ and $\bar{w}^*$ equals to the optimal solution of (5), that is $\bar{w}^* = \arg\min_{\bar{w}} f(\bar{w})$. The additional parameter p allow us to decide the level of sparsity of the solution. Moreover this formulation has the advantage of being λ/q-strongly convex (Kakade et al., 2009), where λ is the regularization parameter in (5). Strong convexity is a key property to design fast batch and online algorithms: the more a problem is strongly convex the easier it is to optimize it (Shalev-Shwartz and Singer, 2007; Kakade et al., 2009). Many optimization problems are strongly convex, as the SVM objective function. When p tends to 1, the solution gets close to the sparse solution obtained by solving (2), but the strong convexity vanishes. Setting p equal to 2 corresponds to using the unweighted sum of the kernels. In the following we will show how to take advantage of the strong convexity to design a fast algorithm to solve (5), and how to have a good convergence rate even when the strong convexity is close to zero. Note that this formulation is similar to the one proposed by Kloft et al. (2009). Indeed, as for (2) and (3), using Lemma 26 in Micchelli and Pontil (2005) it is possible to prove that they are equivalent through a variational argument.

We have chosen to weight the regularization term by λ and divide the loss term by N, instead of the more common formulation with only the loss term weighted by a parameter C. This choice simplifies the math of our algorithm. However the two formulations are equivalent when setting $\lambda = \frac{1}{CN}$, hence a big value of C corresponds to a small value of λ.

3. The OBSCURE Algorithm

Our basic optimization tool is the framework developed in Shalev-Shwartz and Singer (2007); Shalev-Shwartz et al. (2007). It is a general framework to design and analyze stochastic sub-gradient descent algorithms for any strongly convex function. At each step the algorithm takes a random sample of the training set and calculates a sub-gradient of the objective function evaluated on the sample. Then it performs a sub-gradient descent step with decreasing learning rate, followed by a projection inside the space where the solution lives. The algorithm Pegasos, based on this framework, is among the state-of-art solvers for linear SVM (Shalev-Shwartz et al., 2007; Shalev-Shwartz and Srebro, 2008).

Given that the $(2, p)$ group norm is strongly convex, we could use this framework to design an efficient MKL algorithm. It would inherit all the properties of Pegasos (Shalev-Shwartz et al., 2007; Shalev-Shwartz and Srebro, 2008). In particular a Pegasos-like algorithm used to minimize (5) would have a convergence rate, and hence a training time, proportional to $\frac{q}{\lambda}$. Although in general this convergence rate can be quite good, it becomes slow when λ is small and/or p is close to 1. Moreover it is common knowledge that in many real-world problems (e.g., visual learning tasks) the best setting for λ is very small, or equivalently C is very big (the order of $10^2 - 10^3$). Notice that this is a general problem. The same problem also exists in the other SVM optimization algorithms such as SMO and similar approaches, as their training time also depends on the value of the parameter C (Hush et al., 2006).

Do et al. (2009) proposed a variation of the Pegasos algorithm called proximal projected sub-gradient descent. This formulation has a better convergence rate for small values of λ, while retaining the fast convergence rate for big values of λ. A drawback is that the algorithm needs to know in advance an upper bound on the norm of the optimal solution. Do et al. (2009) solve this problem with an algorithm that estimates this bound while training, but it gives a speed-up only when the norm of the optimal solution $\bar{w}^*$ is small. This is not the case in most of the MKL problems for categorization tasks.

Our OBSCURE algorithm takes the best of the two solutions. We first extend the framework of Do et al. (2009) to the generic non-Euclidean norms, to use it with the $(2, p)$ group norm. Then we solve the problem of the upper bound of the norm of the optimal solution using a new online algorithm. This is designed to take advantage of the characteristics of the MKL task and to quickly converge to a solution close to the optimal one. Hence OBSCURE is composed of two steps: the first step is a fast online algorithm (Algorithm 1), used to quickly estimate the region of the space where the optimal solution lives. The second step (Algorithm 2) starts from the approximate solution found by the first stage, and exploiting the information on the estimated region, it uses a stochastic proximal projected sub-gradient descent algorithm. We also found that, even if we cannot guarantee this theoretically, empirically in many cases the solution found by the first stage is extremely close to the optimal one. We will show this in the experiments of Section 5.6.

3.1 Efficient Implementation

The training time of OBSCURE is proportional to the number of steps required to converge to the optimal solution, that will be bounded in Theorem 3, multiplied by the complexity of each step. This

Algorithm 1 OBSCURE stage 1 (online)

```
1: Input: $q, \eta$
2: Initialize: $\bar{\theta}_1 = 0, \bar{w}_1 = 0$
3: for $t = 1, 2, \ldots, T$ do
4:     Sample $(x_t, y_t)$ at random
5:     $\bar{z}_t = \partial\ell(\bar{w}_t, x_t, y_t)$
6:     $\bar{\theta}_{t+1} = \bar{\theta}_t - \eta\bar{z}_t$
7:     $w^j_{t+1} = \frac{1}{q}\left(\frac{\|\theta^j_{t+1}\|_2}{\|\bar{\theta}_{t+1}\|_{2,q}}\right)^{q-2}\theta^j_{t+1}, \ \forall j = 1, \cdots, F$
8: end for
9: return $\bar{\theta}_{T+1}, \bar{w}_{T+1}$
10: return $R = \sqrt{\|\bar{w}_{T+1}\|^2_{2,p} + \frac{2}{\lambda N}\sum_{i=1}^N \ell(\bar{w}_{T+1}, x_i, y_i)}$
```

Algorithm 2 OBSCURE stage 2 (stochastic optimization)

```
1: Input: $q, \bar{\theta}_1, \bar{w}_1, R, \lambda$
2: Initialize: $s_0 = 0$
3: for $t = 1, 2, \ldots, T$ do
4:     Sample $(x_t, y_t)$ at random
5:     $\bar{z}_t = \partial\ell(\bar{w}_t, x_t, y_t)$
6:     $d_t = \lambda t + s_{t-1}$
7:     $s_t = s_{t-1} + 0.5\left(\sqrt{d_t^2 + q\frac{(\frac{\lambda}{q}\|\bar{\theta}_t\|_{2,q} + \|\bar{z}_t\|_{2,q})^2}{R^2}} - d_t\right)$
8:     $\eta_t = \frac{q}{\lambda t + s_t}$
9:     $\bar{\theta}_{t+\frac{1}{2}} = (1 - \frac{\lambda\eta_t}{q})\bar{\theta}_t - \eta_t\bar{z}_t$
10:    $\bar{\theta}_{t+1} = \min\left(1, \frac{qR}{\|\bar{\theta}_{t+\frac{1}{2}}\|_{2,q}}\right)\bar{\theta}_{t+\frac{1}{2}}$
11:    $w^j_{t+1} = \frac{1}{q}\left(\frac{\|\theta^j_{t+1}\|_2}{\|\bar{\theta}_{t+1}\|_{2,q}}\right)^{q-2}\theta^j_{t+1}, \ \forall j = 1, \cdots, F$
12: end for
```

in turn is dominated by the calculation of the gradient of the loss (line 5 in Algorithms 1 and 2). This complexity, for example, for the multiclass hinge loss ℓ^{MC} is $O(NFM)$, given that the number of support vectors is proportional to N. Note that this complexity is common to any other similar algorithm, and it can be reduced using methods like kernel caching (Chang and Lin, 2001).

Following (Shalev-Shwartz et al., 2007, Section 4), it is possible to use Mercer kernels without introducing explicitly the dual formulation of the optimization problem. In both algorithms, $\bar{\theta}_{t+1}$ can be written as a weighted linear summation of $\bar{\phi}(x_t, \cdot)$. For example, when using the multiclass loss function ℓ^{MC}, we have that $\bar{\theta}_{t+1} = -\sum_t \eta_t \bar{z}_t = \sum_t \eta_t(\bar{\phi}(x_t, y_t) - \bar{\phi}(x_t, \hat{y}_t))$. Therefore, the algorithm can easily store η_t, y_t, $\hat{y}_t$, and x_t instead of storing $\bar{\theta}_t$. Observing line 7 in Algorithm 1 and line 11 in the Algorithm 2, we have that at each round, w^j_{t+1} is proportional to θ^j_{t+1}, that is $w^j_{t+1} = \alpha^j_t \theta^j_{t+1}$. Hence $\bar{w}_{t+1}$ can also be represented using $\alpha^j_t \eta_t$, y_t, $\hat{y}_t$ and x_t. In prediction the dot product between $\bar{w}_t$ and $\bar{\phi}(x_t, \cdot)$ can be expressed as a sum of terms $\bar{w}^j_t \cdot \phi^j(x_t, \cdot)$, that can be calculated using the definition of the kernel.

Algorithm 3 Proximal projected sub-gradient descent

```
Input: $R, \sigma, w_1 \in S$
Initialize: $s_0 = 0$
for $t = 1, 2, \dots, T$ do
    Receive $g_t$
    $z_t = \partial g_t(w_t)$
    $s_t = s_{t-1} + \frac{\sqrt{(\alpha\sigma t + s_{t-1})^2 + \frac{\alpha L_t}{R^2}} - \alpha\sigma t - s_{t-1}}{2}$
    $\eta_t = \frac{1}{\sigma t + \frac{s_t}{\alpha}}$
    $w_{t+1} = \nabla h^*(\nabla h(w_t) - \eta_t z_t)$
end for
```

Another important speed-up can be obtained considering the nature of the updates of the second stage. If the optimal solution has a loss equal to zero or close to it, when the algorithm is close to convergence most of the updates will consist just of a scaling. Hence it is possible to cumulate the scalings in a variable, to perform the scaling of the coefficients just before an additive update must be performed, and to take it into account for each prediction. Moreover, when using the multiclass loss (1), each update touches only two classes at a time, so to minimize the number of scalings we can keep a vector of scaling coefficients, one for each class, instead of a single number. For more details on the implementation, we refer the reader to the MATLAB implementation of OBSCURE in DOGMA (Orabona, 2009).

4. Analysis

We now show the theorems that give a theoretical guarantee on the convergence rate of OBSCURE to the optimal solution of (5). The following lemma contains useful properties to prove the performance guarantees of Algorithm 1 and 2.

Lemma 1 *Let $B \in \mathbb{R}^+$, define $S = \{\bar{w} : \|\bar{w}\|_{2,p} \le B\}$. Let $h(\bar{w}) : S \to \mathbb{R}$ defined as $\frac{q}{2}\|\bar{w}\|_{2,p}^2$, define also* $\operatorname{Proj}(\bar{w}, B) = \min\left(1, \frac{B}{\|\bar{w}\|_{2,p}}\right)\bar{w}$*, then*

1. $\nabla h(\bar{w}) = q\left[\left(\frac{\|w^j\|_2}{\|\bar{w}\|_{2,p}}\right)^{p-2} w^j\right]_1^F, \ \forall \bar{w} \in S$

2. $\nabla h^*(\bar{\theta}) = \operatorname{Proj}\left(\frac{1}{q}\left[\left(\frac{\|\theta^j\|_2}{\|\bar{\theta}\|_{2,q}}\right)^{q-2} \theta^j\right]_1^F, B\right)$

3. $\|\bar{w}\|_{2,p} = \frac{1}{q}\|\nabla h(\bar{w})\|_{2,q}, \ \forall \bar{w} \in S$

Proof All the proofs of these relations use the equality $1/p + 1/q = 1$. The first one can be obtained differentiating h. The second relation is obtained using Lemma 2 in Shalev-Shwartz and Singer (2007), through lengthy but straightforward calculations. The last one is obtained from the first one. ■

We now introduce Algorithm 3, that forms the basis for Algorithm 2, and a lemma that bounds its performance, that is a generalization of Theorem 1 in Do et al. (2009) to general norms, using the framework in Shalev-Shwartz and Singer (2007). Hence it can be seen as a particular case of the mirror descent algorithm (Beck and Teboulle, 2003), with an adaptive tuning of the learning rate.

Lemma 2 *Let $h(\cdot) = \frac{\alpha}{2}\|\cdot\|^2$ be a 1-strongly convex function w.r.t. a norm $\|\cdot\|$ over S. Assume that for all t, $g_t(\cdot)$ is a σ-strongly convex function w.r.t. $h(\cdot)$, and $\|z_t\|_* \leq L_t$. Let $R \in \mathbb{R}^+$ such that $\|w - w'\| \leq 2R$ for any $w, w' \in S$. Then for any $u \in S$, and for any sequence of non-negative $\xi_1, \ldots, \xi_T$, Algorithm 3 achieves the following bound for all $T \geq 1$,*

$$\sum_{t=1}^{T} (g_t(w_t) - g_t(u)) \leq \sum_{t=1}^{T} \left[4\xi_t R^2 + \frac{L_t^2}{\sigma t + \frac{\sum_{i=1}^{t} \xi_i}{\alpha}} \right] .$$

Proof Define $g'_t(w) = g_t(w) + \frac{s_t}{2}\|w - w_t\|^2$, where $w, w_t \in S$, and the value of s_t will be specified later. Using the assumptions of this Lemma, we have that g'_t is $(\sigma + \frac{s_t}{\alpha})$-strongly convex w.r.t. to h. Moreover we have that $\partial g'_t(w_t) = \partial g_t(w_t)$, because the gradient of the proximal regularization term is zero when evaluated at w_t (Do et al., 2009). Hence we can apply Theorem 1 from Shalev-Shwartz and Singer (2007) to have

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} \left(g_t(u) + \frac{s_t}{2}\|u - w_t\|^2 \right) = \sum_{t=1}^{T} g'_t(w_t) - \sum_{t=1}^{T} g'_t(u) \leq \frac{1}{2} \sum_{t=1}^{T} \frac{L_t^2}{\sigma t + \frac{\sum_{i=1}^{t} s_i}{\alpha}} .$$

Using the hypothesis of this Lemma we obtain

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(u) \leq \frac{1}{2} \sum_{t=1}^{T} \left(s_t \|u - w_t\|^2 + \frac{\alpha L_t^2}{\alpha\sigma t + \sum_{i=1}^{t} s_i} \right) \leq \frac{1}{2} \sum_{t=1}^{T} \left(4 s_t R^2 + \frac{\alpha L_t^2}{\alpha\sigma t + \sum_{i=1}^{t} s_i} \right) .$$

Using the definition of s_t in the algorithm and Lemma 3.1 in Bartlett et al. (2008), we have

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(u) \leq \min_{\xi_1, \ldots, \xi_T \geq 0} \sum_{t=1}^{T} \left(4\xi_t R^2 + \frac{\alpha L_t^2}{\alpha\sigma t + \sum_{i=1}^{t} \xi_i} \right) .$$

Hence these settings of s_t give us a bound that is only 2 times worse than the optimal one. ■

With this Lemma we can now design stochastic sub-gradient algorithms. In particular, setting $\|\cdot\|_{2,p}$ as the norm, $h(\bar{w}) = \frac{q}{2}\|\bar{w}\|_{2,p}^2$, and $g_t(\bar{w}) = \frac{\lambda}{q} h(\bar{w}) + \ell(\bar{w}, x_t, y_t)$, we obtain Algorithm 2 that solves the p-norm MKL problem in (5). The updates are done on the dual variables $\bar{\theta}_t$, in lines 9-10, and transformed into $\bar{w}_t$ in line 11, through a simple scaling. We can now prove the following bound on the convergence rate for Algorithm 2.

Theorem 3 *Suppose that $\|\partial\ell(\bar{w}, x_t, y_t)\|_{2,q} \leq L$ and $\|\bar{w}^*\|_{2,p} \leq R$, where $\bar{w}^*$ is the optimal solution of (5), that is $\bar{w}^* = \underset{\bar{w}}{\operatorname{argmin}} f(\bar{w})$. Let $1 < p \leq 2$, and $c = \lambda R + L$, then in expectation over the choices of the random samples we have that, after T iterations of the 2nd stage of the OBSCURE algorithm, the difference between $f(\bar{w}_T)$ and $f(\bar{w}^*)$, is less than*

$$c\sqrt{q}\sqrt{1 + \log T} \min\left(\frac{c\sqrt{q}\sqrt{1 + \log T}}{\lambda T}, \frac{4R}{\sqrt{T}} \right) .$$

Proof Let $h(\bar{w}) : S \to \mathbb{R}$ defined as $\frac{q}{2}\|\bar{w}\|^2_{2,p}$, where $S = \{\bar{w} : \|\bar{w}\|_{2,p} \leq R\}$. Define also $g_t(\bar{w}) = \frac{\lambda}{2}\|\bar{w}\|^2_{2,p} + \ell(\bar{w}, x_t, y_t) = \frac{\lambda}{q}h(\bar{w}) + \ell(\bar{w}, x_t, y_t)$. Using Lemma 1 in Shalev-Shwartz and Singer (2007), we can see that these two functions satisfy the hypothesis of Lemma 1, with $\alpha = q$, $\sigma = \frac{\lambda}{q}$. It is easy to verify that $\bar{w}_{t+1}$ is equal to $\nabla h^*(\nabla h(\bar{w}_t) - \eta_t z_t)$. In fact, taking into account Properties 1-3 in Lemma 1 with with $B = R$, lines 9-11 in Algorithm 2 are equivalent to

$$\begin{aligned} \bar{w}_{t+1} &= \nabla h^*(\theta_t - \eta_t z_t) \\ \bar{\theta}_{t+1} &= \nabla h(\bar{w}_{t+1}) \ . \end{aligned}$$

We also have that

$$\|\partial g_t(\bar{w})\|_{2,q} \leq \frac{\lambda}{q}\|\nabla h(\bar{w}_t)\|_{2,q} + \|\bar{z}_t\|_{2,q} = \lambda\|\bar{w}_t\|_{2,p} + \|\bar{z}_t\|_{2,q} \leq c,$$

where the equality is due to Property 3 in Lemma 1. So we have

$$\sum_{t=1}^{T}(g_t(\bar{w}_t) - g_t(\bar{w}^*)) \leq \min_{\xi_1,\cdots,\xi_T}\sum_{t=1}^{T}\left[4\xi_t R^2 + \frac{qc^2}{\lambda t + \sum_{i=1}^{t}\xi_i}\right] \ .$$

Reasoning as in Shalev-Shwartz et al. (2007), we divide by T, take the expectation on both sides. So we obtain that

$$\mathbb{E}[f(\bar{w}_T) - f(\bar{w}^*)] \leq \min_{\xi_1,\cdots,\xi_T}\frac{1}{T}\sum_{t=1}^{T}\left[4\xi_t R^2 + \frac{qc^2}{\lambda t + \sum_{i=1}^{t}\xi_i}\right] \ .$$

Setting $\xi_i = \xi$, $i = 1, \ldots, T$, the last term in the last equation can be upper bounded by

$$A_T = \min_{\xi}\left[4\xi R^2 + \frac{1}{T}\sum_{t=1}^{T}\frac{qc^2}{t(\lambda+\xi)}\right] \ .$$

This term is smaller than any specific setting of ξ, in particular if we set ξ to 0, we have that $A_T \leq \frac{qc^2(1+\log T)}{\lambda T}$. On the other hand setting optimally the expression over ξ and over-approximating we have that $A_T \leq \frac{4cR\sqrt{q}\sqrt{1+\log T}}{\sqrt{T}}$. Taking the minimum of these two quantities we obtain the stated bound. ■

The most important thing to note is that the converge rate is independent from the number of samples, as in Pegasos (Shalev-Shwartz et al., 2007), and the relevant quantities on which it depends are λ and q. Given that for most of the losses, each iteration has a linear complexity in the number of samples, as stated in Section 3.1, the training time will be linearly proportional to the number of samples.

The parameter R is basically an upper bound on the norm of the optimal solution. In the next Section we show how to have a good estimate of R in an efficient way. The theorem first shows that a good estimate of R can speed-up the convergence of the algorithm. In particular if the first term is dominant, the convergence rate is $O(\frac{q\log T}{\lambda T})$. If the second term is predominant, the convergence rate is $O(\frac{R\sqrt{q\log T}}{\sqrt{T}})$, so it becomes independent from λ. The algorithm will always optimally interpolate between these two different rates of convergence. Note that R can also be set to the trivial upper bound of ∞. This would result in a standard Pegasos-like optimization. In fact, s_t would be equal to 0 in Algorithm 2, so the learning rate would be $\frac{1}{\lambda t}$ and the convergence rate would be $O(\frac{q \log T}{\lambda T})$. We will see in Section 5.3 that a tight estimate of R can improve dramatically the convergence rate. Another important point is that Algorithm 2 can start from any vector, while this is not possible in the Pegasos algorithm, where at the very first iteration the starting vector is multiplied by 0 (Shalev-Shwartz et al., 2007).

As said before, the rate of convergence depends on p, through q. A p close to 1 will result in a sparse solution, with a rate of at most $O(\frac{R\sqrt{q \log T}}{\sqrt{T}})$. However in the experiment section we show that the best performance is not always given by the most sparse solution.

This theorem and the pseudocode in Algorithm 2 allows us to design fast and efficient MKL algorithms for a wide range of convex losses. If we consider the multiclass loss ℓ^{MC} with normalized kernels, that is, $\|\phi^j(x_t,y_t)\|_2 \leq 1, \forall j = 1, \cdots, F,\ t = 1, \cdots, N$, we have that $L \leq \sqrt{2} F^{\frac{1}{q}}$. Instead, if we use the hinge loss ℓ^{HL} for binary classification, we have that $L \leq F^{\frac{1}{q}}$. Hence, in both cases, if $p < 2$, the convergence rate has a sublinear dependency on the number of kernels, F, and if the problem is linearly separable it can have a faster convergence rate using more kernels. We will explain this formally in the next section.

4.1 Initialization Through an Online Algorithm

In Theorem 3 we saw that if we have a good estimate of R, the convergence rate of the algorithm can be much faster. Moreover starting from a *good* solution could speed-up the algorithm even more.

We propose to initialize Algorithm 2 with Algorithm 1. It is the online version of problem (5) and it is derived using a recent result in Orabona and Crammer (2010). It is similar to the $2p$-norm matrix Perceptron of Cavallanti et al. (2008), but it overcomes the disadvantage of being used with the same kernel on each feature.

We can run it just for few iterations and then evaluate its norm and its loss. In Algorithm 1 R is then defined as

$$R := \sqrt{\|\bar{w}_{T+1}\|_{2,p}^2 + \frac{2}{\lambda N} \sum_{i=1}^{N} \ell(\bar{w}_{T+1}, x_i, y_i)} \geq \sqrt{\|\bar{w}^*\|_{2,p}^2 + \frac{2}{\lambda N} \sum_{i=1}^{N} \ell(\bar{w}^*, x_i, y_i)} \geq \|\bar{w}^*\|_{2,p},$$

where we remind that $\bar{w}^*$ is solution that minimizes (5), as defined in Section 2.2. So at any moment we can stop the algorithm and obtain an upper bound on $\|\bar{w}^*\|_{2,p}$. However if the problem is linearly separable we can prove that Algorithm 1 will converge in a finite number of updates. In fact, as in Cavallanti et al. (2008), for Algorithm 1 it is possible to prove a relative mistake bound. See also Jie et al. (2010b) and Jin et al. (2010) for similar algorithms for online MKL, with a different update rules and different mistake bounds.

Theorem 4 *Let $(x_1, y_1), \ldots, (x_T, y_T)$ be a sequence of examples where $x_t \in \mathbb{X}$, $y \in \mathbb{Y}$. Suppose that the loss function ℓ has the following properties*

- $\|\partial \ell(\bar{w}, x, y)\|_{2,q} \leq L,\ \forall\ \bar{w} \in \mathbb{X},\ x_t \in \mathbb{X},\ y_t \in \mathbb{Y}$*;*
- $\ell(\bar{u}, x, y) \geq 1 + \bar{u} \cdot \partial \ell(\bar{w}, x, y),\ \forall\ \bar{u} \in \mathbb{X},\ \bar{w} \in \mathbb{X} : \ell(\bar{w}, x, y) > 0,\ x \in \mathbb{X},\ y \in \mathbb{Y}$*;*
- $\bar{w} \cdot \partial \ell(\bar{w}, x, y) \geq -1,\ \forall\ \bar{w} \in \mathbb{X},\ x \in \mathbb{X},\ y \in \mathbb{Y}$.

Denote by $\mathcal{U}$ the set of rounds in which there is an update, and by U its cardinality. Then, for any $\bar{u}$, the number of updates U of Algorithm 1 satisfies

$$U \leq q(2/\eta + L^2)\|\bar{u}\|_{2,p}^2 + \sum_{t\in\mathcal{U}} \ell(\bar{u},x_t,y_t) + \|\bar{u}\|_{2,p}\sqrt{q(2/\eta+L^2)}\sqrt{\sum_{t\in\mathcal{U}} \ell(\bar{u},x_t,y_t)}\ .$$

In particular, if the problem (5) is linearly separable by a hyperplane $\bar{v}$, then the Algorithm 1 will converge to a solution in a finite number of steps less than $q(2/\eta + L^2)\|\bar{v}\|_{2,p}^2$. In this case the returned value of R will be less than $(2+\eta L^2)\|\bar{v}\|_{2,p}$.

Proof The bound on the number of updates can be easily derived using a recent result in Orabona and Crammer (2010), that we report in Appendix for completeness. Let $h(\bar{w}) : \mathbb{X} \to \mathbb{R}$ defined as $\frac{q}{2}\|\bar{w}\|_{2,p}^2$. Notice that, differently from the proof of Theorem 3, here the domain of the function h is the entire Euclidean space $\mathbb{X}$. Using Property 2 in Lemma 1 with $B = \infty$, we have that line 7 in the algorithm's pseudo-code implies that $\bar{w}_t = \nabla h^*(\bar{\theta}_t)$. Using Lemma 5 in the Appendix, we have that

$$\begin{aligned} U &\leq \sum_{t\in\mathcal{U}} \ell(\bar{u},x_t,y_t) + \sqrt{q}\|\bar{u}\|_{2,p}\sqrt{\sum_{t\in\mathcal{U}}\left(\|\bar{z}_t\|_{2,q}^2 - \frac{2\bar{w}_t\cdot\bar{z}_t}{\eta}\right)} \\ &\leq \sum_{t\in\mathcal{U}} \ell(\bar{u},x_t,y_t) + \sqrt{q}\|\bar{u}\|_{2,p}\sqrt{U\left(L^2+\frac{2}{\eta}\right)}\ . \end{aligned}$$

Solving for U and overapproximating we obtain the stated bound.

For the second part of the Theorem, using (Kakade et al., 2009, Corollary 19), we know that $h^*(\bar{w})$ is 1-smooth w.r.t. $\|\cdot\|_{2,q}$. Hence, we obtain

$$\begin{aligned} \|\bar{\theta}_{T+1}\|_{2,q}^2 &\leq \|\bar{\theta}_T\|_{2,q}^2 - 2q\eta\bar{w}_T\cdot\bar{z}_T + q\eta^2\|\bar{z}_T\|_{2,q}^2 \leq \eta^2 q\sum_{t\in\mathcal{U}}\left(\|\bar{z}_t\|_{2,q}^2 - \frac{2\bar{w}_t\cdot\bar{z}_t}{\eta}\right) \\ &\leq \eta^2 qU\left(\frac{2}{\eta}+L^2\right)\ . \end{aligned}$$

So we can write

$$\|\bar{\theta}_{T+1}\|_{2,q} \leq \eta\sqrt{qU(2/\eta+L^2)},$$

and using the bound of U and the hypothesis of linear separability, we have

$$\|\bar{\theta}_{T+1}\|_{2,q} \leq \eta\sqrt{q^2\|\bar{u}\|_{2,p}^2(2/\eta+L^2)^2} = q\|\bar{u}\|_{2,p}\left(2+\eta L^2\right)\ .$$

Using the relation $\|\bar{w}_t\|_{2,p} = \frac{1}{q}\|\bar{\theta}_t\|_{2,q}$, that holds for Property 2 in Lemma 1 with $B = \infty$, we have the stated bound on R. ■

From the theorem it is clear the role of η: a bigger value will speed up the convergence, but it will decrease the quality of the estimate of R. So η governs the trade-off between speed and precision of the first stage.

The multiclass loss ℓ^{MC} and the hinge loss ℓ^{HL} satisfy the conditions of the Theorem, and, as noted for Theorem 1 when p is close to 1, the dependency on the number of kernels in this theorem is strongly sublinear.

Note also that the separability assumption is far from being unrealistic in our setting. In fact the opposite is true: in the greater majority of the cases the problem will be linearly separable. This is due to the fact that in MKL to have separability it is sufficient that only one of the kernel induces a feature space where the problem is separable. So, for example, it is enough to have no repeated samples with different labels and at least one kernel that always produces kernel matrices with full rank, for example, the Gaussian kernel.

Moreover, under the separability assumption, if we increase the number of kernels, we have that $\|\bar{u}\|_{2,p}^2$ cannot increase, and in most of the cases it will decrease. In this case we expect Algorithm 1 to converge to a solution which has null loss on each training sample, in a finite number of steps that is almost independent on F and in some cases even *decreasing* while increasing F. The same consideration holds for the value of R returned by the algorithm, that can decrease when we increase the number of kernels. A smaller value of R will mean a faster convergence of the second stage. We will confirm this statement experimentally in Section 5.

5. Experiments

In this section, we study the behavior of OBSCURE in terms of classification accuracy, computational efficiency and scalability. We implemented our algorithm in MATLAB, in the DOGMA library (Orabona, 2009). We focus on the multiclass loss ℓ^{MC}, being it much harder to be optimized than the binary hinge loss ℓ^{HL}, especially in the MKL setting. Although our MATLAB implementation is not optimized for speed, it is already possible to observe the advantage of the low runtime complexity. This is particularly evident when training on data sets containing large numbers of categories and lots of training samples. Except in the synthetic experiment where we set $p = 1.0001$, in all the other experiments the parameter p is chosen from the set $\{1.01, 1.05, 1.10, 1.25, 1.50, 1.75, 2\}$. The regularization parameter λ is set through CV, as $\frac{1}{CN}$, where $C \in \{0.1, 1, 10, 100, 1000\}$.

We compare our algorithm with the binary SILP algorithm (Sonnenburg et al., 2006), the multi class MKL (MC-MKL) algorithm (Zien and Ong, 2007) and the p-norm MKL algorithm (Kloft et al., 2009), all of them implemented in the SHOGUN-0.9.2 toolbox.[3] For p-norm MKL, it is possible to convert from our p setting to the equivalent setting in Kloft et al. (2009) using $p_{\text{p-norm}} = p_{\text{OBSCURE}}/(2 - p_{\text{OBSCURE}})$. In our experiments, we will compare between OBSCURE and p-norm MKL using the equivalent p parameter. We also compare with the SimpleMKL algorithm[4] (Rakotomamonjy et al., 2008). To train with the unweighted sum of the kernels with an SVM, we use LIBSVM (Chang and Lin, 2001). The cost parameter is selected from the range $C \in \{0.1, 1, 10, 100, 1000\}$ for all the baseline methods. For all the binary classification algorithms, we use the 1-vs-All strategy for their multiple class extensions.

In the following we start by briefly introducing the data sets used in our experiments. Then we present a toy problem on a synthetic data which shows that it is more appropriate to use a multiclass loss instead of dividing the multiclass classification problem into several binary subproblems. We

[3] Available at `http://www.shogun-toolbox.org`, implemented in C++.

[4] Available at `http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html`, implemented in MATLAB. SimpleMKL is more efficient than SILP when uses the same SVM solver (Rakotomamonjy et al., 2008). However, in practice, no efficient SimpleMKL implementation is available. SILP runs much faster compared to SimpleMKL, especially when the size of the problem grows. Moreover, the performance of SimpleMKL and SILP are the same because both algorithm solve an equivalent optimization problem (Rakotomamonjy et al., 2008). Therefore, we only use SILP algorithm as our l_1-norm MKL baseline in experiments whose size of training samples are large than 1000.

then study the convergence rate of OBSCURE and compare it with the original Pegasos algorithm (Shalev-Shwartz et al., 2007; Shalev-Shwartz and Srebro, 2008) as well as the p-norm MKL (Kloft et al., 2009) (Section 5.3). Following that, we study the behaviors of OBSCURE w.r.t different value of p and different number of input kernels (Sections 5.5 and 5.6). Finally we show that OBSCURE achieves state-of-art performance on a challenging image classification task with 102 different classes, and we show its scalability.

5.1 Data Sets

We first briefly introduce the data sets used in this section, and we describe how their kernel matrices are generated.

5.1.1 THE OXFORD FLOWER DATA SET (NILSBACK AND ZISSERMAN, 2006)

contains 17 different categories of flowers. Each class has 80 images with three predefined splits (train, validation and test). The authors also provide 7 precomputed distance matrices.[5] These distance matrices are transformed into kernel using $\exp(-\gamma^{-1}d)$, where γ is the average pairwise distance and d is the distance between two examples. It results in 7 different kernels.

5.1.2 THE PENDIGITS DATA SET (GÖNEN AND ALPAYDIN, 2010)

is on pen-based digit recognition (multiclass classification with 10 classes) and contains four different feature representations.[6] The data set is split into independent training and test sets with 7494 samples for training and 3498 samples for testing. We have generated 4 kernel matrices, one matrix for each feature, using an RBF kernel, $\exp(-\gamma^{-1}\|x_i - x_j\|^2)$. For each feature, γ is equal to the average of the squared pairwise distances between the examples.

5.1.3 THE KTH-IDOL2 DATA SET (PRONOBIS ET AL., 2010)

contains 24 image sequences acquired using a perspective camera mounted on two mobile robot platforms. These sequences were captured with the two robots moving in an indoor laboratory environment consisting of five different rooms under various weather and illumination conditions (sunny, cloudy, and night) and across a time span of six months. For experiments, we used the same setup described in Pronobis et al. (2010); Jie et al. (2010a). We considered the 12 sequences acquired by robot Dumbo, and divided them into training and test sets, where each training sequence has a corresponding one in the test sets captured under roughly similar conditions. In total, we considered twelve different permutations of training and test sets. The images were described using three visual descriptors and a geometric feature from the Laser Scan sensor, as in Jie et al. (2010a), which forms 4 kernels in total.

5.1.4 THE CALTECH-101 DATA SET (FEI-FEI ET AL., 2004)

is a standard benchmark data set for object categorization. In our experiments, we used the pre-computed features and kernels of Gehler and Nowozin (2009b) which the authors made available on their website,[7] with the same training and test split. This allows us to compare against them

[5] Available at `www.robots.ox.ac.uk/~vgg/research/flowers/`.

[6] Available at `http://mkl.ucsd.edu/dataset/pendigits`.

[7] Available at `www.vision.ee.ethz.ch/~pgehler/projects/iccv09/`.

directly. Following that, we report results using all 102 classes of the Caltech-101 data set using five splits. There are five different image descriptors, namely, PHOG Shape Descriptors (PHOG) (Bosch et al., 2007), Appearance Descriptors (App) (Lowe, 2004), Region Covariance (RECOV) (Tuzel et al., 2007), Local Binary Patterns (LBP) (Ojala et al., 2002) and V1S+ (Pinto et al., 2008). All of them but the V1S+ features were computed in a spatial pyramid as proposed by Lazebnik et al. (2006), using several different setup of parameters. This generates several kernels (PHOG, 8 kernels; App, 16 kernels; RECOV, 3 kernels; LBP 3 kernels; V1S+, 1 kernels). We also compute a subwindow kernel, as proposed by Gehler and Nowozin (2009a). In addition to the 32 kernels, the products of the pyramid levels for each feature results in other 7 kernels, for a total of 39 different kernels For brevity, we omit the details of the features and kernels and refer to Gehler and Nowozin (2009a,b).

5.1.5 THE MNIST DATA SET (LECUN ET AL., 1998)

is a handwritten digits data set. It has a training set of 60,000 gray-scale 28x28 pixel digit images for training and 10,000 images for testing. We cut the original digit image into four square blocks (14×14) and obtained an input vector from each block. We used three kernels on each block: a linear kernel, a polynomial kernel and a RBF kernel, resulting in 12 kernels.

5.2 Multiclass Synthetic Data

Multiclass problems are often decomposed into several binary sub-problems using methods like 1-vs-All, however solving the multiclass learning problem jointly using a multiclass loss can yield much sparser solutions. Intuitively, when a l_1-norm is used to impose sparsity in the domain of kernels, different subsets of kernels can be selected for the different binary classification sub-problems. Therefore, the combined multiclass classifier might not obtain the desired properties of sparsity. Moreover, the confidence outputs of the binary classifiers may not lie in the same range, so it is not clear if the winner-takes-all hypothesis is the correct approach for combing them.

To prove our points, we have generated a 3-classes classification problem consisting of 300 samples, with 100 samples for each class. There are in total 4 different features, the kernel matrices corresponding to them are shown in Figure 1 (top). These features are generated in a way that Kernels 1–3 are useful only for distinguishing one class (class 3, class 1 and class 2, respectively) from the other two, while Kernel 4 can separate all the 3 classes. The corresponding kernel combination weights obtained by the SILP algorithm using the 1-vs-All extension and our multiclass OBSCURE are shown in Figure 1 (bottom). It can be observed that each of the binary SILP classifiers pick two kernels. OBSCURE selects only the 4th kernel, achieving a much sparser solution.

5.3 Comparison with $\frac{1}{t}$ Learning Rate

We have compared OBSCURE with a simple one-stage version that uses a $\frac{1}{t}$ learning rate. This can be obtained setting $s_t = 0$, $\forall t$, in Algorithm 2. It can be considered as a straightforward extension of the original Pegasos algorithm (Shalev-Shwartz et al., 2007; Shalev-Shwartz and Srebro, 2008) to the MKL problem of (5), so we denote it Pegasos-MKL.

We first compare the running time performance between OBSCURE and Pegasos-MKL on the Oxford flowers data set. Their generalization performance on the testing data (Figure 2(Top, left)) as well as the value of the objective function (Figure 2(Top, right)) are shown in Figure 2. In the same Figure, we also present the results obtained using SILP, SimpleMKL, p-norm MKL and MC-MKL.

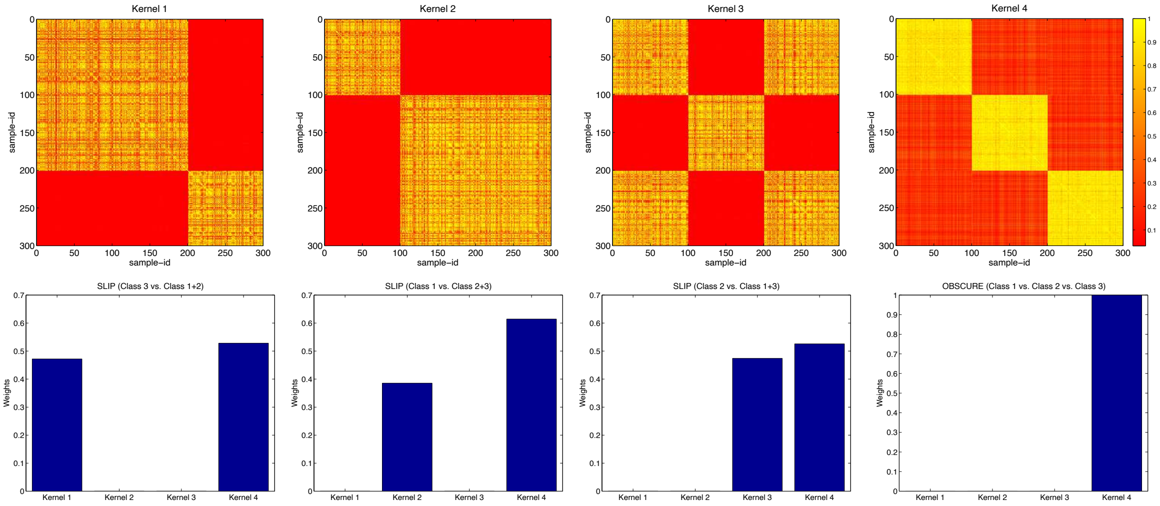


Figure 1: (top) Kernel matrices of the 3-classes synthetic experiments correspond to 4 different features. Sample 1–100, 101–200 and 201–300 are from class 1, 2 and 3 respectively. (bottom) Corresponding kernel combination weights, normalized to have sum equal to 1, obtained by SILP (binary) and by OBSCURE (last figure).

We see that OBSCURE converges much faster compared to Pegasos-MKL. This proves that, as stated in Theorem 3, OBSCURE has a better convergence rate than Pegasos-MKL, as well as faster running time than SILP and SimpleMKL. Note that all the feature combination methods achieve similar results on this data set.

Similar results are shown in Figure 2(Bottom, left) and (Bottom, right) on the Pendigits data sets.

5.4 Comparison with p-norm MKL and Other Baselines

We compare OBSCURE with p-norm MKL (Kloft et al., 2009). Figure 3 reports the results obtained by both algorithms for varying values of p on the Pendigits data set. We can see that all the algorithms (OBSCURE, SILP and p-norm MKL) are order of magnitudes faster than MC-MKL. OBSCURE and p-norm MKL achieve similar performance, but OBSCURE achieve optimal performance in a training time much faster (10^1 and 10^2). The performance of SILP and p-norm MKL are quite close, and their classification rate seems to be more stable on this data set. The difference between OBSCURE and p-norm MKL may be due to the different types of multi class extension they use.

5.5 Experiments with Different Values of p

This experiment aims at showing the behavior of OBSCURE for varying value of p. We consider $p \in (1,2]$, and train OBSCURE on the KTH-IDOL2 and Caltech-101. The results for the two data sets are shown in Figure 4 (top).

For the IDOL2 data set (Figure 4 (top, left)), the best performance is achieved when p is large, which corresponds to give all the kernels similar weights in the decision. On the contrary, a sparse

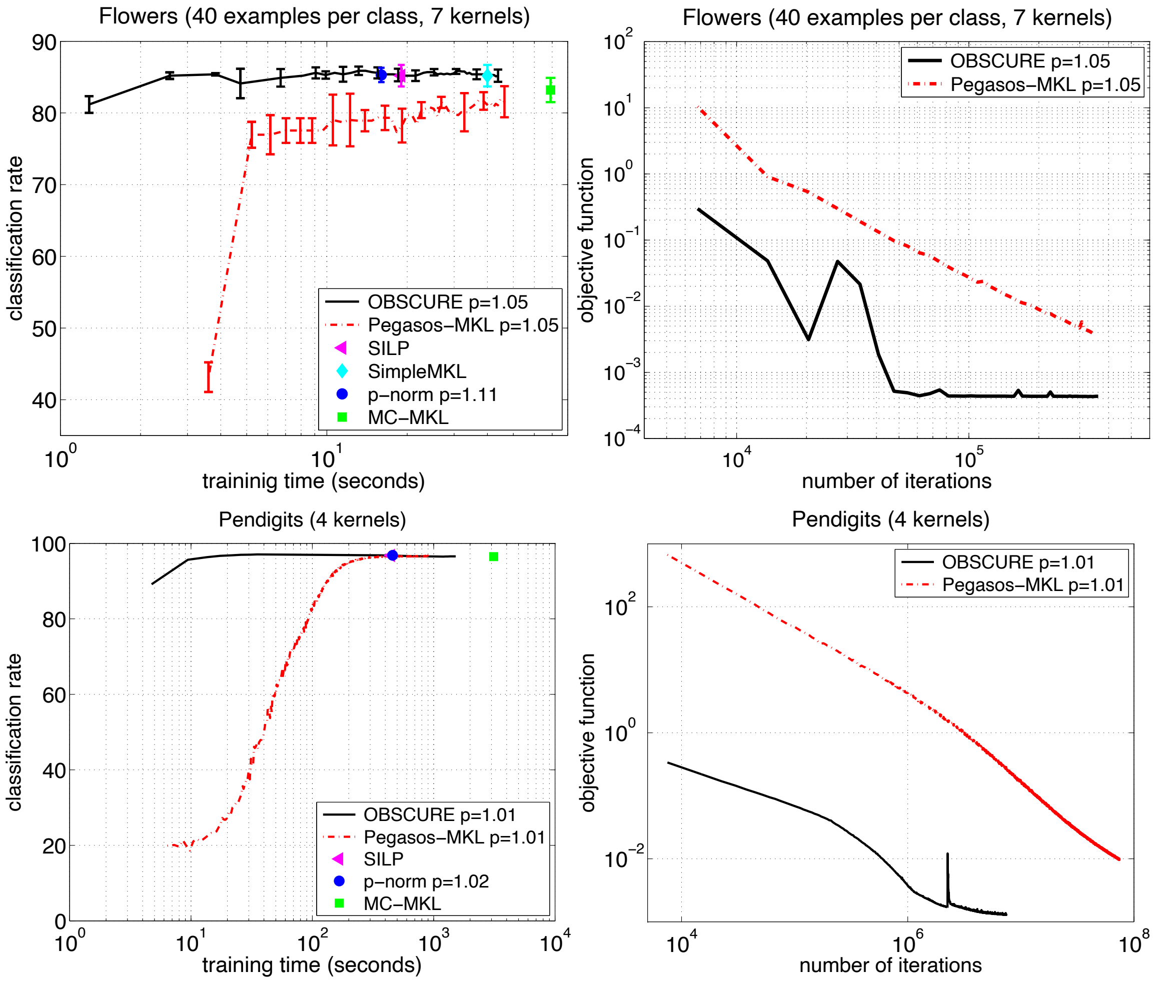


Figure 2: Comparison of running time performance (Left) and objective function value (Right) on the Oxford flowers data set (Top) and Pendigits data set (Bottom).

solution achieves lower accuracy. It indicates that all the kernels carry discriminative information, and excluding some of them can decrease the performance.

For the Caltech-101 data set (Figure 4 (top, right)), following Gehler and Nowozin (2009b), we consider four PHOG (Bosch et al., 2007) kernels computed at different spatial pyramid level. It can be observed that by adjusting p it is possible to improve the performance—sparser solutions (i.e., when p tends to 1) achieve higher accuracy compared to non-sparse solutions (when p tends to 2). However, the optimal p here is 1.10. In other words the optimal performance is achieved for a setting of p different from 1 or 2, fully justifying the presence of this parameter.

Furthermore, Figure 4 (bottom) shows the running time of OBSCURE using the same four kernels, with varying values of p. The dashed lines in the figure correspond to the results obtained by the first online stage of the OBSCURE algorithm. It can be observed that the online stage of OBSCURE converges faster when p is large, and this is consistent with Theorem 4. The online step

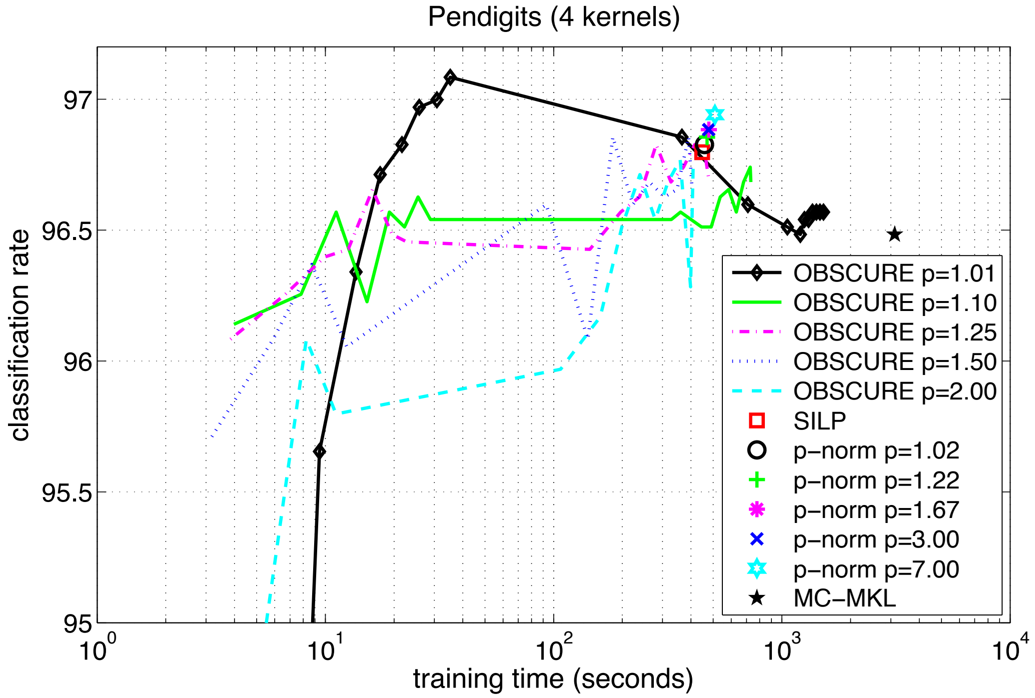


Figure 3: (Best viewed in color.) Comparison of OBSCURE and p-norm MKL with varying value of *p* on Pendigits.

of OBSCURE converges in a training time orders of magnitudes faster (10^1 to 10^3) compared to the full training stage, and in certain cases ($p \leq 1.10$) it can also achieve a performance close to the optimal solution.

5.6 Experiments on Different Number of Kernels

Figure 5 (left) reports the behavior of OBSCURE for different numbers of input kernels. It shows that the algorithm achieves a given accuracy in less iterations when more kernels are given. The dashed line in the figure again corresponds to the results obtained by the first online stage of the OBSCURE algorithm. Figure 5 (right) shows the number of iterations to converge of the online step, proving that the convergence rate improves when there are more kernels, as stated in Theorem 4.

5.7 Multiclass Image Classification

In this experiment, we use the Caltech-101 data set with all the 39 kernels, and the results are shown in Figure 6. The best results for OBSCURE were obtained when *p* is at the smallest value (1.01). This is probably because among these 39 kernels many were redundant or not discriminative enough. For example, the worst single kernel achieves only an accuracy of 13.5% ± 0.6 when trained using 30 images per category, while the best single kernel achieves 69.4% ± 0.4. Thus, sparser solutions are to be favored. The results support our claim in Section 5.2 that multiclass loss function is more suitable for this type of problem, as all the methods that use the multiclass loss

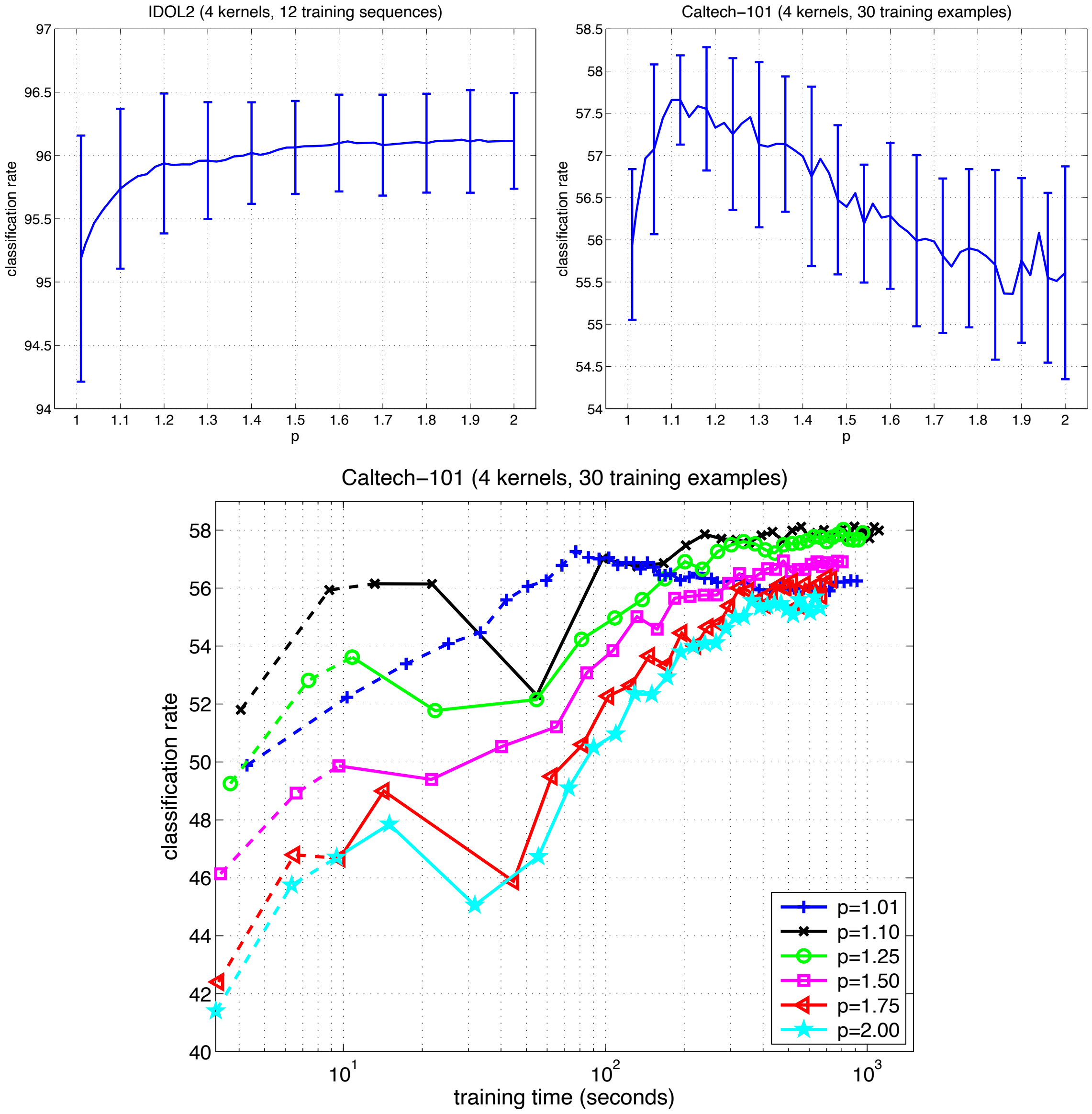


Figure 4: Behaviors of the OBSCURE algorithm w.r.t. p: (top, left) the effect of different values of p on the IDOL2 data set and (top, right) on the Caltech-101 data set using 4 PHOG (Bosch et al., 2007) kernels; (bottom) running time for different values of p on Caltech-101 data set.

outperform SILP and p-norm MKL (p=1.02) using 1-vs-All strategy. MC-MKL is computationally infeasible for 30 sample per category. Its significant gap from OBSCURE seems to indicate that it stops before converging to the optimal solution. Figure 6 (left) reports the training time for different algorithms. Again, OBSCURE reaches optimal solution much faster than the other three baseline algorithms which are implemented in C++. Figure 6 (right) reports the results obtained

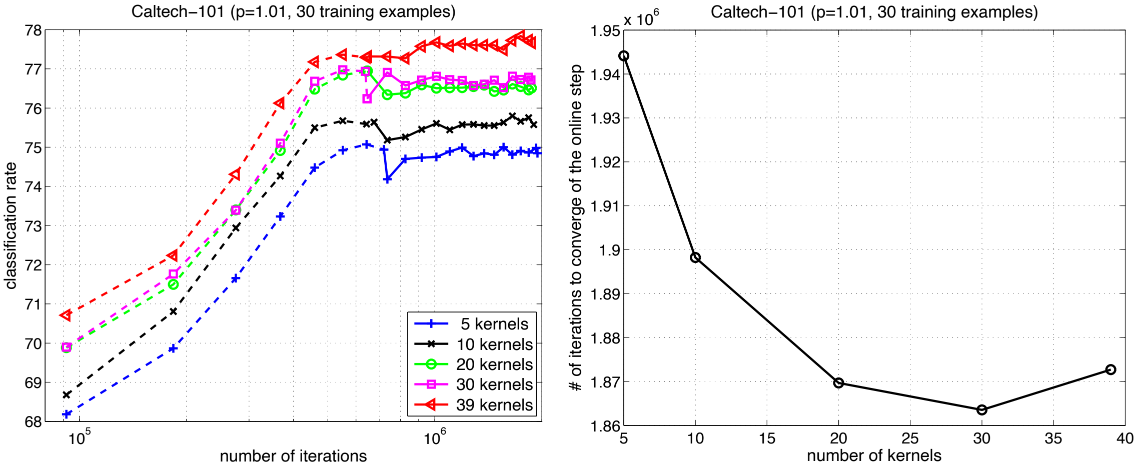


Figure 5: Behaviors of the OBSCURE algorithm w.r.t. the number of kernels: (left) the effect of different number of kernels randomly sampled from the 39 kernels; (right) number of iterations to converge of the online stage.

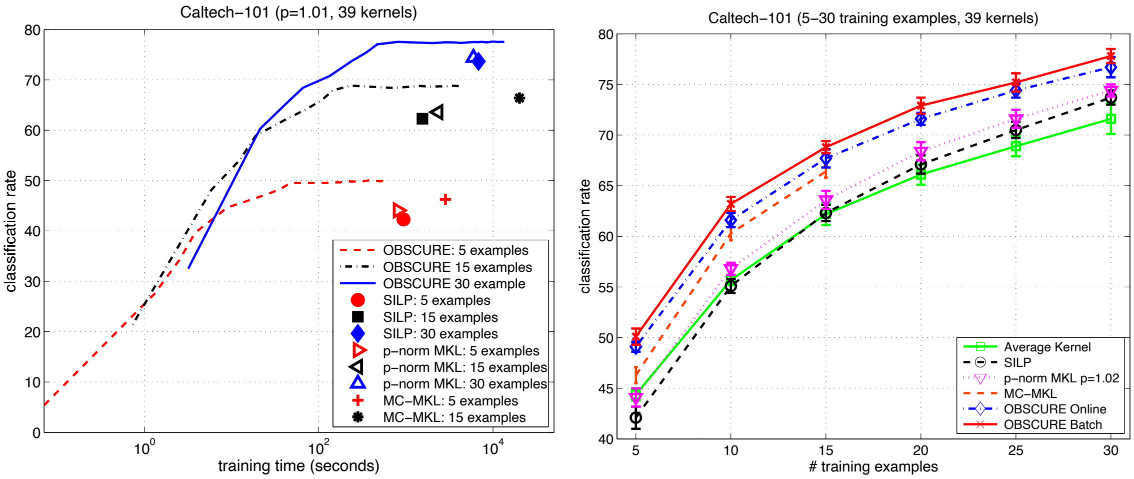


Figure 6: Performance comparison on Caltech-101 using different MKL methods.

using different combination methods for varying size of training samples. It is also interesting to note the performance of the solution generated by the online step of OBSCURE, denoted by "OBSCURE Online", that is very close to the performance of the full training stage, as already noted above.

5.8 Scalability

In this section, we report the experiments on the MNIST data set using varying sizes of training samples. Figure 7 shows the generalization performance on the test set achieved by OBSCURE over time, for various training size. We see that OBSCURE quickly produces solutions with good

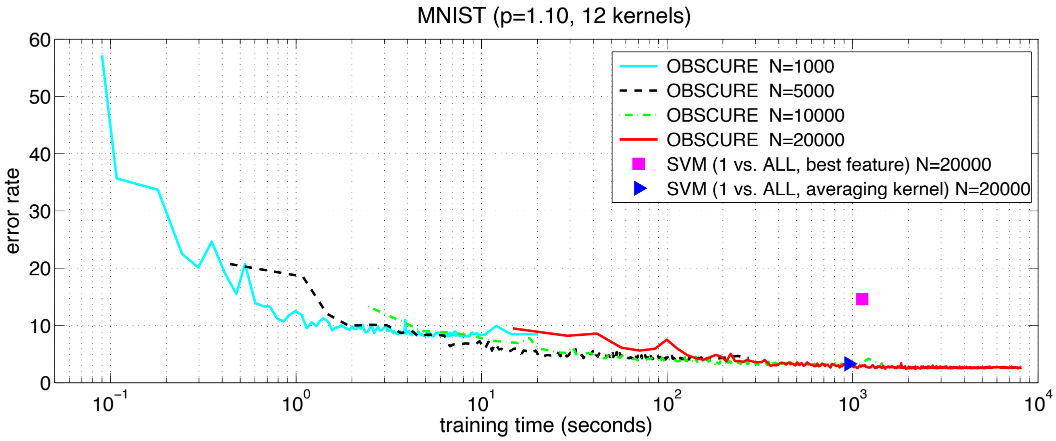


Figure 7: The generalization performance of MNIST data set over different size of training samples.

performance. The performance of the SVM trained using the unweighted sum of the kernels and the best kernel are also plotted. Notice that in the figure we only show the results of up to 20,000 training samples for the sake of comparison, otherwise we could not cache all the 12 kernels in memory. However, by computing the kernel "*on the fly*" we are able to solve the MKL problem using the full 60,000 examples: OBSCURE obtains 1.95% error rate after 10 epochs, which is 0.45% lower compared to the results obtained by OBSCURE with 20,000 training samples after 500 epochs.

6. Conclusions and Discussion

This paper presents OBSCURE, a novel and efficient algorithm for solving p-norm MKL. It uses a hybrid two-stages online-batch approach, optimizing the objective function directly in the primal with a stochastic sub-gradient descent method. Our minimization method allows us to prove convergence rate bounds, proving that the number of iterations required to converge is independent of the number of training samples, and, when a sparse solution is induced, is sub-linear in the number of kernels. Moreover we show that OBSCURE has a faster convergence rate as the number of kernels grows.

Our approach is general, so it can be used with any kind of convex losses, from binary losses to structure output prediction (Tsochantaridis et al., 2004), and even to regression losses.

Experiments show that OBSCURE achieves state-of-art performance on the hard problem of multiclass MKL, with smaller running times than other MKL algorithms. Furthermore, the solution found by the online stage is often very close to the optimal one, while being computed several orders of magnitude faster.

Acknowledgments

The kernel matrixes of Caltech-101 were kindly provided by Peter Gehler, who we also thank for his useful comments. This work was sponsored by the EU project DIRAC IST-027787. FO also acknowledges partial support by the PASCAL2 NoE FP7-216886.

Appendix A.

The following algorithm and Lemma can in found in Orabona and Crammer (2010), stated for the binary case. Here we state them for generic convex losses and report them here for completeness.

Algorithm 4 Prediction algorithm

```
Input: A series of strongly convex functions $h_1, \dots, h_T$.
Initialize: $\theta_1 = 0$
for $t = 1, 2, \dots, T$ do
    Receive $x_t$
    Set $w_t = \nabla h_t^*(\theta_t)$
    $z_t = \partial \ell_t(w_t)$
    $\theta_{t+1} = \theta_t - \eta_t z_t$
end for
```

Lemma 5 *Let $h_t, t = 1, \dots, T$ be β_t-strongly convex functions with respect to the norms $\|\cdot\|_{h_1}, \dots, \|\cdot\|_{h_T}$ over a set S and let $\|\cdot\|_{h_i^*}$ be the respective dual norms. Let $h_0(0) = 0$, and $x_1, \dots, x_T$ be an arbitrary sequence of vectors in $\mathbb{R}^d$. Assume that algorithm in Algorithm 4 is run on this sequence with the functions h_i. If $h_T(\lambda u) \le \lambda^2 h_T(u)$, and ℓ satisfies*

$$\ell(u, x_t, y_t) \ge 1 + u^\top \partial \ell_t(w_t), \ \forall u \in S, w_t : \ell_t(w_t) > 0,$$

then for any $u \in S$, and any $\lambda > 0$ we have

$$\sum_{t=1}^{T} \eta_t \le L + \lambda h_T(u) + \frac{1}{\lambda}\left(D + \sum_{t=1}^{T}\left(\frac{\eta_t^2}{2\beta_t}\|z_t\|_{h_t^*}^2 - \eta_t w_t^\top z_t\right)\right),$$

where $L = \sum_{t \in \mathcal{M} \cup \mathcal{U}} \eta_t \ell_t(u)$, and $D = \sum_{t=1}^{T}(h_t^*(\theta_t) - h_{t-1}^*(\theta_t))$. In particular, choosing the optimal λ, we obtain

$$\sum_{t=1}^{T} \eta_t \le L + \sqrt{2h_T(u)}\sqrt{2D + \sum_{t=1}^{T}\left(\frac{\eta_t^2}{\beta_t}\|z_t\|_{h_t^*}^2 - 2\eta_t w_t^\top z_t\right)}\,.$$

References

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO, algorithm. In *Proc. of the International Conference on Machine Learning*, 2004.

P. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20*, pages 65–72. MIT Press, Cambridge, MA, 2008.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proc. of the 6th ACM International Conference on Image and Video Retrieval*, pages 401–408. ACM, July 2007.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. In *Proc. of the 21st Conference on Learning Theory*, 2008.

C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `www.csie.ntu.edu.tw/~cjlin/libsvm`.

C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *Proc. of the 27th International Conference on Machine Learning*, 2010.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14*, volume 14, pages 367–373, 2002.

C. B. Do, Q. V. Le, and Chuan-Sheng Foo. Proximal regularization for online and batch learning. In *Proc. of the International Conference on Machine Learning*, 2009.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2004.

P. Gehler and S. Nowozin. Let the kernel figure it out: Principled learning of pre-processing for kernel classifiers. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009a.

P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. of the International Conference on Computer Vision*, 2009b.

M. Gönen and E. Alpaydin. Cost-conscious multiple kernel learning. *Pattern Recognition Letters*, 31:959–965, July 2010.

D. Hush, P. Kelly, C. Scovel, and I. Steinwart. QP algorithms with guaranteed accuracy and run time for support vector machines. *Journal of Machine Learning Research*, 7:733–769, 2006.

L. Jie, F. Orabona, and B. Caputo. An online framework for learning novel concepts over multiple cues. In H. Zha, R. Taniguchi, and S. J. Maybank, editors, *Computer Vision - ACCV 2009, 9th Asian Conference on Computer Vision, Xi'an, China, September 23-27, 2009, Revised Selected Papers, Part I*, volume 5994 of *Lecture Notes in Computer Science*, Berlin / Heidelberg, 2010a. Springer.

L. Jie, F. Orabona, M. Fornoni, B. Caputo, and N. Cesa-Bianchi. OM-2: An online multi-class multi-kernel learning algorithm. In *4th IEEE Online Learning for Computer Vision Workshop (in CVPR10)*. IEEE Computer Society, June 2010b.

R. Jin, S. C. H. Hoi, and T. Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *Proc. of the 21st International Conference on Algorithmic Learning Theory*, pages 390–404, 2010.

T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods – Support Vector Learning*, pages 169–185. MIT Press, 1999.

S. Kakade, S. Shalev-Shwartz, and A. Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. Technical report, TTI, 2009.

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate ℓ_p-norm multiple kernel learning. In *Advances in Neural Information Processing Systems 22*, pages 997–1005, 2009.

G. Lanckriet, N. Cristianini, P. Bartlett, and L. E. Ghaoui. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, December 2005.

J. S. Nath, G. Dinesh, S. Ramanand, C. Bhattacharyya, A. Ben-Tal, and K. R. Ramakrishnan. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In *Advances in Neural Information Processing Systems 22*, pages 844–852, 2009.

M. E. Nilsback and B. Caputo. Cue integration through discriminative accumulation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

T. Ojala, M. Pietikaäinen, and T. Maäenpaäaä. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

F. Orabona. *DOGMA: a MATLAB toolbox for Online Learning*, 2009. Software available at `http://dogma.sourceforge.net`.

F. Orabona and K. Crammer. New adaptive algorithms for online classification. In *Advances in Neural Information Processing Systems*, December 2010.

F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kernel learning. In *Proc. of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, June 2010.

N. Pinto, D. D. Cox, and J. J. Dicarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1), January 2008.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.

A. Pronobis, J. Luo, and B. Caputo. The more you learn, the less you store: Memory-controlled incremental SVM for visual place recognition. *Image and Vision Computing (IMAVIS), Special Issue on Online Pattern Recognition and Machine Learning Techniques for Computer-Vision: Theory and Applications*, 28(7):1080–1097, July 2010.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, November 2008.

E. Rubinstein. Support vector machines via advanced optimization techniques. Technical report, Masters thesis, Faculty of Electrical Engineering, Technion, Nov 2005.

C. Sanderson and K. K. Paliwal. Identity verification using speech and face information. *Digital Signal Processing*, 14(5):449–480, 2004.

S. Shalev-Shwartz and Y. Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical Report 2007-42, The Hebrew University, 2007.

S. Shalev-Shwartz and N. Srebro. SVM, optimization: inverse dependence on training set size. In *Proc. of the International Conference on Machine Learning*, 2008.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. of the International Conference on Machine Learning*, 2007.

S. Sonnenburg, G. Rätsch, and C. Schäfer. Learning interpretable SVMs for biological sequence classification. In *Research in Computational Molecular Biology, 9th Annual International Conference, RECOMB 2005, Cambridge, MA, USA, May 14-18, 2005, Proceedings*, volume 3500 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2005.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, December 2006.

R. Tomioka and T. Suzuki. Sparsity-accuracy trade-off in MKL, 2010. URL `http://arxiv.org/abs/1001.2615`.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. of the International Conference on Machine Learning*, 2004.

O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on Riemannian manifolds. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *Proc. of the International Conference on Machine Learning*, 2009.

S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma. Multiple kernel learning and the SMO algorithm. In *Advances in Neural Information Processing Systems*, December 2010.

D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Z. Xu, R. Jin, I. King, and M. R. Lyu. An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems 21*, pages 1825–1832, 2008.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68:49–67, 2006.

A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proc. of the International Conference on Machine Learning*, 2007.

Journal of Machine Learning Research 13 (2012) 255-279 Submitted 10/10; Revised 7/11; Published 2/12

Active Learning via Perfect Selective Classification

Ran El-Yaniv RANI@CS.TECHNION.AC.IL
Yair Wiener WYAIR@TX.TECHNION.AC.IL
Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel

Editor: Yoav Freund

Abstract

We discover a strong relation between two known learning models: stream-based active learning and perfect selective classification (an extreme case of ‘classification with a reject option’). For these models, restricted to the realizable case, we show a reduction of active learning to selective classification that preserves fast rates. Applying this reduction to recent results for selective classification, we derive exponential target-independent label complexity speedup for actively learning general (non-homogeneous) linear classifiers when the data distribution is an arbitrary high dimensional mixture of Gaussians. Finally, we study the relation between the proposed technique and existing label complexity measures, including teaching dimension and disagreement coefficient.

Keywords: classification with a reject option, perfect classification, selective classification, active learning, selective sampling, disagreement coefficient, teaching dimension, exploration vs. exploitation

1. Introduction and Related Work

Active learning is an intriguing learning model that provides the learning algorithm with some control over the learning process, potentially leading to significantly faster learning. In recent years it has been gaining considerable recognition as a vital technique for efficiently implementing inductive learning in many industrial applications where abundance of unlabeled data exists, and/or in cases where labeling costs are high. In this paper we expose a strong relation between active learning and *selective classification*, another known alternative learning model (Chow, 1970; El-Yaniv and Wiener, 2010).

Focusing on binary classification in realizable settings we consider standard *stream-based active learning*, which is also referred to as *online selective sampling* (Atlas et al., 1990; Cohn et al., 1994). In this model the learner is given an error objective ϵ and then sequentially receives unlabeled examples. At each step, after observing an unlabeled example x, the learner decides whether or not to request the label of x. The learner should terminate the learning process and output a binary classifier whose true error is guaranteed to be at most ϵ with high probability. The penalty incurred by the learner is the number of label requests made and this number is called the *label complexity*. A label complexity bound of $O(d\log(d/\epsilon))$ for actively learning ϵ-good classifier from a concept class with VC-dimension d, provides an exponential speedup in terms of $1/\epsilon$ relative to standard (passive) supervised learning where the sample complexity is typically $O(d/\epsilon)$.

The study of (stream-based, realizable) active learning is paved with very interesting theoretical results. Initially, only a few cases were known where active learning provides significant advantage over passive learning. Perhaps the most favorable result was an exponential label complexity speedup for learning homogeneous linear classifiers where the (linearly separable) data is uniformly distributed over the unit sphere. This result was manifested by various authors using various analysis techniques, for a number of strategies that can all be viewed in hindsight as approximations or variations of the "CAL algorithm" of Cohn et al. (1994). Among these studies, the earlier theoretical results (Seung et al., 1992; Freund et al., 1993, 1997; Fine et al., 2002; Gilad-Bachrach, 2007) considered Bayesian settings and studied the speedup obtained by the Query by Committee (QBC) algorithm. The more recent results provided PAC style analyses (Dasgupta et al., 2009; Hanneke, 2007a, 2009).

Lack of positive results for other non-toy problems, as well as various additional negative results that were discovered, led some researchers to believe that active learning is not necessarily advantageous in general. Among the striking negative results is Dasgupta's negative example for actively learning general (non-homogeneous) linear classifiers (even in two dimensions) under the uniform distribution over the sphere (Dasgupta, 2005).

A number of recent innovative papers proposed alternative models for active learning. Balcan et al. (2008) introduced a subtle modification of the traditional label complexity definition, which opened up avenues for new positive results. According to their new definition of "non-verifiable" label complexity, the active learner is not required to know when to stop the learning process with a guaranteed ε-good classifier. Their main result, under this definition, is that active learning is asymptotically better than passive learning in the sense that only $o(1/\varepsilon)$ labels are required for actively learning an ε-good classifier from a concept class that has a finite VC-dimension. Another result they accomplished is an exponential label complexity speedup for (non-verifiable) active learning of non-homogeneous linear classifiers under the uniform distribution over the the unit sphere.

Based on Hanneke's characterization of active learning in terms of the "disagreement coefficient" (Hanneke, 2007a), Friedman (2009) recently extended the Balcan et al. results and proved that a target-dependent exponential speedup can be asymptotically achieved for a wide range of "smooth" learning problems (in particular, the hypothesis class, the instance space and the distribution should all be expressible by smooth functions). He proved that under such smoothness conditions, for any target hypothesis h^*, Hanneke's disagreement coefficient is bounded above in terms of a constant $c(h^*)$ that depends on the unknown target hypothesis h^* (and is independent of δ and ε). The resulting label complexity is $O(c(h^*)\, d \,\text{polylog}(d/\varepsilon))$ (Hanneke, 2011b). This is a very general result but the *target-dependent* constant involved in this bound is only guaranteed to be finite.

With this impressive progress in the case of target-dependent bounds for active learning, the current state of affairs in the *target-independent* bounds for active learning arena leaves much to be desired. To date the most advanced result in this model, which was already essentially established by Seung et al. and Freund et al. more than fifteen years ago (Seung et al., 1992; Freund et al., 1993, 1997), is still a target-independent exponential speed up bound for homogeneous linear classifiers under the uniform distribution over the sphere.

The other learning model we contemplate that will be shown to have strong ties to active learning, is *selective classification*, which is mainly known in the literature as 'classification with a reject option.' This old-timer model, that was already introduced more than fifty years ago (Chow, 1957, 1970), extends standard supervised learning by allowing the classifier to opt out from predictions in cases where it is not confident. The incentive is to increase classification reliability over instances that are not rejected by the classifier. Thus, using selective classification one can potentially achieve a lower error rate using the same labeling "budget." The main quantities that characterize a selective classifier are its (true) error and coverage rate (or its complement, the rejection rate).

There is already substantial volume of research publications on selective classification, that kept emerging through the years. The main theme in many of these publications is the implementation of certain reject mechanisms for specific learning algorithms like support vector machines and neural networks. Among the few theoretical studies on selective classification, there are various excess risk bounds for ERM learning (Herbei and Wegkamp, 2006; Bartlett and Wegkamp, 2008; Wegkamp, 2007), and certain coverage/risk guarantees for selective ensemble methods (Freund et al., 2004). In a recent work (El-Yaniv and Wiener, 2010) the trade-off between error and coverage was examined and in particular, a new extreme case of selective learning was introduced. In this extreme case, termed here "perfect selective classification," the classifier is given m labeled examples and is required to instantly output a classifier whose true error is perfectly zero with certainty. This is of course potentially doable only if the classifier rejects a sufficient portion of the instance space. A non-trivial result for perfect selective classification is a high probability lower bound on the classifier coverage (or equivalently, an upper bound on its rejection rate). Such bounds have recently been presented in El-Yaniv and Wiener (2010).

In Section 3 we present a reduction of active learning to perfect selective classification that preserves "fast rates." This reduction enables the luxury of analyzing *dynamic* active learning problems as *static* problems. Relying on a recent result on perfect selective classification from El-Yaniv and Wiener (2010), in Section 4 we then apply our reduction and conclude that general (non-homogeneous) linear classifiers are actively learnable at exponential (in $1/\varepsilon$) label complexity rate when the data distribution is an arbitrary unknown finite mixture of high dimensional Gaussians. While we obtain exponential label complexity speedup in $1/\varepsilon$, we incur exponential slowdown in d^2, where d is the problem dimension. Nevertheless, in Section 5 we prove a lower bound of $\Omega((\log m)^{(d-1)/2}(1+o(1))$ on the label complexity, when considering the class of unrestricted linear classifiers under a Gaussian distribution. Thus, an exponential slowdown in d is unavoidable in such settings.

Finally, in Section 6 we relate the proposed technique to other complexity measures for active learning. Proving and using a relation to the *teaching dimension* (Goldman and Kearns, 1995) we show, by relying on a known bound for the teaching dimension, that perfect selective classification with meaningful coverage can be achieved for the case of axis-aligned rectangles under a product distribution. We then focus on Hanneke's *disagreement coefficient* and show that the coverage of perfect selective classification can be bounded below using the disagreement coefficient. Conversely, we show that the disagreement coefficient can be bounded above using any coverage bound for perfect selective classification. Consequently, the results here imply that the disagreement coefficient can be sufficiently bounded to ensure fast active learning for the case of linear classifiers under a mixture of Gaussians.

2. Active Learning and Perfect Selective Classification

In *binary classification* the goal is to learn an accurate *binary classifier*, $h : \mathcal{X} \to \{\pm 1\}$, from a finite labeled training sample. Here $\mathcal{X}$ is some instance space and the standard assumption is that the training sample, $S_m = \{(x_i, y_i)\}_{i=1}^m$, containing m labeled examples, is drawn i.i.d. from some unknown distribution $P(X,Y)$ defined over $\mathcal{X} \times \{\pm 1\}$. The classifier h is chosen from some hypothesis class $\mathcal{H}$. In this paper we focus on the *realizable setting* whereby labels are defined by some unknown *target hypothesis* $h^* \in \mathcal{H}$. Thus, the underlying distribution reduces to $P(X)$. The performance of a classifier h is quantified by its true zero-one *error*, $R(h) \triangleq \Pr\{h(X) \neq h^*(X)\}$. A positive result for a classification problem $(\mathcal{H}, P)$ is a learning algorithm that given an error target ε and a confidence parameter δ can output, based on S_m, an hypothesis h whose error $R(h) \leq \varepsilon$, with probability of at least $1-\delta$. A bound $B(\varepsilon, \delta)$ on the size m of labeled training sample sufficient for achieving this is called the *sample complexity* of the learning algorithm. A classical result is that any consistent learning algorithm has sample complexity of $O(\frac{1}{\varepsilon}(d\log(\frac{1}{\varepsilon}) + \log(\frac{1}{\delta})))$, where d is the VC-dimension of $\mathcal{H}$ (see, e.g., Anthony and Bartlett, 1999).

2.1 Active Learning

We consider the following standard active learning model. In this model the learner sequentially observes unlabeled instances, $x_1, x_2, \ldots$, that are sampled i.i.d. from $P(X)$. After receiving each x_i, the learning algorithm decides whether or not to request its label $h^*(x_i)$, where $h^* \in \mathcal{H}$ is an unknown target hypothesis. Before the start of the game the algorithm is provided with some desired error rate ε and confidence level δ. We say that the learning algorithm *actively learned* the problem instance $(\mathcal{H}, P)$ if at some point it can terminate this process, after observing m instances and requesting k labels, and output an hypothesis $h \in \mathcal{H}$ whose error $R(h) \leq \varepsilon$, with probability of at least $1-\delta$. The quality of the algorithm is quantified by the number k of requested labels, which is called the *label complexity*. A positive result for a learning problem $(\mathcal{H}, P)$ is a learning algorithm that can actively learn this problem for any given ε and δ, and for every h^*, with label complexity bounded above by $L(\varepsilon, \delta, h^*)$. If there is a label complexity bound that is $O(polylog(1/\varepsilon))$ we say that the problem is *actively learnable at exponential rate*.

2.2 Selective Classification

Following the formulation in El-Yaniv and Wiener (2010) the goal in selective classification is to learn a pair of functions (h, g) from a labeled training sample S_m (as defined above for passive learning). The pair (h, g), which is called a *selective classifier*, consists of a binary classifier $h \in \mathcal{H}$, and a *selection function*, $g : \mathcal{X} \to \{0, 1\}$, which qualifies the classifier h as follows. For any sample $x \in \mathcal{X}$, the output of the selective classifier is $(h, g)(x) \triangleq h(x)$ iff $g(x) = 1$, and $(h, g)(x) \triangleq$ *abstain* iff $g(x) = 0$. Thus, the function g is a filter that determines a sub-domain of $\mathcal{X}$ over which the selective classifier will abstain from classifications. A selective classifier is thus characterized by its *coverage*, $\Phi(h, g) \triangleq \mathbf{E}_P\{g(x)\}$, which is the P-weighted volume of the sub-domain of $\mathcal{X}$ that is not filtered out, and its *error*, $R(h, g) = \mathbf{E}\{\mathbb{I}(h(X) \neq h^*(X)) \cdot g(X)\}/\Phi(h, g)$, which is the zero-one loss restricted to the covered sub-domain. Note that this is a "smooth" generalization of passive learning and, in particular, $R(h, g)$ reduces to $R(h)$ (standard classification) if $g(x) \equiv 1$. We expect to see a trade-off between $R(h, g)$ and $\Phi(h, g)$ in the sense that smaller error should be obtained by compromising the coverage. A major issue in selective classification is how to optimally control this trade-off. In this paper we are concerned with an extreme case of this trade-off whereby (h, g) is required to achieve a perfect score of *zero error with certainty*. This extreme learning objective is termed *perfect learning* in El-Yaniv and Wiener (2010). Thus, for a *perfect selective classifier* (h, g) we always have $R(h, g) = 0$, and its quality is determined by its guaranteed coverage. A positive result for (perfect) selective classification problem $(\mathcal{H}, P)$ is a learning algorithm that uses a labeled training sample S_m (as in passive learning) to output a perfect selective classifier (h, g) for which $\Phi(h, g) \geq B_\Phi(\mathcal{H}, \delta, m)$ with probability of at least $1-\delta$, for any given δ. The bound

$B_\Phi = B_\Phi(\mathcal{H},\delta,m)$ is called a *coverage bound* (or *coverage rate*) and its complement, $1-B_\Phi$, is called a *rejection bound* (or *rate*). A coverage rate $B_\Phi = 1-O(\frac{polylog(m)}{m})$ (and the corresponding $1-B_\Phi$ rejection rate) are qualified as *fast*.

2.3 The CAL Algorithm and the Consistent Selective Strategy (CSS)

The major players in active learning and in perfect selective classification are the CAL algorithm and the consistent selective strategy (CSS), respectively. To define them we need the following definitions.

Definition 1 (Version space, Mitchell, 1977) *Given an hypothesis class $\mathcal{H}$ and a training sample S_m, the* version space $VS_{\mathcal{H},S_m}$ *is the set of all hypotheses in $\mathcal{H}$ that classify S_m correctly.*

Definition 2 (Disagreement set, Hanneke, 2007a; El-Yaniv and Wiener, 2010) *Let $\mathcal{G}\subset\mathcal{H}$. The* disagreement set *w.r.t. $\mathcal{G}$ is defined as*

$$DIS(\mathcal{G}) \triangleq \{x\in\mathcal{X} : \exists h_1,h_2\in\mathcal{G} \quad s.t. \quad h_1(x)\neq h_2(x)\}.$$

The agreement set *w.r.t. $\mathcal{G}$ is $AGR(\mathcal{G}) \triangleq \mathcal{X}\setminus DIS(\mathcal{G})$.*

The main strategy for active learning in the realizable setting (Cohn et al., 1994) is to request labels only for instances belonging to the disagreement set and output any (consistent) hypothesis belonging to the version space. This strategy is often called the *CAL algorithm*. A related strategy for perfect selective classification was proposed in El-Yaniv and Wiener (2010) and termed *consistent selective strategy (CSS)*. Given a training set S_m, CSS takes the classifier h to be any hypothesis in $VS_{\mathcal{H},S_m}$ (i.e., a consistent learner), and takes a selection function g that equals one for all points in the agreement set with respect to $VS_{\mathcal{H},S_m}$, and zero otherwise.

3. From Coverage Bound to Label Complexity Bound

In this section we present a reduction from stream-based active learning to perfect selective classification. Particularly, we show that if there exists for $\mathcal{H}$ a perfect selective classifier with a fast rejection rate of $O(polylog(m)/m)$, then the CAL algorithm will actively learn $\mathcal{H}$ with exponential label complexity rate of $O(polylog(1/\varepsilon))$.

Lemma 3 *Let $S_m = \{(x_1,y_1),\ldots,(x_m,y_m)\}$ be a sequence of m labeled samples drawn i.i.d. from an unknown distribution $P(X)$ and let $S_i = \{(x_1,y_1),\ldots,(x_i,y_i)\}$ be the i-prefix of S_m. Then, with probability of at least $1-\delta$ over random choices of S_m, the following bound holds simultaneously for all $i=1,\ldots,m-1$,*

$$\Pr\left\{x_{i+1}\in DIS(VS_{\mathcal{H},S_i})|S_i\right\} \leq 1-B_\Phi\left(\mathcal{H},\frac{\delta}{\log_2(m)},2^{\lfloor\log_2(i)\rfloor}\right),$$

where $B_\Phi(\mathcal{H},\delta,m)$ is a coverage bound for perfect selective classification with respect to hypothesis class $\mathcal{H}$, confidence δ and sample size m .

Proof For $j = 1, \ldots, m$, abbreviate $DIS_j \triangleq DIS(VS_{\mathcal{H},S_j})$ and $AGR_j \triangleq AGR(VS_{\mathcal{H},S_j})$. By definition, $DIS_j = \mathcal{X} \setminus AGR_j$. By the definitions of a coverage bound and agreement/disagreement sets, with probability of at least $1-\delta$ over random choices of S_j

$$B_\Phi(\mathcal{H}, \delta, j) \leq \Pr\{x \in AGR_j | S_j\} = \Pr\{x \notin DIS_j | S_j\} = 1 - \Pr\{x \in DIS_j | S_j\}.$$

Applying the union bound we conclude that the following inequality holds simultaneously with high probability for $t = 0, \ldots, \lfloor \log_2(m) \rfloor - 1$,

$$\Pr\{x_{2^t+1} \in DIS_{2^t} | S_{2^t}\} \leq 1 - B_\Phi\left(\mathcal{H}, \frac{\delta}{\log_2(m)}, 2^t\right). \qquad (1)$$

For all $j \leq i$, $S_j \subseteq S_i$, so $DIS_i \subseteq DIS_j$. Therefore, since the samples in S_m are all drawn i.i.d., for any $j \leq i$,

$$\Pr\left\{x_{i+1} \in DIS_i | S_i\right\} \leq \Pr\left\{x_{i+1} \in DIS_j | S_j\right\} = \Pr\left\{x_{j+1} \in DIS_j | S_j\right\}.$$

The proof is complete by setting $j = 2^{\lfloor \log_2(i) \rfloor} \leq i$, and applying inequality (1). ■

Lemma 4 (Bernstein's inequality Hoeffding, 1963) *Let $X_1, \ldots, X_n$ be independent zero-mean random variables. Suppose that $|X_i| \leq M$ almost surely, for all i. Then, for all positive t,*

$$\Pr\left\{\sum_{i=1}^{n} X_i > t\right\} \leq \exp\left(-\frac{t^2/2}{\sum \mathbf{E}\left\{X_j^2\right\} + Mt/3}\right).$$

Lemma 5 *Let Z_i, $i = 1, \ldots, m$, be independent Bernoulli random variables with success probabilities p_i. Then, for any $0 < \delta < 1$, with probability of at least $1-\delta$,*

$$\sum_{i=1}^{m} (Z_i - \mathbf{E}\{Z_i\}) \leq \sqrt{2 \ln \frac{1}{\delta} \sum p_i} + \frac{2}{3} \ln \frac{1}{\delta}.$$

Proof Define $W_i \triangleq Z_i - \mathbf{E}\{Z_i\} = Z_i - p_i$. Clearly,

$$\mathbf{E}\{W_i\} = 0, \quad |W_i| \leq 1, \quad \mathbf{E}\{W_i^2\} = p_i(1-p_i).$$

Applying Bernstein's inequality (Lemma 4) on the W_i,

$$\begin{aligned}\Pr\left\{\sum_{i=1}^{n} W_i > t\right\} &\leq \exp\left(-\frac{t^2/2}{\sum \mathbf{E}\left[W_j^2\right] + t/3}\right) = \exp\left(-\frac{t^2/2}{\sum p_i(1-p_i) + t/3}\right) \\ &\leq \exp\left(-\frac{t^2/2}{\sum p_i + t/3}\right).\end{aligned}$$

Equating the right-hand side to δ and solving for t, we have

$$\frac{t^2/2}{\sum p_i + t/3} = \ln \frac{1}{\delta} \iff t^2 - t \cdot \frac{2}{3} \ln \frac{1}{\delta} - 2 \ln \frac{1}{\delta} \sum p_i = 0,$$

and the positive solution of this quadratic equation is

$$t = \frac{1}{3}\ln\frac{1}{\delta} + \sqrt{\frac{1}{9}\ln^2\frac{1}{\delta} + 2\ln\frac{1}{\delta}\sum p_i} < \frac{2}{3}\ln\frac{1}{\delta} + \sqrt{2\ln\frac{1}{\delta}\sum p_i}.$$

■

Lemma 6 *Let $Z_1, Z_2, \ldots, Z_m$ be a high order Markov sequence of dependent binary random variables defined in the same probability space. Let $X_1, X_2, \ldots, X_m$ be a sequence of independent random variables such that,*

$$\Pr\{Z_i = 1 | Z_{i-1}, \ldots, Z_1, X_{i-1}, \ldots, X_1\} = \Pr\{Z_i = 1 | X_{i-1}, \ldots, X_1\}.$$

Define $P_1 \triangleq \Pr\{Z_1 = 1\}$, and for $i = 2, \ldots, m$,

$$P_i \triangleq \Pr\{Z_i = 1 | X_{i-1}, \ldots, X_1\}.$$

Let $b_1, b_2 \ldots b_m$ be given constants independent of $X_1, X_2, \ldots, X_m$.[1] *Assume that $P_i \le b_i$ simultaneously for all i with probability of at least $1 - \delta/2$, $\delta \in (0,1)$. Then, with probability of at least $1 - \delta$,*

$$\sum_{i=1}^{m} Z_i \le \sum_{i=1}^{m} b_i + \sqrt{2\ln\frac{2}{\delta}\sum b_i} + \frac{2}{3}\ln\frac{2}{\delta}.$$

We proceed with a direct proof of Lemma 6. An alternative proof of this lemma, using supermartingales, appears in Appendix B.

Proof For $i = 1, \ldots, m$, let W_i be binary random variables satisfying

$$\begin{aligned}
\Pr\{W_i = 1 | Z_i = 1, X_{i-1}, \ldots, X_1\} &\triangleq \frac{b_i + \mathbb{I}(P_i \le b_i) \cdot (P_i - b_i)}{P_i}, \\
\Pr\{W_i = 1 | Z_i = 0, X_{i-1}, \ldots, X_1\} &\triangleq \max\left\{\frac{b_i - P_i}{1 - P_i}, 0\right\}, \\
\Pr\{W_i = 1 | W_{i-1}, \ldots, W_1, X_{i-1}, \ldots, X_1\} &= \Pr\{W_i = 1 | X_{i-1}, \ldots, X_1\}.
\end{aligned}$$

We notice that

$$\begin{aligned}
\Pr\{W_i = 1 | X_{i-1}, \ldots, X_1\} &= \Pr\{W_i = 1, Z_i = 1 | X_{i-1}, \ldots, X_1\} \\
&+ \Pr\{W_i = 1, Z_i = 0 | X_{i-1}, \ldots, X_1\} \\
&= \Pr\{W_i = 1 | Z_i = 1, X_{i-1}, \ldots, X_1\} \Pr\{Z_i = 1 | X_{i-1}, \ldots, X_1\} \\
&+ \Pr\{W_i = 1 | Z_i = 0, X_{i-1}, \ldots, X_1\} \Pr\{Z_i = 0 | X_{i-1}, \ldots, X_1\} \\
&= \begin{cases} P_i + \frac{b_i - P_i}{1 - P_i}(1 - P_i) = b_i, & P_i \le b_i; \\ \frac{b_i}{P_i} \cdot P_i + 0 = b_i, & \text{else}. \end{cases}
\end{aligned}$$

Hence the distribution of each W_i is independent of $X_{i-1}, \ldots, X_1$, and the W_i are independent Bernoulli random variables with success probabilities b_i. By construction if $P_i \le b_i$ then

$$\Pr\{W_i = 1 | Z_i = 1\} = \int_X \Pr\{W_i = 1 | Z_i = 1, X_{i-1}, \ldots, X_1\} = 1.$$

[1] Precisely we require that each of the b_i were selected before X_i are chosen

By assumption $P_i \leq b_i$ for all i simultaneously with probability of at least $1-\delta/2$. Therefore, $Z_i \leq W_i$ simultaneously with probability of at least $1-\delta/2$. We now apply Lemma 5 on the W_i. The proof is then completed using the union bound. ■

Theorem 7 *Let S_m be a sequence of m unlabeled samples drawn i.i.d. from an unknown distribution P. Then with probability of at least $1-\delta$ over choices of S_m, the number of label requests k by the CAL algorithm is bounded by*

$$k \leq \Psi(\mathcal{H},\delta,m)+\sqrt{2\ln\frac{2}{\delta}\Psi(\mathcal{H},\delta,m)}+\frac{2}{3}\ln\frac{2}{\delta},$$

where

$$\Psi(\mathcal{H},\delta,m) \triangleq \sum_{i=1}^{m}\left(1-B_\Phi\left(\mathcal{H},\frac{\delta}{2\log_2(m)},2^{\lfloor\log_2(i)\rfloor}\right)\right)$$

and $B_\Phi(\mathcal{H},\delta,m)$ is a coverage bound for perfect selective classification with respect to hypothesis class $\mathcal{H}$, confidence δ and sample size m .

Proof According to CAL, the label of sample x_i will be requested iff $x_i \in DIS(VS_{\mathcal{H},S_{i-1}})$. For $i=1,\ldots,m$, let Z_i be binary random variables such that $Z_i \triangleq 1$ iff CAL requests a label for sample x_i. Applying Lemma 3 we get that for all $i=2,\ldots,m$, with probability of at least $1-\delta/2$

$$\Pr\{Z_i=1|S_{i-1}\} \quad = \quad \Pr\left\{x_i \in DIS(VS_{\mathcal{H},S_{i-1}})|S_{i-1}\right\} \leq 1-B_\Phi\left(\mathcal{H},\frac{\delta}{2\log_2(m)},2^{\lfloor\log_2(i-1)\rfloor}\right).$$

For $i=1$, $B_\Phi(\mathcal{H},\delta,1)=0$ and the above inequality trivially holds. An application of Lemma 6 on the variables Z_i completes the proof. ■

Theorem 7 states an upper bound on the label complexity expressed in terms of m, the size of the sample provided to CAL. This upper bound is very convenient for directly analyzing the active learning speedup relative to supervised learning. A standard label complexity upper bound, which depends on $1/\varepsilon$, can be extracted using the following simple observation.

Lemma 8 (Hanneke, 2009; Anthony and Bartlett, 1999) *Let S_m be a sequence of m unlabeled samples drawn i.i.d. from an unknown distribution P. Let $\mathcal{H}$ be a hypothesis class whose finite VC dimension is d, and let ε and δ be given. If*

$$m \geq \frac{4}{\varepsilon}\left(d\ln\frac{12}{\varepsilon}+\ln\frac{2}{\delta}\right),$$

then, with probability of at least $1-\delta$, CAL will output a classifier whose true error is at most ε.

Proof Hanneke (2009) observed that since CAL requests a label whenever there is a disagreement in the version space, it is guaranteed that after processing m examples, CAL will output a classifier that is consistent with all the m examples introduced to it. Therefore, CAL is a consistent learner. A classical result (Anthony and Bartlett, 1999, Theorem 4.8) is that any consistent learner will achieve, with probability of at least $1-\delta$, a true error not exceeding ε after observing at most $\frac{4}{\varepsilon}\left(d\ln\frac{12}{\varepsilon}+\ln\frac{2}{\delta}\right)$ labeled examples. ■

Theorem 9 *Let $\mathcal{H}$ be a hypothesis class whose finite VC dimension is d. If the rejection rate of CSS (see definition in Section 2.3) is $O\left(\frac{polylog\left(\frac{m}{\delta}\right)}{m}\right)$, then $(\mathcal{H},P)$ is actively learnable with exponential label complexity speedup.*

Proof Plugging this rejection rate into Ψ (defined in Theorem 7) we have,

$$\Psi(\mathcal{H},\delta,m) \triangleq \sum_{i=1}^{m}\left(1-B_{\Phi}(\mathcal{H},\frac{\delta}{\log_2(m)},2^{\lfloor\log_2(i)\rfloor})\right) = \sum_{i=1}^{m} O\left(\frac{polylog\left(\frac{i\log(m)}{\delta}\right)}{i}\right).$$

Applying Lemma 41 we get

$$\Psi(\mathcal{H},\delta,m) = O\left(polylog\left(\frac{m\log(m)}{\delta}\right)\right).$$

By Theorem 7, $k = O\left(polylog\left(\frac{m}{\delta}\right)\right)$, and an application of Lemma 8 concludes the proof. ■

4. Label Complexity Bounding Technique and Its Applications

In this section we present a novel technique for deriving target-independent label complexity bounds for active learning. The technique combines the reduction of Theorem 7 and a general data-dependent coverage bound for selective classification from El-Yaniv and Wiener (2010). For some learning problems it is a straightforward technical exercise, involving VC-dimension calculations, to arrive with exponential label complexity bounds. We show a few applications of this technique resulting in both reproductions of known label complexity exponential rates as well as a new one. The following definitions (El-Yaniv and Wiener, 2010) are required for introducing the technique.

Definition 10 (Version space compression set) *For any hypothesis class $\mathcal{H}$, let S_m be a labeled sample of m points inducing a version space $VS_{\mathcal{H},S_m}$. The* version space compression set*, $S' \subseteq S_m$, is a smallest subset of S_m satisfying $VS_{\mathcal{H},S_m} = VS_{\mathcal{H},S'}$. The (unique) number $\hat{n} = \hat{n}(\mathcal{H},S_m) = |S'|$ is called the version space compression set* size.

Remark 11 *Our "version space compression set" is precisely Hanneke's "minimum specifying set" (Hanneke, 2007b) for f on U with respect to V, where,*

$$f = h^*, \quad U = S_m, \quad V = \mathcal{H}[S_m] \quad \text{(see Definition 23).}$$

Definition 12 (Characterizing hypothesis) *For any subset of hypotheses $\mathcal{G} \subseteq \mathcal{H}$, the* characterizing hypothesis *of $\mathcal{G}$, denoted $f_{\mathcal{G}}(x)$, is a binary hypothesis over X (not restricted to $\mathcal{H}$) obtaining positive values over the agreement set $AGR(\mathcal{G})$ (Definition 2), and zero otherwise.*

Definition 13 (Order-n characterizing set) *For each n, let Σ_n be the set of all possible labeled samples of size n (all n-subsets, each with all 2^n possible labelings). The order-n* characterizing set *of $\mathcal{H}$, denoted $\mathcal{F}_n$, is the set of all characterizing hypotheses $f_{\mathcal{G}}(x)$, where $\mathcal{G} \subseteq \mathcal{H}$ is a version space induced by some member of Σ_n.*

Definition 14 (Characterizing set complexity) *Let $\mathcal{F}_n$ be the order-n characterizing set of $\mathcal{H}$. The order-n* characterizing set complexity *of $\mathcal{H}$, denoted $\gamma(\mathcal{H},n)$, is the VC-dimension of $\mathcal{F}_n$.*

The following theorem, credited to (El-Yaniv and Wiener, 2010, Theorem 21), is a powerful data-dependent coverage bound for perfect selective learning, expressed in terms of the version space compression set size and the characterizing set complexity.

Theorem 15 (Data-dependent coverage guarantee) *For any m, let $a_1, a_2, \ldots, a_m \in \mathbb{R}$ be given, such that $a_i \geq 0$ and $\sum_{i=1}^{m} a_i \leq 1$. Let (h,g) be perfect selective classifier (CSS, see Section 2.3). Then, $R(h,g) = 0$, and for any $0 \leq \delta \leq 1$, with probability of at least $1-\delta$,*

$$\Phi(h,g) \geq 1 - \frac{2}{m}\left[\gamma(\mathcal{H},\hat{n}) \ln_{+}\left(\frac{2em}{\gamma(\mathcal{H},\hat{n})}\right) + \ln\frac{2}{a_{\hat{n}}\delta}\right],$$

where $\hat{n}$ is the size of the version space compression set and $\gamma(\mathcal{H},\hat{n})$ is the order-$\hat{n}$ characterizing set complexity of $\mathcal{H}$.

Given an hypothesis class $\mathcal{H}$, our recipe to deriving active learning label complexity bounds for $\mathcal{H}$ is: (i) calculate both $\hat{n}$ and $\gamma(\mathcal{H},\hat{n})$; (ii) apply Theorem 15, obtaining a bound B_{Φ} for the coverage; (iii) plug B_{Φ} in Theorem 7 to get a label complexity bound expressed as a summation; (iv) Apply Lemma 41 to obtain a label complexity bound in a closed form.

4.1 Examples

In the following example we derive a label complexity bound for the concept class of thresholds (linear separators in $\mathbb{R}$). Although this is a toy example (for which an exponential rate is well known) it does exemplify the technique, and in many other cases the application of the technique is not much harder. Let $\mathcal{H}$ be the class of thresholds. We first show that the corresponding version space compression set size $\hat{n} \leq 2$. Assume w.l.o.g. that $h^*(x) \triangleq \mathbb{I}(x > w)$ for some $w \in (0,1)$. Let $x_{-} \triangleq \max\{x_i \in S_m | y_i = -1\}$ and $x_{+} \triangleq \min(x_i \in S_m | y_i = +1)$. At least one of x_{-} or x_{+} exist. Let $S'_m = \{(x_{-},-1),(x_{+},+1)\}$. Then $VS_{\mathcal{H},S_m} = VS_{\mathcal{H},S'_m}$, and $\hat{n} = |S'_m| \leq 2$. Now, $\gamma(\mathcal{H},2) = 2$, because the order-2 characterizing set of $\mathcal{H}$ is the class of intervals in $\mathbb{R}$ whose VC-dimension is 2. Plugging these numbers in Theorem 15, and using the assignment $a_1 = a_2 = 1/2$,

$$B_{\Phi}(\mathcal{H},\delta,m) = 1 - \frac{2}{m}\left[2\ln(em) + \ln\frac{4}{\delta}\right] = 1 - O\left(\frac{\ln(m/\delta)}{m}\right).$$

Next we plug B_{Φ} in Theorem 7 obtaining a raw label complexity

$$\Psi(\mathcal{H},\delta,m) = \sum_{i=1}^{m}\left(1 - B_{\Phi}\left(\mathcal{H}, \frac{\delta}{2\log_2(m)}, 2^{\lfloor \log_2(i) \rfloor}\right)\right) = \sum_{i=1}^{m} O\left(\frac{\ln(\log_2(m) \cdot i/\delta)}{i}\right).$$

Finally, by applying Lemma 41, with $a = 1$ and $b = \log_2 m/\delta$, we conclude that

$$\Psi(\mathcal{H},\delta,m) = O\left(\ln^2\left(\frac{m}{\delta}\right)\right).$$

Thus, $\mathcal{H}$ is actively learnable with exponential speedup, and this result applies to any distribution. In Table 1 we summarize the $\hat{n}$ and $\gamma(\mathcal{H},\hat{n})$ values we calculated for four other hypothesis classes. The

Hypothesis class	Distribution	$\hat{n}$	$\gamma(\mathcal{H},\hat{n})$
Linear separators in $\mathbb{R}$	any	2	2
Intervals in $\mathbb{R}$	any (target-dependent)[2]	4	4
Linear separators in $\mathbb{R}^2$	any distribution on the unit circle (target-dependent)[2]	4	4
Linear separators in $\mathbb{R}^d$	mixture of Gaussians	$O\left((\log m)^{d-1}/\delta\right)$	$O\left(\hat{n}^{d/2+1}\right)$
Balanced axis-aligned rectangles in $\mathbb{R}^d$	product distribution	$O(\log(dm/\delta))$	$O(d\hat{n}\log\hat{n})$

Table 1: The $\hat{n}$ and γ of various hypothesis spaces achieving exponential rates.

last two cases are fully analyzed in Sections 4.2 and 6.1, respectively. For the other classes, where γ and $\hat{n}$ are constants, it is clear (Theorem 15) that exponential rates are obtained. We emphasize that the bounds for these two classes are target-dependent as they require that S_m include at least one sample from each class.

4.2 Linear Separators in $\mathbb{R}^d$ Under Mixture of Gaussians

In this section we state and prove our main example, an exponential label complexity bound for linear classifiers in $\mathbb{R}^d$.

Theorem 16 *Let $\mathcal{H}$ be the class of all linear binary classifiers in $\mathbb{R}^d$, and let the underlying distribution be any mixture of a fixed number of Gaussians in $\mathbb{R}^d$. Then, with probability of at least $1-\delta$ over choices of S_m, the number of label requests k by CAL is bounded by*

$$k = O\left(\frac{(\log m)^{d^2+1}}{\delta^{(d+3)/2}}\right).$$

Therefore by Lemma 8 we get $k = O\left(poly(1/\delta)\cdot polylog(1/\varepsilon)\right)$.

Proof The following is a coverage bound for linear classifiers in d dimensions that holds in our setting with probability of at least $1-\delta$ (El-Yaniv and Wiener, 2010, Corollary 33),[3]

$$\Phi(h,g) \geq 1 - O\left(\frac{(\log m)^{d^2}}{m}\cdot\frac{1}{\delta^{(d+3)/2}}\right). \tag{2}$$

[2] Target-dependent with at least one sample in each class.

[3] This bound uses the fact that for linear classifiers in d dimensions $\hat{n} = O\left((\log m)^{d-1}/\delta\right)$ (El-Yaniv and Wiener, 2010, Lemma 32), and that $\gamma(\mathcal{H},\hat{n}) = O\left(\hat{n}^{d/2+1}\right)$ (El-Yaniv and Wiener, 2010, Lemma 27).

Plugging this bound in Theorem 7 we obtain,

$$\begin{aligned}\Psi(\mathcal{H},\delta,m) &= \sum_{i=1}^{m}\left(1-B_{\Phi}\left(\mathcal{H},\frac{\delta}{2\log_2(m)},2^{\lfloor\log_2(i)\rfloor}\right)\right)\\ &= \sum_{i=1}^{m} O\left(\frac{(\log i)^{d^2}}{i}\cdot\left(\frac{\log_2(m)}{\delta}\right)^{\frac{d+3}{2}}\right)\\ &= O\left(\left(\frac{\log_2(m)}{\delta}\right)^{\frac{d+3}{2}}\cdot\sum_{i=1}^{m}\frac{(\log(i))^{d^2}}{i}\right).\end{aligned}$$

Finally, an application of Lemma 41 with $a = d^2$ and $b = 1$ completes the proof. ■

5. Lower Bound on Label Complexity

In the previous section we have derived an upper bound on the label complexity of CAL for various classifiers and distributions. In the case of linear classifiers in $\mathbb{R}^d$ we have shown an exponential speed up in terms of $1/\varepsilon$ but also an exponential slow down in terms of the dimension d. In passive learning there is a linear dependency in the dimension while in our case (active learning using CAL) there is an exponential one. Is it an artifact of our bounding technique or a fundamental phenomenon?

To answer this question we derive an asymptotic lower bound on the label complexity. We show that the exponential dependency in d is unavoidable (at least asymptotically) for every bounding technique when considering linear classifier even under a single Gaussian (isotropic) distribution. The argument is obtained by the observation that CAL has to request a label to any point on the convex hull of a sample S_m. The bound is obtained using known results from probabilistic geometry, which bound the first two moments of the number of vertices of a random polytope under the Gaussian distribution.

Definition 17 (Gaussian polytope) *Let $X_1,\ldots,X_m$ be i.i.d. random points in $\mathbb{R}^d$ with common standard normal distribution (with zero mean and covariance matrix $\frac{1}{2}I_d$). A Gaussian polytope P_m is the convex hull of these random points.*

Denote by $f_k(P_m)$ the number of k-faces in the Gaussian polytope P_m. Note that $f_0(P_m)$ is the number of vertices in P_m. The following two Theorems asymptotically bound the average and variance of $f_k(P_m)$.

Theorem 18 (Hug et al., 2004, Theorem 1.1) *Let $X_1,\ldots,X_m$ be i.i.d. random points in $\mathbb{R}^d$ with common standard normal distribution. Then*

$$\mathbf{E}f_k(P_m) = c_{(k,d)}\,(\log m)^{\frac{d-1}{2}}\cdot(1+o(1))$$

as $m\to\infty$, where $c_{(k,d)}$ is a constant depending only on k and d.

Theorem 19 (Hug and Reitzner, 2005, Theorem 1.1) *Let $X_1,\dots,X_m$ be i.i.d. random points in $\mathbb{R}^d$ with common standard normal distribution. Then there exists a positive constant c_d, depending only on the dimension, such that*

$$Var\left(f_k(P_m)\right) \le c_d \left(\log m\right)^{\frac{d-1}{2}}$$

for all $k \in \{0,\dots,d-1\}$.

We can now use Chebyshev's inequality to lower bound the number of vertices in P_m ($f_0(P_m)$) with high probability.

Theorem 20 *Let $X_1,\dots,X_m$ be i.i.d. random points in $\mathbb{R}^d$ with common standard normal distribution and $\delta > 0$ be given. Then with probability of at least $1-\delta$,*

$$f_0(P_m) \ge \left(c_d \left(\log m\right)^{\frac{d-1}{2}} - \frac{\tilde{c}_d}{\sqrt{\delta}} \left(\log m\right)^{\frac{d-1}{4}}\right) \cdot \left(1+o(1)\right)$$

as $m \to \infty$, where c_d and $\tilde{c}_d$ are constants depending only on d.

Proof Using Chebyshev's inequality (in the second inequality), as well as Theorem 19 we get

$$\begin{aligned}\Pr\left(f_0(P_m) > \mathbf{E}f_0(P_m) - t\right) &= 1 - \Pr\left(f_0(P_m) \le \mathbf{E}f_0(P_m) - t\right) \\ &\ge 1 - \Pr\left(\left|f_0(P_m) - \mathbf{E}f_0(P_m)\right| \ge t\right) \\ &\ge 1 - \frac{\mathrm{Var}\left(f_0(P_m)\right)}{t^2} \ge 1 - \frac{c_d}{t^2} \left(\log m\right)^{\frac{d-1}{2}}.\end{aligned}$$

Equating the RHS to $1-\delta$ and solving for t we get

$$t = \sqrt{c_d \frac{\left(\log m\right)^{\frac{d-1}{2}}}{\delta}}.$$

Applying Theorem 18 completes the proof. ■

Theorem 21 (Lower bound) *Let $\mathcal{H}$ be the class of linear binary classifiers in $\mathbb{R}^d$, and let the underlying distribution be standard normal distribution in $\mathbb{R}^d$. Then there exists a target hypothesis such that, with probability of at least $1-\delta$ over choices of S_m, the number of label requests k by CAL is bounded by*

$$k \ge \frac{c_d}{2} \left(\log m\right)^{\frac{d-1}{2}} \cdot \left(1+o(1)\right).$$

as $m \to \infty$, where c_d is a constant depending only on d.

Proof Let us look at the Gaussian polytope P_m induced by the random sample S_m. As long as all labels requested by CAL have the same value (the case of minuscule minority class) we note that every vertex of P_m falls in the region of disagreement with respect to any subset of S_m that do not include that specific vertex. Therefore, CAL will request label at least for each vertex of P_m. For sufficiently large m, in particular,

$$\log m \ge \left(\frac{2\tilde{c}_d}{c_d\sqrt{\delta}}\right)^{\frac{4}{d-1}},$$

we conclude the proof by applying Theorem 20. ■

6. Relation to Existing Label Complexity Measures

A number of complexity measures to quantify the speedup in active learning have been proposed. In this section we show interesting relations between our techniques and two well known measures, namely the teaching dimension (Goldman and Kearns, 1995) and the disagreement coefficient (Hanneke, 2009).

Considering first the teaching dimension, we prove in Lemma 26 that the version space compression set size is bounded above, with high probability, by the extended teaching dimension growth function (introduced by Hanneke, 2007b). Consequently, it follows that perfect selective classification with meaningful coverage can be achieved for the case of axis-aligned rectangles under a product distribution.

We then focus on Hanneke's disagreement coefficient and show in Theorem 34 that the coverage of CSS can be bounded below using the disagreement coefficient. Conversely, in Corollary 39 we show that the disagreement coefficient can be bounded above using any coverage bound for CSS. Consequently, the results here imply that the disagreement coefficient, $\theta(\varepsilon)$ grows slowly with $1/\varepsilon$ for the case of linear classifiers under a mixture of Gaussians.

6.1 Teaching Dimension

The teaching dimension is a label complexity measure proposed by Goldman and Kearns (1995). The dimension of the hypothesis class $\mathcal{H}$ is the minimum number of examples required to present to any consistent learner in order to uniquely identify any hypothesis in the class.

We now define the following variation of the extended teaching dimension (Hegedüs, 1995) due to Hanneke. Throughout we use the notation $h_1(S) = h_2(S)$ to denote the fact that the two hypotheses agree on the classification of all instances in S.

Definition 22 (Extended Teaching Dimension, Hegedüs, 1995; Hanneke, 2007b) *Let $V \subseteq \mathcal{H}$, $m \geq 0$, $U \in \mathcal{X}^m$,*

$$\forall f \in \mathcal{H}, \quad XTD(f,V,U) = \inf\{t \,|\, \exists R \subseteq U : |\{h \in V : h(R) = f(R)\}| \leq 1 \wedge |R| \leq t\}.$$

Definition 23 (Hanneke, 2007b) *For $V \subseteq \mathcal{H}$, $V[S_m]$ denotes any subset of V such that*

$$\forall h \in V, \quad |\{h' \in V[S_m] : h'(S_m) = h(S_m)\}| = 1.$$

Claim 24 *Let S_m be a sample of size m, $\mathcal{H}$ an hypothesis class, and $\hat{n} = n(\mathcal{H}, S_m)$, the version space compression set size. Then,*

$$XTD(h^*, \mathcal{H}[S_m], S_m) = \hat{n}.$$

Proof Let $S_{\hat{n}} \subseteq S_m$ be a version space compression set. Assume, by contradiction, that there exist two hypotheses $h_1, h_2 \in \mathcal{H}[S_m]$, each of which agrees on the given classifications of all examples in $S_{\hat{n}}$. Therefore, $h_1, h_2 \in VS_{\mathcal{H},S_{\hat{n}}}$, and by the definition of version space compression set, we get $h_1, h_2 \in VS_{\mathcal{H},S_m}$. Hence,

$$|\{h \in \mathcal{H}[S_m] : h(S_m) = h^*(S_m)\}| \geq 2,$$

which contradicts definition 23. Therefore,

$$|\{h \in \mathcal{H}[S_m] : h(S_{\hat{n}}) = h^*(S_{\hat{n}})\}| \leq 1,$$

and

$$XTD(h^*, \mathcal{H}[S_m], S_m) \leq |S_{\hat{n}}| = \hat{n}.$$

Let $R \subset S_m$ be any subset of size $|R| < \hat{n}$. Consequently, $VS_{\mathcal{H},S_m} \subset VS_{\mathcal{H},R}$, and there exist hypothesis, $h' \in VS_{\mathcal{H},R}$, that agrees with all labeled examples in R, but disagrees with at least one example in S_m. Thus,

$$h'(S_m) \neq h^*(S_m),$$

and according to definition 23, there exist hypotheses $h_1, h_2 \in \mathcal{H}[S_m]$ such that $h_1(S_m) = h'(S_m) \neq h^*(S_m) = h_2(S_m)$. But $h_1(R) = h_2(R) = h^*(R)$, so

$$|\{h \in V[S_m] : h(R) = h^*(R)\}| \geq 2.$$

It follows that $XTD(h^*, \mathcal{H}[S_m], S_m) \geq \hat{n}$. ■

Definition 25 (XTD Growth Function, Hanneke, 2007b) *For $m \geq 0$, $V \subseteq \mathcal{H}$, $\delta \in [0,1]$,*

$$XTD(V, P, m, \delta) = \inf\left\{t | \forall h \in \mathcal{H}, Pr\left\{XTD(h, V[S_m], S_m) > t\right\} \leq \delta\right\}.$$

Lemma 26 *Let $\mathcal{H}$ be an hypothesis class, P an unknown distribution, and $\delta > 0$. Then, with probability of at least $1-\delta$,*

$$\hat{n} \leq XTD(\mathcal{H}, P, m, \delta).$$

Proof According to Definition 25, with probability of at least $1-\delta$,

$$XTD(h^*, \mathcal{H}[S_m], S_m) \leq XTD(\mathcal{H}, P, m, \delta).$$

Applying Claim 24 completes the proof. ■

Lemma 27 (Balanced Axis-Aligned Rectangles, Hanneke, 2007b, Lemma 4) *If P is a product distribution on $\mathbb{R}^d$ with continuous CDF, and $\mathcal{H}$ is the set of axis-aligned rectangles such that $\forall h \in \mathcal{H}$, $Pr_{X \sim P}\{h(X) = +1\} \geq \lambda$, then,*

$$XTD(\mathcal{H}, P, m, \delta) \leq O\left(\frac{d^2}{\lambda} \log \frac{dm}{\delta}\right).$$

Lemma 28 Blumer et al., 1989, Lemma 3.2.3 *Let $\mathcal{F}$ be a binary hypothesis class of finite VC dimension $d \geq 1$. For all $k \geq 1$, define the k-fold union,*

$$\mathcal{F}_{k\cup} \triangleq \left\{\cup_{i=1}^{k} f_i : f_i \in \mathcal{F}, 1 \leq i \leq k\right\}.$$

Then, for all $k \geq 1$,

$$VC(\mathcal{F}_{k\cup}) \leq 2dk \log_2(3k).$$

Lemma 29 (order-n characterizing set complexity) *Let $\mathcal{H}$ be the class of axis-aligned rectangles in $\mathbb{R}^d$. Then,*

$$\gamma(\mathcal{H}, n) \leq O(dn \log n).$$

Proof Let $S_n = S_k^- \cup S_{n-k}^+$ be a sample of size n composed of k negative examples, $\{x_1, x_2, \dots x_k\}$, and $n-k$ positive ones. Let $\mathcal{H}$ be the class of axis-aligned rectangles. We define,

$$\forall 1 \leq i \leq k, \qquad R_i \triangleq S_{n-k}^+ \cup \{(x_i, -1)\}.$$

Notice that $VS_{\mathcal{H},R_i}$ includes all axis aligned rectangles that classify all samples in S^+ as positive, and x_i as negative. Therefore, the agreement region of $VS_{\mathcal{H},R_i}$ is composed of two components as depicted in Figure 1. The first component is the smallest rectangle that bounds the positive samples, and the second is an unbounded convex polytope defined by up to d hyperplanes intersecting at x_i. Let AGR_i be the agreement region of $VS_{\mathcal{H},R_i}$ and AGR the agreement region of $VS_{\mathcal{H},S_n}$. Clearly, $R_i \subseteq S_n$, so $VS_{\mathcal{H},S_n} \subseteq VS_{\mathcal{H},R_i}$, and $AGR_i \subseteq AGR$, and it follows that

$$\bigcup_{i=1}^{k} AGR_i \subseteq AGR.$$

Assume, by contradiction, that $x \in AGR$ but $x \notin \bigcup_{i=1}^{k} AGR_i$. Therefore, for any $1 \leq i \leq k$, there exist two hypotheses $h_1^{(i)}, h_2^{(i)} \in VS_{\mathcal{H},R_i}$, such that, $h_1^{(i)}(x) \neq h_2^{(i)}(x)$. Assume, without loss of generality, that $h_1^{(i)}(x) = 1$. We define

$$h_1 \triangleq \bigwedge_{i=1}^{k} h_1^{(i)} \quad \text{and} \quad h_2 \triangleq \bigwedge_{i=1}^{k} h_2^{(i)},$$

meaning that h_1 classifies a sample as positive if and only if all hypotheses $h_1^{(i)}$ classify it as positive. Noting that the intersection of axis-aligned rectangles is itself an axis-aligned rectangle, we know that $h_1, h_2 \in \mathcal{H}$. Moreover, for any x_i we have, $h_1^{(i)}(x_i) = h_2^{(i)}(x_i) = -1$, so also $h_1(x_i) = h_2(x_i) = -1$, and $h_1, h_2 \in VS_{\mathcal{H},S_n}$. But $h_1(x) \neq h_2(x)$. Contradiction. Therefore,

$$AGR = \bigcup_{i=1}^{k} AGR_i.$$

It is well known that the VC dimension of a hyper-rectangle in $\mathbb{R}^d$ is $2d$. The VC dimension of AGR_i is bounded by the VC dimension of the union of two hyper-rectangles in $\mathbb{R}^d$. Furthermore, the VC dimension of AGR is bounded by the VC dimension of the union of all AGR_i. Applying Lemma 28 twice we get,

$$VCdim\{AGR\} \leq 42dk \log_2(3k) \leq 42dn \log_2(3n).$$

If $k = 0$ then the entire sample is positive and the region of agreement is an hyper-rectangle. Therefore, $VCdim\{AGR\} = 2d$. If $k = n$ then the entire sample is negative and the region of agreement is the points of the samples themselves. Hence, $VCdim\{AGR\} = n$. Overall we get that in all cases,

$$VCdim\{AGR\} \leq 42dn \log_2(3n) = O(dn \log n).$$

■

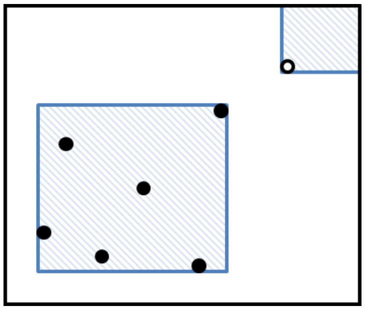


Figure 1: Agreement region of $VS_{\mathcal{H},R_i}$.

Corollary 30 (Balanced Axis-Aligned Rectangles) *Under the same conditions of Lemma 27, the class of balanced axis-aligned rectangles in $\mathbb{R}^d$ can be perfectly selectively learned with fast coverage rate.*

Proof Applying Lemmas 26 and 27 we get that with probability of at least $1-\delta$,

$$\hat{n} \leq O\left(\frac{d^2}{\lambda}\log\frac{dm}{\delta}\right).$$

Any balanced axis-aligned rectangle belongs to the class of all axis-aligned rectangles. Therefore, the coverage of CSS for the class of balanced axis-aligned rectangles is bounded bellow by the coverage of the class of axis-aligned rectangles. Applying Lemma 29, and assuming $m \geq d$, we obtain,

$$\gamma(\mathcal{H},\hat{n}) \quad \leq \quad O\left(d\frac{d^2}{\lambda}\log\frac{dm}{\delta}\log\left(\frac{d^2}{\lambda}\log\frac{dm}{\delta}\right)\right) \leq O\left(\frac{d^3}{\lambda}\log^2\frac{dm}{\lambda\delta}\right).$$

Applying Theorem 15 completes the proof. ■

6.2 Disagreement Coefficient

In this section we show interesting relations between the disagreement coefficient and coverage bounds in perfect selective classification. We begin by defining, for an hypothesis $h \in \mathcal{H}$, the set of all hypotheses that are r-close to h.

Definition 31 (Hanneke, 2011b, p.337) *For any hypothesis $h \in \mathcal{H}$, distribution P over $\mathcal{X}$, and $r > 0$, define the set $B(h,r)$ of all hypotheses that reside in a ball of radius r around h,*

$$B(h,r) \triangleq \left\{h' \in \mathcal{H} : \Pr_{X\sim P}\left\{h'(X) \neq h(X)\right\} \leq r\right\}.$$

Theorem 32 (Vapnik and Chervonenkis, 1971; Anthony and Bartlett, 1999, p.53) *Let $\mathcal{H}$ be a hypothesis class with VC-dimension d. For any probability distribution P on $\mathcal{X} \times \{\pm 1\}$, with probability of at least $1-\delta$ over the choice of S_m, any hypothesis $h \in \mathcal{H}$ consistent with S_m satisfies*

$$R(h) \leq \eta(d,m,\delta) \triangleq \frac{2}{m}\left[d\ln\frac{2em}{d} + \ln\frac{2}{\delta}\right].$$

For any $G \subseteq \mathcal{H}$ and distribution P we denote by ΔG the volume of the disagreement region of G,

$$\Delta G \triangleq \Pr\{DIS(G)\}.$$

Definition 33 (Disagreement coefficient, Hanneke, 2009) *Let $\varepsilon \geq 0$. The disagreement coefficient of the hypothesis class $\mathcal{H}$ with respect to the target distribution P is*

$$\theta(\varepsilon) \triangleq \theta_{h^*}(\varepsilon) = \sup_{r>\varepsilon} \frac{\Delta B(h^*, r)}{r}.$$

The following theorem formulates an intimate relation between active learning (disagreement coefficient) and selective classification.

Theorem 34 *Let $\mathcal{H}$ be an hypothesis class with VC-dimension d, P an unknown distribution, $\varepsilon \geq 0$, and $\theta(\varepsilon)$, the corresponding disagreement coefficient. Let (h,g) be a perfect selective classifier (CSS, see Section 2.3). Then, $R(h,g) = 0$, and for any $0 \leq \delta \leq 1$, with probability of at least $1-\delta$,*

$$\Phi(h,g) \geq 1 - \theta(\varepsilon) \cdot \max\{\eta(d,m,\delta), \varepsilon\}.$$

Proof Clearly, $R(h,g) = 0$, and it remains to prove the coverage bound. By Theorem 32, with probability of at least $1-\delta$,

$$\forall h \in VS_{\mathcal{H},S_m} \quad R(h) \leq \eta(d,m,\delta) \leq \max\{\eta(d,m,\delta), \varepsilon\}.$$

Therefore,

$$VS_{\mathcal{H},S_m} \subseteq B\left(h^*, \max\{\eta(d,m,\delta), \varepsilon\}\right),$$
$$\Delta VS_{\mathcal{H},S_m} \leq \Delta B\left(h^*, \max\{\eta(d,m,\delta), \varepsilon\}\right).$$

By Definition 33, for any $r' > \varepsilon$,

$$\Delta B(h^*, r') \leq \theta(\varepsilon) r'.$$

Thus, the proof is complete by recalling that

$$\Phi(h,g) = 1 - \Delta VS_{\mathcal{H},S_m}.$$

■

Theorem 34 tells us that whenever our learning problem (specified by the pair $(\mathcal{H},P)$) has a disagreement coefficient that grows slowly with respect to $1/\varepsilon$, it can be (perfectly) selectively learned with a "fast" coverage bound. Consequently, through Theorem 9 we also know that in each case where there exists a disagreement coefficient that grows slowly with respect to $1/\varepsilon$, active learning with a fast rate can also be deduced directly through a reduction from perfect selective classification. It follows that as far as fast rates in active learning are concerned, whatever can be accomplished by bounding the disagreement coefficient, can be accomplished also using perfect selective classification. This result is summarized in the following corollary.

Corollary 35 *Let $\mathcal{H}$ be an hypothesis class with VC-dimension d, P an unknown distribution, and $\theta(\varepsilon)$, the corresponding disagreement coefficient. If $\theta(\varepsilon) = O(polylog(1/\varepsilon))$, there exists a coverage bound such that an application of Theorem 7 ensures that $(\mathcal{H},P)$ is actively learnable with exponential label complexity speedup.*

Proof The proof is established by straightforward applications of Theorems 34 with $\varepsilon = 1/m$ and 9. ■

The following result, due to Hanneke (2011a), implies a coverage upper bound for CSS.

Lemma 36 (Hanneke, 2011a, Proof of Lemma 47) *Let $\mathcal{H}$ be an hypothesis class, P an unknown distribution, and $r \in (0,1)$. Then,*

$$\mathbf{E}_P \Delta D_m \geq (1-r)^m \Delta B(h^*, r),$$

where

$$D_m \triangleq VS_{\mathcal{H},S_m} \cap B(h^*, r). \tag{3}$$

Theorem 37 (Coverage upper bound) *Let $\mathcal{H}$ be an hypothesis class, P an unknown distribution, and $\delta \in (0,1)$. Then, for any $r \in (0,1)$, $1 > \alpha > \delta$,*

$$B_\Phi(\mathcal{H}, \delta, m) \leq 1 - \frac{(1-r)^m - \alpha}{1-\alpha} \Delta B(h^*, r),$$

where $B_\Phi(\mathcal{H}, \delta, m)$ is any coverage bound.

Proof Recalling the definition of D_m (3), clearly $D_m \subseteq VS_{\mathcal{H},S_m}$ and $D_m \subseteq B(h^*, r)$. These inclusions imply (respectively), by the definition of disagreement set,

$$\Delta D_m \leq \Delta VS_{\mathcal{H},S_m}, \text{ and } \Delta D_m \leq \Delta B(h^*, r). \tag{4}$$

Using Markov's inequality (in inequality (5) of the following derivation) and applying (4) (in equality (6)), we thus have,

$$\begin{aligned}
& Pr\left\{\Delta VS_{\mathcal{H},S_m} \leq \frac{(1-r)^m - \alpha}{1-\alpha} \Delta B(h^*, r)\right\} \leq Pr\left\{\Delta D_m \leq \frac{(1-r)^m - \alpha}{1-\alpha} \Delta B(h^*, r)\right\} \\
= \;& Pr\left\{\Delta B(h^*, r) - \Delta D_m \geq \frac{1-(1-r)^m}{1-\alpha} \Delta B(h^*, r)\right\} \\
\leq \;& Pr\left\{|\Delta B(h^*, r) - \Delta D_m| \geq \frac{1-(1-r)^m}{1-\alpha} \Delta B(h^*, r)\right\} \\
\leq \;& (1-\alpha) \cdot \frac{\mathbf{E}\{|\Delta B(h^*, r) - \Delta D_m|\}}{(1-(1-r)^m)\, \Delta B(h^*, r)} && (5) \\
= \;& (1-\alpha) \cdot \frac{\Delta B(h^*, r) - \mathbf{E} \Delta D_m}{(1-(1-r)^m)\, \Delta B(h^*, r)}. && (6)
\end{aligned}$$

Applying Lemma 36 we therefore obtain,

$$\leq \quad (1-\alpha) \cdot \frac{\Delta B(h^*, r) - (1-r)^m \Delta B(h^*, r)}{(1-(1-r)^m)\, \Delta B(h^*, r)} = 1 - \alpha < 1 - \delta.$$

Observing that for any coverage bound,

$$Pr\left\{\Delta VS_{\mathcal{H},S_m} \leq 1 - B_\Phi(\mathcal{H}, \delta, m)\right\} \geq 1 - \delta,$$

completes the proof. ■

Corollary 38 *Let $\mathcal{H}$ be an hypothesis class, P an unknown distribution, and $\delta \in (0, 1/8)$. Then for any $m \geq 2$,*

$$B_\Phi(\mathcal{H}, \delta, m) \leq 1 - \frac{1}{7}\Delta B\left(h^*, \frac{1}{m}\right),$$

where $B_\Phi(\mathcal{H}, \delta, m)$ is any coverage bound.

Proof The proof is established by a straightforward application of Theorem 37 with $\alpha = 1/8$ and $r = 1/m$. ■

With Corollary 38 we can bound the disagreement coefficient for settings whose coverage bound is known.

Corollary 39 *Let $\mathcal{H}$ be an hypothesis class, P an unknown distribution, and $B_\Phi(\mathcal{H}, \delta, m)$ a coverage bound. Then the disagreement coefficient is bounded by,*

$$\theta(\varepsilon) \leq \max\left\{ \sup_{r \in (\varepsilon, 1/2)} 7 \cdot \frac{1 - B_\Phi(\mathcal{H}, 1/9, \lfloor 1/r \rfloor)}{r}, 2 \right\}.$$

Proof Applying Corollary 38 we get that for any $r \in (0, 1/2)$,

$$\frac{\Delta B(h^*, r)}{r} \leq \frac{\Delta B(h^*, 1/\lfloor 1/r \rfloor)}{r} \leq 7 \cdot \frac{1 - B_\Phi(\mathcal{H}, 1/9, \lfloor 1/r \rfloor)}{r}.$$

Therefore,

$$\theta(\varepsilon) = \sup_{r > \varepsilon} \frac{\Delta B(h^*, r)}{r} \leq \max\left\{ \sup_{r \in (\varepsilon, 1/2)} 7 \cdot \frac{1 - B_\Phi(\mathcal{H}, 1/9, \lfloor 1/r \rfloor)}{r}, 2 \right\}.$$

■

Corollary 40 *Let $\mathcal{H}$ be the class of all linear binary classifiers in $\mathbb{R}^d$, and let the underlying distribution be any mixture of a fixed number of Gaussians in $\mathbb{R}^d$. Then*

$$\theta(\varepsilon) \leq O\left(polylog\left(\frac{1}{\varepsilon}\right)\right).$$

Proof Applying Corollary 39 together with inequality 2 we get that

$$\begin{aligned} \theta(\varepsilon) &\leq \max\left\{ \sup_{r \in (\varepsilon, 1/2)} 7 \cdot \frac{1 - B_\Phi(\mathcal{H}, 1/9, \lfloor 1/r \rfloor)}{r}, 2 \right\} \\ &\leq \max\left\{ \sup_{r \in (\varepsilon, 1/2)} \frac{7}{r} \cdot O\left(\frac{(\log \lfloor 1/r \rfloor)^{d^2}}{\lfloor 1/r \rfloor} \cdot 9^{\frac{d+3}{2}} \right), 2 \right\} \leq O\left(\left(log \frac{1}{\varepsilon} \right)^{d^2} \right). \end{aligned}$$

■

7. Concluding Remarks

For quite a few years, since its inception, the theory of target-independent bounds for noise-free active learning managed to handle relatively simple settings, mostly revolving around homogeneous linear classifiers under the uniform distribution over the sphere. It is likely that this distributional uniformity assumption was often adapted to simplify analyses. However, it was shown by Dasgupta (2005) that under this distribution, exponential speed up cannot be achieved when considering general (non homogeneous) linear classifiers.

The reason for this behavior is related to the two tasks that a good active learner should successfully accomplish: *exploration* and *exploitation*. Intuitively (and oversimplifying things) exploration is the task of obtaining at least one sample in each class, and exploitation is the process of refining the decision boundary by requesting labels of points around the boundary. Dasgupta showed that exploration cannot be achieved fast enough under the uniform distribution on the sphere. The source of this difficulty is the fact that under this distribution all training points reside on their convex hull. In general, the speed of exploration (using linear classifiers) depends on the size (number of vertices) of the convex hull of the training set. When using homogeneous linear classifiers, exploration is trivially achieved (under the uniform distribution) and exploitation can achieve exponential speedup.

So why in the *non-verifiable* model (Balcan et al., 2008) it is possible to achieve exponential speedup even when using non homogeneous linear classifiers under the uniform distribution? The answer is that in the non-verifiable model, label complexity attributed to exploration is encapsulated in a target-dependent "constant." Specifically, in Balcan et al. (2008) this constant is explicitly defined to be the probability mass of the minority class. Indeed, in certain noise free settings using linear classifiers, where the minority class is large enough, exploration is a non issue. In general, however, exploration is a major bottleneck in practical active learning (Baram et al., 2004; Begleiter et al., 2008). The present results show how exponential speedup can be achieved, including exploration, when using different (and perhaps more natural) distributions.

With these good news, a somewhat pessimistic picture arises from the lower bound we obtained for the exponential dependency on the dimension d. This negative result is not restricted to stream-based active learning and readily applies also to the pool-based model. While the bound is only asymptotic, we conjecture that it also holds for finite samples. Moreover, we believe that within the stream- or pool-based settings a similar statement should hold true for any active learning method (and not necessarily CAL-based querying strategies). This result indicates that when performing noise free active learning of linear classifiers, aggressive feature selection is beneficial for exploration speedup. We note, however, that it remains open whether a slowdown exponent of d (rather than d^2) is achievable.

We have exposed interesting relations of the present technique to well known complexity measures for active learning, namely, the teaching dimension and the disagreement coefficient. These developments were facilitated by observations made by Hanneke on the teaching dimension and the disagreement coefficient. These relations gave rise to further observations on active learning, which are discussed in Section 6 and include exponential speedup for balanced axis-aligned rectangles. Finally, we note that the intimate relation between selective classification and the disagreement coefficient was recently exposed in another result for selective classification where the disagreement coefficient emerged as a dominating factor in a coverage bound for agnostic selective classification (El-Yaniv and Wiener, 2011).

Acknowledgments

We thank the anonymous reviewers for their good comments. This paper particularly benefited from insightful observations made by one of the reviewers, which are summarized in Section 6, including the proof of Theorem 37 and the link between our $\hat{n}$ and the extended teaching dimension (Lemmas 26 and 27).

Appendix A.

Lemma 41 *For any $m \geq 3$, $a \geq 1$, $b \geq 1$ we get*

$$\sum_{i=1}^{m} \left(\frac{\ln^{a}(bi)}{i} \right) < \frac{4}{a} \ln^{a+1}(b(m+1)).$$

Proof Setting $f(x) \triangleq \frac{\ln^{a}(bx)}{x}$, we have

$$\frac{df}{dx} = (a - \ln bx) \cdot \frac{\ln^{a-1}(bx)}{x^2}.$$

Therefore, f is monotonically increasing when $x < e^a/b$, monotonically decreasing function when $x \geq e^a/b$ and its attains its maximum at $x = e^a/b$. Consequently, for $i < e^a/b - 1$, or $i \geq e^a/b + 1$,

$$f(i) \leq \int_{x=i-1}^{i+1} f(x)dx.$$

For $e^a/b - 1 \leq i < e^a/b + 1$,

$$f(i) \leq f(e^a/b) = b\left(\frac{a}{e}\right)^{a} \leq a^a. \tag{7}$$

Therefore, if $m < e^a - 1$ we have,

$$\sum_{i=1}^{m} f(i) = \ln^{a}(b) + \sum_{i=2}^{m} f(i) < 2 \cdot \int_{x=1}^{m+1} f(x)dx \leq \frac{2}{a+1} \ln^{a+1}(b(m+1)).$$

Otherwise, $m \geq e^a/b$, in which case we overcome the change of slope by adding twice the (upper bound on the) maximal value (7),

$$\begin{aligned} \sum_{i=1}^{m} f(i) &< \frac{2}{a+1} \ln^{a+1}(b(m+1)) + 2a^a = \frac{2}{a+1} \ln^{a+1}(b(m+1)) + \frac{2}{a} a^{a+1} \\ &\leq \frac{2}{a+1} \ln^{a+1}(b(m+1)) + \frac{2}{a} \ln^{a+1} bm \leq \frac{4}{a} \ln^{a+1}(b(m+1)). \end{aligned}$$

■

Appendix B. Alternative Proof of Lemma 6 Using Super Martingales

Define $W_k \triangleq \sum_{i=1}^{k}(Z_i - b_i)$. We assume that with probability of at least $1-\delta/2$, $\Pr\{Z_i|Z_1,\ldots,Z_{i-1}\} \le b_i$, simultaneously for all i. Since Z_i is a binary random variable it is easy to see that (w.h.p.),

$$E_{Z_i}\{W_i|Z_1,\ldots,Z_{i-1}\} = \Pr\{Z_i|Z_1,\ldots,Z_{i-1}\} - b_i + W_{i-1} \le W_{i-1},$$

and the sequence $W_1^m \triangleq W_1,\ldots,W_m$ is a super-martingale with high probability. We apply the following theorem by McDiarmid that refers to martingales (but can be shown to apply to super-martingales, by following its original proof).

Theorem 42 (McDiarmid, 1998, Theorem 3.12) *Let $Y_1,\ldots,Y_n$ be a martingale difference sequence with $-a_k \le Y_k \le 1-a_k$ for each k; let $A = \frac{1}{n}\sum a_k$. Then, for any $\varepsilon > 0$,*

$$\Pr\left\{\sum Y_k \ge An\varepsilon\right\} \le \exp\left(-[(1+\varepsilon)\ln(1+\varepsilon)-\varepsilon]An\right) \le \exp\left(-\frac{An\varepsilon^2}{2(1+\varepsilon/3)}\right).$$

In our case, $Y_k = W_k - W_{k-1} = Z_k - b_k \le 1 - b_k$ and we apply the (revised) theorem with $a_k \triangleq b_k$ and $An \triangleq \sum b_k \triangleq B$. We thus obtain, for any $0 < \varepsilon < 1$,

$$\Pr\left\{\sum Z_k \ge B + B\varepsilon\right\} \le \exp\left(-\frac{B\varepsilon^2}{2(1+\varepsilon/3)}\right).$$

Equating the right-hand side to $\delta/2$, we obtain

$$\begin{aligned}
\varepsilon &= \left(\frac{2}{3}\ln\frac{2}{\delta} \pm \sqrt{\frac{4}{9}\ln^2\frac{2}{\delta} + 8B\ln\frac{2}{\delta}}\right)/2B \\
&\le \left(\frac{1}{3}\ln\frac{2}{\delta} + \sqrt{\frac{1}{9}\ln^2\frac{2}{\delta}} + \sqrt{2B\ln\frac{2}{\delta}}\right)/B \\
&= \left(\frac{2}{3}\ln\frac{2}{\delta} + \sqrt{2B\ln\frac{2}{\delta}}\right)/B.
\end{aligned}$$

Applying the union bound completes the proof.

References

M. Anthony and P.L. Bartlett. *Neural Network Learning; Theoretical Foundations*. Cambridge University Press, 1999.

L. Atlas, D. Cohn, R. Ladner, A.M. El-Sharkawi, and R.J. Marks. Training connectionist networks with queries and selective sampling. In *Neural Information Processing Systems (NIPS)*, pages 566–573, 1990.

M.F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *21st Annual Conference on Learning Theory (COLT)*, pages 45–56, 2008.

Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5:255–291, 2004.

P.L. Bartlett and M.H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

R. Begleiter, R. El-Yaniv, and D. Pechyony. Repairing self-confident active-transductive learners using systematic exploration. *Pattern Recognition Letters*, 29(9):1245–1251, 2008.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36, 1989.

C.K. Chow. An optimum character recognition system using decision function. *IEEE Transactions on Computers*, 6(4):247–254, 1957.

C.K. Chow. On optimum recognition error and reject trade-off. *IEEE Transactions on Information Theory*, 16:41–36, 1970.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18*, pages 235–242, 2005.

S. Dasgupta, A. Tauman Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

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. Agnostic selective classification. In *Neural Information Processing Systems (NIPS)*, 2011.

S. Fine, R. Gilad-Bachrach, and E. Shamir. Query by committee, linear separation and random walks. *Theoretical Computer Science*, 284(1):25–51, 2002.

Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Information, prediction, and Query by Committee. In *Advances in Neural Information Processing Systems (NIPS) 5*, pages 483–490, 1993.

Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

Y. Freund, Y. Mansour, and R.E. Schapire. Generalization bounds for averaged classifiers. *Annals of Statistics*, 32(4):1698–1722, 2004.

E. Friedman. Active learning for smooth problems. *In Proceedings of the* 22^{nd} *Annual Conference on Learning Theory (COLT)*, 2009.

R. Gilad-Bachrach. *To PAC and Beyond*. PhD thesis, the Hebrew University of Jerusalem, 2007.

S. Goldman and M. Kearns. On the complexity of teaching. *JCSS: Journal of Computer and System Sciences*, 50, 1995.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007a.

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT)*, volume 4539 of *Lecture Notes in Artificial Intelligence*, pages 66–81, 2007b.

S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Carnegie Mellon University, 2009.

S. Hanneke. Activized learning: Transforming passive to active with improved label complexity. *CoRR*, abs/1108.1766, 2011a. URL `http://arxiv.org/abs/1108.1766`. informal publication.

S. Hanneke. Rates of convergence in active learning. *Annals of Statistics*, 37(1):333–361, 2011b.

T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1995.

R. Herbei and M.H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics*, 34(4):709–721, 2006.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

D. Hug and M. Reitzner. Gaussian polytopes: variances and limit theorems, June 2005.

D. Hug, G. O. Munsonious, and M. Reitzner. Asymptotic mean values of Gaussian polytopes. *Beiträge Algebra Geom.*, 45:531–548, 2004.

C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16, pages 195–248. Springer-Verlag, 1998.

T. Mitchell. Version spaces: a candidate elimination approach to rule learning. In *IJCAI'77: Proceedings of the 5th international joint conference on Artificial Intelligence*, pages 305–310, 1977.

H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning theory (COLT)*, pages 287–294, 1992.

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.

M.H. Wegkamp. Lasso type classifiers with a reject option. *Electronic Journal of Statistics*, 1: 155–168, 2007.

Journal of Machine Learning Research 13 (2012) 281-305 Submitted 3/11; Revised 9/11; Published 2/12

Random Search for Hyper-Parameter Optimization

James Bergstra JAMES.BERGSTRA@UMONTREAL.CA
Yoshua Bengio YOSHUA.BENGIO@UMONTREAL.CA
Département d'Informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC, H3C 3J7, Canada

Editor: Leon Bottou

Abstract

Grid search and manual search are the most widely used strategies for hyper-parameter optimization. This paper shows empirically and theoretically that randomly chosen trials are more efficient for hyper-parameter optimization than trials on a grid. Empirical evidence comes from a comparison with a large previous study that used grid search and manual search to configure neural networks and deep belief networks. Compared with neural networks configured by a pure grid search, we find that random search over the same domain is able to find models that are as good or better within a small fraction of the computation time. Granting random search the same computational budget, random search finds better models by effectively searching a larger, less promising configuration space. Compared with deep belief networks configured by a thoughtful combination of manual search and grid search, purely random search over the same 32-dimensional configuration space found statistically equal performance on four of seven data sets, and superior performance on one of seven. A Gaussian process analysis of the function from hyper-parameters to validation set performance reveals that for most data sets only a few of the hyper-parameters really matter, but that different hyper-parameters are important on different data sets. This phenomenon makes grid search a poor choice for configuring algorithms for new data sets. Our analysis casts some light on why recent "High Throughput" methods achieve surprising success—they appear to search through a large number of hyper-parameters because most hyper-parameters do not matter much. We anticipate that growing interest in large hierarchical models will place an increasing burden on techniques for hyper-parameter optimization; this work shows that random search is a natural baseline against which to judge progress in the development of adaptive (sequential) hyper-parameter optimization algorithms.

Keywords: global optimization, model selection, neural networks, deep learning, response surface modeling

1. Introduction

The ultimate objective of a typical learning algorithm $\mathcal{A}$ is to find a function f that minimizes some expected loss $\mathcal{L}(x;f)$ over i.i.d. samples x from a natural (grand truth) distribution $\mathcal{G}_x$. A *learning algorithm* $\mathcal{A}$ is a functional that maps a data set $\mathcal{X}^{(\text{train})}$ (a finite set of samples from $\mathcal{G}_x$) to a function f. Very often a learning algorithm produces f through the optimization of a training criterion with respect to a set of *parameters* θ. However, the learning algorithm itself often has bells and whistles called *hyper-parameters* λ, and the actual learning algorithm is the one obtained after choosing λ, which can be denoted $\mathcal{A}_\lambda$, and $f = \mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})$ for a training set $\mathcal{X}^{(\text{train})}$. For example, with a

Gaussian kernel SVM, one has to select a regularization penalty C for the training criterion (which controls the margin) and the bandwidth σ of the Gaussian kernel, that is, $\lambda = (C, \sigma)$.

What we really need in practice is a way to choose λ so as to minimize generalization error $\mathbb{E}_{x\sim\mathcal{G}_x}[\mathcal{L}(x;\mathcal{A}_\lambda(\mathcal{X}^{(\text{train})}))]$. Note that the computation performed by $\mathcal{A}$ itself often involves an inner optimization problem, which is usually iterative and approximate. The problem of identifying a good value for hyper-parameters λ is called the problem of *hyper-parameter optimization*. This paper takes a look at algorithms for this difficult outer-loop optimization problem, which is of great practical importance in empirical machine learning work:

$$\lambda^{(*)} = \operatorname*{argmin}_{\lambda\in\Lambda} \mathbb{E}_{x\sim\mathcal{G}_x}[\mathcal{L}\left(x;\mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right)]. \tag{1}$$

In general, we do not have efficient algorithms for performing the optimization implied by Equation 1. Furthermore, we cannot even evaluate the expectation over the unknown natural distribution $\mathcal{G}_x$, the value we wish to optimize. Nevertheless, we must carry out this optimization as best we can. With regards to the expectation over $\mathcal{G}_x$, we will employ the widely used technique of *cross-validation* to estimate it. Cross-validation is the technique of replacing the expectation with a mean over a *validation set* $\mathcal{X}^{(\text{valid})}$ whose elements are drawn i.i.d $x \sim \mathcal{G}_x$. Cross-validation is unbiased as long as $\mathcal{X}^{(\text{valid})}$ is independent of any data used by $\mathcal{A}_\lambda$ (see Bishop, 1995, pp. 32-33). We see in Equations 2-4 the hyper-parameter optimization problem as it is addressed in practice:

$$\begin{aligned}
\lambda^{(*)} &\approx \operatorname*{argmin}_{\lambda\in\Lambda} \operatorname*{mean}_{x\in\mathcal{X}^{(\text{valid})}} \mathcal{L}\left(x;\mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right). && (2)\\
&\equiv \operatorname*{argmin}_{\lambda\in\Lambda} \Psi(\lambda) && (3)\\
&\approx \operatorname*{argmin}_{\lambda\in\{\lambda^{(1)}\ldots\lambda^{(S)}\}} \Psi(\lambda) \equiv \hat{\lambda} && (4)
\end{aligned}$$

Equation 3 expresses the hyper-parameter optimization problem in terms of a *hyper-parameter response function*, Ψ. Hyper-parameter optimization is the minimization of $\Psi(\lambda)$ over $\lambda \in \Lambda$. This function is sometimes called the *response surface* in the experiment design literature. Different data sets, tasks, and learning algorithm families give rise to different sets Λ and functions Ψ. Knowing in general very little about the response surface Ψ or the search space Λ, the dominant strategy for finding a good λ is to choose some number (S) of *trial* points $\{\lambda^{(1)}\ldots\lambda^{(S)}\}$, to evaluate $\Psi(\lambda)$ for each one, and return the $\lambda^{(i)}$ that worked the best as $\hat{\lambda}$. This strategy is made explicit by Equation 4.

The critical step in hyper-parameter optimization is to choose the set of trials $\{\lambda^{(1)}\ldots\lambda^{(S)}\}$. The most widely used strategy is a combination of grid search and manual search (e.g., LeCun et al., 1998b; Larochelle et al., 2007; Hinton, 2010), as well as machine learning software packages such as libsvm (Chang and Lin, 2001) and scikits.learn.[1] If Λ is a set indexed by K configuration variables (e.g., for neural networks it would be the learning rate, the number of hidden units, the strength of weight regularization, etc.), then grid search requires that we choose a set of values for each variable ($L^{(1)}\ldots L^{(K)}$). In grid search the set of trials is formed by assembling every possible combination of values, so the number of trials in a grid search is $S = \prod_{k=1}^{K} |L^{(k)}|$ elements. This product over K sets makes grid search suffer from the *curse of dimensionality* because the number of joint values grows exponentially with the number of hyper-parameters (Bellman, 1961). Manual

[1] `scikits.learn`: Machine Learning in Python can be found at `http://scikit-learn.sourceforge.net`.

search is used to identify regions in Λ that are promising and to develop the intuition necessary to choose the sets $L^{(k)}$. A major drawback of manual search is the difficulty in *reproducing results*. This is important both for the progress of scientific research in machine learning as well as for ease of application of learning algorithms by non-expert users. On the other hand, grid search alone does very poorly in practice (as discussed here). We propose random search as a substitute and baseline that is both reasonably efficient (roughly equivalent to or better than combinining manual search and grid search, in our experiments) and keeping the advantages of implementation simplicity and reproducibility of pure grid search. Random search is actually more practical than grid search because it can be applied even when using a cluster of computers that can fail, and allows the experimenter to change the "resolution" on the fly: adding new trials to the set or ignoring failed trials are both feasible because the trials are i.i.d., which is not the case for a grid search. Of course, random search can probably be improved by automating what manual search does, i.e., a sequential optimization, but this is left to future work.

There are several reasons why manual search and grid search prevail as the state of the art despite decades of research into global optimization (e.g., Nelder and Mead, 1965; Kirkpatrick et al., 1983; Powell, 1994; Weise, 2009) and the publishing of several hyper-parameter optimization algorithms (e.g., Nareyek, 2003; Czogiel et al., 2005; Hutter, 2009):

- Manual optimization gives researchers some degree of insight into Ψ;
- There is no technical overhead or barrier to manual optimization;
- Grid search is simple to implement and parallelization is trivial;
- Grid search (with access to a compute cluster) typically finds a better $\hat{\lambda}$ than purely manual sequential optimization (in the same amount of time);
- Grid search is reliable in low dimensional spaces (e.g., 1-d, 2-d).

We will come back to the use of global optimization algorithms for hyper-parameter selection in our discussion of future work (Section 6). In this paper, we focus on random search, that is, independent draws from a uniform density from the same configuration space as would be spanned by a regular grid, as an alternative strategy for producing a trial set $\{\lambda^{(1)}...\lambda^{(S)}\}$. We show that random search has all the practical advantages of grid search (conceptual simplicity, ease of implementation, trivial parallelism) and trades a small reduction in efficiency in low-dimensional spaces for a large improvement in efficiency in high-dimensional search spaces.

In this work we show that random search is more efficient than grid search in high-dimensional spaces because functions Ψ of interest have a *low effective dimensionality*; essentially, Ψ of interest are more sensitive to changes in some dimensions than others (Caflisch et al., 1997). In particular, if a function f of two variables could be approximated by another function of one variable ($f(x_1,x_2) \approx g(x_1)$), we could say that f has a *low effective dimension*. Figure 1 illustrates how point grids and uniformly random point sets differ in how they cope with low effective dimensionality, as in the above example with f. A grid of points gives even coverage in the original 2-d space, but projections onto either the x_1 or x_2 subspace produces an inefficient coverage of the subspace. In contrast, random points are slightly less evenly distributed in the original space, but far more evenly distributed in the subspaces.

If the researcher could know ahead of time which subspaces would be important, then he or she could design an appropriate grid. However, we show the failings of this strategy in Section 2. For a

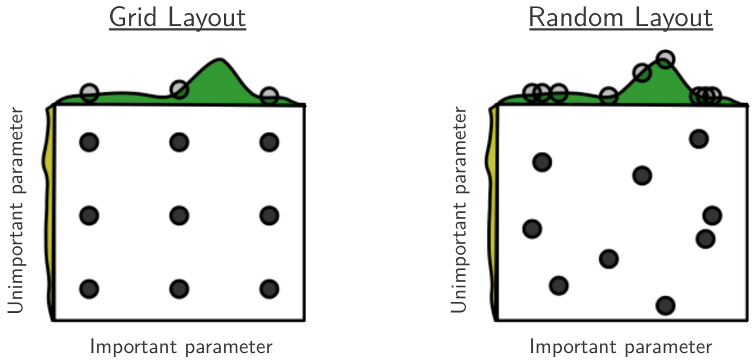


Figure 1: Grid and random search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g. This failure of grid search is the rule rather than the exception in high dimensional hyper-parameter optimization.

given learning algorithm, looking at several relatively similar data sets (from different distributions) reveals that on different data sets, different subspaces are important, and to different degrees. A grid with sufficient granularity to optimizing hyper-parameters for all data sets must consequently be inefficient for each individual data set because of the curse of dimensionality: the number of wasted grid search trials is exponential in the number of search dimensions that turn out to be irrelevant for a particular data set. In contrast, random search thrives on low effective dimensionality. Random search has the same efficiency in the relevant subspace as if it had been used to search only the relevant dimensions.

This paper is organized as follows. Section 2 looks at the efficiency of random search in practice vs. grid search as a method for optimizing neural network hyper-parameters. We take the grid search experiments of Larochelle et al. (2007) as a point of comparison, and repeat similar experiments using random search. Section 3 uses Gaussian process regression (GPR) to analyze the results of the neural network trials. The GPR lets us characterize what Ψ looks like for various data sets, and establish an empirical link between the low effective dimensionality of Ψ and the efficiency of random search. Section 4 compares random search and grid search with more sophisticated point sets developed for Quasi Monte-Carlo numerical integration, and argues that in the regime of interest for hyper-parameter selection grid search is inappropriate and more sophisticated methods bring little advantage over random search. Section 5 compares random search with the expert-guided manual sequential optimization employed in Larochelle et al. (2007) to optimize Deep Belief Networks. Section 6 comments on the role of global optimization algorithms in future work. We conclude in Section 7 that random search is generally superior to grid search for optimizing hyper-parameters.

2. Random vs. Grid for Optimizing Neural Networks

In this section we take a second look at several of the experiments of Larochelle et al. (2007) using random search, to compare with the grid searches done in that work. We begin with a look at hyper-parameter optimization in neural networks, and then move on to hyper-parameter optimization in Deep Belief Networks (DBNs). To characterize the efficiency of random search, we present two techniques in preliminary sections: Section 2.1 explains how we estimate the generalization performance of the *best* model from a set of candidates, taking into account our uncertainty in which model is actually best; Section 2.2 explains the random experiment efficiency curve that we use to characterize the performance of random search experiments. With these preliminaries out of the way, Section 2.3 describes the data sets from Larochelle et al. (2007) that we use in our work. Section 2.4 presents our results optimizing neural networks, and Section 5 presents our results optimizing DBNs.

2.1 Estimating Generalization

Because of finite data sets, test error is not monotone in validation error, and depending on the set of particular hyper-parameter values λ evaluated, the test error of the best-validation error configuration may vary. When reporting performance of learning algorithms, it can be useful to take into account the uncertainty due to the choice of hyper-parameters values. This section describes our procedure for estimating test set accuracy, which takes into account any uncertainty in the choice of which trial is actually the best-performing one. To explain this procedure, we must distinguish between estimates of performance $\Psi^{(\text{valid})} = \Psi$ and $\Psi^{(\text{test})}$ based on the validation and test sets respectively:

$$\Psi^{(\text{valid})}(\lambda) = \text{mean}_{x \in \mathcal{X}^{(\text{valid})}} \; \mathcal{L}\left(x; \mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right),$$
$$\Psi^{(\text{test})}(\lambda) = \text{mean}_{x \in \mathcal{X}^{(\text{test})}} \; \mathcal{L}\left(x; \mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right).$$

Likewise, we must define the estimated variance $\mathbb{V}$ about these means on the validation and test sets, for example, for the zero-one loss (Bernoulli variance):

$$\mathbb{V}^{(\text{valid})}(\lambda) = \frac{\Psi^{(\text{valid})}(\lambda)\left(1 - \Psi^{(\text{valid})}(\lambda)\right)}{|\mathcal{X}^{(\text{valid})}| - 1}, \text{ and}$$
$$\mathbb{V}^{(\text{test})}(\lambda) = \frac{\Psi^{(\text{test})}(\lambda)\left(1 - \Psi^{(\text{test})}(\lambda)\right)}{|\mathcal{X}^{(\text{test})}| - 1}.$$

With other loss functions the estimator of variance will generally be different.

The standard practice for evaluating a model found by cross-validation is to report $\Psi^{(\text{test})}(\lambda^{(s)})$ for the $\lambda^{(s)}$ that minimizes $\Psi^{(\text{valid})}(\lambda^{(s)})$. However, when different trials have nearly optimal validation means, then it is not clear which test score to report, and a slightly different choice of λ could have yielded a different test error. To resolve the difficulty of choosing a winner, we report a weighted average of all the test set scores, in which each one is weighted by the probability that its particular $\lambda^{(s)}$ is in fact the best. In this view, the uncertainty arising from $\mathcal{X}^{(\text{valid})}$ being a finite sample of $\mathcal{G}_x$ makes the test-set score of the best model among $\lambda^{(1)}, ..., \lambda^{(S)}$ a random variable, z. This score z is modeled by a Gaussian mixture model whose S components have means $\mu_s = \Psi^{(\text{test})}(\lambda^{(s)})$, variances $\sigma_s^2 = \mathbb{V}^{(\text{test})}(\lambda^{(s)})$, and weights w_s defined by

$$w_s = P\left(Z^{(s)} < Z^{(s')},\ \ \forall s' \neq s\right),\ \text{where}$$
$$Z^{(i)} \sim \mathcal{N}\left(\Psi^{(\text{valid})}(\lambda^{(i)}), \mathbb{V}^{(\text{valid})}(\lambda^{(i)})\right).$$

To summarize, the performance z of the best model in an experiment of S trials has mean μ_z and standard error σ_z^2,

$$\mu_z = \sum_{s=1}^{S} w_s \mu_s,\ \text{and} \tag{5}$$

$$\sigma_z^2 = \sum_{s=1}^{S} w_s \left(\mu_s^2 + \sigma_s^2\right) - \mu_z^2. \tag{6}$$

It is simple and practical to estimate weights w_s by simulation. The procedure for doing so is to repeatedly draw hypothetical validation scores $Z^{(s)}$ from Normal distributions whose means are the $\Psi^{(\text{valid})}(\lambda^{(s)})$ and whose variances are the squared standard errors $\mathbb{V}^{(\text{valid})}(\lambda^{(s)})$, and to count how often each trial generates a winning score. Since the test scores of the best validation scores are typically relatively close, w_s need not be estimated very precisely and a few tens of hypothetical draws suffice.

In expectation, this technique for estimating generalization gives a higher estimate than the traditional technique of reporting the test set error of the best model in validation. The difference is related to the variance $\Psi^{(\text{valid})}$ and the density of validation set scores $\Psi(\lambda^{(i)})$ near the best value. To the extent that $\Psi^{(\text{valid})}$ casts doubt on which model was best, this technique averages the performance of the best model together with the performance of models which were not the best. The next section (Random Experiment Efficieny Curve) illustrates this phenomenon and discusses it in more detail.

2.2 Random Experiment Efficiency Curve

Figure 2 illustrates the results of a random experiment: an experiment of 256 trials training neural networks to classify the rectangles data set. Since the trials of a random experiment are independently identically distributed (i.i.d.), a random search experiment involving S i.i.d. trials can also be interpreted as N independent experiments of s trials, as long as $sN \leq S$. This interpretation allows us to estimate statistics such as the minimum, maximum, median, and quantiles of any random experiment of size s, where s is a divisor of S.

There are two general trends in random experiment efficiency curves, such as the one in Figure 2: a sharp upward slope of the lower extremes as experiments grow, and a gentle downward slope of the upper extremes. The sharp upward slope occurs because when we take the maximum over larger subsets of the S trials, trials with poor performance are rarely the best within their subset. It is natural that larger experiments find trials with better scores. The shape of this curve indicates the frequency of good models under random search, and quantifies the relative volumes (in search space) of the various levels of performance.

The gentle downward slope occurs because as we take the maximum over larger subsets of trials (in Equation 6), we are less sure about which trial is actually the best. Large experiments average together good validation trials with unusually high test scores with other good validation trials with unusually low test scores to arrive at a more accurate estimate of generalization. For example,

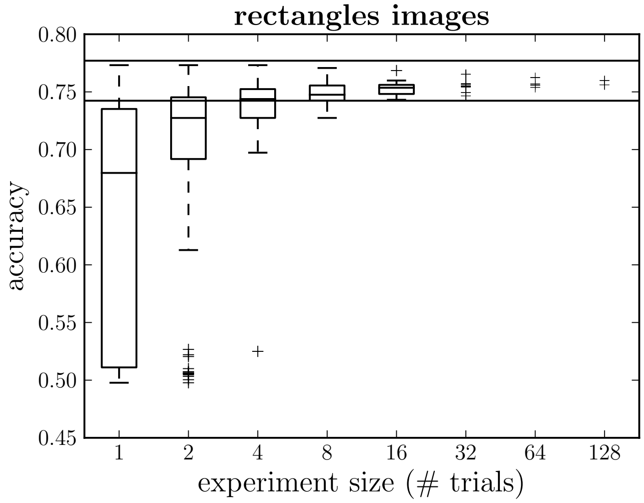


Figure 2: A random experiment efficiency curve. The trials of a random experiment are i.i.d, so an experiment of many trials (here, 256 trials optimizing a neural network to classify the **rectangles basic** data set, Section 2.3) can be interpreted as several independent smaller experiments. For example, at horizontal axis position 8, we consider our 256 trials to be 32 experiments of 8 trials each. The vertical axis shows the test accuracy of the best trial(s) from experiments of a given size, as determined by Equation 5. When there are sufficiently many experiments of a given size (i.e., 10), the distribution of performance is illustrated by a box plot whose boxed section spans the lower and upper quartiles and includes a line at the median. The whiskers above and below each boxed section show the position of the most extreme data point within 1.5 times the inter-quartile range of the nearest quartile. Data points beyond the whiskers are plotted with '+' symbols. When there are not enough experiments to support a box plot, as occurs here for experiments of 32 trials or more, the best generalization score of each experiment is shown by a scatter plot. The two thin black lines across the top of the figure mark the upper and lower boundaries of a 95% confidence interval on the generalization of the best trial overall (Equation 6).

consider what Figure 2 would look like if the experiment had included *lucky trial* whose validation score were around 77% as usual, but whose test score were 80%. In the bar plot for trials of size 1, we would see the top performer scoring 80%. In larger experiments, we would average that 80% performance together with other test set performances because 77% is not clearly the best validation score; this averaging would make the upper envelope of the efficiency curve slope downward from 80% to a point very close to the current test set estimate of 76%.

Figure 2 characterizes the range of performance that is to be expected from experiments of various sizes, which is valuable information to anyone trying to reproduce these results. For example, if we try to repeat the experiment and our first four random trials fail to find a score better than 70%, then the problem is likely not in hyper-parameter selection.

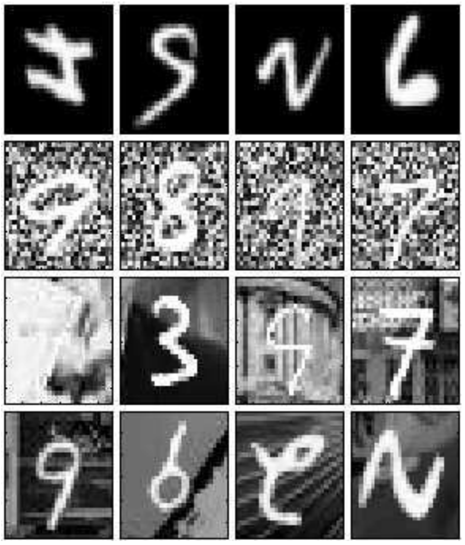


Figure 3: From top to bottom, samples from the **mnist rotated**, **mnist background random**, **mnist background images**, **mnist rotated background images** data sets. In all data sets the task is to identify the digit (0 - 9) and ignore the various distracting factors of variation.

2.3 Data Sets

Following the work of Larochelle et al. (2007) and Vincent et al. (2008), we use a variety of classification data sets that include many factors of variation.[2]

The **mnist basic** data set is a subset of the well-known MNIST handwritten digit data set (LeCun et al., 1998a). This data set has 28x28 pixel grey-scale images of digits, each belonging to one of ten classes. We chose a different train/test/validation splitting in order to have faster experiments and see learning performance differences more clearly. We shuffled the original splits randomly, and used 10 000 training examples, 2000 validation examples, and 50 000 testing examples. These images are presented as white (1.0-valued) foreground digits against a black (0.0-valued) background.

The **mnist background images** data set is a variation on **mnist basic** in which the white foreground digit has been composited on top of a 28x28 natural image patch. Technically this was done by taking the maximum of the original MNIST image and the patch. Natural image patches with very low pixel variance were rejected. As with **mnist basic** there are 10 classes, 10 000 training examples, 2000 validation examples, and 50 000 test examples.

The **mnist background random** data set is a similar variation on **mnist basic** in which the white foreground digit has been composited on top of random uniform (0,1) pixel values. As with **mnist basic** there are 10 classes, 10 000 training examples, 2000 validation examples, and 50 000 test examples.

The **mnist rotated** data set is a variation on **mnist basic** in which the images have been rotated by an amount chosen randomly between 0 and 2π radians. This data set included 10000 training examples, 2000 validation examples, 50 000 test examples.

[2] Data sets can be found at `http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/DeepVsShallowComparisonICML2007`.

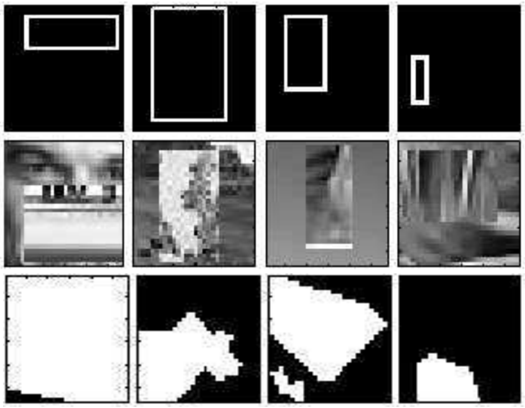


Figure 4: *Top:* Samples from the **rectangles** data set. *Middle:* Samples from the **rectangles images** data set. *Bottom:* Samples from the **convex** data set. In rectangles data sets, the image is formed by overlaying a small rectangle on a background. The task is to label the small rectangle as being either tall or wide. In **convex**, the task is to identify whether the set of white pixels is convex (images 1 and 4) or not convex (images 2 and 3).

The **mnist rotated background images** data set is a variation on **mnist rotated** in which the images have been rotated by an amount chosen randomly between 0 and 2π radians, and then subsequently composited onto natural image patch backgrounds. This data set included 10000 training examples, 2000 validation examples, 50 000 test examples.

The **rectangles** data set (Figure 4, top) is a simple synthetic data set of outlines of rectangles. The images are 28x28, the outlines are white (1-valued) and the backgrounds are black (0-valued). The height and width of the rectangles were sampled uniformly, but when their difference was smaller than 3 pixels the samples were rejected. The top left corner of the rectangles was also sampled uniformly, with the constraint that the whole rectangle fits in the image. Each image is labelled as one of two classes: tall or wide. This task was easier than the MNIST digit classification, so we only used 1000 training examples, and 200 validation examples, but we still used 50 000 testing examples.

The **rectangles images** data set (Figure 4, middle) is a variation on **rectangles** in which the foreground rectangles were filled with one natural image patch, and composited on top of a different background natural image patch. The process for sampling rectangle shapes was similar to the one used for **rectangles**, except a) the area covered by the rectangles was constrained to be between 25% and 75% of the total image, b) the length and width of the rectangles were forced to be of at least 10 pixels, and c) their difference was forced to be of at least 5 pixels. This task was harder than **rectangles**, so we used 10000 training examples, 2000 validation examples, and 50 000 testing examples.

The **convex** data set (Figure 4, bottom) is a binary image classification task. Each 28x28 image consists entirely of 1-valued and 0-valued pixels. If the 1-valued pixels form a convex region in image space, then the image is labelled as being convex, otherwise it is labelled as non-convex. The convex sets consist of a single convex region with pixels of value 1.0. Candidate convex images were constructed by taking the intersection of a number of half-planes whose location and orientation were chosen uniformly at random. The number of intersecting half-planes was also sampled randomly according to a geometric distribution with parameter 0.195. A candidate convex image was rejected if there were less than 19 pixels in the convex region. Candidate non-convex images were constructed by taking the union of a random number of convex sets generated as above, but with the number of half-planes sampled from a geometric distribution with parameter 0.07 and with a minimum number of 10 pixels. The number of convex sets was sampled uniformly from 2 to 4. The candidate non-convex images were then tested by checking a convexity condition for every pair of pixels in the non-convex set. Those sets that failed the convexity test were added to the data set. The parameters for generating the convex and non-convex sets were balanced to ensure that the conditional overall pixel mean is the same for both classes.

2.4 Case Study: Neural Networks

In Larochelle et al. (2007), the hyper-parameters of the neural network were optimized by search over a grid of trials. We describe the hyper-parameter configuration space of our neural network learning algorithm in terms of the distribution that we will use to randomly sample from that configuration space. The first hyper-parameter in our configuration is the type of data preprocessing: with equal probability, one of (a) none, (b) normalize (center each feature dimension and divide by its standard deviation), or (c) PCA (after removing dimension-wise means, examples are projected onto principle components of the data whose norms have been divided by their eigenvalues). Part of PCA preprocessing is choosing how many components to keep. We choose a fraction of variance to keep with a uniform distribution between 0.5 and 1.0. There have been several suggestions for how the random weights of a neural network should be initialized (we will look at unsupervised learning *pretraining* algorithms later in Section 5). We experimented with two distributions and two scaling heuristics. The possible distributions were (a) uniform on $(-1,1)$, and (b) unit normal. The two scaling heuristics were (a) a hyper-parameter multiplier between 0.1 and 10.0 divided by the square root of the number of inputs (LeCun et al., 1998b), and (b) the square root of 6 divided by the square root of the number of inputs plus hidden units (Bengio and Glorot, 2010). The weights themselves were chosen using one of three random seeds to the Mersenne Twister pseudo-random number generator. In the case of the first heuristic, we chose a multiplier uniformly from the range $(0.2, 2.0)$. The number of hidden units was drawn geometrically[3] from 18 to 1024. We selected either a sigmoidal or tanh nonlinearity with equal probability. The output weights from hidden units to prediction units were initialized to zero. The cost function was the mean error over minibatches of either 20 or 100 (with equal probability) examples at a time: in expectation these give the same gradient directions, but with more or less variance. The optimization algorithm was stochastic gradient descent with [initial] learning rate ε_0 drawn geometrically from 0.001 to 10.0. We offered the possibility of an annealed learning rate via a time point t_0 drawn geometrically from 300 to 30000. The effective learning rate ε_t after t minibatch iterations was

$$\varepsilon_t = \frac{t_0 \varepsilon_0}{\max(t, t_0)}. \tag{7}$$

We permitted a minimum of 100 and a maximum of 1000 iterations over the training data, stopping if ever, at iteration t, the best validation performance was observed before iteration $t/2$. With 50%

[3] We will use the phrase *drawn geometrically* from A to B for $0 < A < B$ to mean drawing uniformly in the log domain between $\log(A)$ and $\log(B)$, exponentiating to get a number between A and B, and then rounding to the nearest integer. The phrase *drawn exponentially* means the same thing but without rounding.

probability, an ℓ_2 regularization penalty was applied, whose strength was drawn exponentially from 3.1×10^{-7} to 3.1×10^{-5}. This sampling process covers roughly the same domain with the same density as the grid used in Larochelle et al. (2007), except for the optional preprocessing steps. The grid optimization of Larochelle et al. (2007) did not consider normalizing or keeping only leading PCA dimensions of the inputs; we compare to random sampling with and without these restrictions.[4]

We formed experiments for each data set by drawing $S = 256$ trials from this distribution. The results of these experiments are illustrated in Figures 5 and 6. Random sampling of trials is surprisingly effective in these settings. Figure 5 shows that even among the fraction of jobs ($71/256$) that used no preprocessing, the random search with 8 trials is better than the grid search employed in Larochelle et al. (2007).

Typically, the extent of a grid search is determined by a computational budget. Figure 6 shows what is possible if we use random search in a larger space that requires more trials to explore. The larger search space includes the possibility of normalizing the input or applying PCA preprocessing. In the larger space, 32 trials were necessary to consistently outperform grid search rather than 8, indicating that there are many harmful ways to preprocess the data. However, when we allowed larger experiments of 64 trials or more, random search found superior results to those found more quickly within the more restricted search. This tradeoff between exploration and exploitation is central to the design of an effective random search.

The efficiency curves in Figures 5 and 6 reveal that different data sets give rise to functions Ψ with different shapes. The **mnist basic** results converge very rapidly toward what appears to be a global maximum. The fact that experiments of just 4 or 8 trials often have the same maximum as much larger experiments indicates that the region of Λ that gives rise to the best performance is approximately a quarter or an eighth respectively of the entire configuration space. Assuming that the random search has not missed a tiny region of significantly better performance, we can say that random search has solved this problem in 4 or 8 guesses. It is hard to imagine any optimization algorithm doing much better on a non-trivial 7-dimensional function. In contrast the **mnist rotated background images** and **convex** curves show that even with 16 or 32 random trials, there is considerable variation in the generalization of the reportedly best model. This indicates that the Ψ function in these cases is more peaked, with small regions of good performance.

3. The Low Effective Dimension of Ψ

Section 2 showed that random sampling is more efficient than grid sampling for optimizing functions Ψ corresponding to several neural network families and classification tasks. In this section we show that indeed Ψ has a low effective dimension, which explains why randomly sampled trials found better values. One simple way to characterize the shape of a high-dimensional function is to look at how much it varies in each dimension. Gaussian process regression gives us the statistical machinery to look at Ψ and measure its effective dimensionality (Neal, 1998; Rasmussen and Williams, 2006).

We estimated the sensitivity of Ψ to each hyper-parameter by fitting a Gaussian process (GP) with squared exponential kernels to predict $\Psi(\lambda)$ from λ. The squared exponential kernel (or Gaussian kernel) measures similarity between two real-valued hyper-parameter values a and b by $\exp(-\left(\frac{a-b}{l}\right)^2)$. The positive-valued l governs the sensitivity of the GP to change in this hyper-

[4] Source code for the simulations is available at `https://github.com/jaberg/hyperopt`.

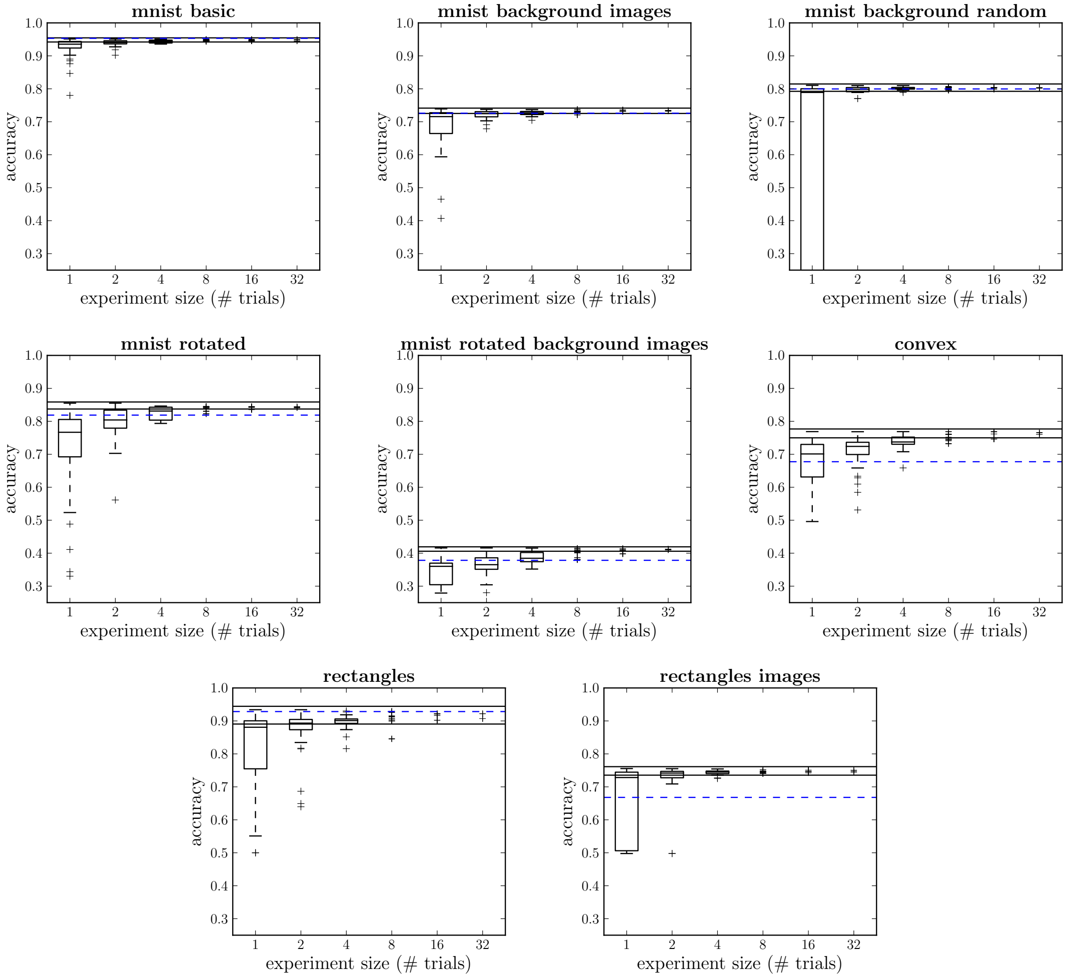


Figure 5: Neural network performance without preprocessing. Random experiment efficiency curves of a single-layer neural network for eight of the data sets used in Larochelle et al. (2007), looking only at trials with no preprocessing (7 hyper-parameters to optimize). The vertical axis is test-set accuracy of the best model by cross-validation, the horizontal axis is the experiment size (the number of models compared in cross-validation). The dashed blue line represents grid search accuracy for neural network models based on a selection by grids averaging 100 trials (Larochelle et al., 2007). Random searches of 8 trials match or outperform grid searches of (on average) 100 trials.

parameter. The kernels defined for each hyper-parameter were combined by multiplication (joint Gaussian kernel). We fit a GP to samples of Ψ by finding the *length scale* (l) for each hyper-parameter that maximized the marginal likelihood. To ensure relevance could be compared between hyper-parameters, we shifted and scaled each one to the unit interval. For hyper-parameters that were drawn geometrically or exponentially (e.g., learning rate, number of hidden units), kernel calculations were based on the logarithm of the effective value.

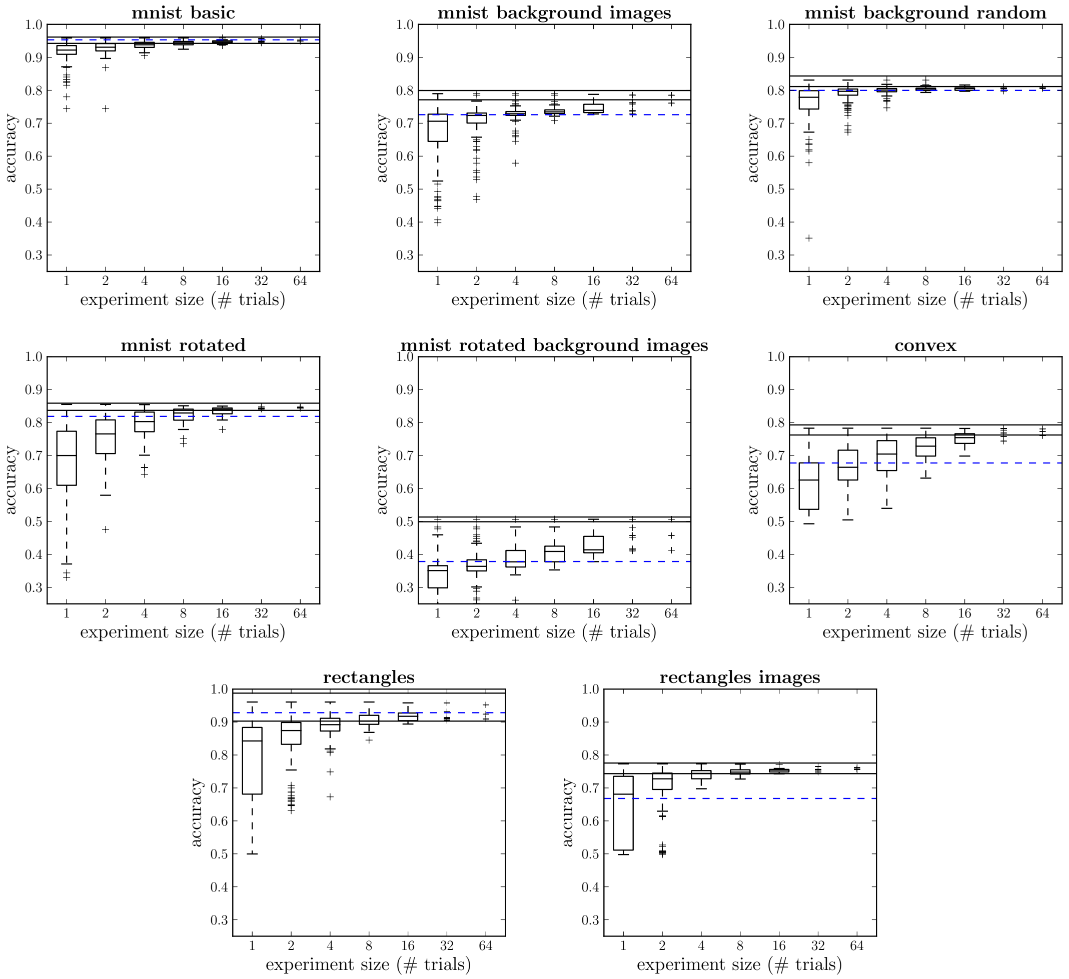


Figure 6: Neural network performance when standard preprocessing algorithms are considered (9 hyper-parameters). Dashed blue line represents grid search accuracy using (on average) 100 trials (Larochelle et al., 2007), in which no preprocessing was done. Often the extent of a search is determined by a computational budget, and with random search 64 trials are enough to find better models in a larger less promising space. Exploring just four PCA variance levels by grid search would have required 5 times as many (average 500) trials per data set.

Figure 7 shows the relevance of each component of Λ in modelling $\Psi(\lambda)$. Finding the length scales that maximize marginal likelihood is not a convex problem and many local minima exist. To get a sense of what length scales were supported by the data, we fit each set of samples from Ψ 50 times, resampling different subsets of 80% of the observations every time, and reinitializing the length scale estimates randomly between 0.1 and 2. Figure 7 reveals two important properties of Ψ for neural networks that suggest why grid search performs so poorly relative to random experiments:

1. a small fraction of hyper-parameters matter for any one data set, but

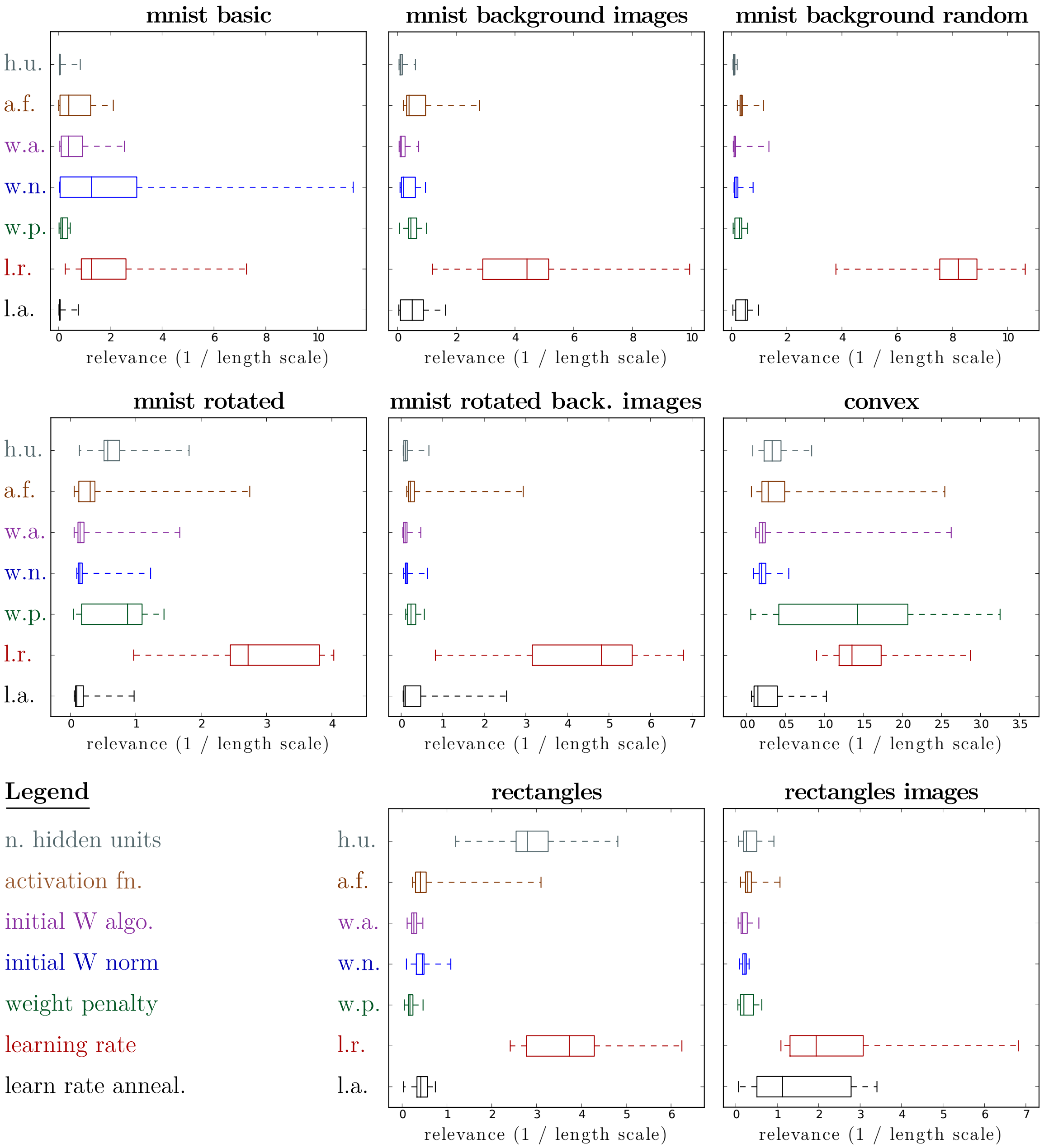


Figure 7: Automatic Relevance Determination (ARD) applied to hyper-parameters of neural network experiments (with raw preprocessing). For each data set, a small number of hyper-parameters dominate performance, but the relative importance of each hyper-parameter varies from each data set to the next. Section 2.4 describes the seven hyper-parameters in each panel. Boxplots are obtained by randomizing the subset of data used to fit the length scales, and randomizing the length scale initialization. (*Best viewed in color.*)

2. different hyper-parameters matter on different data sets.

Even in this simple 7-d problem, Ψ has a much lower effective dimension of between 1 and 4, depending on the data set. It would be impossible to cover just these few dimensions with a reliable grid however, because different data sets call for grids on different dimensions. The learning rate is always important, but sometimes the learning rate annealing rate was important (**rectangles images**), sometimes the ℓ_2-penalty was important (**convex**, **mnist rotated**), sometimes the number of hidden units was important (**rectangles**), and so on. While random search optimized these Ψ functions with 8 to 16 trials, a grid with, say, four values in each of these axes would already require 256 trials, and yet provide no guarantee that Ψ for a new data set would be well optimized.

Figure 7 also allows us to establish a correlation between effective dimensionality and ease of optimization. The data sets for which the effective dimensionality was lowest (1 or 2) were **mnist basic**, **mnist background images**, **mnist background random**, and **rectangles images**. Looking back at the corresponding efficiency curves (Figure 5) we find that these are also the data sets whose curves plateau most sharply, indicating that these functions are the easiest to optimize. They are often optimized reasonably well by just 2 random trials. Looking to Figure 7 at the data sets with largest effective dimensionality (3 or 4), we identify **convex**, **mnist rotated**, **rectangles**. Looking at their efficiency curves in Figure 5 reveals that they consistently required at least 8 random trials. This correlation offers another piece of evidence that the effective dimensionality of Ψ is playing a strong role in determining the difficulty of hyper-parameter optimization.

4. Grid Search and Sets with Low Effective Dimensionality

It is an interesting mathematical challenge to choose a set of trials for sampling functions of unknown, but low effective dimensionality. We would like it to be true that no matter which dimensions turn out to be important, our trials sample the important dimensions evenly. Sets of points with this property are well studied in the literature of Quasi-Random methods for numerical integration, where they are known as *low-discrepancy sets* because they try to match (minimize discrepancy with) the uniform distribution. Although there are several formal definitions of low discrepancy, they all capture the intuition that the points should be roughly equidistant from one another, in order that there be no “clumps” or “holes” in the point set.

Several procedures for constructing low-discrepancy point sets in multiple dimensions also try to ensure as much as possible that subspace projections remain low-discrepancy sets in the subspace. For example, the Sobol (Antonov and Saleev, 1979), Halton (Halton, 1960), and Niederreiter (Bratley et al., 1992) sequences, as well as latin hypercube sampling (McKay et al., 1979) are all more or less deterministic schemes for getting point sets that are more representative of random uniform draws than actual random uniform draws. In Quasi Monte-Carlo integration, such point sets are shown to asymptotically minimize the variance of finite integrals faster than true random uniform samples, but in this section, we will look at these point sets in the setting of relatively small sample sizes, to see if they can be used for more efficient search than random draws.

Rather than repeat the very computationally expensive experiments conducted in Section 2, we used an artificial simulation to compare the efficiency of grids, random draws, and the four low-discrepancy point sets mentioned in the previous paragraph. The artificial search problem was to find a uniformly randomly placed multi-dimensional target interval, which occupies 1% of the volume of the unit hyper-cube. We looked at four variants of the search problem, in which the target was

1. a cube in a 3-dimensional space,

2. a hyper-rectangle in a 3-dimensional space,

3. a hyper-cube in a 5-dimensional space,

4. a hyper-rectangle in a 5-dimensional space.

The shape of the target rectangle in variants (2) and (4) was determined by sampling side lengths uniformly from the unit interval, and then scaling the rectangle to have a volume of 1%. This process gave the rectangles a shape that was often wide or tall - much longer along some axes than others. The position of the target was drawn uniformly among the positions totally inside the unit hyper-cube. In the case of tall or wide targets (2) and (4), the indicator function [of the target] had a lower effective dimension than the dimensionality of the overall space because the dimensions in which the target is elongated can be almost ignored.

The simulation experiment began with the generation of 100 random search problems. Then for each experiment design method (random, Sobol, latin hypercube, grid) we created experiments of $1, 2, 3$, and so on up to 512 trials.[5] The Sobol, Niederreiter, and Halton sequences yielded similar results, so we used the Sobol sequence to represent the performance of these low-discepancy set construction methods. There are many possible grid experiments of any size in multiple dimensions (at least for non-prime experiment sizes). We did not test every possible grid, instead we tested every grid with a monotonic resolution. For example, for experiments of size 16 in 5 dimensions we tried the five grids with resolutions (1, 1, 1, 1, 16), (1, 1, 1, 2, 8), (1, 1, 2, 2, 4), (1, 1, 1, 4, 4), (1, 2, 2, 2, 2); for experiments of some prime size P in 3 dimensions we tried one grid with resolution $(1, 1, P)$. Since the target intervals were generated in such a way that rectangles identical up to a permutation of side lengths have equal probability, grids with monotonic resolution are representative of all grids. The score of an experiment design method for each experiment size was the fraction of the 100 targets that it found.

To characterize the performance of random search, we used the analytic form of the expectation. The expected probability of finding the target is 1.0 minus the probability of missing the target with every single one of T trials in the experiment. If the volume of the target relative to the unit hypercube is $(v/V = 0.01)$ and there are T trials, then this probability of finding the target is

$$1 - (1 - \frac{v}{V})^T = 1 - 0.99^T.$$

Figure 8 illustrates the efficiency of each kind of point set at finding the multidimensional intervals. There were some grids that were best at finding cubes and hyper-cubes in 3-d and 5-d, but most grids were the worst performers. No grid was competitive with the other methods at finding the rectangular-shaped intervals, which had low effective dimension (cases 2 and 4; Figure 8, right panels). Latin hypercubes, commonly used to initialize experiments in Bayesian optimization, were no more efficient than the expected performance of random search. Interestingly, the Sobol sequence was consistently best by a few percentage points. The low-discrepancy property that makes the Sobol useful in integration helps here, where it has the effect of minimizing the size of holes where the target might pass undetected. The advantage of the Sobol sequence is most pronounced in experiments of 100-300 trials, where there are sufficiently many trials for the structure in the Sobol

[5] Samples from the Sobol sequence were provided by the GNU Scientific Library (M. Galassi et al., 2009).

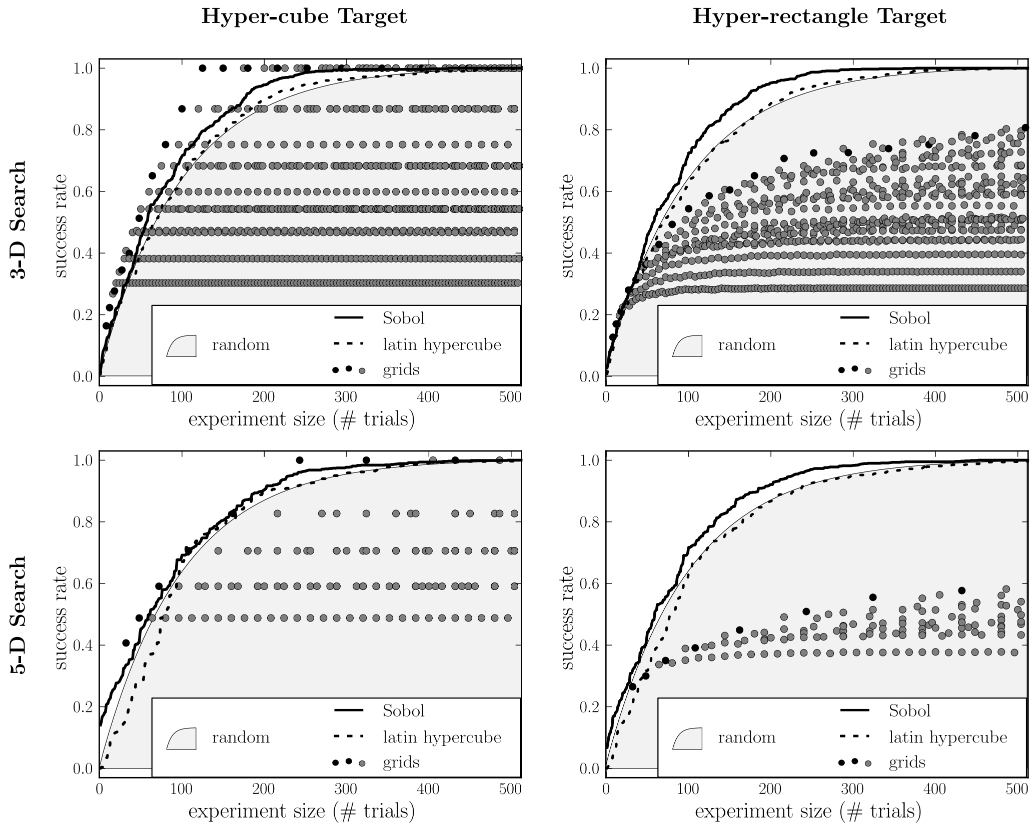


Figure 8: The efficiency in simulation of low-discrepancy sequences relative to grid and pseudo-random experiments. The simulation tested how reliably various experiment design methods locate a multidimensional interval occupying 1% of a unit hyper-cube. There is one grey dot in each sub-plot for every grid of every experiment size that has at least two ticks in each dimension. The black dots indicate near-perfect grids whose finest and coarsest dimensional resolutions differ by either 0 or 1. Hyper-parameter search is most typically like the bottom-right scenario. Grid search experiments are inefficient for finding axis-aligned elongated regions in high dimensions (i.e., bottom-right). Pseudo-random samples are as efficient as latin hypercube samples, and slightly less efficient than the Sobol sequence.

depart significantly from i.i.d points, but not sufficiently many trials for random search to succeed with high probability.

A thought experiment gives some intuition for why grid search fails in the case of rectangles. Long thin rectangles tend to intersect with several points if they intersect with any, reducing the effective sample size of the search. If the rectangles had been rotated away from the axes used to build the grid, then depending on the angle the efficiency of grid could approach the efficiency of random or low-discrepancy trials. More generally, if the target manifold were not systematically aligned with subsets of trial points, then grid search would be as efficient as the random and quasi-random searches.

5. Random Search vs. Sequential Manual Optimization

To see how random search compares with a careful combination of grid search and hand-tuning in the context of a model with many hyper-parameters, we performed experiments with the Deep Belief Network (DBN) model (Hinton et al., 2006). A DBN is a multi-layer graphical model with directed and undirected components. It is parameterized similarly to a multilayer neural network for classification, and it has been argued that *pretraining* a multilayer neural network by unsupervised learning as a DBN acts both to regularize the neural network toward better generalization, and to ease the optimization associated with *finetuning* the neural network for a classification task (Erhan et al., 2010).

A DBN classifier has many more hyper-parameters than a neural network. Firstly, there is the number of units and the parameters of random initialization for each layer. Secondly, there are hyper-parameters governing the unsupervised pretraining algorithm for each layer. Finally, there are hyper-parameters governing the global finetuning of the whole model for classification. For the details of how DBN models are trained (stacking restricted Boltzmann machines trained by contrastive divergence), the reader is referred to Larochelle et al. (2007), Hinton et al. (2006) or Bengio (2009). We evaluated random search by training 1-layer, 2-layer and 3-layer DBNs, sampling from the following distribution:

- We chose 1, 2, or 3 layers with equal probability.
- For each layer, we chose:
  - a number of hidden units (log-uniformly between 128 and 4000),
  - a weight initialization heuristic that followed from a distribution (uniform or normal), a multiplier (uniformly between 0.2 and 2), a decision to divide by the fan-out (true or false),
  - a number of iterations of contrastive divergence to perform for pretraining (log-uniformly from 1 to 10000),
  - whether to treat the real-valued examples used for unsupervised pretraining as Bernoulli means (from which to draw binary-valued training samples) or as a samples themselves (even though they are not binary),
  - an initial learning rate for contrastive divergence (log-uniformly between 0.0001 and 1.0),
  - a time point at which to start annealing the contrastive divergence learning rate as in Equation 7 (log-uniformly from 10 to 10 000).
- There was also the choice of how to preprocess the data. Either we used the raw pixels or we removed some of the variance using a ZCA transform (in which examples are projected onto principle components, and then multiplied by the transpose of the principle components to place them back in the inputs space).
- If using ZCA preprocessing, we kept an amount of variance drawn uniformly from 0.5 to 1.0.
- We chose to seed our random number generator with one of 2, 3, or 4.
- We chose a learning rate for finetuning of the final classifier log-uniformly from 0.001 to 10.

- We chose an anneal start time for finetuning log-uniformly from 100 to 10000.
- We chose ℓ_2 regularization of the weight matrices at each layer during finetuning to be either 0 (with probability 0.5), or log-uniformly from 10^{-7} to 10^{-4}.

This hyper-parameter space includes 8 global hyper-parameters and 8 hyper-parameters for each layer, for a total of 32 hyper-parameters for 3-layer models.

A grid search is not practical for the 32-dimensional search problem of DBN model selection, because even just 2 possible values for each of 32 hyper-parameters would yield more trials than we could conduct ($2^{32} > 10^9$ trials and each can take hours). For many of the hyper-parameters, especially real valued ones, we would really like to try more than two values. The approach taken in Larochelle et al. (2007) was a combination of manual search, multi-resolution grid search and coordinate descent. The algorithm (including manual steps) is somewhat elaborate, but sensible, and we believe that it is representative of how model search is typically done in several research groups, if not the community at large. Larochelle et al. (2007) describe it as follows:

> “The hyper-parameter search procedure we used alternates between fixing a neural network architecture and searching for good optimization hyper-parameters similarly to coordinate descent. More time would usually be spent on finding good optimization parameters, given some empirical evidence that we found indicating that the choice of the optimization hyper-parameters (mostly the learning rates) has much more influence on the obtained performance than the size of the network. We used the same procedure to find the hyper-parameters for DBN-1, which are the same as those of DBN-3 except the second hidden layer and third hidden layer sizes. We also allowed ourselves to test for much larger first-hidden layer sizes, in order to make the comparison between DBN-1 and DBN-3 fairer.
>
> “We usually started by testing a relatively small architecture (between 500 and 700 units in the first and second hidden layer, and between 1000 and 2000 hidden units in the last layer). Given the results obtained on the validation set (compared to those of NNet for instance) after selecting appropriate optimization parameters, we would then consider growing the number of units in all layers simultaneously. The biggest networks we eventually tested had up to 3000, 4000 and 6000 hidden units in the first, second and third hidden layers respectively.
>
> “As for the optimization hyper-parameters, we would proceed by first trying a few combinations of values for the stochastic gradient descent learning rate of the supervised and unsupervised phases (usually between 0.1 and 0.0001). We then refine the choice of tested values for these hyper-parameters. The first trials would simply give us a trend on the validation set error for these parameters (is a change in the hyper-parameter making things worse of better) and we would then consider that information in selecting appropriate additional trials. One could choose to use learning rate adaptation techniques (e.g., slowly decreasing the learning rate or using momentum) but we did not find these techniques to be crucial.

There was large variation in the number of trials used in Larochelle et al. (2007) to optimize the DBN-3. One data set (**mnist background images**) benefited from 102 trials, while another (**mnist background random**) only 13 because a good result was found more quickly. The average number

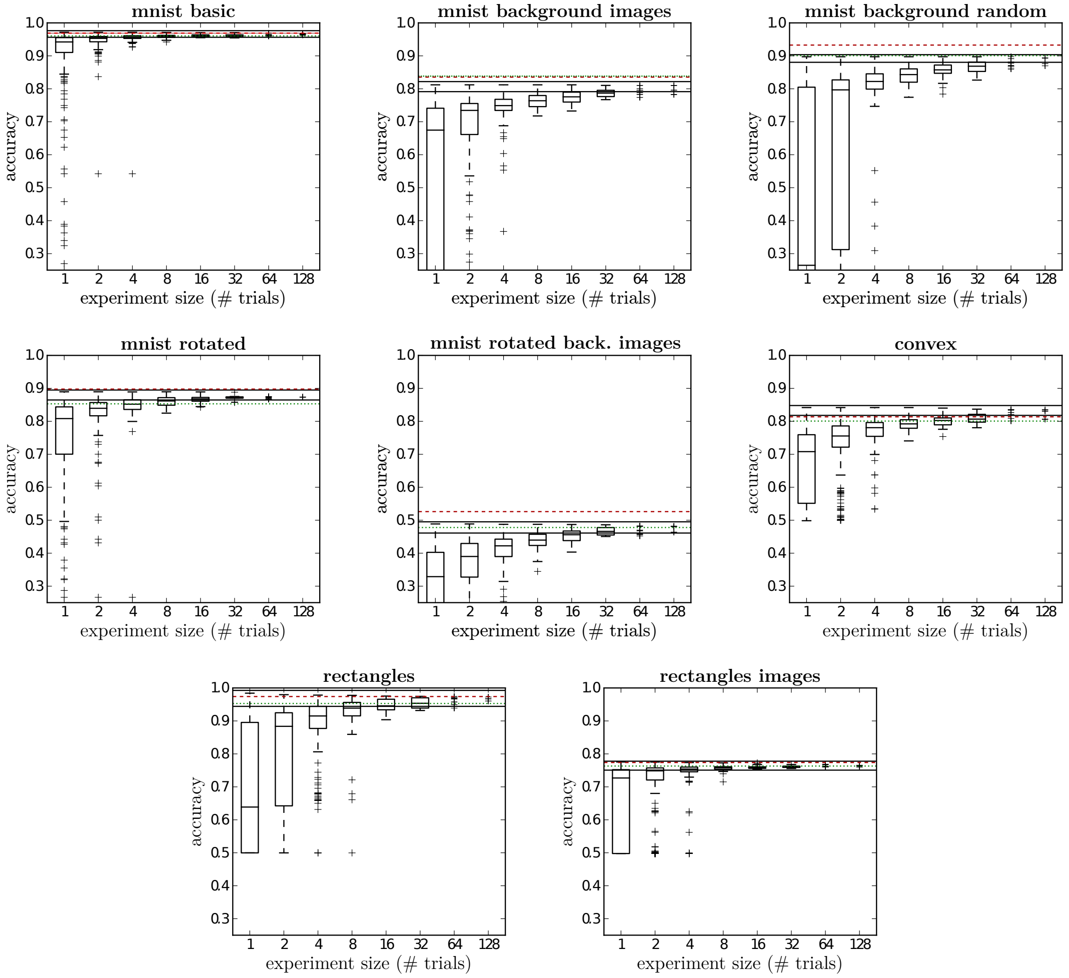


Figure 9: Deep Belief Network (DBN) performance according to random search. Here random search is used to explore up to 32 hyper-parameters. Results obtained by grid-assisted manual search using an average of 41 trials are marked in finely-dashed green (1-layer DBN) and coarsely-dashed red (3-layer DBN). Random experiments of 128 random trials found an inferior best model for three data sets, a competitive model in four, and superior model in one (**convex**). (*Best viewed in color.*)

of trials across data sets for the DBN-3 model was 41. In considering the number of trials per data set, it is important to bear in mind that the experiments on different data sets were not performed independently. Rather, later experiments benefited from the experience the authors had drawn from earlier ones. Although grid search was part of the optimization loop, the manual intervention turns the overall optimization process into something with more resemblance to an adaptive sequential algorithm.

Random search versions of the DBN experiments from Larochelle et al. (2007) are shown in Figure 9. In this more challenging optimization problem random search is still effective, but not superior as it was as in the case of neural network optimization. Comparing to the 3-layer DBN results in Larochelle et al. (2007), random search found a better model than the manual search in one data set (**convex**), an equally good model in four (**mnist basic**, **mnist rotated**, **rectangles**, and **rectangles images**), and an inferior model in three (**mnist background images**, **mnist background random**, **mnist rotated background images**). Comparing to the 1-layer DBN results, random search of the 1-layer, 2-layer and 3-layer configuration space found at least a good a model in all cases. In comparing these scores, the reader should bear in mind that the scores in the original experiments were not computed using the same score-averaging technique that we described in Section 2.1, and our averaging technique is slightly biased toward underestimation. In the DBN efficiency curves we see that even experiments with larger numbers of trials (64 and larger) feature significant variability. This indicates that the regions of the search space with the best performance are small, and randomly chosen i.i.d. trials do not reliably find them.

6. Future Work

Our result on the multidimensional interval task, together with the GPR characterization of the shape of Ψ, together with the computational constraint that hyper-parameter searches only draw on a few hundred trials, all suggest that pseudo-random or quasi-random trials are optimal for non-adaptive hyper-parameter search. There is still work to be done for each model family, to establish how it should be parametrized for i.i.d. random search to be as reliable as possible, but the most promising and interesting direction for future work is certainly in adaptive algorithms.

There is a large body of literature on global optimization, a great deal of which bears on the application of hyper-parameter optimization. General numeric methods such as simplex optimization (Nelder and Mead, 1965), constrained optimization by linear approximation (Powell, 1994; Weise, 2009), finite difference stochastic approximation and simultaneous prediction stochastic approximation (Kleinman et al., 1999) could be useful, as well as methods for search in discrete spaces such as simulated annealing (Kirkpatrick et al., 1983) and evolutionary algorithms (Rechenberg, 1973; Hansen et al., 2003). Drew and de Mello (2006) have already proposed an optimization algorithm that identifies effective dimensions, for more efficient search. They present an algorithm that distinguishes between important and unimportant dimensions: a low-discrepancy point set is used to choose points in the important dimensions, and unimportant dimensions are "padded" with thinner coverage and cheaper samples. Their algorithm's success hinges on the rapid and successful identification of important dimensions. Sequential model-based optimization methods and particularly Bayesian optimization methods are perhaps more promising because they offer principled approaches to weighting the importance of each dimension (Hutter, 2009; Hutter et al., 2011; Srinivasan and Ramakrishnan, 2011).

With so many sophisticated algorithms to draw on, it may seem strange that grid search is still widely used, and, with straight faces, we now suggest using random search instead. We believe the reason for this state of affairs is a technical one. Manual optimization followed by grid search is easy to implement: grid search requires very little code infrastructure beyond access to a cluster of computers. Random search is just as simple to carry out, uses the same tools, and fits in the same workflow. Adaptive search algorithms on the other hand require more code complexity. They require client-server architectures in which a master process keeps track of the trials that have completed, the trials that are in progress, the trials that were started but failed to complete. Some kind of shared database and inter-process communication mechanisms are required. Trials in an adaptive experiment cannot be queued up all at once; the master process must be involved somehow in the scheduling and timing of jobs on the cluster. These technical hurdles are not easy to jump with the standard tools of the trade such as MATLAB or Python; significant software engineering is required. Until that engineering is done and adopted by a community of researchers, progress on the study of sophisticated hyper-parameter optimization algorithms will be slow.

7. Conclusion

Grid search experiments are common in the literature of empirical machine learning, where they are used to optimize the hyper-parameters of learning algorithms. It is also common to perform multi-stage, multi-resolution grid experiments that are more or less automated, because a grid experiment with a fine-enough resolution for optimization would be prohibitively expensive. We have shown that random experiments are more efficient than grid experiments for hyper-parameter optimization in the case of several learning algorithms on several data sets. Our analysis of the hyper-parameter response surface (Ψ) suggests that random experiments are more efficient because not all hyper-parameters are equally important to tune. Grid search experiments allocate too many trials to the exploration of dimensions that do not matter and suffer from poor coverage in dimensions that are important. Compared with the grid search experiments of Larochelle et al. (2007), random search found better models in most cases and required less computational time.

Random experiments are also easier to carry out than grid experiments for practical reasons related to the statistical independence of every trial.

- The experiment can be stopped any time and the trials form a complete experiment.

- If extra computers become available, new trials can be added to an experiment without having to adjust the grid and commit to a much larger experiment.

- Every trial can be carried out asynchronously.

- If the computer carrying out a trial fails for any reason, its trial can be either abandoned or restarted without jeopardizing the experiment.

Random search is not incompatible with a controlled experiment. To investigate the effect of one hyper-parameter of interest X, we recommend random search (instead of grid search) for optimizing over other hyper-parameters. Choose one set of random values for these remaining hyper-parameters and use that same set for each value of X.

Random experiments with large numbers of trials also bring attention to the question of how to measure test error of an experiment when many trials have some claim to being best. When using a relatively small validation set, the uncertainty involved in selecting the best model by cross-validation can be larger than the uncertainty in measuring the test set performance of any one model. It is important to take both of these sources of uncertainty into account when reporting the uncertainty around the best model found by a model search algorithm. This technique is useful to all experiments (including both random and grid) in which multiple models achieve approximately the best validation set performance.

Low-discrepancy sequences developed for QMC integration are also good alternatives to grid-based experiments. In low dimensions (e.g., 1-5) our simulated results suggest that they can hold some advantage over pseudo-random experiments in terms of search efficiency. However, the trials of a low-discrepancy experiment are not i.i.d. which makes it inappropriate to analyze performance with the random efficiency curve. It is also more difficult in practice to conduct a quasi-random experiment because like a grid experiment, the omission of a single point can be more severe. Finally, when there are many hyper-parameter dimensions relative to the computational budget for the experiment, a low-discrepancy trial set is not expected to behave very differently from a pseudo-random one.

Finally, the hyper-parameter optimization strategies considered here are non-adaptive: they do not vary the course of the experiment by considering any results that are already available. Random search was not generally as good as the sequential combination of manual and grid search from an expert (Larochelle et al., 2007) in the case of the 32-dimensional search problem of DBN optimization, because the efficiency of sequential optimization overcame the inefficiency of the grid search employed at each step of the procedure. Future work should consider sequential, adaptive search/optimization algorithms in settings where many hyper-parameters of an expensive function must be optimized jointly and the effective dimensionality is high. We hope that future work in that direction will consider random search of the form studied here as a baseline for performance, rather than grid search.

Acknowledgments

This work was supported by the National Science and Engineering Research Council of Canada and Compute Canada, and implemented with Theano (Bergstra et al., 2010).

References

I. A. Antonov and V. M. Saleev. An economic method of computing LP_τ-sequences. *USSR Computational Mathematics and Mathematical Physics*, 19(1):252–256, 1979.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, New Jersey, 1961.

Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1–127, 2009. doi: 10.1561/2200000006.

Y. Bengio and X. Glorot. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proc. of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, pages 249–256, 2010.

J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral.

C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, London, UK, 1995.

P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *Transactions on Modeling and Computer Simulation, (TOMACS)*, 2(3):195–213, 1992.

R. E. Caflisch, W. Morokoff, and A. Owen. Valuation of mortgage backed securities using brownian bridges to reduce effective dimension, 1997.

C. Chang and C. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001.

I. Czogiel, K. Luebke, and C. Weihs. Response surface methodology for optimizing hyper parameters. Technical report, Universität Dortmund Fachbereich Statistik, September 2005.

S. S. Drew and T. Homem de Mello. Quasi-Monte Carlo strategies for stochastic optimization. In *Proc. of the 38th Conference on Winter Simulation*, pages 774 – 782, 2006.

D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, 2010.

J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11 (1):1–18, 2003.

G. E. Hinton. A practical guide to training restricted Boltzmann machines. Technical Report 2010-003, University of Toronto, 2010. version 1.

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

F. Hutter. *Automated Configuration of Algorithms for Solving Hard Computational Problems*. PhD thesis, University of British Columbia, 2009.

F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION-5*, 2011. Extended version as UBC Tech report TR-2010-10.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983.

N. L. Kleinman, J. C. Spall, and D. Q. Naiman. Simulation-based optimization with stochastic approximation using common random numbers. *Management Science*, 45(11):1570–1578, November 1999. doi: doi:10.1287/mnsc.45.11.1570.

H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In Z. Ghahramani, editor, *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML'07)*, pages 473–480. ACM, 2007.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998a.

Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. In G. Orr and K. Muller, editors, *Neural Networks: Tricks of the Trade*. Springer, 1998b.

M. Galassi et al. *GNU Scientific Library Reference Manual*, 3rd edition, 2009.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2): 239–245, May 1979. doi: doi:10.2307/1268522.

A. Nareyek. Choosing search heuristics by non-stationary reinforcement learning. *Applied Optimization*, 86:523–544, 2003.

R. M. Neal. Assessing relevance determination methods using DELVE. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 97–129. Springer-Verlag, 1998.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7: 308–313, 1965.

M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis*, pages 51–67, 1994.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Ingo Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fommann-Holzboog, Stuttgart, 1973.

A. Srinivasan and G. Ramakrishnan. Parameter screening and optimisation for ILP using designed experiments. *Journal of Machine Learning Research*, 12:627–662, February 2011.

P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In W. W. Cohen, A. McCallum, and S. T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, pages 1096–1103. ACM, 2008.

T. Weise. *Global Optimization Algorithms - Theory and Application*. Self-Published, second edition, 2009. Online available at http://www.it-weise.de/.

Journal of Machine Learning Research 13 (2012) 307-361 Submitted 12/10; Revised 11/11; Published 2/12

Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics

Michael U. Gutmann MICHAEL.GUTMANN@HELSINKI.FI
Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Department of Computer Science
Department of Mathematics and Statistics
Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland

Editor: Yoshua Bengio

Abstract

We consider the task of estimating, from observed data, a probabilistic model that is parameterized by a finite number of parameters. In particular, we are considering the situation where the model probability density function is unnormalized. That is, the model is only specified up to the partition function. The partition function normalizes a model so that it integrates to one for any choice of the parameters. However, it is often impossible to obtain it in closed form. Gibbs distributions, Markov and multi-layer networks are examples of models where analytical normalization is often impossible. Maximum likelihood estimation can then not be used without resorting to numerical approximations which are often computationally expensive. We propose here a new objective function for the estimation of both normalized and unnormalized models. The basic idea is to perform nonlinear logistic regression to discriminate between the observed data and some artificially generated noise. With this approach, the normalizing partition function can be estimated like any other parameter. We prove that the new estimation method leads to a consistent (convergent) estimator of the parameters. For large noise sample sizes, the new estimator is furthermore shown to behave like the maximum likelihood estimator. In the estimation of unnormalized models, there is a trade-off between statistical and computational performance. We show that the new method strikes a competitive trade-off in comparison to other estimation methods for unnormalized models. As an application to real data, we estimate novel two-layer models of natural image statistics with spline nonlinearities.

Keywords: unnormalized models, partition function, computation, estimation, natural image statistics

1. Introduction

This paper is about parametric density estimation, where the general setup is as follows. A sample $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ of a random vector $\mathbf{x} \in \mathbb{R}^n$ is observed which follows an unknown probability density function (pdf) p_d. The data-pdf p_d is modeled by a parameterized family of functions $\{p_m(.;\boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ where $\boldsymbol{\theta}$ is a vector of parameters. It is commonly assumed that p_d belongs to this family. In other words, $p_d(.) = p_m(.;\boldsymbol{\theta}^\star)$ for some parameter $\boldsymbol{\theta}^\star$. The parametric density estimation problem is then about finding $\boldsymbol{\theta}^\star$ from the observed sample X. Any estimate $\hat{\boldsymbol{\theta}}$ must yield a properly normalized pdf $p_m(.;\hat{\boldsymbol{\theta}})$ which satisfies

$$\int p_m(\mathbf{u};\hat{\boldsymbol{\theta}})\mathrm{d}\mathbf{u} = 1, \qquad\qquad p_m(.;\hat{\boldsymbol{\theta}}) \geq 0. \tag{1}$$

These are two constraints in the estimation.

If the model $p_m(.;\boldsymbol{\theta})$ is such that the constraints hold for all $\boldsymbol{\theta}$, and not only $\hat{\boldsymbol{\theta}}$, we say that the model is normalized. The maximum likelihood principle can then be used to estimate $\boldsymbol{\theta}$. If the model is specified such that the positivity constraint but not the normalization constraint is satisfied for all parameters, we say that the model is unnormalized. By assumption there is, however, at least one value of the parameters for which an unnormalized model integrates to one, namely $\boldsymbol{\theta}^\star$. In order to highlight that a model, parameterized by some $\boldsymbol{\alpha}$, is unnormalized, we denote it by $p_m^0(.;\boldsymbol{\alpha})$. Unnormalized models are easy to specify by taking, for example, the exponential transform of a suitable function.

The partition function $Z(\boldsymbol{\alpha})$,

$$Z(\boldsymbol{\alpha}) = \int p_m^0(\mathbf{u};\boldsymbol{\alpha})\mathrm{d}\mathbf{u}, \tag{2}$$

can be used to convert an unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ into a normalized one: $p_m^0(.;\boldsymbol{\alpha})/Z(\boldsymbol{\alpha})$ integrates to one for every value of $\boldsymbol{\alpha}$. Examples of distributions which are often specified by means of an unnormalized model and the partition function are Gibbs distributions, Markov networks or multilayer networks. The function $\boldsymbol{\alpha} \mapsto Z(\boldsymbol{\alpha})$ is, however, defined via an integral. Unless $p_m^0(.;\boldsymbol{\alpha})$ has some particularly convenient form, the integral cannot be computed analytically so that the function $Z(\boldsymbol{\alpha})$ is not available in closed form. For low-dimensional problems, numerical integration can be used to approximate the function $Z(\boldsymbol{\alpha})$ to a very high accuracy but for high-dimensional problems this is computationally expensive. Our paper deals with density estimation in this case, that is, with density estimation when the computation of the partition function is analytically intractable and computationally expensive.

Several solutions for the estimation of unnormalized models which cannot be normalized in closed form have been suggested so far. Geyer (1994) proposed to approximate the calculation of the partition function by means of importance sampling and then to maximize the approximate log-likelihood (Monte Carlo maximum likelihood). Approximation of the gradient of the log-likelihood led to another estimation method (contrastive divergence by Hinton, 2002). Estimation of the parameter $\boldsymbol{\alpha}$ directly from an unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ has been proposed by Hyvärinen (2005). This approach, called score matching, avoids the problematic integration to obtain the partition function altogether. All these methods need to balance the accuracy of the estimate and the time to compute the estimate.

In this paper,[1] we propose a new estimation method for unnormalized models. The idea is to consider Z, or $c = \ln 1/Z$, not any more as a function of $\boldsymbol{\alpha}$ but as an additional parameter of the model. That is, we extend the unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ to include a normalizing parameter c and estimate

$$\ln p_m(.;\boldsymbol{\theta}) = \ln p_m^0(.;\boldsymbol{\alpha}) + c,$$

with parameter vector $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. The estimate $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{c})$ is then such that the unnormalized model $p_m^0(.;\hat{\boldsymbol{\alpha}})$ matches the shape of p_d, while $\hat{c}$ provides the proper scaling so that Equation (1) holds.

[1] Preliminary versions were presented at AISTATS (Gutmann and Hyvärinen, 2010) and ICANN (Gutmann and Hyvärinen, 2009).

Unlike in the approach based on the partition function, we aim not at normalizing $p_m^0(.;\boldsymbol{\alpha})$ for all $\boldsymbol{\alpha}$ but only for $\hat{\boldsymbol{\alpha}}$. This avoids the problematic integration in the definition of the partition function $\boldsymbol{\alpha} \mapsto Z(\boldsymbol{\alpha})$. Such a separate estimation of shape and scale is, however, not possible for maximum likelihood estimation (MLE). The reason is that the likelihood can be made arbitrarily large by setting the normalizing parameter c to larger and larger numbers. The new estimation method which we propose here is based on the maximization of a well defined objective function. There are no constraints in the optimization so that powerful optimization techniques can be employed. The intuition behind the new objective function is to learn to classify between the observed data and some artificially generated noise. We approach thus the density estimation problem, which is an unsupervised learning problem, via supervised learning. The new method relies on noise which the data is contrasted to, so that we will refer to it as "noise-contrastive estimation".

The paper is organized in four main sections. In Section 2, we present noise-contrastive estimation and prove fundamental statistical properties such as consistency. In Section 3, we validate and illustrate the derived properties on artificial data. We use artificial data also in Section 4 in order to compare the new method to the aforementioned estimation methods with respect to their statistical and computational efficiency. In Section 5, we apply noise-contrastive estimation to real data. We estimate two-layer models of natural images and also learn the nonlinearities from the data. This section is fairly independent from the other ones. The reader who wants to focus on natural image statistics may not need to go first through the previous sections. On the other hand, the reader whose interest is in estimation theory only can skip this section without missing pieces of the theory although the section provides, using real data, a further illustration of the workings of unnormalized models and the new estimation method. Section 6 concludes the paper.

2. Noise-Contrastive Estimation

This section presents the theory of noise-contrastive estimation. In Section 2.1, we motivate noise-contrastive estimation and relate it to supervised learning. The definition of noise-contrastive estimation is given in Section 2.2. In Section 2.3, we prove that the estimator is consistent for both normalized and unnormalized models, and derive its asymptotic distribution. In Section 2.4, we discuss practical aspects of the estimator and show that, in some limiting case, the estimator performs as well as MLE.

2.1 Density Estimation by Comparison

Density estimation is much about characterizing properties of the observed data X. A convenient way to describe properties is to describe them relative to the properties of some reference data Y. Let us assume that the reference (noise) data Y is an i.i.d. sample $(\mathbf{y}_1, \dots \mathbf{y}_{T_n})$ of a random variable $\mathbf{y} \in \mathbb{R}^n$ with pdf p_n. A relative description of the data X is then given by the ratio p_d/p_n of the two density functions. If the reference distribution p_n is known, one can, of course, obtain p_d from the ratio p_d/p_n. In other words, if one knows the differences between X and Y, and also the properties of Y, one can deduce from the differences the properties of X.

Comparison between two data sets can be performed via classification: In order to discriminate between two data sets, the classifier needs to compare their properties. In the following, we show that training a classifier based on logistic regression provides a relative description of X in the form of an estimate of the ratio p_d/p_n.

Denote by $U = (\mathbf{u}_1, \dots, \mathbf{u}_{T_d+T_n})$ the union of the two sets X and Y, and assign to each data point $\mathbf{u}_t$ a binary class label C_t: $C_t = 1$ if $\mathbf{u}_t \in X$ and $C_t = 0$ if $\mathbf{u}_t \in Y$. In logistic regression, the posterior probabilities of the classes given the data are estimated. As the pdf p_d of the data $\mathbf{x}$ is unknown, we model the class-conditional probability $p(.|C = 1)$ with $p_m(.;\boldsymbol{\theta})$.[2] The class-conditional probability densities are thus

$$p(\mathbf{u}|C=1;\boldsymbol{\theta}) = p_m(\mathbf{u};\boldsymbol{\theta}), \qquad p(\mathbf{u}|C=0) = p_n(\mathbf{u}).$$

The prior probabilities are $P(C=1) = T_d/(T_d+T_n)$ and $P(C=0) = T_n/(T_d+T_n)$. The posterior probabilities for the classes are therefore

$$P(C=1|\mathbf{u};\boldsymbol{\theta}) = \frac{p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \qquad P(C=0|\mathbf{u};\boldsymbol{\theta}) = \frac{\nu p_n(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})}, \tag{3}$$

where ν is the ratio $P(C=0)/P(C=1) = T_n/T_d$. In the following, we denote $P(C=1|\mathbf{u};\boldsymbol{\theta})$ by $h(\mathbf{u};\boldsymbol{\theta})$. Introducing the log-ratio $G(.;\boldsymbol{\theta})$ between $p_m(.;\boldsymbol{\theta})$ and p_n,

$$G(\mathbf{u};\boldsymbol{\theta}) = \ln p_m(\mathbf{u};\boldsymbol{\theta}) - \ln p_n(\mathbf{u}), \tag{4}$$

$h(\mathbf{u};\boldsymbol{\theta})$ can be written as

$$h(\mathbf{u};\boldsymbol{\theta}) = r_\nu\left(G(\mathbf{u};\boldsymbol{\theta})\right), \tag{5}$$

where

$$r_\nu(u) = \frac{1}{1+\nu\exp(-u)} \tag{6}$$

is the logistic function parameterized by ν.

The class labels C_t are assumed Bernoulli distributed and independent. The conditional log-likelihood is given by

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{t=1}^{T_d+T_n} C_t \ln P(C_t=1|\mathbf{u}_t;\boldsymbol{\theta}) + (1-C_t)\ln P(C_t=0|\mathbf{u}_t;\boldsymbol{\theta}) \\
&= \sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \sum_{t=1}^{T_n} \ln\left[1-h(\mathbf{y}_t;\boldsymbol{\theta})\right].
\end{aligned} \tag{7}$$

Optimizing $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ leads to an estimate $G(.;\hat{\boldsymbol{\theta}})$ of the log-ratio $\ln(p_d/p_n)$. That is, an approximate description of X relative to Y can be obtained by optimization of Equation (7). The sign-flipped objective function, $-\ell(\boldsymbol{\theta})$, is also known as the cross-entropy error function (Bishop, 1995).

Thus, density estimation, which is an unsupervised learning problem, can be performed by logistic regression, that is, supervised learning. While this connection has been discussed earlier by Hastie et al. (2009, Chapter 14.2.4, pp. 495–497), in the next sections, we will prove that even unnormalized models can be estimated with the same principle.

[2] Classically, $p_m(.;\boldsymbol{\theta})$ would, in the context of this section, be a normalized pdf. In our paper, however, $\boldsymbol{\theta}$ may include a parameter for the normalization of the model.

2.2 Definition of the Estimator

Given an unnormalized statistical model $p_m^0(.;\boldsymbol{\alpha})$, we include for normalization an additional parameter c into the model. That is, we define the model as

$$\ln p_m(.;\boldsymbol{\theta}) = \ln p_m^0(.;\boldsymbol{\alpha}) + c,$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. The parameter c scales the unnormalized model $p_m^0(.;\boldsymbol{\alpha})$ so that Equation (1) can be fulfilled. After learning, $\hat{c}$ provides an estimate for $\ln 1/Z(\hat{\boldsymbol{\alpha}})$. If the initial model is normalized in the first place, no such inclusion of a normalizing parameter c is needed.

In line with the notation so far, we denote by $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ the observed data set that consists of T_d independent observations of $\mathbf{x} \in \mathbb{R}^n$. We denote by $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_{T_n})$ an artificially generated data set that consists of $T_n = \nu T_d$ independent observations of noise $\mathbf{y} \in \mathbb{R}^n$ with known distribution p_n. The estimator is defined to be the argument $\hat{\boldsymbol{\theta}}_T$ which maximizes

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\left\{\sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right]\right\}, \tag{8}$$

where the nonlinearity $h(.;\boldsymbol{\theta})$ was defined in Equation (5). The objective function J_T is, up to the division by T_d, the log-likelihood in Equation (7). It can also be written as

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d} \ln\left[h(\mathbf{x}_t;\boldsymbol{\theta})\right] + \nu \frac{1}{T_n}\sum_{t=1}^{T_n} \ln\left[1 - h(\mathbf{y}_t;\boldsymbol{\theta})\right]. \tag{9}$$

Note that $h(.;\boldsymbol{\theta}) \in (0\ 1)$, where zero is obtained in the limit of $G(.;\boldsymbol{\theta}) \to -\infty$ and one in the limit of $G(.;\boldsymbol{\theta}) \to \infty$. Zero is an upper bound for J_T, which is reached if, for all t, $h(\mathbf{x}_t;\boldsymbol{\theta})$ and $h(\mathbf{y}_t;\boldsymbol{\theta})$ tend to one and zero, respectively. Therefore, the optimal parameter $\hat{\boldsymbol{\theta}}_T$ is such that $G(\mathbf{u}_t;\hat{\boldsymbol{\theta}}_T)$ is as large as possible for $\mathbf{u}_t \in X$ and as small as possible for $\mathbf{u}_t \in Y$. Intuitively, this means that logistic regression has learned to discriminate between the two sets as well as possible.

2.3 Properties of the Estimator

We characterize here the behavior of the estimator $\hat{\boldsymbol{\theta}}_T$ for large sample sizes T_d and fixed ratio ν. Since ν is kept fixed, $T_n = \nu T_d$ will also increase as T_d increases. The weak law of large numbers shows that as T_d increases the objective function $J_T(\boldsymbol{\theta})$ converges in probability to J,

$$J(\boldsymbol{\theta}) = \mathrm{E}\left\{\ln\left[h(\mathbf{x};\boldsymbol{\theta})\right]\right\} + \nu\,\mathrm{E}\left\{\ln\left[1 - h(\mathbf{y};\boldsymbol{\theta})\right]\right\}. \tag{10}$$

Let us denote by $\tilde{J}$ the objective J seen as a function of $f_m(.) = \ln p_m(.;\boldsymbol{\theta})$,

$$\tilde{J}(f_m) \quad = \quad \mathrm{E}\left\{\ln\left[r_\nu\left(f_m(\mathbf{x}) - \ln p_n(\mathbf{x})\right)\right]\right\} + \nu\,\mathrm{E}\left\{\ln\left[1 - r_\nu\left(f_m(\mathbf{y}) - \ln p_n(\mathbf{y})\right)\right]\right\}. \tag{11}$$

We start the characterization of the estimator $\hat{\boldsymbol{\theta}}_T$ by describing the optimization landscape for f_m. The following theorem shows that the data-pdf p_d can be found by maximization of $\tilde{J}$, that is by learning a nonparametric classifier under the ideal situation of an infinite amount of data.

Theorem 1 (Nonparametric estimation) *$\tilde{J}$ attains a maximum at $f_m = \ln p_d$. There are no other extrema if the noise density p_n is chosen such that it is nonzero whenever p_d is nonzero.*

The proof is given in Appendix A.2. A fundamental point in the theorem is that the maximization is performed without any normalization constraint for f_m. This is in stark contrast to MLE, where $\exp(f_m)$ must integrate to one. With our objective function, no such constraints are necessary. The maximizing pdf is found to have unit integral automatically.

The positivity condition for p_n in the theorem tells us that the data-pdf p_d cannot be inferred at regions in the data space where there are no contrastive noise samples. For example, the estimation of a pdf p_d which is nonzero only on the positive real line by means of a noise distribution p_n that has its support on the negative real line is impossible. The positivity condition can be easily fulfilled by taking, for example, a Gaussian as contrastive noise distribution.

In practice, the amount of data is limited and a finite number of parameters $\boldsymbol{\theta} \in \mathbb{R}^m$ specify $p_m(.;\boldsymbol{\theta})$. This has two consequences for any estimation method that is based on optimization: First, it restricts the space where the data-pdf p_d is searched for. Second, it may introduce local maxima into the optimization landscape. For the characterization of the estimator in this situation, it is normally assumed that p_d follows the model, so that there is a $\boldsymbol{\theta}^\star$ with $p_d(.) = p_m(.;\boldsymbol{\theta}^\star)$. In the following, we make this assumption.

Our second theorem shows that $\hat{\boldsymbol{\theta}}_T$, the value of $\boldsymbol{\theta}$ which (globally) maximizes J_T, converges to $\boldsymbol{\theta}^\star$. The correct estimate of p_d is thus obtained as the sample size T_d increases. For unnormalized models, the conclusion of the theorem is that maximization of J_T leads to the correct estimates for both the parameter $\boldsymbol{\alpha}$ in the unnormalized pdf $p_m^0(.;\boldsymbol{\alpha})$ and the normalizing parameter c.

Theorem 2 (Consistency) *If conditions (a) to (c) are fulfilled then $\hat{\boldsymbol{\theta}}_T$ converges in probability to $\boldsymbol{\theta}^\star$, $\hat{\boldsymbol{\theta}}_T \xrightarrow{P} \boldsymbol{\theta}^\star$.*

(a) p_n *is nonzero whenever* p_d *is nonzero*

(b) $\sup_\theta |J_T(\boldsymbol{\theta}) - J(\boldsymbol{\theta})| \xrightarrow{P} 0$

(c) The matrix $\boldsymbol{\mathcal{I}}_\nu = \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$ *has full rank, where*

$$\mathbf{g}(\mathbf{u}) = \nabla_{\boldsymbol{\theta}} \ln p_m(\mathbf{u};\boldsymbol{\theta})|_{\boldsymbol{\theta}^\star}, \qquad\qquad P_\nu(\mathbf{u}) = \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}.$$

The proof is given in Appendix A.3. Condition (a) is inherited from Theorem 1. Conditions (b) and (c) have their counterparts in MLE (see for example Wasserman, 2004, Theorem 9.13): We need in (b) uniform convergence in probability of J_T to J; in MLE, uniform convergence of the log-likelihood to the Kullback-Leibler divergence is required likewise. Condition (c) assures that for large sample sizes, the objective function J_T becomes peaked enough around the true value $\boldsymbol{\theta}^\star$. This imposes a constraint on the model $p_m(.;\boldsymbol{\theta})$ via the vector $\mathbf{g}$. A similar constraint is required in MLE.

The next theorem describes the distribution of the estimation error $(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ for large sample sizes. The proof is given in Appendix A.4.

Theorem 3 (Asymptotic normality) *$\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ is asymptotically normal with mean zero and covariance matrix $\boldsymbol{\Sigma}$,*

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right) \boldsymbol{\mathcal{I}}_\nu^{-1} \mathrm{E}(P_\nu \mathbf{g}) \mathrm{E}(P_\nu \mathbf{g})^T \boldsymbol{\mathcal{I}}_\nu^{-1},$$

where $\mathrm{E}(P_\nu \mathbf{g}) = \int P_\nu(\mathbf{u}) \mathbf{g}(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$.

From the distribution of $\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$, we can easily evaluate the asymptotic mean squared error (MSE) of the estimator.

Corollary 4 *For large sample sizes* T_d*, the mean squared error* $\mathrm{E}\left(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2\right)$ *equals* $\mathrm{tr}(\boldsymbol{\Sigma})/T_d$.

Proof Using that for any vector $\mathbf{v}$, $||\mathbf{v}||^2 = \mathrm{tr}(\mathbf{v}\mathbf{v}^T)$, the corollary follows directly from the definition of the MSE and Theorem 3. ■

2.4 Choosing the Noise

Theorem 3 shows that the noise distribution p_n and the ratio $\nu = T_n/T_d$ have an influence on the accuracy of the estimate $\hat{\boldsymbol{\theta}}_T$. A natural question to ask is what, from a statistical standpoint, the best choice of p_n and ν is. Our result on consistency (Theorem 2) also includes a technical constraint for p_n but this one is so mild that many distributions will satisfy it.

Theorem 2 shows that, for a given samples size T_d, P_ν tends to one as the size T_n of the contrastive noise sample is made larger and larger. This implies that for large ν, the covariance matrix $\boldsymbol{\Sigma}$ does not depend on the choice of the noise distribution p_n. We have thus the following corollary.

Corollary 5 *For* $\nu \to \infty$, $\boldsymbol{\Sigma}$ *is independent of the choice of* p_n *and equals*

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}^{-1} - \boldsymbol{\mathcal{I}}^{-1}\mathrm{E}(\mathbf{g})\,\mathrm{E}(\mathbf{g})^T\boldsymbol{\mathcal{I}}^{-1},$$

where $\mathrm{E}(\mathbf{g}) = \int \mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}$ *and* $\boldsymbol{\mathcal{I}} = \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T p_d(\mathbf{u})\mathrm{d}\mathbf{u}$.

The asymptotic distribution of the estimation error becomes thus independent from p_n. Hence, as the size of the contrastive-noise sample Y increases, the choice of the contrastive-noise distribution becomes less and less important. Moreover, for normalized models, we have the result that the estimation error has the same distribution as the estimation error in MLE.

Corollary 6 *For normalized models, noise-contrastive estimation is, in the limit of* $\nu \to \infty$*, asymptotically Fisher-efficient for all choices of* p_n.

Proof For normalized models, no normalizing parameter c is needed. In Corollary 5, the function $\mathbf{g}$ is then the score function as in MLE, and the matrix $\boldsymbol{\mathcal{I}}$ is the Fisher information matrix. Since the expectation $\mathrm{E}(\mathbf{g})$ is zero, the covariance matrix $\boldsymbol{\Sigma}$ is the inverse of the Fisher information matrix. ■

The corollaries above give one answer to the question on how to choose the noise distribution p_n and the ratio ν: If ν is made large enough, the actual choice of p_n is not of great importance. Note that this answer considers only estimation accuracy and ignores the computational load associated with the processing of noise. In Section 4, we will analyze the trade-off between estimation accuracy and computation time.

For any given ν, one could try to find the noise distribution which minimizes the MSE $\mathrm{E}\,||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2$. However, this minimization turns out to be quite difficult. Intuitively, one could think that a good candidate for the noise distribution p_n is a distribution which is close to the data distribution p_d. If p_n is too different from p_d, the classification problem might be too easy and would not require the system to learn much about the structure of the data. This intuition is partly justified by the following theoretical result:

Corollary 7 *If* $p_n = p_d$ *then* $\mathbf{\Sigma} = \left(1+\frac{1}{\nu}\right)\left(\boldsymbol{\mathcal{I}}^{-1} - \boldsymbol{\mathcal{I}}^{-1}\,\mathrm{E}(\mathbf{g})\,\mathrm{E}(\mathbf{g})^T\boldsymbol{\mathcal{I}}^{-1}\right)$.

Proof The corollary follows from Theorem 3 and the fact that P_ν equals $\nu/(1+\nu)$ for $p_n = p_d$. ■

For normalized models, we see that for $\nu = 1$, $\mathbf{\Sigma}$ is two times the inverse of the Fisher information matrix, and that for $\nu = 10$, the ratio is already down to 1.1. For a noise distribution that is close to the data distribution, we have thus even for moderate values of ν some guarantee that the MSE is reasonably close to the theoretical optimum.

To get estimates with a small estimation error, the foregoing discussion suggests the following

1. Choose noise for which an analytical expression for $\ln p_n$ is available.

2. Choose noise that can be sampled easily.

3. Choose noise that is in some aspect, for example with respect to its covariance structure, similar to the data.

4. Make the noise sample size as large as computationally possible.

Some examples for suitable noise distributions are Gaussian distributions, Gaussian mixture distributions, or ICA distributions. Uniform distributions are also suitable as long as their support includes the support of the data distribution so that condition (a) in Theorem 2 holds.

3. Simulations to Validate and Illustrate the Theory

In this section,[3] we validate and illustrate the theoretical properties of noise-contrastive estimation. In Section 3.1, we focus on the consistency of the estimator. In Section 3.2, we validate our theoretical results on the distribution of the estimation error, and investigate its dependency on the ratio ν between noise and data sample size. In Section 3.3, we study how the performance of the estimator scales with the dimension of the data.

3.1 Consistency

For the illustration of consistency, we estimate here the parameters of a zero mean multivariate Gaussian. Its log-pdf is

$$\ln p_d(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T\mathbf{\Lambda}^\star\mathbf{x} + c^\star, \qquad c^\star = \left(-\frac{1}{2}\ln|\det\mathbf{\Lambda}^\star| - \frac{n}{2}\ln(2\pi)\right), \tag{12}$$

where $c^\star$ does not depend on $\mathbf{x}$ and normalizes p_d to integrate to one. The precision matrix $\mathbf{\Lambda}^\star$ is the inverse of the covariance matrix. It is thus a symmetric matrix. The dimension of $\mathbf{x}$ is here $n = 5$.

As we are mostly interested in the estimation of unnormalized models, we consider here the hypothetical situation where we want to estimate the model

$$\ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) = -\frac{1}{2}\mathbf{x}^T\mathbf{\Lambda}\mathbf{x}$$

without knowing how to normalize it in closed form. This unnormalized model is a pairwise Markov network with quadratic node and edge potentials (see for example Koller and Friedman, 2009, Chapter 7). The parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^{15}$ contains the coefficients of the lower-triangular part of $\mathbf{\Lambda}$ as the

[3] Matlab code for this and the other sections can be downloaded from the homepage of the first author.

matrix is symmetric. For noise-contrastive estimation, we add an additional normalizing parameter c to the model. The model that we estimate is thus

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) + c.$$

The model has 16 parameters given by $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. They are estimated by maximization of the objective function $J_T(\boldsymbol{\theta})$ in Equation (8). We used a standard normal distribution for p_n. The optimization was performed with the nonlinear conjugate gradient algorithm of Rasmussen (2006).

3.1.1 RESULTS

The presented results are an average over 500 estimation problems where the true precision matrix $\boldsymbol{\Lambda}^\star$ was drawn at random with the condition number being controlled to be smaller than ten. The sampling of $\boldsymbol{\Lambda}^\star$ was performed by randomly sampling its eigenvalues and eigenvectors: We drew the eigenvalues from an uniform distribution on the interval $[0.1\ 0.9]$. The orthonormal matrix $\mathbf{E}$ with the eigenvectors was created by orthogonally projecting a matrix $\mathbf{M}$ with elements drawn independently from a standard Gaussian onto the set of orthonormal matrices: $\mathbf{E} = (\mathbf{M}\mathbf{M}^T)^{-1/2}\mathbf{M}$.

Figure 1(a) and (b) show the mean squared error (MSE) for $\boldsymbol{\alpha}$, which contains the elements of the precision matrix $\boldsymbol{\Lambda}$, and the normalizing parameter c, respectively. The MSE as a function of the data sample size T_d decays linearly on a log-log scale. This illustrates our result of consistency of the estimator, stated as Theorem 2, as convergence in quadratic mean implies convergence in probability. The plots also show that taking more noise samples T_n than data samples T_d leads to more and more accurate estimates. The performance for noise-contrastive estimation with $\nu = T_n/T_d$ equal to one is shown in blue with circles as markers. For that value of ν, there is a clear difference compared to MLE (black triangles in Figure 1(a)). However, the accuracy of the estimate improves strongly for $\nu = 5$ (green squares) or $\nu = 10$ (red diamonds) where the performance is rather close to the performance of MLE.

Another way to visualize the results is by showing the Kullback-Leibler divergences between the 500 true and estimated distributions. Figure 2 shows boxplots of the divergences for $\nu = 1$ (blue) and $\nu = 10$ (red). The results for MLE are shown in black. In line with the visualization in Figure 1, the estimated distribution becomes closer to the true distribution as the sample size increases. Moreover, the divergences become clearly smaller as ν is increased from one to ten.

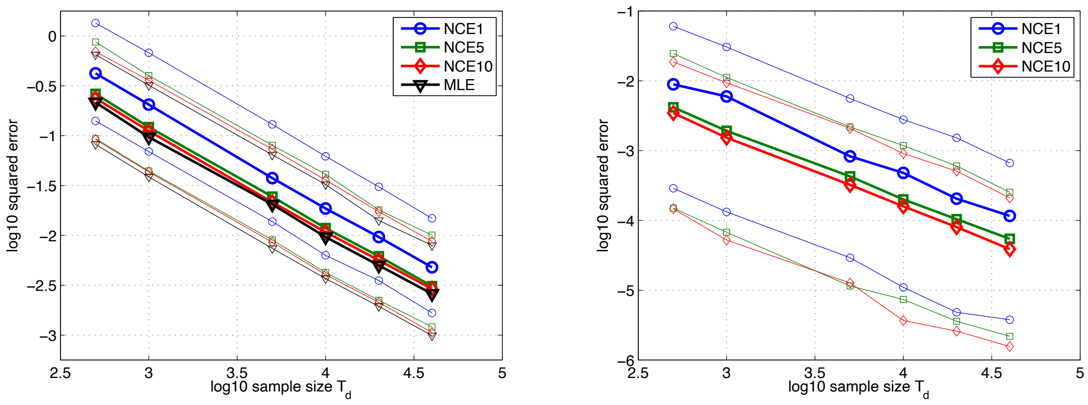
For unnormalized models, there is a subtlety in the computation of the divergence. With a validation set of size T_v, a sample version D_{KL} of the Kullback-Leibler divergence is given by the difference

$$D_{\mathrm{KL}} = \frac{1}{T_v}\sum_{t=1}^{T_v} \ln p_d(\mathbf{x}_t) - \left(\frac{1}{T_v}\sum_{t=1}^{T_v} \ln p_m^0(\mathbf{x}_t;\hat{\boldsymbol{\alpha}}) + \ln 1/Z(\hat{\boldsymbol{\alpha}})\right).$$

The first term is the rescaled log-likelihood (average, sign-inverted log-loss) for the true distribution. The term in parentheses is the rescaled log-likelihood L of the estimated model. In the estimation of unnormalized models, we do not assume to know the mapping $\boldsymbol{\alpha} \to Z(\boldsymbol{\alpha})$ so that L cannot be computed. With noise-contrastive estimation, we can obtain an estimate $\hat{L}$,

$$\hat{L} = \frac{1}{T_v}\sum_{t=1}^{T_v} \ln p_m^0(\mathbf{x}_t;\hat{\boldsymbol{\alpha}}) + \hat{c}, \tag{13}$$

by using $\hat{c}$ in lieu of $\ln 1/Z(\hat{\boldsymbol{\alpha}})$, see Section 2.2. Figure 2(a) shows that the estimated D_{KL} is sometimes negative which means that $\hat{L}$ is sometimes larger than the rescaled log-likelihood of



(a) Precision matrix

(b) Normalizing parameter

Figure 1: Validation of the theory of noise-contrastive estimation: Estimation errors for a 5 dimensional Gaussian distribution. Figures (a) and (b) show the mean squared error for the precision matrix $\mathbf{\Lambda}$ and the normalizing parameter c, respectively. The performance of noise-contrastive estimation (NCE) approaches the performance of maximum likelihood estimation (MLE, black triangles) as the ratio $\nu = T_n/T_d$ increases: the case of $\nu = 1$ is shown with blue circles, $\nu = 5$ with green squares, and $\nu = 10$ with red diamonds. The thicker curves are the median of the performance for 500 random precision matrices with condition number smaller than ten. The finer curves show the 0.9 and 0.1 quantiles of the logarithm of the squared estimation error.

the true distribution. This happens because $\hat{c}$ can be an over or underestimate of $\ln 1/Z(\hat{\boldsymbol{\alpha}})$. This result follows from Figure 2(b) where we have computed D_{KL} with the analytical expression for $\ln 1/Z(\hat{\boldsymbol{\alpha}})$, which is available for the Gaussian model considered here, see Equation (12).

3.2 Distribution of the Estimation Error

We validate and illustrate further properties of our estimator using the ICA model (see for example Hyvärinen et al., 2001b)

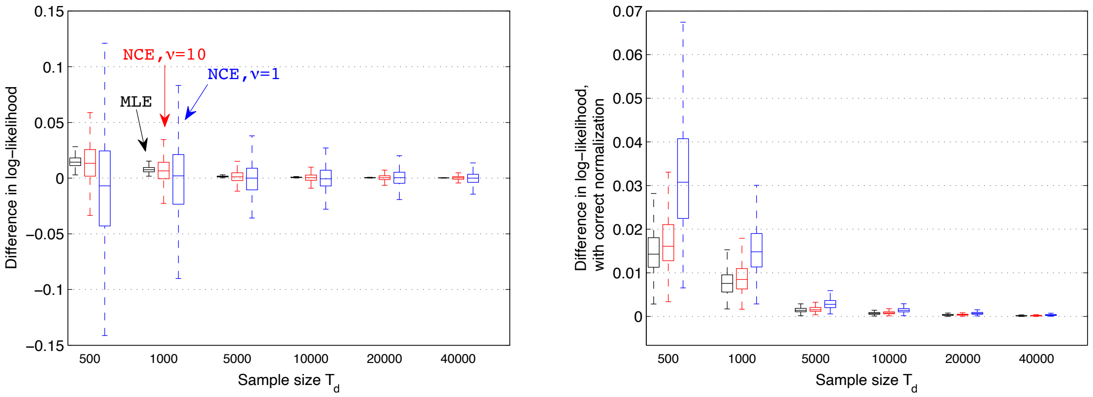
$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{14}$$

In this subsection, $n = 4$, that is $\mathbf{x} \in \mathbb{R}^4$, and $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_4)$ is a 4×4 mixing matrix. The sources in the vector $\mathbf{s} \in \mathbb{R}^4$ are identically distributed and independent from each other so that the data log-pdf $\ln p_d$ is

$$\ln p_d(\mathbf{x}) = \sum_{i=1}^{n} f(\mathbf{b}_i^{\star}\mathbf{x}) + c^{\star}. \tag{15}$$

The i-th row of the matrix $\mathbf{B}^{\star} = \mathbf{A}^{-1}$ is denoted by $\mathbf{b}_i^{\star}$. We consider here Laplacian sources of unit variance and zero mean. The nonlinearity f and the constant $c^{\star}$, which normalizes p_d to integrate to one, are in this case given by

$$f(u) = -\sqrt{2}|u|, \qquad\qquad c^{\star} = \ln|\det \mathbf{B}^{\star}| - \frac{n}{2}\ln 2. \tag{16}$$



(a) Estimated normalization

(b) Analytical normalization

Figure 2: Validation of the theory of noise-contrastive estimation: Distributions of the Kullback-Leibler divergences between the true and estimated 5 dimensional Gaussians. For each sample size, from left to right, the results for maximum likelihood estimation (MLE) are shown in black, the results for noise-contrastive estimation (NCE) with $\nu = 10$ in red, and the results for $\nu = 1$ in blue. The size T_v of the validation set was 100000. For MLE, the results shown in Figures (a) and (b) are the same. For NCE, the divergences in Figure (a) were computed using the estimate $\hat{c}$ of $\ln 1/Z(\hat{\boldsymbol{\alpha}})$. In Figure (b), the analytical expression for $\ln 1/Z(\hat{\boldsymbol{\alpha}})$ was used.

As in Section 3.1, we apply noise-contrastive estimation to the hypothetical situation where we want to estimate the unnormalized model

$$\ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) = \sum_{i=1}^{n} f(\mathbf{b}_i \mathbf{x}) \tag{17}$$

without knowing how to normalize it in closed form. The parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^{16}$ contains the elements of the row vectors $\mathbf{b}_i$. For noise-contrastive estimation, we add an additional normalizing parameter c and estimate the model

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) + c,$$

with $\boldsymbol{\theta} = (\boldsymbol{\alpha}, c)$. As for the Gaussian case, we estimate $\boldsymbol{\theta}$ by maximizing $J_T(\boldsymbol{\theta})$ in Equation (8) with the nonlinear conjugate gradient algorithm of Rasmussen (2006). For the noise distribution p_n, we used a Gaussian distribution with covariance matrix given by the sample covariance of the data.

3.2.1 RESULTS

In Figures 3 and 4, we illustrate Theorem 2 on consistency and Theorem 3 on the asymptotic distribution of the estimator, as well as its corollaries. The results are averages over 500 random estimation problems. The mixing matrices $\mathbf{A}$ were drawn at random by drawing their elements independently from a standard Gaussian and only accepting matrices which had a condition number smaller than ten.

Figure 3(a) and (b) show the mean squared error (MSE) for $\boldsymbol{\alpha}$, corresponding to the mixing matrix, and the normalizing parameter c, respectively. As illustrated for the Gaussian case in Figure 1, this figure visualizes the consistency of noise-contrastive estimation. Furthermore, we see again that making $\nu = T_n/T_d$ larger leads to a reduction of the error. The reduction gets, however, smaller as ν increases. On average, changing ν from one (red curve with asterisks as markers) to ten (light blue squares) reduces the MSE for the mixing matrix by 53%; relative to $\nu = 10$, $\nu = 100$ (magenta diamonds) leads to a reduction of 18%. For c, the relative decrease in the MSE is 60% and 17%, respectively.

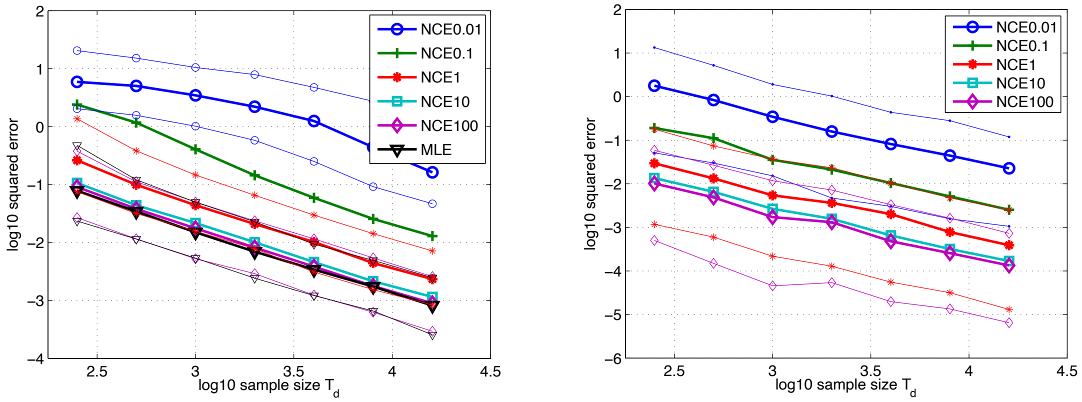
In Figure 4(a), we test the theoretical prediction of Corollary 4 that, for large samples sizes T_d, the MSE decays like $\operatorname{tr}\boldsymbol{\Sigma}/T_d$. The covariance matrix $\boldsymbol{\Sigma}$ can be numerically evaluated according to its definition in Theorem 3.[4] This allows for a prediction of the MSE that can be compared to the MSE obtained in the simulations. The figure shows that the MSE from the simulations (labelled "sim" in the figure) matches the prediction ("pred") for large T_d. Furthermore, we see again that for large ν, the performance of noise-contrastive estimation is close to the performance of MLE. In other words, the trace of $\boldsymbol{\Sigma}$ is close to the trace of the Fisher information matrix. Note that for clarity, we only show the curves for $\nu \in \{0.1, 1, 100\}$. The curve for $\nu = 10$ was, as in Figure 3(a) and (b), very close to the curve for $\nu = 100$.

In Figure 4(b), we investigate how the value of $\operatorname{tr}\boldsymbol{\Sigma}$ (the asymptotic variance) depends on the ratio ν. Note that the covariance matrix $\boldsymbol{\Sigma}$ includes terms related to the parameter c. The Fisher information matrix includes, in contrast to $\boldsymbol{\Sigma}$, only terms related to the mixing matrix. For better comparison with MLE, we show thus in the figure the trace of $\boldsymbol{\Sigma}$ both with the contribution of the normalizing parameter c (blue squares) and without (red circles). For the latter case, the reduced trace of $\boldsymbol{\Sigma}$, which we will denote by $\operatorname{tr}\boldsymbol{\Sigma}_B$, approaches the trace of the Fisher information matrix. Corollary 6 stated that noise-contrastive estimation is asymptotically Fisher-efficient for large values of ν if the normalizing constant is not estimated. Here, we see that this result also approximately holds for our unnormalized model where the normalizing constant needs to be estimated.

Figure 4(c) gives further details to which extent the estimation becomes more difficult if the model is unnormalized. We computed numerically the asymptotic variance $\operatorname{tr}\tilde{\boldsymbol{\Sigma}}$ if the model is correctly normalized, and compared it to the asymptotic variance $\operatorname{tr}\boldsymbol{\Sigma}_B$ for the unnormalized model. The figure shows the distribution of the ratio $\operatorname{tr}\boldsymbol{\Sigma}_B/\operatorname{tr}\tilde{\boldsymbol{\Sigma}}$ for different values of ν. Interestingly, the ratio is almost equal to one for all tested values of ν. Hence, additional estimation of the normalizing constant does not really seem to have had a negative effect on the accuracy of the estimates for the mixing matrix.

In Corollary 7, we have considered the hypothetical case where the noise distribution p_n is the same as the data distribution p_d. In Figure 4(d), we plot for that situation the asymptotic variance as a function of ν (green curve). For reference, we plot again the curve for Gaussian contrastive noise (red circles, same as in Figure 4(b)). In both cases, we only show the asymptotic variance $\operatorname{tr}\boldsymbol{\Sigma}_B$ for the parameters that correspond to the mixing matrix. The asymptotic variance for $p_n = p_d$ is, for a given value of ν, always smaller than the asymptotic variance for the case where the noise is Gaussian. However, by choosing ν large enough for the case of Gaussian noise, it is possible to get estimates which are as accurate as those obtained in the hypothetical situation where $p_n = p_d$. Moreover, for larger ν, the performance is the same for both cases: both converge to the performance of MLE.

[4] See Appendix B.1 for the calculations in the special case of orthogonal mixing matrices.



(a) Mixing matrix

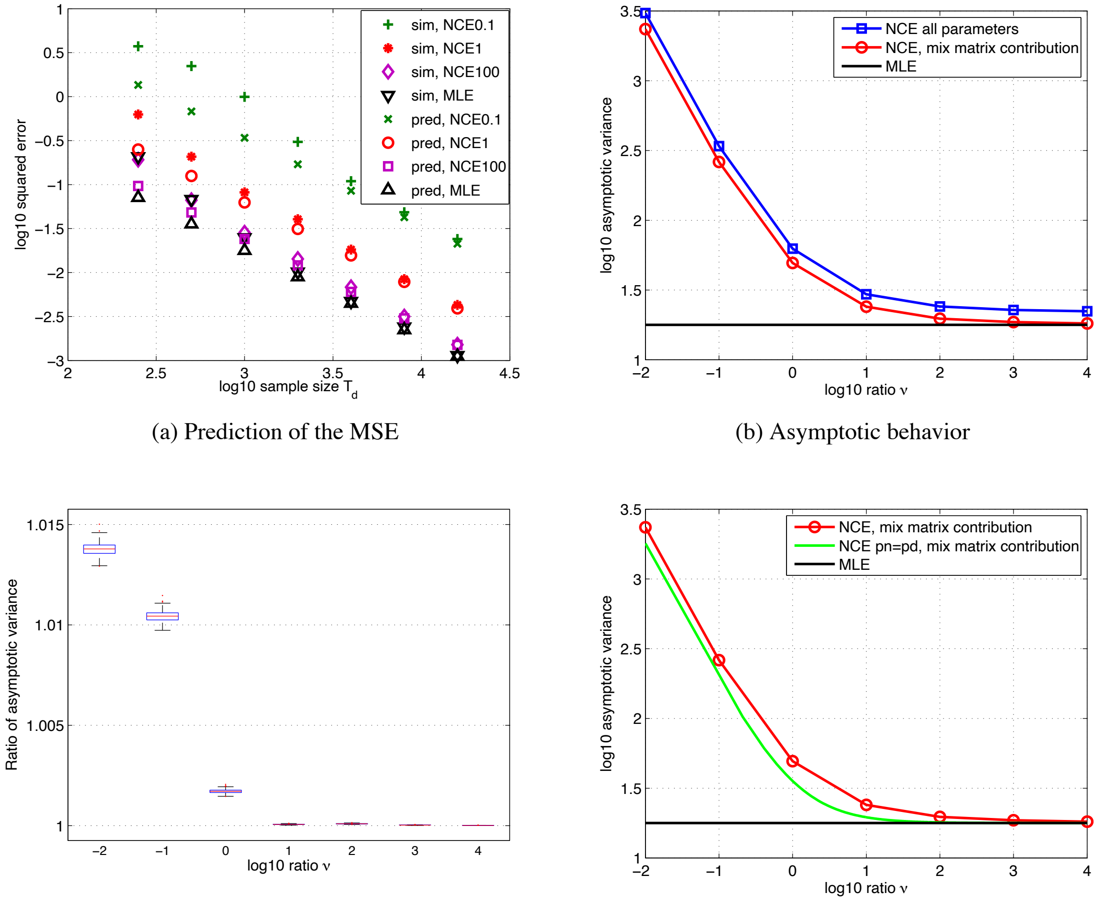
(b) Normalizing parameter

Figure 3: Validation of the theory of noise-contrastive estimation: Estimation errors for an ICA model with four sources. Figures (a) and (b) show the mean squared error for the mixing matrix $\mathbf{B}$ and the normalizing parameter c, respectively. The performance of noise-contrastive estimation (NCE) approaches the performance of maximum likelihood estimation (MLE, black triangles) as the ratio $\nu = T_n/T_d$ increases: the case of $\nu = 0.01$ is shown with blue circles, $\nu = 0.1$ with green crosses, $\nu = 1$ with red asterisks, $\nu = 10$ with light blue squares, and $\nu = 100$ with magenta diamonds. The thicker curves are the median of the performance for 500 random precision matrices with condition number smaller than ten. The finer curves show the 0.9 and 0.1 quantiles of the logarithm of the squared estimation error. To increase readability of the plots, the quantiles for $\nu = 0.1$ and $\nu = 10$ are not shown.

3.3 Scaling Properties

We use the ICA model from the previous subsection to study the behavior of noise-contrastive estimation as the dimension n of the data increases. As before, we estimate the parameters by maximizing $J_T(\boldsymbol{\theta})$ in Equation (8) with the nonlinear conjugate gradient algorithm of Rasmussen (2006). Again, we use a Gaussian with the same covariance structure as the data as noise distribution p_n.

The randomly chosen $n \times n$ mixing matrices $\mathbf{A}$ are restricted to be orthogonal. Orthogonality is only used to set up the estimation problem; in the estimation, the orthogonality property is not used. A reason for this restriction is that drawing mixing matrices at random as in the previous subsection leads more and more often to badly conditioned matrices as the dimension increases. Another reason is that the estimation error for orthogonal mixing matrices depends only on the dimension n and not on the particular mixing matrix chosen, see Appendix B.1 for a proof. Hence, this restriction allows us to isolate the effect of dimension n on the estimation accuracy.



(a) Prediction of the MSE

(b) Asymptotic behavior

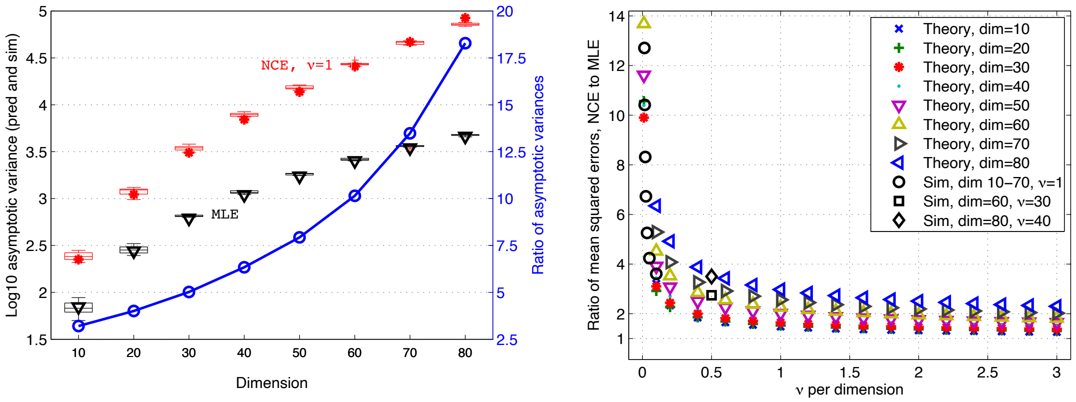
(c) Normalized vs unnormalized model

(d) Effect of the noise distribution

Figure 4: Validation of the theory of noise-contrastive estimation: Estimation error for large sample sizes. Figure (a) shows that Corollary 4 correctly predicts the MSE for large samples sizes T_d. Figure (b) shows the asymptotic variance $\operatorname{tr} \mathbf{\Sigma}$ as a function of ν. Figure (c) shows a boxplot of the ratio between the asymptotic variance when the model is unnormalized and the asymptotic variance when the model is normalized. Figure (d) compares noise-contrastive estimation with Gaussian noise to the hypothetical case where p_n equals the data distribution p_d. As in Figure 3, the curves in all figures but in Figure (c) are the median of the results for 500 random mixing matrices. The boxplot in Figure (c) shows the distribution for all the 500 matrices.

3.3.1 RESULTS

Figure 5(a) shows the asymptotic variance $\operatorname{tr} \mathbf{\Sigma}_B$ related to the mixing matrix as a function of the dimension n. Noise-contrastive estimation (NCE) with $\nu = T_n/T_d$ =1 is shown in red with asterisks as markers, maximum likelihood estimation (MLE) in black using triangles as markers. The markers show the theoretical prediction based on Corollary 4; the boxplots the simulation results for ten



(a) NCE and MLE in higher dimensions

(b) Relative estimation error and amount of noise per dimension

Figure 5: Investigating how noise-contrastive estimation (NCE) scales with the dimension of the data. Figure (a) shows the logarithm of the asymptotic variance for NCE ($\nu = T_n/T_d = 1$, in red) and MLE (in black). The boxplots show simulation results; the asterisks and triangles theoretical predictions for NCE and MLE, respectively. The same figure shows the ratio of the two asymptotic variances (blue circles, right scale). Figure (b) plots the ratio of the mean squared errors of the two estimators as a function of ν per dimension n. The value of ν needs to be increased as the dimensions increases; a linear increase leads to acceptable results.

random mixing matrices with $T_d = 80000$. The simulation results match the predictions well, which validates the theory of noise-contrastive estimation in large dimensions.

Since the number of parameters increases with larger n, it is natural that $\operatorname{tr}\mathbf{\Sigma}_B$ increases with n. However, for noise-contrastive estimation, the increase is larger than for MLE. This is more clearly visible by considering the blue curve in Figure 5(a) (circles as markers, scale on the right axis). The curve shows the ratio between the asymptotic variance for noise-contrastive estimation and for MLE. By definition of the asymptotic variance, this ratio is equal to the ratio of the two estimation errors obtained with the two different methods. The ratio does not depend on the number of parameters and the sample size T_d. It is hence a suitable performance indicator to investigate how noise-contrastive estimation scales with the dimension n of the data. The plot shows that for fixed ν, the performance deteriorates as the dimension increases. In order to counteract this decline in performance, the parameter ν needs to be increased as the dimension increases.

Figure 5(b) shows the ratio of the squared errors as a function of ν/n where we varied n from ten to eighty dimensions as in Figure 5(a). Importantly, both theoretical results, where we numerically calculated the asymptotic variances, and simulation results show that for a reasonable performance in comparison to MLE, ν does not need to be increased exponentially as the dimension n increases; a linear increase with, for instance, $\nu \in [n/2\ n]$ suffices to lead to estimation errors of about 2-4 times of those that are obtained by estimating normalized models with MLE.

4. Investigating the Trade-Off between Statistical and Computational Performance

We have seen that for large ratios ν of noise sample size T_n to data sample size T_d, the estimation error for noise-contrastive estimation behaves like the error in MLE. For large ν, however, the computational load becomes also heavier because more noise samples need to be processed. There is thus a trade-off between statistical and computational performance. Such a trade-off exists also in other estimation methods for unnormalized models. In this section, we investigate the trade-off in noise-contrastive estimation, and compare it to the trade-off in Monte Carlo maximum likelihood estimation (Geyer, 1994), contrastive divergence (Hinton, 2002) and persistent contrastive divergence[5] (Younes, 1989; Tieleman, 2008), as well as score matching (Hyvärinen, 2005).

In Section 4.1, we comment on the data which we use in the comparison. In Section 4.2, we review the different estimation methods with focus on the trade-off between statistical and computational performance. In Section 4.3, we point out the limitations of our comparison before presenting the simulation results in Section 4.4.

4.1 Data Used in the Comparison

For the comparison, we use artificial data which follows the ICA model in Equation (14) with the data log-pdf $\ln p_d$ being given by Equation (15). We set the dimension n to ten and use $T_d = 8000$ observations to estimate the parameters. In a first comparison, we assume Laplacian sources in the ICA model. The log-pdf $\ln p_d$ is then specified with Equation (16). Note that this log-pdf has a sharp peak around zero where it is not continuously differentiable. In a second comparison, we use sources that follow the smoother logistic density. The nonlinearity f and the log normalizing constant c^* in Equation (15) are in that case

$$f(u) = -2\ln\cosh\left(\frac{\pi}{2\sqrt{3}}u\right), \qquad\qquad c^* = \ln|\det\mathbf{B}^\star| + n\ln\left(\frac{\pi}{4\sqrt{3}}\right),$$

respectively. We are thus making the comparison for a relatively nonsmooth and smooth density. Both comparisons are based on 100 randomly chosen mixing matrices with condition number smaller than 10.

4.2 Estimation Methods Used in the Comparison

We introduce here briefly the different methods and comment on our implementation and choices of parameters.

4.2.1 NOISE-CONTRASTIVE ESTIMATION

To estimate the parameters, we maximize J_T in Equation (8). We use here a Gaussian noise density p_n with a covariance matrix equal to the sample covariance of the data. As before, J_T is maximized using the nonlinear conjugate gradient method of Rasmussen (2006). To map out the trade-off between statistical and computational performance, we measured the estimation error and the time needed to optimize J_T for $\nu \in \{1,2,5,10,20,50,100,200,400,1000\}$.

[5] Persistent contrastive divergence is also known under the name stochastic MLE.

4.2.2 MONTE CARLO MAXIMUM LIKELIHOOD ESTIMATION

For normalized models, an estimate for the parameters $\boldsymbol{\alpha}$ can be obtained by choosing them such that the probability of the observed data is maximized. This is done by maximization of

$$J_{\text{MLE}}(\boldsymbol{\alpha}) = \frac{1}{T_d}\sum_{t=1}^{T_d} \ln p_m^0(\mathbf{x}_t;\boldsymbol{\alpha}) - \ln Z(\boldsymbol{\alpha}). \tag{18}$$

If no analytical expression for the partition function $Z(\boldsymbol{\alpha})$ is available, importance sampling can be used to numerically approximate $Z(\boldsymbol{\alpha})$ via its definition in Equation (2), that is

$$Z(\boldsymbol{\alpha}) \approx \frac{1}{T_n}\sum_{t=1}^{T_n} \frac{p_m^0(\mathbf{n}_t;\boldsymbol{\alpha})}{p_{\text{IS}}(\mathbf{n}_t)}.$$

The $\mathbf{n}_t$ are independent observations of "noise" with distribution p_{IS}. Note that more sophisticated ways exist to numerically calculate the value of Z at a given $\boldsymbol{\alpha}$ (see for example Robert and Casella, 2004, in particular Chapter 3 and Chapter 4). The simple approach above leads to the objective function $J_{\text{IS}}(\boldsymbol{\alpha})$ known as Monte Carlo maximum likelihood (Geyer, 1994),

$$J_{\text{IS}}(\boldsymbol{\alpha}) = \frac{1}{T_d}\sum_{t=1}^{T_d} \ln p_m^0(\mathbf{x}_t;\boldsymbol{\alpha}) - \ln\left(\frac{1}{T_n}\sum_{t=1}^{T_n} \frac{p_m^0(\mathbf{n}_t;\boldsymbol{\alpha})}{p_{\text{IS}}(\mathbf{n}_t)}\right).$$

We maximized $J_{\text{IS}}(\boldsymbol{\alpha})$ with the nonlinear conjugate gradient algorithm of Rasmussen (2006).

Like in noise-contrastive estimation, there is a trade-off between statistical performance and running time: The larger T_n gets the better the approximation of the log-likelihood. Hence, the estimates become more accurate but the optimization of J_{IS} takes also more time. To map out the trade-off curve, we used the same values of $T_n = \nu T_d$ as in noise-contrastive estimation, and also the same noise distribution, that is $p_{\text{IS}} = p_n$.

4.2.3 CONTRASTIVE DIVERGENCE

If J_{MLE} is maximized with a steepest ascent algorithm, the update rule for $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \mu_k \nabla_{\boldsymbol{\alpha}} J_{\text{MLE}}(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}_k}, \tag{19}$$

where μ_k is the step-size. For the calculation of $\nabla_{\boldsymbol{\alpha}} J_{\text{MLE}}$, the gradient of the log partition function $\ln Z(\boldsymbol{\alpha})$ is needed, see Equation (18). Above, importance sampling was used to evaluate $\ln Z(\boldsymbol{\alpha})$ and its gradient $\nabla_{\boldsymbol{\alpha}} \ln Z(\boldsymbol{\alpha})$. The gradient of the log partition function can, however, also be expressed as

$$\nabla_{\boldsymbol{\alpha}} \ln Z(\boldsymbol{\alpha}) = \frac{\nabla_{\boldsymbol{\alpha}} Z(\boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})} = \int \frac{p_m^0(\mathbf{n};\boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})} \nabla_{\boldsymbol{\alpha}} \ln p_m^0(\mathbf{n};\boldsymbol{\alpha}) \mathrm{d}\mathbf{n}. \tag{20}$$

If we had data $\mathbf{n}_t$ at hand which follows the normalized model density $p_m^0(.;\boldsymbol{\alpha})/Z(\boldsymbol{\alpha})$, the last equation could be evaluated by taking the sample average. The parameter vector $\boldsymbol{\alpha}$ could then be learned based on Equation (19). In general, sampling from the model density is, however, only possible by means of Markov chain Monte Carlo methods. In contrastive divergence (Hinton, 2002), to compute $\boldsymbol{\alpha}_{k+1}$, Markov chains are started at the data points $\mathbf{x}_t$ and stopped after a few Monte Carlo steps before they actually reach the stationary distribution $p_m^0(.;\boldsymbol{\alpha}_k)/Z(\boldsymbol{\alpha}_k)$. The data points

$\mathbf{n}_t$ that are created in that way follow thus only approximately $p_m^0(.;\boldsymbol{\alpha}_k)/Z(\boldsymbol{\alpha}_k)$. For every update of $\boldsymbol{\alpha}$ the Markov chains are restarted from the $\mathbf{x}_t$. Note that this update rule for $\boldsymbol{\alpha}$ is not directly optimizing a known objective function.

In our implementation, we used Hamiltonian Monte Carlo (see for example Neal, 2010) with a rejection ratio of 10% for the sampling (like in Teh et al., 2004; Ranzato and Hinton, 2010). There are then four tuning parameters for contrastive divergence: The number of Monte Carlo steps, the number of "leapfrog" steps in Hamiltonian Monte Carlo, the choice of the step sizes μ_k, as well as the number of data points $\mathbf{x}_t$ and noise points $\mathbf{n}_t$ used in each update step of $\boldsymbol{\alpha}$. The choice of the tuning parameters will affect the estimation error and the computation time. For our comparison here, we used contrastive divergence with one and three Monte Carlo steps (denoted by CD1 and CD3 in the figures below), together with either three or twenty leapfrog steps. Ranzato and Hinton (2010) used CD1 with twenty leapfrog steps (below denoted by CD1 20), while Teh et al. (2004) used CD1 30 to estimate unnormalized models from natural image data. For the μ_k, we considered constant step sizes, as well as linearly and exponentially decaying step sizes.[6] For each update step, we chose an equal number of data and noise points. We considered the case of using all data in each update step, and the case of using minibatches of only 100 randomly chosen data points.

We selected the step size μ_k and the number of data points used in each update by means of preliminary simulations on five data sets. We limited ourselves to contrastive divergence with one Monte Carlo and three leapfrog steps (CD1 3). For both Laplacian and logistic sources, using minibatches with an exponential decaying step size gave the best results. The results are reported below in Section 4.4. The use of minibatches led to faster estimation results without affecting their accuracy. Exponentially decaying step sizes are advocated by the theory of stochastic approximation; in some cases, however, linear decay was found to be more appropriate (Tieleman, 2008, Section 4.5). For Laplacian sources, the initial step size μ_0 was 0.005; for logistic sources, it was $\mu_0 = 0.01$. Note that in this selection of the tuning parameters, we used the true parameters to compute the estimation error. Clearly, this cannot be done in real applications since the true parameter values are not known. The choice of the tuning parameters must then solely be based on experience, as well as trial and error.

4.2.4 PERSISTENT CONTRASTIVE DIVERGENCE

As contrastive divergence, persistent contrastive divergence (Younes, 1989; Tieleman, 2008) uses the update rule in Equation (19) together with an approximative evaluation of the integral in Equation (20) to learn the parameters $\boldsymbol{\alpha}$. The integral is also computed based on Markov chain Monte Carlo sampling. Unlike contrastive divergence, however, the Markov chains are not restarted at the data points $\mathbf{x}_t$. For the computation of $\boldsymbol{\alpha}_{k+1}$, the Markov chains are initialized with the samples $\mathbf{n}_t$ that were obtained in the previous iteration by running Markov chains converging to $p_m^0(.;\boldsymbol{\alpha}_{k-1})/Z(\boldsymbol{\alpha}_{k-1})$. As in contrastive divergence, the Markov chains are only run for a short time and stopped before having actually converged.

Since persistent contrastive divergence differs from contrastive divergence only by the initialization of the Markov chains, it has the same tuning parameters. As in contrastive divergence, we used preliminary simulations to select suitable parameters: again, exponentially decaying step sizes μ_k together with minibatches of size 100 gave the best performance. The preliminary simulations yielded also the same initial step sizes μ_0 as in contrastive divergence. It turned out, however,

[6] Linear decay: $\mu_k = \mu_0(1 - k/maxIteration)$, exponential decay: $\mu_k = \mu_0 C/(C+k)$ with $C = 5000$.

that the number of leapfrog steps in persistent contrastive divergence needs to be larger than in contrastive divergence: using, for example, only three leapfrog steps as in contrastive divergence resulted in a poor performance in terms of estimation accuracy. For the results reported below in Section 4.4, we used 20 and 40 leapfrog steps, together with one and three Monte Carlo steps.

4.2.5 SCORE MATCHING

In score matching (Hyvärinen, 2005), the parameter vector $\boldsymbol{\alpha}$ is estimated by minimization of the cost function J_{SM},

$$J_{\text{SM}}(\boldsymbol{\alpha}) = \frac{1}{T_d}\sum_{t=1}^{T_d}\sum_{i=1}^{n}\frac{1}{2}\Psi_i^2(\mathbf{x}_t;\boldsymbol{\alpha}) + \Psi_i'(\mathbf{x}_t;\boldsymbol{\alpha}).$$

The term $\Psi_i(\mathbf{x};\boldsymbol{\alpha})$ is the derivative of the unnormalized model with respect to $\mathbf{x}(i)$, the i-th element of the vector $\mathbf{x}$,

$$\Psi_i(\mathbf{x};\boldsymbol{\alpha}) = \frac{\partial \ln p_m^0(\mathbf{x};\boldsymbol{\alpha})}{\partial \mathbf{x}(i)}.$$

The term $\Psi_i'(\mathbf{x};\boldsymbol{\alpha})$ denotes the derivative of $\Psi_i(\mathbf{x};\boldsymbol{\alpha})$ with respect to $\mathbf{x}(i)$. The presence of this derivative may make the objective function and its gradient algebraically rather complicated if a sophisticated model is estimated. For the ICA model with Laplacian sources, $\Psi_i(\mathbf{x};\boldsymbol{\alpha})$ equals

$$\Psi_i(\mathbf{x};\boldsymbol{\alpha}) = \sum_{j=1}^{n} -\sqrt{2}\text{sign}(\mathbf{b}_j\mathbf{x})B_{ji} \tag{21}$$

which is not smooth enough to be used in score matching. Using the smooth approximation $\text{sign}(u) \approx \tanh(10u)$ is a way to obtain a smooth enough $\Psi_i(\mathbf{x};\boldsymbol{\alpha})$ and $\Psi_i'(\mathbf{x};\boldsymbol{\alpha})$. The optimization of J_{SM} is done by the nonlinear conjugate gradient algorithm of Rasmussen (2006). Note that, unlike the estimation methods considered above, score matching does not have a tuning parameter which controls the trade-off between statistical and computational performance. Moreover, score matching does not rely on sampling.

4.3 Limitations of the Comparison

For all considered methods but contrastive and persistent contrastive divergence, the algorithm which is used to optimize the given objectives can be rather freely chosen. This choice will influence the trade-off between statistical and computational performance. Here, we use the optimization algorithm by Rasmussen (2006). Our results below show thus the trade-off of the different estimation methods in combination with this particular optimization algorithm. With this optimization algorithm, we used for each update all data. The algorithm is not suitable for stochastic optimization with minibatches (see for example Schraudolph and Graepel, 2002). Optimization based on minibatches may well lead not only for (persistent) contrastive divergence to gains in speed but also for the other estimation methods, including noise-contrastive estimation.

It is well known that a Gaussian as noise (proposal) distribution is not the optimal choice for importance sampling if the data has heavy tails (see for example Wasserman, 2004, Chapter 24). Gaussian noise is not the optimal choice for noise-contrastive estimation either. The presented results should thus not be considered as a general comparison of the two estimation methods per se. Importantly, however, the chosen setup allows one to assess how noise-contrastive estimation behaves when the data has heavier tails than the noise, which is often the case in practice.

Finally, the reader may want to keep in mind that for other kinds of data, in particular also in very high dimensions, differences may occur.

4.4 Results

We first compare noise-contrastive estimation with the methods for which we use the same optimization algorithm, that is Monte Carlo maximum likelihood estimation and score matching. Then, we compare it with contrastive and persistent contrastive divergence.

4.4.1 COMPARISON WITH MONTE CARLO MLE AND SCORE MATCHING

Figure 6 shows the comparison of noise-contrastive estimation (NCE, red squares), Monte Carlo maximum likelihood (IS, blue circles) and score matching (SM, black triangles). The left panels show the simulation results in form of “result points” where the x-coordinate represents the time till the algorithm converged and the y-coordinate the estimation error at convergence. Convergence in the employed nonlinear conjugate gradient algorithm by Rasmussen (2006) means that the line search procedure failed twice in a row to meet the strong Wolfe-Powell conditions (see for example Sun and Yuan, 2006, Chapter 2.5.2). For score matching, 100 result points corresponding to 100 different random mixing matrices are shown in each figure. For noise-contrastive estimation and Monte Carlo maximum likelihood, we used ten different values of ν so that for these methods, each figure shows 1000 result points. The panels on the right present the simulation result in a more schematic way. For noise-contrastive estimation and Monte Carlo maximum likelihood, the different ellipses represent the outcomes for different values of ν. Each ellipse contains 90% of the result points. We can see that increasing ν reduces the estimation error but it also increases the running time. For score matching, there is no such trade-off.

Figure 6(a) shows that for Laplacian sources, noise-contrastive estimation outperforms the other methods in terms of the trade-off between statistical and computational performance. The large estimation error of score matching is likely to be due to the smooth approximation of the sign function in Equation (21). The figure also shows that noise-contrastive estimation handles noise that has lighter tails than the data more gracefully than Monte Carlo maximum likelihood estimation. The reason is that the nonlinearity $h(\mathbf{u};\boldsymbol{\theta})$ in the objective function in Equation (8) is bounded even if data and noise distribution do not match well (see also Pihlaja et al., 2010).

For logistic sources, shown in Figure 6(b), noise-contrastive estimation and Monte Carlo maximum likelihood perform equally. Score matching reaches its level of accuracy about 20 times faster than the other methods. Noise-contrastive estimation and Monte Carlo maximum likelihood can, however, have a higher estimation accuracy than score matching if ν is large enough. Score matching can thus be considered to have a built-in trade-off between estimation performance and computation time: Computations are fast but the speed comes at the cost of not being able to reach an estimation accuracy as high as, for instance, noise-contrastive estimation.

4.4.2 COMPARISON WITH CONTRASTIVE AND PERSISTENT CONTRASTIVE DIVERGENCE

Since contrastive and persistent contrastive divergence do not have an objective function and given the randomness that is introduced by the minibatches, it is difficult to choose a reliable stopping criterion. Hence, we did not impose any stopping criterion but the maximal number of iterations. The two algorithms had always converged before this maximal number of iterations was reached in the sense that the estimation error did not visibly decrease any more.

We base our comparison on the estimation error as a function of the running time of the algorithm. This makes the comparison independent from the stopping criterion that is used in noise-contrastive estimation. For noise-contrastive estimation, the parameter ν controls the trade-off between computational and statistical performance; for contrastive and persistent contrastive divergence, it is the number of leapfrog steps and the number of Markov steps taken in each update. We compiled a trade-off curve for each of the one hundred estimation problems by taking at any time point the minimum estimation error over the various estimation errors that are obtained for different values of the trade-off parameters.[7] Figure 7 shows an example for noise-contrastive estimation and contrastive divergence. The distribution of the trade-off curves is shown in Figure 8. For large running times, the distribution of the estimation error is for all estimation methods similar to the one for maximum likelihood estimation. For shorter running times, noise-contrastive estimation is seen to have for Laplacian sources a better trade-off than the other methods. For logistic sources, however, the situation is reversed.

4.4.3 SUMMARY

The foregoing simulation results and discussion suggest that all estimation methods trade, in one form or the other, estimation accuracy against computation speed. In terms of this trade-off, noise-contrastive estimation is particularly well suited for the estimation of data distributions with heavy tails. In case of thin tails, noise-contrastive estimation performs similarly to Monte Carlo maximum likelihood, and contrastive or persistent contrastive divergence has a better trade-off. If the data distribution is particularly smooth and the model algebraically not too complicated, score matching may, depending on the required estimation accuracy, be the best option.

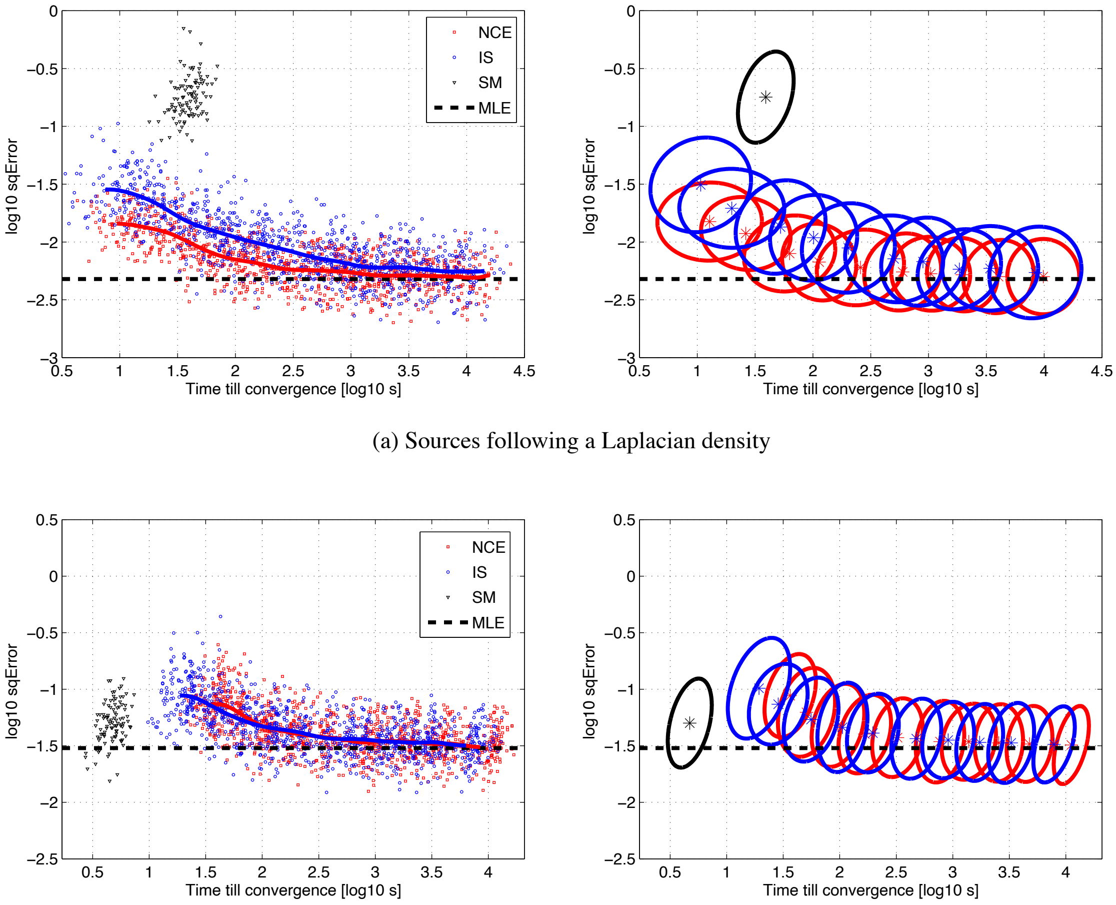
5. Simulations with Natural Images

In this section, we estimate with our new estimation method models of natural images. In the theory of noise-contrastive estimation, we have assumed that all variables can be observed. Noise-contrastive estimation can thus not be used for models with latent variables which cannot be integrated out analytically. Such models occur for example in the work by Olshausen and Field (1996), Hyvärinen et al. (2001a), Karklin and Lewicki (2005), Lücke and Sahani (2008) and Osindero and Hinton (2008). We are here considering models which avoid latent variables. Recent models which are related to the models that we are considering here can be found in the work by Osindero et al. (2006), Köster and Hyvärinen (2010) and Ranzato and Hinton (2010). For a comprehensive introduction to natural image statistics, see for example the textbook by Hyvärinen et al. (2009).

The presented models will consist of two processing layers, like in a multilayer neural network. The output of the network for a given input image gives the value of the model-pdf at that image. Because of the two processing layers, we call the models “two-layer models”.

We start with giving some preliminaries in Section 5.1. In Section 5.2, we present the settings of noise-contrastive estimation. In Section 5.3, we properly define the two-layer model and estimate a version with more than 50000 parameters. In Section 5.4, we present an extension of the model where the learned output nonlinearity of the network belongs to the flexible family of splines. The different models are compared in Section 5.5.

[7] A comparison of CD and PCD for different settings can be found in Appendix C.1.



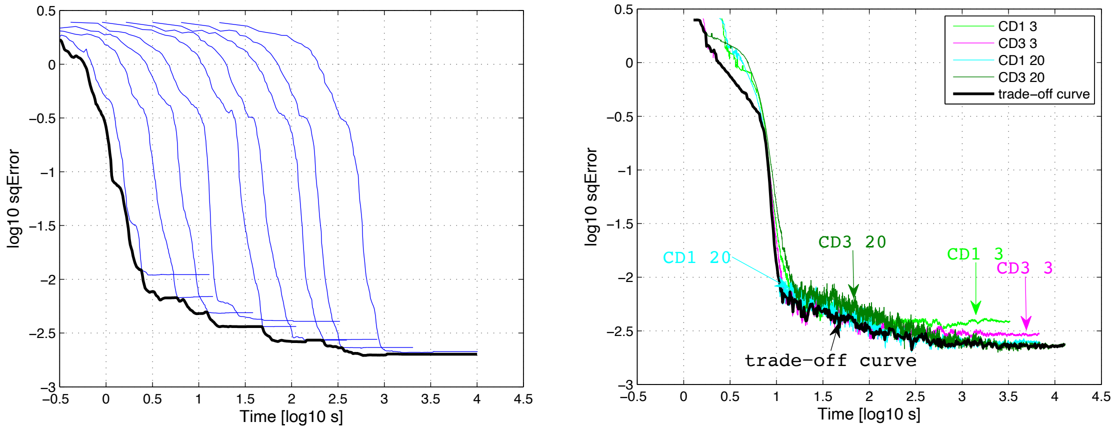
(a) Sources following a Laplacian density

(b) Sources following a logistic density

Figure 6: Trade-off between statistical and computational performance for noise-contrastive estimation (NCE, red squares), Monte Carlo maximum likelihood (IS, blue circles) and score matching (SM, black triangles). Each point represents the result of one simulation. Performing local linear kernel smoothing regression on the result points yields the thick curves. For noise-contrastive estimation and Monte Carlo maximum likelihood, the ten ellipses represent the outcomes for the ten different values of $\nu \in \{1,2,5,10,20,50,100,200,400,1000\}$. The ellipses were obtained by fitting a Gaussian to the distribution of the result points, each one contains 90% of the results points for a given ν. The asterisks mark their center. For an ICA model with Laplacian sources, NCE has the best trade-off between statistical and computational performance. For logistic sources, NCE and IS perform equally well. For medium estimation accuracy, score matching outperforms the other two estimation methods.

5.1 Data, Preprocessing and Modeling Goal

Our basic data are a random sample of 25px × 25px image patches that we extracted from a subset of van Hateren's image database (van Hateren and van der Schaaf, 1998). The images in the subset showed wildlife scenes only. The sample size T_d is 160000.



(a) Noise-contrastive estimation

(b) Contrastive divergence

Figure 7: Example of a trade-off curve for noise-contrastive estimation and contrastive divergence. (a) The different curves in blue show the estimation error which is obtained for the various values of ν. The thicker curve in black shows the trade-off curve. It is is obtained by taking at any time point the minimum estimation error. (b) The trade-off curve, shown in black, is similarly obtained by taking the minimum over the estimation errors which are obtained with different settings of contrastive divergence.

As preprocessing, we removed from each image patch its average value (local mean, DC component), whitened the data and reduced the dimension from $d = 25 \cdot 25 = 625$ to $n = 160$. This retains 93% of the variance of the image patches. After dimension reduction, we additionally centered each data point and rescaled it to unit variance. In order to avoid division by small numbers, we avoided taking small variance patches. This gave our data $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$. Because of the centering and rescaling, each data point $\mathbf{x}_t$ satisfies

$$\sum_{k=1}^{n} \mathbf{x}_t(k) = 0, \qquad \frac{1}{n-1} \sum_{k=1}^{n} \mathbf{x}_t(k)^2 = 1. \tag{22}$$

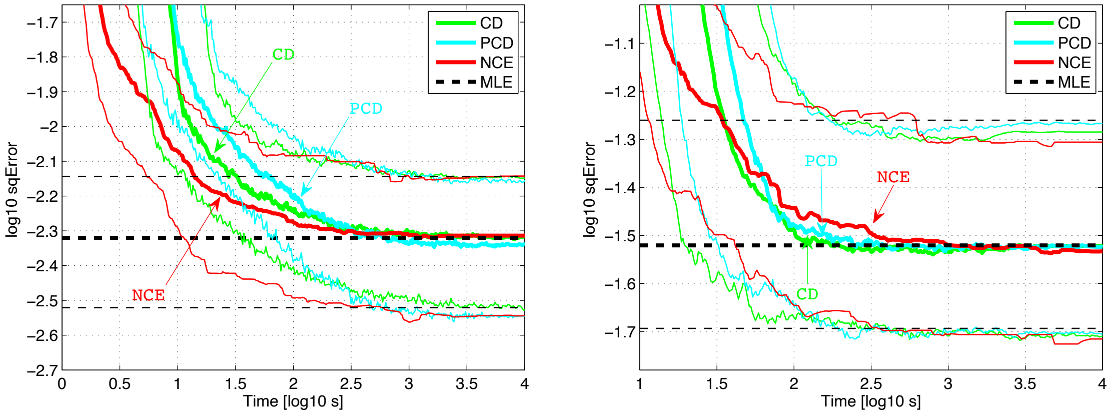
This means that each data point lies on the surface of a $n-1$ dimensional sphere $\mathbb{S}$.

This kind of preprocessing is a form of luminance and contrast gain control which aim at canceling out the effects of the lighting conditions (see for example Hyvärinen et al., 2009, Chapter 9, where also the statistical effects of such a preprocessing are analyzed). Centering and rescaling to unit variance has also been used in image quality assessment in order to access the structural component of an image, which is related to the reflectance of the depicted objects (Wang et al., 2004, in particular Section III.B). By modeling the data X, we are thus modeling the structure in the image patches.

Given a data point $\mathbf{x}_t$, we can reconstruct the original (vectorized) image patch via

$$\mathbf{i}_t = \mathbf{V}^{-}\mathbf{x}_t, \qquad \mathbf{V}^{-} = \mathbf{E}\mathbf{D}^{1/2}, \tag{23}$$

where $\mathbf{E}$ is the $d \times n$ matrix formed by the leading n eigenvectors of the covariance matrix of the image patches. The diagonal $n \times n$ matrix $\mathbf{D}$ contains the corresponding eigenvalues. The matrix



(a) Laplacian sources

(b) Logistic sources

Figure 8: Distribution of the trade-off curves for contrastive divergence (CD, green), persistent contrastive divergence (PCD, cyan), and noise-contrastive estimation (NCE, red). The distribution of the estimation error for maximum likelihood estimation is shown in black. The thick curves show the median, the finer curves the 0.9 and 0.1 quantiles.

$\mathbf{V}^-$ defined above is the pseudoinverse of the whitening matrix $\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^T$. Since the column vectors of $\mathbf{V}^-$ form a basis for a n dimensional subspace of $\mathbb{R}^d$, $\mathbf{x}$ is the coordinate vector of $\mathbf{i}$ with respect to that basis. The dimension reduction implies that the reconstruction cannot be perfect; the reconstruction can also only performed up to the scale and average value of the patch because of the the luminance and contrast gain control. Figure 9(a) shows examples of natural image patches after extraction from the data base; Figure 9(b) shows the corresponding reconstructions $\mathbf{i}$. Since all image patches in Figure 9 were rescaled to use the full colormap, the effects of luminance and contrast gain control are not visible. The effect of the dimension reduction is low-pass filtering.

5.2 Settings for Noise-Contrastive Estimation

Matlab code for the simulations is available from the authors' homepage so that our description here will not be exhaustive. All the models considered in the next subsections are estimated with noise-contrastive estimation. We learn the parameters by optimization of the objective J_T in Equation (8). The two-layer models are estimated by first estimating one-layer models. The learned parameters are used as initial values for the first layer in the estimation of the complete two-layer model. The second layer is initialized to small random values.

For the contrastive noise distribution p_n, we take a uniform distribution on the surface of the $n-1$ dimensional sphere $\mathbb{S}$ on which $\mathbf{x}$ is defined.[8] Examples of image patches with coordinates following p_n are shown in Figure 9(c). Samples from p_n can easily be created by sampling from a standard normal distribution, followed by centering and rescaling such that Equation (22) holds. Since p_n is a constant, the log-ratio $G(.;\boldsymbol{\theta})$ in Equation (4) is up to an additive constant equal to

[8] $\ln p_n = -\ln(2) - \frac{n-1}{2}\ln(\pi) - (n-2)\ln(r) + \ln\Gamma\left(\frac{n-1}{2}\right)$ with $r = \sqrt{n-1}$.

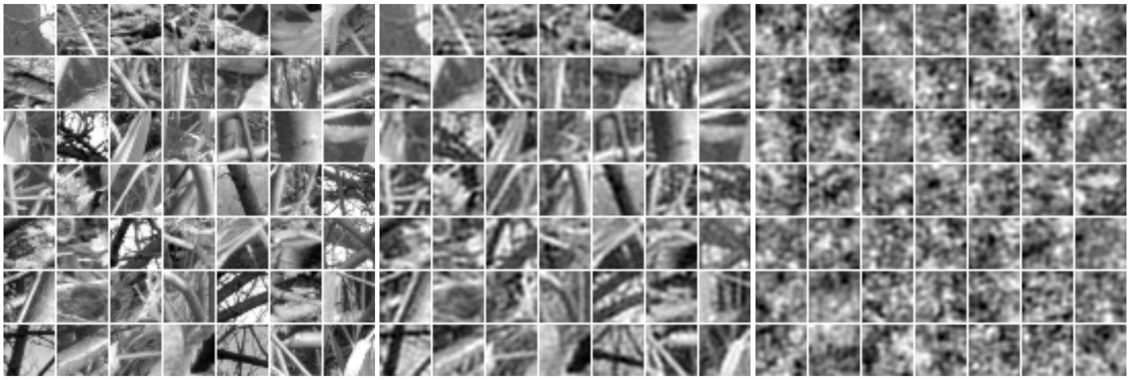

(a) Image patches (b) Reconstructions (c) Noise

Figure 9: (a) Natural image patches of size $25\text{px} \times 25\text{px}$. (b) Reconstructed image patches after preprocessing. These are examples of the image patches denoted by $\mathbf{i}$ in Equation (23) with coordinate vectors $\mathbf{x} \in \mathbb{R}^{160}$. (c) Noise images which are obtained via Equation (23) if the coordinates are uniformly distributed on the sphere $\mathbb{S}$. Comparison with Figure (b) shows that the coordinate vectors $\mathbf{x}$ for natural images are clearly not uniformly distributed on the sphere. In the next subsections, we model their distribution.

$\ln p_m(.;\boldsymbol{\theta})$,

$$G(.;\boldsymbol{\theta}) = \ln p_m(.;\boldsymbol{\theta}) + \text{constant}.$$

As pointed out in Section 2.2, $\boldsymbol{\theta}$ evolves in the maximization of J_T such that $G(\mathbf{u};\hat{\boldsymbol{\theta}}_T)$ is as large as possible for $\mathbf{u} \in X$ (natural images) but as small as possible for $\mathbf{u} \in Y$ (noise). For uniform noise, the same must thus also hold for $\ln p_m(\mathbf{u};\hat{\boldsymbol{\theta}}_T)$. This observation will be a useful guiding tool for the interpretation of the models below.

The factor $\nu = T_n/T_d$ was set to 10. We found that an iterative optimization procedure where we separate the data into subsets and optimize J_T for increasingly larger values of ν reduced computation time. The optimization for each ν is done with the nonlinear conjugate gradient method of Rasmussen (2006). The size of the subsets is rather large, for example 80000 in the simulation of the next subsection.[9] A more detailed discussion of this optimization procedure can be found in Appendix C.2.

5.3 Two-Layer Model with Thresholding Nonlinearities

The first model that we consider is

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k; a_k, b_k) + c, \qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2, \tag{24}$$

where f is a smooth, compressive thresholding function that is parameterized by a_k and b_k. See Figure 10 for details regarding the parameterization and the formula for f. The parameters $\boldsymbol{\theta}$ of

[9] As pointed out in Section 4.3, the used nonlinear conjugate gradient algorithm is not suitable for stochastic optimization with small minibatches.

the model are the second-layer weights $Q_{ki} \geq 0$, the first-layer weights $\mathbf{w}_i \in \mathbb{R}^n$, the normalizing parameter $c \in \mathbb{R}$, as well as $a_k > 0$ and $b_k \in \mathbb{R}$ for the nonlinearity f. The definition of y_k shows that multiplying Q_{ki} by a factor γ_i^2 and $\mathbf{w}_i$ at the same time by the factor $1/\gamma_i$ does not change the value of y_k. There is thus some ambiguity in the parameterization which could be resolved by imposing a norm constraint either on the $\mathbf{w}_i$ or on the columns of the matrix $\mathbf{Q}$ formed by the weights Q_{ki}. It turned out that for the estimation of the model such constraints were not necessary. For the visualization and interpretation of the results, we chose γ_i such that all the $\mathbf{w}_i$ had norm one.

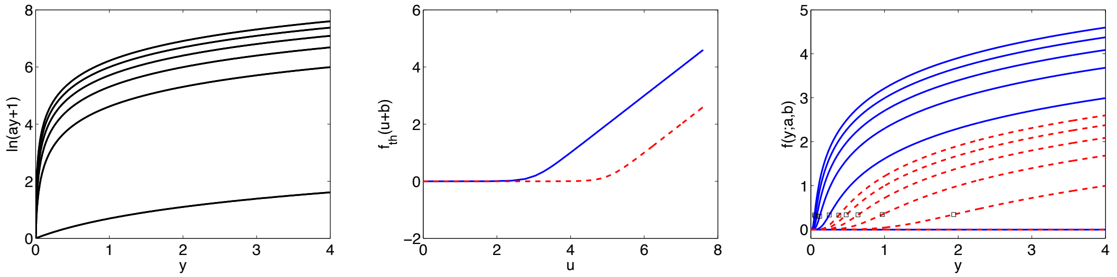
The motivation for the thresholding property of f is that, in line with Section 5.2, $\ln p_m(.;\boldsymbol{\theta})$ can easily be made large for natural images and small for noise. The y_k must just be above the thresholds for natural image input and below for noise. This occurs when the vectors $\mathbf{w}_i$ detect features (regularities) in the input which are specific to natural images, and when, in turn, the second-layer weights Q_{ki} detect characteristic regularities in the squared first-layer feature outputs $\mathbf{w}_i^T\mathbf{x}$. The squaring implements the assumption that the regularities in $\mathbf{x}$ and $(-\mathbf{x})$ are the same so that the pdf of $\mathbf{x}$ should be an even function of the $\mathbf{w}_i^T\mathbf{x}$. Another property of the nonlinearity is its compressive log-like behavior for inputs above the threshold. The motivation for this is to "counteract" the squaring in the computation of y_k. The compression of large values of y_k leads to numerical robustness in the computation of $\ln p_m$.

A model like the one in Equation (24) has been studied before by Osindero et al. (2006) and Köster and Hyvärinen (2010). There are, however, a number of differences. The main difference is that in our case $\mathbf{x}$ lies on a sphere while in the cited work, $\mathbf{x}$ was defined in the whole space $\mathbb{R}^n$. This difference allows us to use nonlinearities that do not decay asymptotically to $-\infty$ which is necessary if $\mathbf{x}$ is defined in $\mathbb{R}^n$. A smaller difference is that we do not need to impose norm constraints to facilitate the learning of the parameters.

5.3.1 RESULTS

For the visualization of the first-layer feature detectors $\mathbf{w}_i$, note that the inner product $\mathbf{w}_i^T\mathbf{x}$ equals $(\mathbf{w}_i^T\mathbf{V})\mathbf{i} = \tilde{\mathbf{w}}_i^T\mathbf{i}$. The $\mathbf{w}_i \in \mathbb{R}^n$ are coordinate vectors with respect to the basis given by the columns of $\mathbf{V}^-$, see Section 5.1, while the $\tilde{\mathbf{w}}_i \in \mathbb{R}^d$ are the coordinate vectors with respect to the pixel basis. The latter vectors can thus be visualized as images. This is done in Figure 11(a). Another way to visualize the first-layer feature detectors $\mathbf{w}_i$ is to show the images which yield the largest feature output while satisfying the constraints in Equation (22). These optimal stimuli are proportional to $\mathbf{V}^-(\mathbf{w}_i - \langle\mathbf{w}_i\rangle)$, where $\langle\mathbf{w}_i\rangle \in \mathbb{R}$ is the average value of the elements in the vector $\mathbf{w}_i$, see Appendix B.2 for a proof. The optimal stimuli are shown in Figure 11(b). Both visualizations show that the first layer computes "Gabor-like" features, which is in line with previous research on natural image statistics.

Figure 12 shows a random selection of the learned second-layer weights Q_{ik}. Figure 12(a) shows that the weights are extremely sparse. The optimization started with the weights being randomly assigned to small values, with the optimization most of them shrank to zero; few selected ones, however, increased in magnitude. Note that this result was obtained without any norm constraints on $\mathbf{Q}$. From Figure 12(b), we see that the learned second-layer weights Q_{ik} are such that they combine first-layer features of similar orientation, which are centered at nearby locations ("complex cells"). The same figure shows also a condensed representation of the feature detectors using icons. This form of visualization is used in Figure 13 to visualize all the second-layer feature detectors.



(a) Compression

(b) Rectification

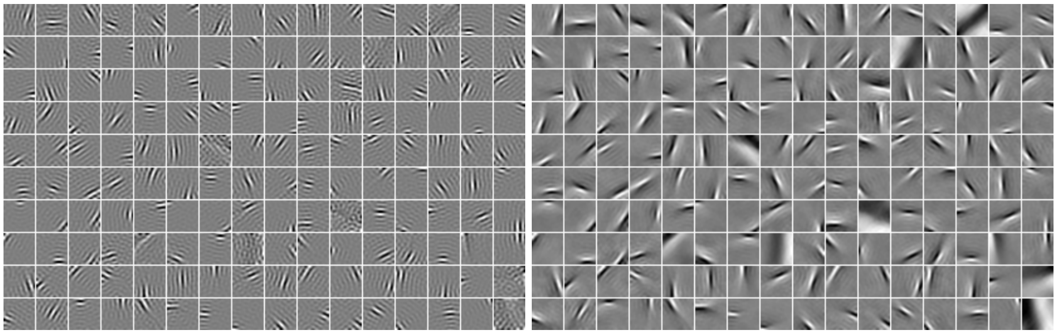
(c) Resulting nonlinearity

Figure 10: Two-layer model with thresholding nonlinearities. The family of nonlinearities used in the modeling is $f(y;a,b) = f_{\text{th}}(\ln(ay+1)+b)$, $y \geq 0$. The parameterized function is composed of a compressive nonlinearity $\ln(ay+1)$, shown in Figure (a), and a smooth rectification function $f_{\text{th}}(u+b)$ shown in Figure (b). Figure (c) shows examples of $f(y;a,b)$ for different values of a and b. Parameter b sets the threshold, and parameter a controls the steepness of the function. Since the scale of the weights in Equation (24) is not restrained, the parameters a_k do not need to be learned explicitly. After learning, they can be identified by dividing y_k in Equation (24) by a_k so that its expectation is one for natural images. The formula for the thresholding function is $f_{\text{th}}(u) = 0.25\ln(\cosh(2u)) + 0.5u + 0.17$. The curves shown in blue are for $b = -3$ and $a \in \{1, 50, 100, 200, \ldots, 500\}$. For the dashed curves in red, $b = -5$. The small squares in Figure (c) indicate where f changes from convex to concave.

Figure 14(a) shows the learned nonlinearities $f(.;a_k,b_k)$. Note that we incorporated the learned normalizing parameter c as an offset c/n for each nonlinearity. The learned thresholding is similar for feature outputs of mid- and high-frequency feature detectors (black, solid curves). For the feature detectors tuned to low frequencies, the thresholds tend to be smaller (green, dashed curves). The nonlinearities in black are convex for arguments y smaller than two (see red rectangle in the figure). That is, they show a squashing behavior for $y < 2$. Looking at the distribution of the second-layer outputs y_k in Figure 14(b), we see that it is more likely that noise rather than natural images was the input when the second-layer feature outputs y_k are approximately between 0.5 and 2. In this regime, the squashing nonlinearities map thus more often the noise input to small values than natural images so that $\ln p_m(\mathbf{u};\hat{\boldsymbol{\theta}}_T)$ tends to be larger when input $\mathbf{u}$ is a natural image than when it is noise (see Section 5.2). One could, however, think that the thresholding nonlinearities are suboptimal because they ignore the fact that natural images lead, compared to the noise, rather often to y_k which are close to zero, see Figure 14(b). An optimal nonlinearity should, unlike the thresholding nonlinearities, assign a large value to both large and small y_k while mapping intermediate values of y_k to small numbers. The next subsection shows that such kinds of mappings emerge naturally when splines are used to learn the nonlinearities from the data.

5.4 Two-Layer Model with Spline Nonlinearities

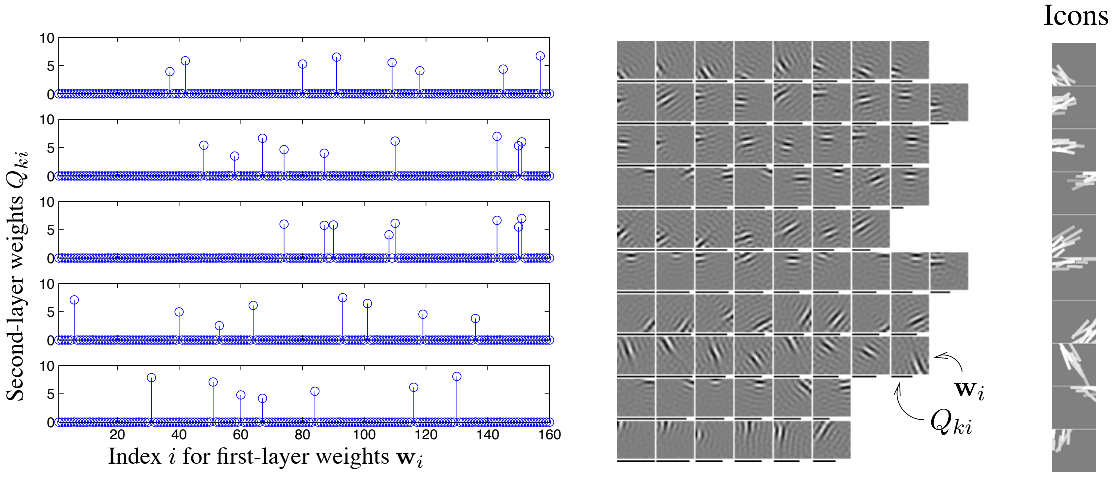
In the previous subsection, the family of nonlinearities f in Equation (24) was rather limited. Here, we look for f in the larger family of cubic splines where we consider the location of the knots to



(a) Feature detectors

(b) Optimal stimuli

Figure 11: Two-layer model with thresholding nonlinearities: Visualization of the learned first-layer feature detectors $\mathbf{w}_i$. (a) The feature detectors in the pixel basis. (b) The corresponding optimal stimuli. The feature detectors in the first layer are "Gabor-like" (localized, oriented, bandpass). Comparison of the two figures shows that feature detectors which appear noisy in the pixel basis are tuned to low-frequency input.



(a) Raw result

(b) Graphical visualization

Figure 12: Two-layer model with thresholding nonlinearities: Random selection of second layer units. (a) Second-layer weights Q_{ki} for five different k (five different rows of the matrix $\mathbf{Q}$) are shown. The weights are extremely sparse so that in the sum $\sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2$ only few selected squared first-layer outputs are added together. (b) Every row shows one second-layer feature detector. The first-layer feature detectors $\mathbf{w}_i$ are shown as image patches like in Figure 11, and the black bar under each patch indicates the strength Q_{ki} by which a certain $\mathbf{w}_i$ is pooled by the k-th second-layer feature detector. The numerical values Q_{ki} for the first five rows are shown in Figure (a). The right-most column shows a condensed visualization. The icons were created by representing each first-layer feature by a bar of the same orientation and similar length as the feature, and then superimposing them with weights given by Q_{ki}.

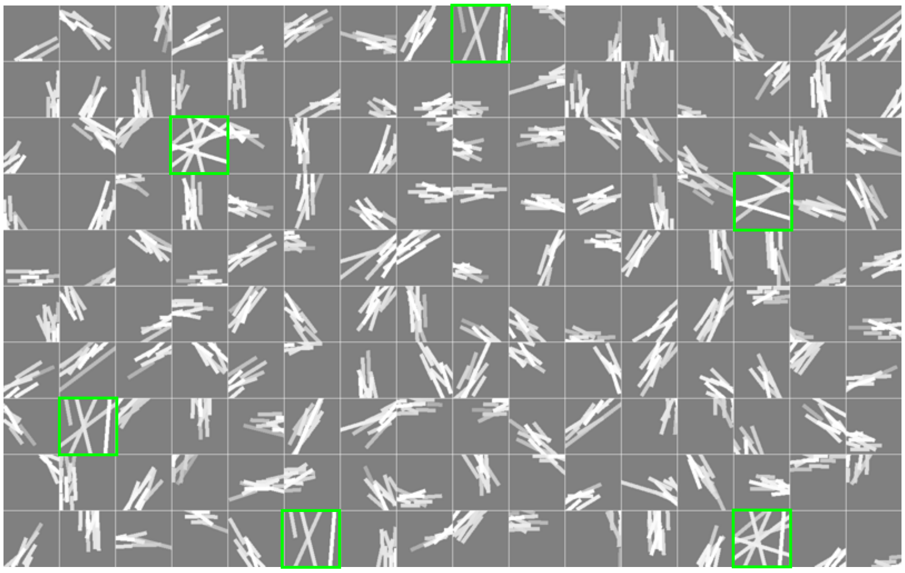
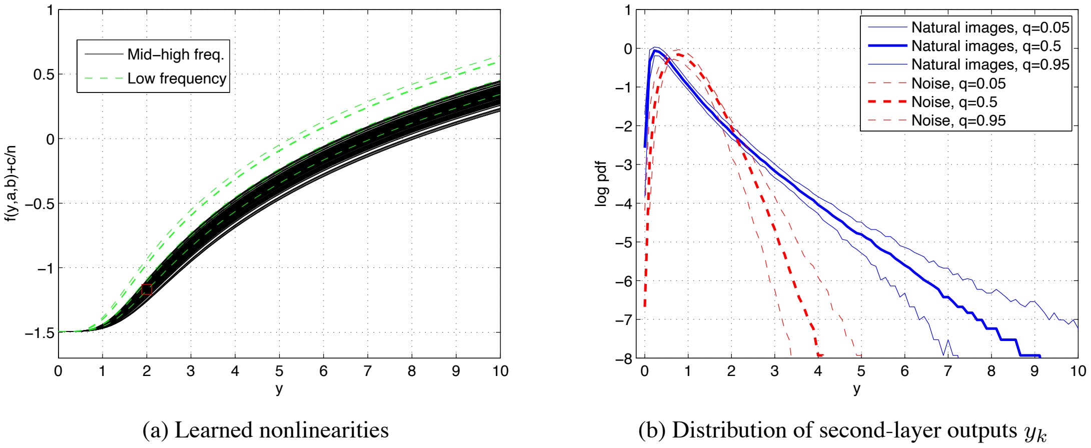


Figure 13: Two-layer model with thresholding nonlinearities: Visualization of the first- and second-layer feature detectors with icons. In the second layer, first-layer features of similar orientations are pooled together. See Figure 12 for details of how the icons were created. The feature detectors marked with a green frame are tuned to low frequencies.



(a) Learned nonlinearities

(b) Distribution of second-layer outputs y_k

Figure 14: Two-layer model with thresholding nonlinearities: Learned nonlinearities and interpretation. Natural images tend to have larger second-layer outputs y_k than noise input since the two processing layers, visualized in Figures 11 to 13, detect structure inherent to natural images. Thresholding the y_k provides a way to assign to natural images large values in the model-pdf and to noise small values. In Figure (a), the nonlinearities acting on pooled low-frequency feature detectors are shown in green (dashed lines), those for medium and high frequency feature detectors in black (solid lines). The bold curves in Figure (b) show the median, the other curves the 5% and 95% quantiles. The solid curves in blue relate to natural images, the dashed curves in red to noise. As explained in Figure 10, the y_k have expectation one for natural images.

be fixed (regression splines represented with B-spline basis functions, see for example Hastie et al., 2009, Chapter 5).

The model that we consider here is

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k; a_1, a_2, \ldots) + c, \qquad\qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2. \tag{25}$$

The difference between this and the model of the previous subsection is that the output nonlinearity f is a cubic spline. Part of the parameters $\boldsymbol{\theta}$ are thus as previously the $\mathbf{w}_i \in \mathbb{R}^n$, $Q_{ki} \geq 0$, and $c \in \mathbb{R}$. Additional parameters are the $a_i \in \mathbb{R}$ which are the coefficients of the B-spline basis functions of the cubic spline f. As before, we denote the matrix formed by the Q_{ki} by $\mathbf{Q}$.

For the modeling of the nonlinearity f, we must define its domain, which is the range of its arguments y_k. A way to control the range of y_k is to constrain the norm of the columns of $\mathbf{Q}$ and also to constrain the vectors $\mathbf{w}_k$ such that

$$\max_i \mathrm{E}\left\{(\mathbf{w}_i^T\mathbf{x})^2\right\} = 1, \tag{26}$$

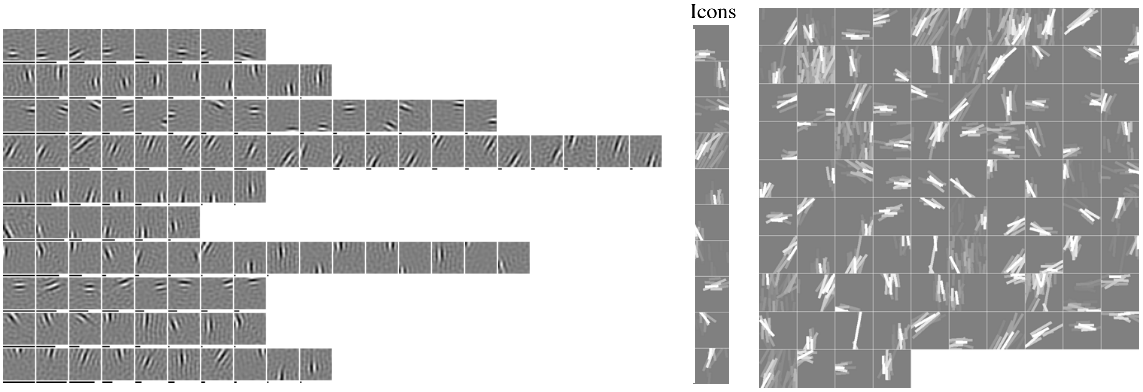
where the expectation is taken over the natural images.

We estimated the model in Equation (25) by first estimating a spline-based one-layer model which is presented in Appendix C.3. In brief, in this model, we did not square the first-layer feature outputs $\mathbf{w}_i^T\mathbf{x}$ and the matrix $\mathbf{Q}$ was the identity. The arguments of the spline nonlinearity f were thus the feature outputs $\mathbf{w}_i^T\mathbf{x}$ without additional processing. The learned nonlinearity is shown in Figure 16(a). In the following, we denote it by f_1. In Appendix C.3, we point out that the shape of f_1 is closely related to the sparsity of the feature outputs when natural images are the input. Because f_1 is an even function, and because of the squaring in the definition of y_k, we initialized f for the estimation of the two-layer model as $f(u) = f_1(\sqrt{u})$. This function is shown in Figure 16(b) (blue, dashes). The learned $\mathbf{w}_i$ of the one-layer model were used as initial points for the estimation of the two-layer model. The Q_{ki} were randomly initialized to small values. It turned out that imposing Equation (26) was enough for the learning to work and no norm constraint for the columns of $\mathbf{Q}$ was necessary. The results were very similar whether there were norm constraints or not. In the following, we report the results without any norm constraints.

5.4.1 RESULTS

Figure 15 visualizes the learned parameters $\mathbf{w}_i$ and Q_{ki} in the same way as in Figures 12 and 13 for the two-layer model with thresholding nonlinearities. The learned feature extraction stage is qualitatively very similar, up to two differences. The first difference is that many second-layer weights Q_{ki} shrank to zero: 66 out of 160 rows of the matrix $\mathbf{Q}$ had so small values that we could omit them while accounting for 99.9% of the sum $\sum_{ki} Q_{ki}$. The second difference is that the pooling in the second layer is sometimes less sparse. In that case, the second layer still combines first-layer feature detectors of the same orientation but they are not all centered at the same location.

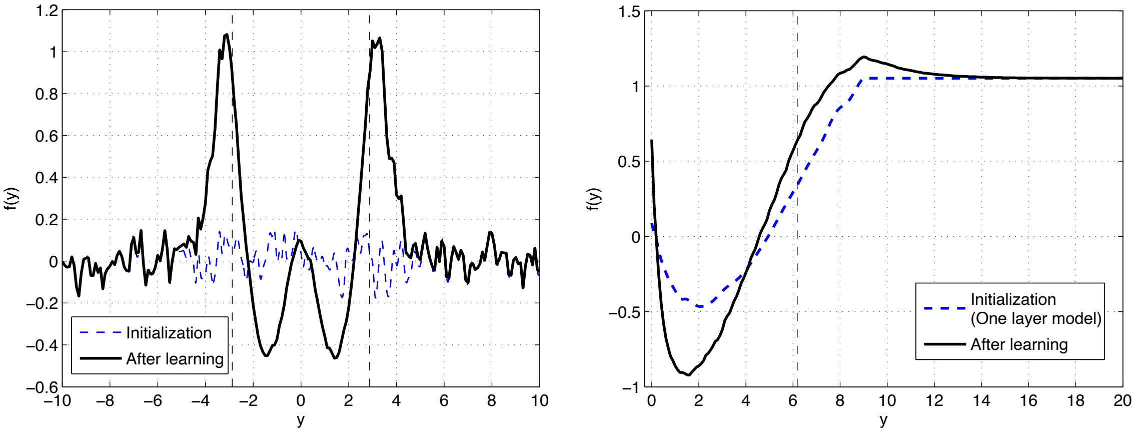
The learned nonlinearity f is shown in Figure 16(b) (black, solid). The nonlinearity from the one-layer model, shown in blue as a dashed curve, is altered so that small and large inputs are assigned to larger numbers while intermediate inputs are mapped to smaller numbers. Compared to the thresholding nonlinearities from the previous subsection, the learned nonlinearity has also for small inputs large outputs. Since the second-layer feature outputs y_k are sparser (that is, more often very small or large) for natural images than for the noise, the shape of the learned nonlinearity



(a) Pooling in the second layer

(b) Representation with icons

Figure 15: Two-layer model with spline nonlinearities. (a) Random selection of the learned second-layer units. (b) Representation of all the learned second-layer feature detectors as iconic images.



(a) Learned nonlinearity, one-layer model

(b) Learned nonlinearity, two-layer model

Figure 16: Two-layer model with spline nonlinearities. (a) Learned nonlinearity (black, solid) and its random initialization (blue, dashes) for the one-layer model. The learned nonlinearity is used as starting point in the learning of the two-layer model. (b) Learned nonlinearity (black, solid) and its initialization (blue, dashes) for the two-layer model. The dashed vertical lines indicate the 99% quantile for all the feature outputs for natural images. Due to the lack of training examples, the nonlinearities should not be considered valid beyond these lines.

implies that the estimated model assigns more often a higher probability density to natural images than to the noise.

5.5 Model Comparison

We have estimated models for natural images, both with thresholding nonlinearities and with splines. We make here a simple model comparison.

A quantitative comparison is done by calculating for ten validation sets the value of the objective function J_T of noise-contrastive estimation (see Equation (8) for the definition). The sample size of each validation set was $T_v = 100000$, and ν was set to 10, as in the estimation of the models. For the same validation data, we also computed the performance measure $\hat{L} = 1/T_v \sum_t \ln p_m(\mathbf{x}_t; \hat{\boldsymbol{\theta}}_T)$, which is an estimate for the rescaled log-likelihood, see Equation (13) in Section 3.1. As pointed out there, $\hat{L}$ is only an estimate of the rescaled log-likelihood because $\hat{c}$, which is an element of the parameter vector $\hat{\boldsymbol{\theta}}_T$, is used instead of the correct normalizing constant. Both J_T and the log-likelihood have the property that models which fit the data better have a higher score.

Comparing the structure of data points which are considered likely by the different models is a way to make a qualitative model comparison. Another approach would be to sample from the models, which we do in Appendix C.5. In order to get the likely points, we drew random samples that followed the noise distribution p_n (uniform on the sphere), and used them as initial points in the optimization of the various log-densities $\ln p_m(\mathbf{x}; \hat{\boldsymbol{\theta}}_T)$ with respect to $\mathbf{x}$ under the constraint of Equation (22). We used the same initial points for all models and visualized the likely points $\hat{\mathbf{x}}$ via Equation (23) as images $\hat{\mathbf{i}} = \mathbf{V}^{-}\hat{\mathbf{x}}$.

The ICA model with Laplacian sources is a simple model for natural images. It has previously also been used to model natural images after they have been projected on a sphere (Hyvärinen et al., 2009, Chapter 9). The unnormalized model has been defined in Section 3.2 in Equation (17) and consists of one processing layer with the fixed nonlinearity $f(u) = -\sqrt{2}|u|$. We include it in our comparison and refer to it as one-layer model with "Laplacian nonlinearity".

5.5.1 RESULTS

Table 1 shows that the spline-based two-layer model of Section 5.4 gives, on average, the largest value of the objective function J_T, and also L_T. To investigate the merits of the spline output-nonlinearity, we fixed the feature extraction stage of the thresholding model in Section 5.3 and learned only the nonlinearity f using splines (for details, see Appendix C.4). The resulting model, labeled "refinement" in the table, performs nearly as good as the best model. The one-layer models with thresholding or Laplacian nonlinearities have the smallest objectives J_T and L_T. The two models achieve the objectives in different, complimentary ways. For the thresholding model, the absolute value of the feature outputs $\mathbf{w}_i^T\mathbf{x}$ must be large to yield a large objective while for the model with the Laplacian nonlinearity $f(\mathbf{w}_i^T\mathbf{x}) = -\sqrt{2}|\mathbf{w}_i^T\mathbf{x}|$, the feature outputs must have small absolute values. The two models consider thus different aspects of the, for natural images, typically sparse feature outputs $\mathbf{w}_i^T\mathbf{x}$. The one-layer model with spline nonlinearity combines both aspects, see Figure 16(a), and yields also a higher score in the comparison. The same reason explains why spline-based two-layer models have higher scores than the two-layer model with the thresholding nonlinearity.

Figure 17 shows the likely data points from the various models p_m. The models with large objectives in Table 1 lead to image patches with particularly clear structure. The emergence of structure can be explained in terms of sparse coding since image patches which lead to sparse activations of the feature detectors are typically highly structured. Sparseness of the feature outputs

	One-layer model			Two-layer model		
	Thresholding	Laplacian	Spline	Thresholding	Refinement	Spline
J_T, av	-1.871	-1.518	-1.062	-0.8739	-0.6248	**-0.6139**
J_T, std	0.0022	0.0035	0.0030	0.0029	0.0030	0.0037
L_T, av	-223.280	-222.714	-219,786	-220.739	-213.303	**-212.598**
L_T, std	0.0029	0.0077	0.0137	0.0088	0.0282	0.0273

Table 1: Quantitative model comparison. The objective J_T of noise-contrastive estimation, see Equation (8), and the estimate $\hat{L}$ of the (rescaled) log-likelihood, see Equation (13), are used to measure the performance. Larger values indicate better performance. The table gives the average (av) and the standard deviation (std) for ten validation sets. All models are defined on a sphere and learned with noise-contrastive estimation. The features for the one-layer models with thresholding and Laplacian nonlinearity are not shown in the paper. The "one-layer, thresholding" model is identical to the "two-layer, thresholding" model when the second layer is fixed to the identity matrix. With Laplacian nonlinearity we mean the function $f(u) = -\sqrt{2}|u|$. The "two-layer, thresholding" model has been presented in Section 5.3, and the "two-layer, spline" model in Section 5.4. The "one-layer, spline" and "two-layer, refinement" models are presented in the Appendix C.3 and C.4, respectively.

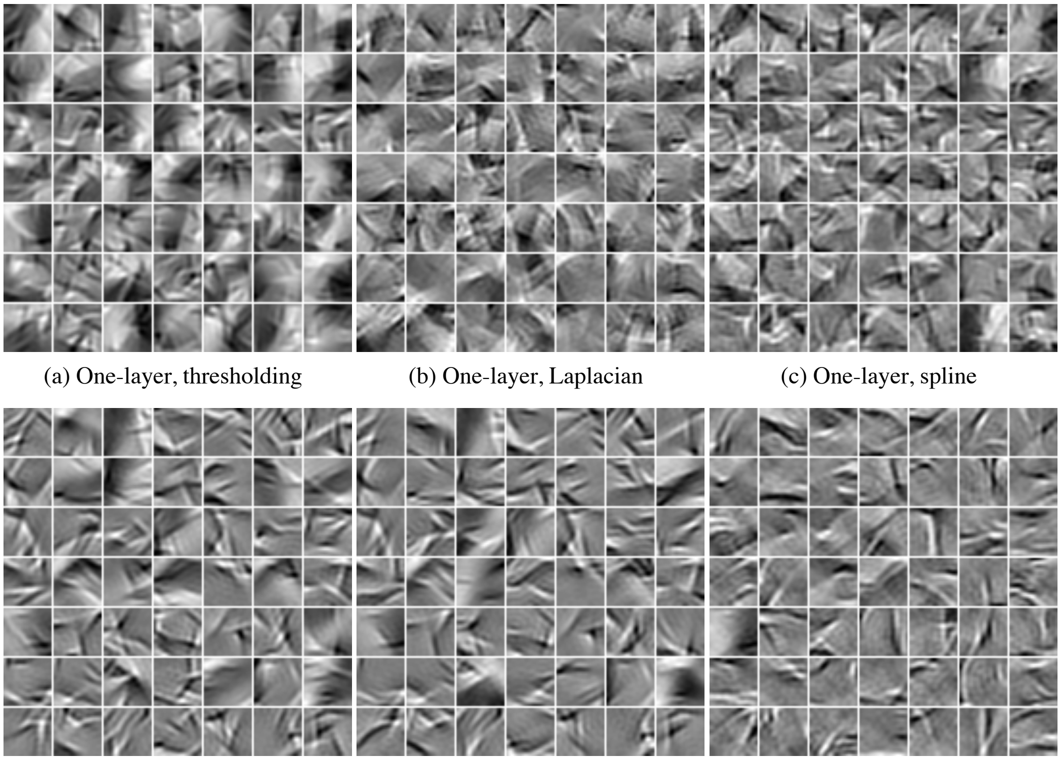
is facilitated by the nonlinearities in the models, and through the competition between the features by means of the sphere-constraint on the coordinates $\mathbf{x}$, as specified in Equation (22).

6. Conclusions

In this paper, we have considered the problem of estimating unnormalized statistical models for which the normalizing partition function cannot be computed in closed form. Such models cannot be estimated by maximization of the likelihood without resorting to numerical approximations which are often computationally expensive. The main contribution of the paper is a new estimation method for unnormalized models. A further contribution is made in the modeling of natural image statistics.

We have proven that our new estimation method, noise-contrastive estimation, provides a consistent estimator for both normalized and unnormalized statistical models. The assumptions that must be fulfilled to have consistency are not stronger than the assumptions that are needed in maximum likelihood estimation. We have further derived the asymptotic distribution of the estimation error which shows that, in the limit of arbitrarily many contrastive noise samples, the estimator performs like the maximum likelihood estimator. The new method has a very intuitive interpretation in terms of supervised learning: The estimation is performed by discriminating between the observed data and some artificially generated noise by means of logistic regression.

All theoretical results were illustrated and validated on artificial data where ground truth is known. We have also used artificial data to assess the balance between statistical and computational performance. In particular, we have compared the new estimation method to a number of other estimation methods for unnormalized models: Simulations suggest that noise-contrastive estimation strikes a highly competitive trade-off. We have used the mean squared error of the estimated param-



(a) One-layer, thresholding (b) One-layer, Laplacian (c) One-layer, spline

(d) Two-layer, thresholding (e) Two-layer, refinement (f) Two-layer, spline

Figure 17: Likely points under the learned models for natural images. See caption of Table 1 for information on the models.

eters as statistical performance measure. It should be noted that this is only one possible criterion among many (see Hyvärinen, 2008, for a recently proposed alternative measure of performance).

Noise-contrastive estimation as presented here extends the previous definition given by Gutmann and Hyvärinen (2010) since it allows for more noise samples than data points. We have also previously considered such a generalization (Pihlaja et al., 2010). Unlike in that preliminary version, our method here is asymptotically Fisher-efficient for all admissible noise densities when the number of noise samples becomes arbitrarily large. Pihlaja et al. (2010) has established links of noise-contrastive estimation to importance sampling which remain valid for this paper.

We applied noise-contrastive estimation to the modeling of natural images. Besides validating the method on a large two-layer model, we have, as a new contribution to the understanding of natural image statistics, presented spline-based extensions: In previous models, the output nonlinearity in the pdf was hand-picked. Here, we have parameterized it as a spline and learned it from the data. The statistical models were all unnormalized and had several ten-thousands of parameters which demonstrates that our new method can handle demanding estimation problems.

Acknowledgments

We would like to thank the anonymous referees for their comments which helped to improve the paper. We are also grateful for the interesting discussions with Yoshua Bengio, Ian Goodfellow, Pascal Vincent, Geoffrey Hinton, Nicolas Le Roux, Marc'Aurelio Ranzato, and Ilya Sutskever. This work was funded by the Centre-of-Excellence in Algorithmic Data Analysis and the Computational Sciences program, both of the Academy of Finland.

Appendix A. Proofs of the Theorems

We give here detailed proofs for Theorem 1, 2 and 3 on nonparametric estimation, consistency and the asymptotic distribution of the estimator, respectively.

A.1 Preliminaries

In the proofs, we often use the following properties of the function $r_\nu(u)$,

$$r_\nu(u) = \frac{1}{1+\nu\exp(-u)},$$

which was introduced in Equation (6):

$$\begin{aligned}
1-r_\nu(u) &= r_{\frac{1}{\nu}}(-u)\\
\frac{\partial r_\nu(u)}{\partial u} &= r_{\frac{1}{\nu}}(-u)r_\nu(u)\\
\frac{\partial}{\partial u}\ln r_\nu(u) &= r_{\frac{1}{\nu}}(-u)\\
\frac{\partial^2}{\partial u^2}\ln r_\nu(u) &= -r_{\frac{1}{\nu}}(-u)r_\nu(u)\\
\frac{\partial}{\partial u}\ln[1-r_\nu(u)] &= -r_\nu(u)\\
\frac{\partial^2}{\partial u^2}\ln[1-r_\nu(u)] &= -r_{\frac{1}{\nu}}(-u)r_\nu(u)
\end{aligned}$$

The functions $h(\mathbf{u};\boldsymbol{\theta}) = r_\nu(G(\mathbf{u};\boldsymbol{\theta}))$ and $1-h(\mathbf{u};\boldsymbol{\theta}) = r_{\frac{1}{\nu}}(-G(\mathbf{u};\boldsymbol{\theta}))$ are equal to

$$h(\mathbf{u};\boldsymbol{\theta}) = \frac{p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta})+\nu p_n(\mathbf{u})}, \qquad 1-h(\mathbf{u};\boldsymbol{\theta}) = \frac{\nu p_n(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta})+\nu p_n(\mathbf{u})}, \tag{27}$$

see Equation (3). It follows that

$$\nu p_n(\mathbf{u})r_\nu(G(\mathbf{u};\boldsymbol{\theta})) = \frac{\nu p_n(\mathbf{u})p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta})+\nu p_n(\mathbf{u})},, \tag{28}$$

$$p_d(\mathbf{u})r_{\frac{1}{\nu}}(-G(\mathbf{u};\boldsymbol{\theta})) = \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta})+\nu p_n(\mathbf{u})}, \tag{29}$$

which are key properties for the proofs below.

The first and second order derivatives are used in the following Taylor expansions

$$\begin{aligned}
\ln r_\nu(u+\epsilon u_1+\epsilon^2 u_2) &= \ln r_\nu(u)+\epsilon r_{\frac{1}{\nu}}(-u)u_1+ \\
&\quad \epsilon^2\left[r_{\frac{1}{\nu}}(-u)u_2-\frac{1}{2}r_{\frac{1}{\nu}}(-u)r_\nu(u)u_1^2\right]+ \\
&\quad O(\epsilon^3), \qquad (30)\\
\ln\left[1-r_\nu(u+\epsilon u_1+\epsilon^2 u_2)\right] &= \ln[1-r_\nu(u)]-\epsilon r_\nu(u)u_1+ \\
&\quad \epsilon^2\left[-r_\nu(u)u_2-\frac{1}{2}r_{\frac{1}{\nu}}(-u)r_\nu(u)u_1^2\right]+ \\
&\quad O(\epsilon^3). \qquad (31)
\end{aligned}$$

A.2 Proof of Theorem 1 (Nonparametric Estimation)

For clarity of the proof, we state an important stepping stone as a lemma.

A.2.1 LEMMA

The Taylor expansions in Equation (30) and Equation (31) are used to prove the following lemma.

Lemma 8 *For $\epsilon > 0$ and $\phi(\mathbf{x})$ a perturbation of the log-pdf $f_m(\mathbf{x}) = \ln p_m(\mathbf{x})$,*

$$\begin{aligned}
\tilde{J}(f_m+\epsilon\phi) &= \tilde{J}(f_m)+\epsilon\int[p_d(\mathbf{u})r_{\frac{1}{\nu}}(-f_m(\mathbf{u})+\ln p_n(\mathbf{u}))- \\
&\quad \nu p_n(\mathbf{u})r_\nu(f_m(\mathbf{u})-\ln p_n(\mathbf{u}))]\phi(\mathbf{u})\mathrm{d}\mathbf{u}- \\
&\quad \frac{\epsilon^2}{2}\int r_{\frac{1}{\nu}}(-f_m(\mathbf{u})+\ln p_n(\mathbf{u}))r_\nu(f_m(\mathbf{u})-\ln p_n(\mathbf{u})) \\
&\quad (p_d(\mathbf{u})+\nu p_n(\mathbf{u}))\phi(\mathbf{u})^2\mathrm{d}\mathbf{u}+O(\epsilon^3).
\end{aligned}$$

Proof The proof is obtained by evaluating the objective function $\tilde{J}$ in Equation (11) at $f_m+\epsilon\phi$, and making then use of the Taylor expansions in Equation (30) and Equation (31) with $u = f_m(\mathbf{x})-\ln p_n(\mathbf{x})$, $u_1 = \phi(\mathbf{x})$ and $u_2 = 0$. ■

A.2.2 PROOF OF THE THEOREM

Proof A necessary condition for optimality is that in the expansion of $\tilde{J}(f_m+\epsilon\phi)$, the term of order ϵ is zero for any perturbation ϕ. This happens if and only if

$$p_d(\mathbf{u})r_{\frac{1}{\nu}}(-f_m(\mathbf{u})+\ln p_n(\mathbf{u})) = \nu p_n(\mathbf{u})r_\nu(f_m(\mathbf{u})-\ln p_n(\mathbf{u})).$$

With Equation (28) and Equation (29), this implies that $\tilde{J}$ has an extremum at p_m if and only if

$$\frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_m(\mathbf{u})+\nu p_n(\mathbf{u})} = \frac{\nu p_n(\mathbf{u})p_m(\mathbf{u})}{p_m(\mathbf{u})+\nu p_n(\mathbf{u})}.$$

That is, as $\nu > 0$, $p_m(\mathbf{u}) = p_d(\mathbf{u})$ at all points $\mathbf{u}$ where $p_n(\mathbf{u}) \neq 0$. At points where $p_n(\mathbf{u}) = 0$, the equation is trivially fulfilled. Hence, $p_m = p_d$, or $f_m = \ln p_d$, leads to an extremum of $\tilde{J}$.

Inserting $f_m = \ln p_d$ into $\tilde{J}$ in Lemma 8 leads to

$$\tilde{J}(\ln p_d + \epsilon\phi) = \tilde{J}(\ln p_d) - \frac{\epsilon^2}{2}\left\{\int \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}\phi(u)^2 \mathrm{d}\mathbf{u}\right\} + O(\epsilon^3).$$

Since the term of order ϵ^2 is negative for all choices of ϕ, the extremum is a maximum. The assumption that $p_n(\mathbf{u}) \neq 0$ whenever $p_d(\mathbf{u}) \neq 0$ shows that $f_m = \ln p_d$ is the only extremum and completes the proof. ■

A.3 Proof of Theorem 2 (Consistency)

For clarity of the proof, we state important stepping stones as lemmata.

A.3.1 LEMMATA

The Taylor expansions in Equation (30) and Equation (31) are used to prove the following lemma which is like Lemma 8 for $\tilde{J}$ but for the objective function J in Equation (10).

Lemma 9 *For $\epsilon > 0$ and $\boldsymbol{\varphi} \in \mathbb{R}^m$,*

$$\begin{aligned} J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) &= J(\boldsymbol{\theta}) + \epsilon \int u_1 \left[p_d(\mathbf{u})(1 - h(\mathbf{u};\boldsymbol{\theta})) - \nu p_n(\mathbf{u})h(\mathbf{u};\boldsymbol{\theta})\right] \mathrm{d}\mathbf{u} + \\ &\quad \epsilon^2 \left\{ \int -\frac{1}{2}u_1^2(1 - h(\mathbf{u};\boldsymbol{\theta}))h(\mathbf{u};\boldsymbol{\theta})\left(p_d(\mathbf{u}) + \nu p_n(\mathbf{u})\right) \mathrm{d}\mathbf{u} + \right. \\ &\quad \left. \int u_2 \left(p_d(\mathbf{u})(1 - h(\mathbf{u};\boldsymbol{\theta})) - \nu p_n(\mathbf{u})h(\mathbf{u};\boldsymbol{\theta})\right) \mathrm{d}\mathbf{u} \right\} + O(\epsilon^3), \end{aligned}$$

where

$$\begin{aligned} u_1 &= \boldsymbol{\varphi}^T \mathbf{g}(\mathbf{u};\boldsymbol{\theta}), \\ u_2 &= \frac{1}{2}\boldsymbol{\varphi}^T \mathbf{H}_G(\mathbf{u};\boldsymbol{\theta})\boldsymbol{\varphi}. \end{aligned}$$

The term $\mathbf{g}(\mathbf{u};\boldsymbol{\theta})$ is $\nabla G(\mathbf{u};\boldsymbol{\theta})$, and $\mathbf{H}_G$ denotes the Hessian matrix of $G(\mathbf{u};\boldsymbol{\theta})$ where the derivatives are taken with respect to $\boldsymbol{\theta}$.

Proof With the definition of J in Equation (10), we have

$$\begin{aligned} J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) &= \int \ln\left[r_\nu\left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})\right)\right] p_d(\mathbf{u})\mathrm{d}\mathbf{u} + \\ &\quad \nu \int \ln\left[1 - r_\nu\left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})\right)\right] p_n(\mathbf{u})\mathrm{d}\mathbf{u}. \end{aligned}$$

Developing $G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})$ till terms of order ϵ^2 yields

$$G(\mathbf{u};\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi}) = G(\mathbf{u};\boldsymbol{\theta}) + \epsilon\boldsymbol{\varphi}^T\mathbf{g}(\mathbf{u};\boldsymbol{\theta}) + \epsilon^2\frac{1}{2}\boldsymbol{\varphi}^T\mathbf{H}_G(\mathbf{u};\boldsymbol{\theta})\boldsymbol{\varphi} + O(\epsilon^3).$$

Defining u_1 and u_2 as in the lemma, we obtain

$$\ln r_\nu\left(G(\mathbf{u};\boldsymbol{\theta} + \epsilon\nu)\right) = \ln r_\nu\left(G(\mathbf{u};\boldsymbol{\theta}) + \epsilon u_1 + \epsilon^2 u_2 + O(\epsilon^3)\right).$$

Using now the Taylor expansions in Equation (30) and Equation (31) for $u = G(\mathbf{u};\boldsymbol{\theta})$, and the identities $h(\mathbf{u};\boldsymbol{\theta}) = r_\nu(G(\mathbf{u};\boldsymbol{\theta}))$ as well as $1-h(\mathbf{u};\boldsymbol{\theta}) = r_{\frac{1}{\nu}}(-G(\mathbf{u};\boldsymbol{\theta}))$ proves the lemma. ■

Lemma 10 *If* $p_n(\mathbf{u}) \neq 0$ *whenever* $p_d(\mathbf{u}) \neq 0$ *and if*

$$\boldsymbol{\mathcal{I}}_\nu = \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$$

is full rank, where

$$\begin{aligned} P_\nu(\mathbf{u}) &= \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}, \\ \mathbf{g}(\mathbf{u}) &= \nabla_{\boldsymbol{\theta}} \ln p_m(\mathbf{u};\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}, \end{aligned}$$

then

$$J(\boldsymbol{\theta}^\star) > J(\boldsymbol{\theta}^\star + \boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \neq \mathbf{0}.$$

Proof A necessary condition for optimality is that in the expansion of $J(\boldsymbol{\theta} + \epsilon\boldsymbol{\varphi})$ in Lemma 9, the term of order ϵ is zero for any $\boldsymbol{\varphi}$. This happens if

$$p_d(\mathbf{u})(1-h(\mathbf{u};\boldsymbol{\theta})) = \nu p_n(\mathbf{u}) h(\mathbf{u};\boldsymbol{\theta}),$$

that is, if

$$\frac{\nu p_n(\mathbf{u}) p_d(\mathbf{u})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})} = \frac{\nu p_n(\mathbf{u}) p_m(\mathbf{u};\boldsymbol{\theta})}{p_m(\mathbf{u};\boldsymbol{\theta}) + \nu p_n(\mathbf{u})},$$

where we have used Equation (28) and Equation (29) as in the proof for Lemma 8. The assumption that $\nu > 0$ and $p_d(.) = p_m(.;\boldsymbol{\theta}^\star)$ implies together with the above equation that the term of order ϵ is zero if $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$.

The objective function $J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi})$ becomes thus

$$\begin{aligned} J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi}) &= J(\boldsymbol{\theta}^\star) - \frac{\epsilon^2}{2}\int u_1^2 (1-h(\mathbf{u};\boldsymbol{\theta}^\star)) h(\mathbf{u};\boldsymbol{\theta}^\star) \\ &\quad (p_d(\mathbf{u}) + \nu p_n(\mathbf{u}))\, \mathrm{d}\mathbf{u} + O(\epsilon^3). \end{aligned}$$

The terms $h(\mathbf{u};\boldsymbol{\theta}^\star)$ and $1-h(\mathbf{u};\boldsymbol{\theta}^\star)$ are with Equation (27)

$$h(\mathbf{u};\boldsymbol{\theta}^\star) = \frac{p_d(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}, \qquad\qquad 1-h(\mathbf{u};\boldsymbol{\theta}^\star) = \frac{\nu p_n(\mathbf{u})}{p_d(\mathbf{u}) + \nu p_n(\mathbf{u})}.$$

The expression for $J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi})$ becomes then

$$J(\boldsymbol{\theta}^\star + \epsilon\boldsymbol{\varphi}) = J(\boldsymbol{\theta}^\star) - \frac{\epsilon^2}{2}\boldsymbol{\varphi}^T \left[\int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}\right] \boldsymbol{\varphi} + O(\epsilon^3)$$

by inserting the definition of u_1 evaluated at $\boldsymbol{\theta}^\star$, and making use of the definitions for $P_\nu(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ in the statement of the lemma. The term of order ϵ^2 defines the nature of the extremum at $\boldsymbol{\theta}^\star$. If $\boldsymbol{\mathcal{I}}_\nu$ is positive definite, $J(\boldsymbol{\theta}^\star)$ is a maximum. As $\boldsymbol{\mathcal{I}}_\nu$ is a positive semi-definite matrix, it is positive definite if it is full rank.

Depending on the parameterization, there might be other values $\check{\boldsymbol{\theta}}$ which make the term of order ϵ zero. Note that, by definition, $J(\boldsymbol{\theta}) = \tilde{J}(\ln p_m(.;\boldsymbol{\theta}))$ for any $\boldsymbol{\theta}$ so that $J(\check{\boldsymbol{\theta}}) = \tilde{J}(\ln p_m(.;\check{\boldsymbol{\theta}}))$ and $J(\boldsymbol{\theta}^\star) = \tilde{J}(\ln p_m(.;\boldsymbol{\theta}^\star)) = \tilde{J}(\ln p_d)$. Now, by Theorem 1, $J(\check{\boldsymbol{\theta}}) < J(\boldsymbol{\theta}^\star)$ for a suitable noise density p_n so that J attains a global maximum at $\boldsymbol{\theta}^\star$. ■

A.3.2 PROOF OF THE THEOREM

The proof of consistency goes along the same lines as the proof of consistency for MLE (see for example Wasserman, 2004, Chapter 9).

Proof To prove consistency, we have to show that given $\epsilon > 0$, $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ tends to zero as $T_d \to \infty$. In what follows, it is sometimes useful to make the underlying probability space explicit and write $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ as $P(\{\omega : ||\hat{\boldsymbol{\theta}}_T(\omega) - \boldsymbol{\theta}^\star|| > \epsilon\})$.

Since, by Lemma 10, $J(\boldsymbol{\theta}^\star)$ is a global maximum, $||\boldsymbol{\theta} - \boldsymbol{\theta}^\star|| > \epsilon$ implies that there is a $\delta(\epsilon)$ such that $J(\boldsymbol{\theta}) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)$. Hence,

$$\{\omega : ||\hat{\boldsymbol{\theta}}_T(\omega) - \boldsymbol{\theta}^\star|| > \epsilon\} \subset \{\omega : J(\hat{\boldsymbol{\theta}}_T(\omega)) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)\}$$

and thus

$$P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon) < P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)). \tag{32}$$

Next, we investigate what happens to $P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon))$ when T_d goes to infinity. We have

$$\begin{aligned} J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T) &= J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star) + J_T(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T) \\ &\leq J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star) + J_T(\hat{\boldsymbol{\theta}}_T) - J(\hat{\boldsymbol{\theta}}_T) \end{aligned}$$

as $\hat{\boldsymbol{\theta}}_T$ has been defined as the argument which maximizes J_T. Using the triangle inequality we obtain further

$$|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| \leq |J(\boldsymbol{\theta}^\star) - J_T(\boldsymbol{\theta}^\star)| + |J_T(\hat{\boldsymbol{\theta}}_T) - J(\hat{\boldsymbol{\theta}}_T)|,$$

and

$$|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| \leq 2 \sup_{\boldsymbol{\theta}} |J(\boldsymbol{\theta}) - J_T(\boldsymbol{\theta})|,$$

from which follows that

$$P(|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| > \delta(\epsilon)) \leq P(2 \sup_{\boldsymbol{\theta}} |J(\boldsymbol{\theta}) - J_T(\boldsymbol{\theta})| > \delta(\epsilon)).$$

Using the assumption that $J_T(\boldsymbol{\theta})$ converges in probability uniformly over $\boldsymbol{\theta}$ to $J(\boldsymbol{\theta})$, we obtain that for sufficiently large T_d

$$P(|J(\boldsymbol{\theta}^\star) - J(\hat{\boldsymbol{\theta}}_T)| > \delta(\epsilon)) < \epsilon_2$$

for any $\epsilon_2 > 0$. As $J(\boldsymbol{\theta}^\star) > J(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$, we have thus the result that

$$P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon)) < \epsilon_2$$

for any $\epsilon_2 > 0$. The probability $P(J(\hat{\boldsymbol{\theta}}_T) < J(\boldsymbol{\theta}^\star) - \delta(\epsilon))$ can thus be made arbitrarily small by choosing T_d large enough. Combining this result with Equation (32), we conclude that $P(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star|| > \epsilon)$ tends to zero as $T_d \to \infty$. ■

A.4 Proof of Theorem 3 (Asymptotic Normality)

For clarity of the proof, we state important stepping stones as lemmata.

A.4.1 LEMMATA

In the following lemma, we use the definitions of the score function $\mathbf{g}(\mathbf{x};\boldsymbol{\theta})$ and $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x};\boldsymbol{\theta}^\star)$, as well as the definition of the Hessian $\mathbf{H}_G$, which were given in Lemma 9 and Lemma 10.

Lemma 11

$$0 = \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) + \mathbf{H}_J(\boldsymbol{\theta}^\star)(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) + O(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2)$$

where

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d}\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t) - \nu\frac{1}{T_n}\sum_{t=1}^{T_n} h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t),\\
\mathbf{H}_J(\boldsymbol{\theta}^\star) &= \frac{1}{T_d}\sum_{t=1}^{T_d}\Big\{-(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))h(\mathbf{x}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{x}_t)\mathbf{g}(\mathbf{x}_t)^T + \\
&\quad (1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{H}_G(\mathbf{x}_t;\boldsymbol{\theta}^\star)\Big\} - \\
&\quad \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\Big\{(1-h(\mathbf{y}_t;\boldsymbol{\theta}^\star))h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\mathbf{g}(\mathbf{y}_t)^T + \\
&\quad h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{H}_G(\mathbf{y}_t;\boldsymbol{\theta}^\star)\Big\}.
\end{aligned}$$

Proof Using the chain rule, it follows from the relations in Section A.1 that

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \ln h(\mathbf{x}_t;\boldsymbol{\theta}) &= (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\mathbf{g}(\mathbf{x}_t;\boldsymbol{\theta})\\
\nabla_{\boldsymbol{\theta}} \ln[1-h(\mathbf{y}_t;\boldsymbol{\theta})] &= -h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta}).
\end{aligned}$$

The derivative $\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta})$ of $J_T(\boldsymbol{\theta})$, defined in Equation (9) as

$$J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d}\ln h(\mathbf{x}_t;\boldsymbol{\theta}) + \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\ln[1-h(\mathbf{y}_t;\boldsymbol{\theta})],$$

is

$$\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}) = \frac{1}{T_d}\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}))\mathbf{g}(\mathbf{x};\boldsymbol{\theta}) - \nu\frac{1}{T_n}\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta}).$$

As $\hat{\boldsymbol{\theta}}_T$ is the value of $\boldsymbol{\theta}$ which maximizes $J_T(\boldsymbol{\theta})$, we must have $\nabla_{\boldsymbol{\theta}} J_T(\hat{\boldsymbol{\theta}}_T) = 0$. Doing a Taylor series around $\hat{\boldsymbol{\theta}}_T$, we have

$$0 = \nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) + \mathbf{H}_J(\boldsymbol{\theta}^\star)(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) + O((||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2).$$

Half of the lemma is proved when $\nabla_{\boldsymbol{\theta}} J_T$ is evaluated at $\boldsymbol{\theta}^\star$. To prove the other half, we need to calculate the Hessian $\mathbf{H}_J$ at $\boldsymbol{\theta}^\star$. The k-th row of the Hessian $\mathbf{H}_J(\boldsymbol{\theta})$ is $\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta})^T$ where F_k is the k-th element of the vector $\nabla_{\boldsymbol{\theta}} J_T$. Denoting by g_k the k-th element of the score function $\mathbf{g}$, we have

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta}) &= \frac{1}{T_d}\sum_{t=1}^{T_d}\{-\nabla_{\boldsymbol{\theta}} h(\mathbf{x}_t;\boldsymbol{\theta})g_k(\mathbf{x}_t;\boldsymbol{\theta}) + (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\}\\
&\quad -\nu\frac{1}{T_n}\sum_{t=1}^{T_n}\{\nabla_{\boldsymbol{\theta}} h(\mathbf{y}_t;\boldsymbol{\theta})g_k(\mathbf{y}_t;\boldsymbol{\theta}) + h(\mathbf{y}_t;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\}.
\end{aligned}$$

Using the chain rule, it follows from the relations in Section A.1 that

$$\nabla_{\boldsymbol{\theta}} h(\mathbf{u};\boldsymbol{\theta}) = (1-h(\mathbf{u};\boldsymbol{\theta}))h(\mathbf{u};\boldsymbol{\theta})\mathbf{g}(\mathbf{u};\boldsymbol{\theta}).$$

Hence,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} F_k(\boldsymbol{\theta}) = & \frac{1}{T_d}\sum_{t=1}^{T_d}\{-(1-h(\mathbf{x}_t;\boldsymbol{\theta}))h(\mathbf{x}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{x}_t;\boldsymbol{\theta})g_k(\mathbf{x}_t;\boldsymbol{\theta})+ \\ & (1-h(\mathbf{x}_t;\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}} g_k(\mathbf{x}_t;\boldsymbol{\theta})\} - \\ & \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\{(1-h(\mathbf{y}_t;\boldsymbol{\theta}))h(\mathbf{y}_t;\boldsymbol{\theta})\mathbf{g}(\mathbf{y}_t;\boldsymbol{\theta})g_k(\mathbf{y}_t;\boldsymbol{\theta})+ \\ & h(\mathbf{y}_t;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} g_k(\mathbf{y}_t;\boldsymbol{\theta})\},\end{aligned}$$

which proves the lemma. ■

For the next lemma, recall the definition of $\boldsymbol{\mathcal{I}}_\nu$ given in Lemma 10 or Theorem 2.

Lemma 12 $\mathbf{H}_J(\boldsymbol{\theta}^\star)$ *converges in probability to* $-\boldsymbol{\mathcal{I}}_\nu$ *as the sample size* T_d *tends to infinity.*

Proof As $T_n = \nu T_d, T_n$ also tends to infinity when T_d tends to infinity. As the sample sizes become arbitrarily large, the sample averages become integration over the corresponding densities so that

$$\begin{aligned}\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) \overset{P}{\to} & \int -(1-h(\mathbf{x};\boldsymbol{\theta}^\star))h(\mathbf{x};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^T p_d(\mathbf{x})\mathrm{d}\mathbf{x} + \\ & \int (1-h(\mathbf{x};\boldsymbol{\theta}^\star))\mathbf{H}_G(\mathbf{x};\boldsymbol{\theta}^\star)p_d(\mathbf{x})\mathrm{d}\mathbf{x} - \\ & \int (1-h(\mathbf{y};\boldsymbol{\theta}^\star))h(\mathbf{y};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y})^T \nu p_n(\mathbf{y})\mathrm{d}\mathbf{y} - \\ & \int h(\mathbf{y};\boldsymbol{\theta}^\star)\mathbf{H}_G(\mathbf{y};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{y})\mathrm{d}\mathbf{y}.\end{aligned}$$

Reordering of the terms and changing the names of the integration variables to $\mathbf{u}$ gives

$$\begin{aligned}\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) \overset{P}{\to} & -\int (1-h(\mathbf{u};\boldsymbol{\theta}^\star))h(\mathbf{u};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T(p_d(\mathbf{u})+\nu p_n(\mathbf{u}))\mathrm{d}\mathbf{u} + \\ & \int \left((1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) - h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right)\mathbf{H}_G(\mathbf{u};\boldsymbol{\theta}^\star)\mathrm{d}\mathbf{u}.\end{aligned}$$

With Equation (28) and Equation (29), we have

$$\begin{aligned}(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) &= h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u}), \qquad (33) \\ (1-h(\mathbf{u};\boldsymbol{\theta}^\star))h(\mathbf{u};\boldsymbol{\theta}^\star)(p_d(\mathbf{u})+\nu p_n(\mathbf{u})) &= \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})}.\end{aligned}$$

Hence,

$$\lim_{T_d\to\infty}\mathbf{H}_J(\boldsymbol{\theta}^\star) \overset{P}{\to} -\int \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})}\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T\mathrm{d}\mathbf{u},$$

which is $-\boldsymbol{\mathcal{I}}_\nu$. ■

Lemma 13 *The expectation* $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ *is zero.*

Proof We calculate

$$\begin{aligned}
\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d}\sum_{t=1}^{T_d}\mathrm{E}\,\mathbf{g}(\mathbf{x}_t)(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star)) - \\
&\quad \nu\frac{1}{T_n}\sum_{t=1}^{T_n}\mathrm{E}\,\mathbf{g}(\mathbf{y}_t)h(\mathbf{y}_t;\boldsymbol{\theta}^\star) \\
&= \mathrm{E}\,\mathbf{g}(\mathbf{x})(1-h(\mathbf{x};\boldsymbol{\theta}^\star)) - \nu\,\mathrm{E}\,\mathbf{g}(\mathbf{y})h(\mathbf{y};\boldsymbol{\theta}^\star) \\
&= \int \mathbf{g}(\mathbf{u})(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u})\mathrm{d}\mathbf{u} - \\
&\quad \nu\int \mathbf{g}(\mathbf{u})h(\mathbf{u};\boldsymbol{\theta}^\star)p_n(\mathbf{u})\mathrm{d}\mathbf{u},
\end{aligned}$$

where the second equality follows from the i.i.d. assumption of the sample X and Y, respectively. Reordering leads to

$$\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) = \int \mathbf{g}(\mathbf{u})\left((1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) - h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right)\mathrm{d}\mathbf{u},$$

which is, with Equation (33), zero. ■

Lemma 14 *The variance* $\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ *is*

$$\frac{1}{T_d}\left(\boldsymbol{\mathcal{I}}_\nu - \left(1+\frac{1}{\nu}\right)\mathrm{E}(P_\nu\mathbf{g})\,\mathrm{E}(P_\nu\mathbf{g})^T\right),$$

where $\boldsymbol{\mathcal{I}}_\nu$, P_ν *and* $\mathbf{g}$ *were defined in Lemma 10, and the expectation is taken over the data-pdf* p_d.

Proof As the expectation $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$ is zero, the variance is given by $\mathrm{E}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)^T$. Multiplying out gives

$$\begin{aligned}
\mathrm{Var}\,\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)^T\right] - \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)^T\right] - \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\sum_{t=1}^{T_d}(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)^T\right] + \\
&\quad \frac{1}{T_d^2}\mathrm{E}\left[\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\sum_{t=1}^{T_n}h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)^T\right].
\end{aligned}$$

Since the samples are all independent from each other, we have

$$\begin{aligned}\operatorname{Var}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d^2}\sum_{t=1}^{T_d}\mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))^2\mathbf{g}(\mathbf{x}_t)\mathbf{g}(\mathbf{x}_t)^T\right]+\\ &\frac{1}{T_d^2}\sum_{\substack{t,\tau=1\\ t\neq\tau}}^{T_d}\mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\right]\mathrm{E}\left[(1-h(\mathbf{x}_\tau;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_\tau)^T\right]-\\ &\frac{1}{T_d^2}\sum_{t=1}^{T_d}\sum_{\tau=1}^{T_n}\mathrm{E}\left[(1-h(\mathbf{x}_t;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_t)\right]\mathrm{E}\left[h(\mathbf{y}_\tau;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_\tau)^T\right]-\\ &\frac{1}{T_d^2}\sum_{t=1}^{T_n}\sum_{\tau=1}^{T_d}\mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\right]\mathrm{E}\left[(1-h(\mathbf{x}_\tau;\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{x}_\tau)^T\right]+\\ &\frac{1}{T_d^2}\sum_{\substack{t,\tau=1\\ t\neq\tau}}^{T_n}\mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_t)\right]\mathrm{E}\left[h(\mathbf{y}_\tau;\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{y}_\tau)^T\right]+\\ &\frac{1}{T_d^2}\sum_{t=1}^{T_n}\mathrm{E}\left[h(\mathbf{y}_t;\boldsymbol{\theta}^\star)^2\mathbf{g}(\mathbf{y}_t)\mathbf{g}(\mathbf{y}_t)^T\right].\end{aligned}$$

As we assume that all $\mathbf{x}_t$, and also $\mathbf{y}_t$, are identically distributed, the above expression simplifies to

$$\begin{aligned}\operatorname{Var}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star) &= \frac{1}{T_d}\int(1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T p_d(\mathbf{u})\mathrm{d}\mathbf{u}+\\ &\frac{T_d^2-T_d}{T_d^2}\mathbf{m}_x\mathbf{m}_x^T-\frac{T_dT_n}{T_d^2}\mathbf{m}_x\mathbf{m}_y^T-\\ &\frac{T_dT_n}{T_d^2}\mathbf{m}_y\mathbf{m}_x^T+\frac{T_n^2-T_n}{T_d^2}\mathbf{m}_y\mathbf{m}_y^T+\\ &\frac{T_n}{T_d^2}\int h(\mathbf{u};\boldsymbol{\theta}^\star)^2\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T p_n(\mathbf{u})\mathrm{d}\mathbf{u}, \end{aligned}\tag{34}$$

where

$$\begin{aligned}\mathbf{m}_x &= \int(1-h(\mathbf{u};\boldsymbol{\theta}^\star))\mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u},\\ \mathbf{m}_y &= \int h(\mathbf{u};\boldsymbol{\theta}^\star)\mathbf{g}(\mathbf{u})p_n(\mathbf{u})\mathrm{d}\mathbf{u}.\end{aligned}$$

Denoting by A the sum of the first and last line of Equation (34), we have

$$A = \frac{1}{T_d}\int\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T\left[(1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2p_d(\mathbf{u})+h(\mathbf{u};\boldsymbol{\theta}^\star)^2\nu p_n(\mathbf{u})\right]\mathrm{d}\mathbf{u}$$

since $T_n=\nu T_d$. Now, Equation (27) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star)=p_d(\mathbf{u})$ imply that

$$\begin{aligned}(1-h(\mathbf{u};\boldsymbol{\theta}^\star))^2p_d(\mathbf{u})+h(\mathbf{u};\boldsymbol{\theta}^\star)^2\nu p_n(\mathbf{u}) &= \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})}\\ &= P_\nu p_d(\mathbf{u}),\end{aligned}$$

so that

$$\begin{aligned} A &= \frac{1}{T_d}\int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu p_d(\mathbf{u})\mathrm{d}\mathbf{u} \\ &= \frac{1}{T_d}\boldsymbol{\mathcal{I}}_\nu. \end{aligned}$$

Denote by B the second line of Equation (34). Rearranging the terms, we have

$$\begin{aligned} B &= \mathbf{m}_x\int \left[(1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u})-h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u})\right]\mathbf{g}(\mathbf{u})^T\mathrm{d}\mathbf{u}- \\ & \frac{1}{T_d}\mathbf{m}_x\mathbf{m}_x^T. \end{aligned} \tag{35}$$

Again, Equation (27) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star)=p_d(\mathbf{u})$ imply that

$$\begin{aligned} (1-h(\mathbf{u};\boldsymbol{\theta}^\star))p_d(\mathbf{u}) &= h(\mathbf{u};\boldsymbol{\theta}^\star)\nu p_n(\mathbf{u}) \\ &= \frac{\nu p_n(\mathbf{u})p_d(\mathbf{u})}{p_d(\mathbf{u})+\nu p_n(\mathbf{u})} \\ &= P_\nu p_d(\mathbf{u}), \end{aligned}$$

so that the first line in Equation (35) is zero and

$$\mathbf{m}_x = \int P_\nu \mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}.$$

The term B is thus

$$B = -\frac{1}{T_d}\int P_\nu\mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}\int P_\nu\mathbf{g}(\mathbf{u})^T p_d(\mathbf{u})\mathrm{d}\mathbf{u}.$$

Denote by C the third line of Equation (34). Rearranging the terms, we have with $T_n=\nu T_d$

$$C = -\frac{\nu}{T_d}\mathbf{m}_y\mathbf{m}_y^T+\nu\mathbf{m}_y(\nu\mathbf{m}_y^T-\mathbf{m}_x^T).$$

The term $\nu\mathbf{m}_y$ is with Equation (27) and $p_m(\mathbf{u};\boldsymbol{\theta}^\star)=p_d(\mathbf{u})$

$$\nu\mathbf{m}_y = \int P_\nu\mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u},$$

so that $\nu\mathbf{m}_y=\mathbf{m}_x$, and hence

$$\begin{aligned} C &= -\frac{1}{\nu T_d}(\nu\mathbf{m}_y)(\nu\mathbf{m}_y^T) \\ &= \frac{1}{\nu}B. \end{aligned}$$

All in all, the variance $\operatorname{Var}\nabla_{\boldsymbol{\theta}}J_T(\boldsymbol{\theta}^\star)$ is thus

$$\begin{aligned} \operatorname{Var}\nabla_{\boldsymbol{\theta}}J_T(\boldsymbol{\theta}^\star) &= A+B+C \\ &= \frac{1}{T_d}\left(\boldsymbol{\mathcal{I}}_\nu-\left(1+\frac{1}{\nu}\right)\mathrm{E}\left(P_\nu\mathbf{g}\right)\mathrm{E}\left(P_\nu\mathbf{g}^T\right)\right), \end{aligned}$$

where

$$\mathrm{E}\left(P_\nu\mathbf{g}\right) = \int P_\nu\mathbf{g}(\mathbf{u})p_d(\mathbf{u})\mathrm{d}\mathbf{u}.$$

■

A.4.2 PROOF OF THE THEOREM

We are now ready to give the proof of Theorem 3.

Proof Up to terms of order $O(||\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star||^2)$, we have with Lemma 11

$$\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star) = -\mathbf{H}_J^{-1}\sqrt{T_d}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star).$$

By Lemma 12, $\mathbf{H}_J \overset{P}{\to} -\boldsymbol{\mathcal{I}}_\nu$ for large sample sizes T_d. Using Lemma 13 and Lemma 14, we see that

$$\sqrt{T_d}\nabla_{\boldsymbol{\theta}} J_T(\boldsymbol{\theta}^\star)$$

converges in distribution to a normal distribution of mean zero and covariance matrix

$$\boldsymbol{\mathcal{I}}_\nu - \left(1 + \frac{1}{\nu}\right)\mathrm{E}(P_\nu \mathbf{g})\,\mathrm{E}(P_\nu \mathbf{g})^T,$$

which implies that $\sqrt{T_d}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^\star)$ converges in distribution to a normal distribution of mean zero and covariance matrix $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right)\boldsymbol{\mathcal{I}}_\nu^{-1}\,\mathrm{E}(P_\nu \mathbf{g})\,\mathrm{E}(P_\nu \mathbf{g})^T \boldsymbol{\mathcal{I}}_\nu^{-1}.$$

■

Appendix B. Calculations

The following sections contain calculations needed in Section 3.3 and Section 5.3.

B.1 Theory, Section 3.3: Asymptotic Variance for Orthogonal ICA Model

We calculate here the asymptotic covariance matrix of the estimation error for an orthogonal ICA model when a Gaussian distribution is used as noise distribution in noise-contrastive estimation. This result is used to make the predictions about the estimation error in Section 3.3. The calculations show that the asymptotic variance does not depend on the mixing matrix but only on the dimension of the data. Similar calculations can be used to show that this also holds for maximum likelihood estimation.

A random variable $\mathbf{x}$ following an ICA model with orthogonal mixing matrix $\mathbf{A} = (\mathbf{a}_1 \ldots \mathbf{a}_n)$ has the distribution

$$p_d(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{n} f(\mathbf{a}_i^T \mathbf{x}),$$

where Z is the partition function. By orthogonality of $\mathbf{A}$,

$$p_d(\mathbf{A}\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{n} f(x_i),$$

which equals $p_s(\mathbf{x})$ where p_s is the distribution of the sources $\mathbf{s}$ of the ICA model. Also by orthogonality of $\mathbf{A}$, the noise distribution p_n with the same covariance as $\mathbf{x}$ is the standard normal distribution. In particular, $p_n(\mathbf{A}\mathbf{x}) = p_n(\mathbf{x})$.

For the calculation of the asymptotic variance, we need to compute the matrix $\boldsymbol{\mathcal{I}}_\nu$ which occurs in Theorem 2, $\boldsymbol{\mathcal{I}}_\nu = \int \mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) p_d(\mathbf{u}) \mathrm{d}\mathbf{u}$. With the above data and noise distribution, $P_\nu(\mathbf{u})$ has the property that

$$\begin{aligned} P_\nu(\mathbf{Au}) &= \frac{\nu p_n(\mathbf{Au})}{p_d(\mathbf{Au}) + \nu p_n(\mathbf{Au})} \\ &= \frac{\nu p_n(\mathbf{u})}{p_s(\mathbf{u}) + \nu p_n(\mathbf{u})}. \end{aligned}$$

Hence $P_\nu(\mathbf{Au})$ does not depend on $\mathbf{A}$. Below, we will denote $P_\nu(\mathbf{Au})$ by $\tilde{P}_\nu(\mathbf{u})$. For the ICA model, the vector $\mathbf{g}(\mathbf{u})$ has the form

$$\mathbf{g}(\mathbf{u}) = (\mathbf{g}_1(\mathbf{u}), \ldots, \mathbf{g}_n(\mathbf{u}), g_c(\mathbf{u}))^T$$

where $\mathbf{g}_i(\mathbf{u}) = \nabla_{\mathbf{a}_i} \ln p_m(\mathbf{u}) = f'(\mathbf{a}_i^T\mathbf{u})\mathbf{u}$ and $g_c(\mathbf{u}) = \partial_c \ln p_m(\mathbf{u}) = 1$. By orthogonality of $\mathbf{A}$, we have

$$\mathbf{g}_i(\mathbf{Au}) = \mathbf{A} f'(u_i)\mathbf{u}.$$

We denote the vector $f'(u_i)\mathbf{u}$ by $\tilde{\mathbf{g}}_i(\mathbf{u})$ so that $\mathbf{g}_i(\mathbf{Au}) = \mathbf{A}\tilde{\mathbf{g}}_i(\mathbf{u})$. Hence,

$$\mathbf{g}(\mathbf{Au}) = \boldsymbol{\mathcal{A}}(\tilde{\mathbf{g}}_1(\mathbf{u}), \ldots, \tilde{\mathbf{g}}_n(\mathbf{u}), 1)^T$$

where $\boldsymbol{\mathcal{A}}$ is a block-diagonal matrix with n matrices $\mathbf{A}$ on the diagonal and a single 1 in the $(n+1)$-th slot. As a shorthand, we will denote $\mathbf{g}(\mathbf{Au})$ by $\boldsymbol{\mathcal{A}}\tilde{\mathbf{g}}(\mathbf{u})$.

With these preliminaries, using the change of variables $\mathbf{u} = \mathbf{Av}$,

$$\begin{aligned} \boldsymbol{\mathcal{I}}_\nu &= \int p_d(\mathbf{u})\mathbf{g}(\mathbf{u})\mathbf{g}(\mathbf{u})^T P_\nu(\mathbf{u}) \mathrm{d}\mathbf{u} \\ &= \int p_s(\mathbf{v}) \boldsymbol{\mathcal{A}}\tilde{\mathbf{g}}(\mathbf{v})\tilde{\mathbf{g}}(\mathbf{v})^T \boldsymbol{\mathcal{A}}^T \tilde{P}_\nu(\mathbf{v}) \mathrm{d}\mathbf{v} \\ &= \boldsymbol{\mathcal{A}}\tilde{\boldsymbol{\mathcal{I}}}_\nu \boldsymbol{\mathcal{A}}^T, \end{aligned}$$

where the matrix

$$\tilde{\boldsymbol{\mathcal{I}}}_\nu = \int p_s(\mathbf{v})\tilde{\mathbf{g}}(\mathbf{v})\tilde{\mathbf{g}}(\mathbf{v})^T \tilde{P}_\nu(\mathbf{v}) \mathrm{d}\mathbf{v}$$

does not depend on the mixing matrix $\mathbf{A}$ but only on the distribution of the sources $\mathbf{s}$, the noise distribution p_n, and ν. Moreover, by orthogonality of $\mathbf{A}$, the inverse of $\boldsymbol{\mathcal{I}}_\nu$ is given by

$$\boldsymbol{\mathcal{I}}_\nu^{-1} = \boldsymbol{\mathcal{A}}\tilde{\boldsymbol{\mathcal{I}}}_\nu^{-1} \boldsymbol{\mathcal{A}}^T.$$

The same reasoning shows that

$$\int p_d(\mathbf{u}) P_\nu(\mathbf{u}) \mathbf{g}(\mathbf{u}) \mathrm{d}\mathbf{u} = \boldsymbol{\mathcal{A}} \int p_s(\mathbf{v}) \tilde{\mathbf{g}}(\mathbf{v}) \tilde{P}_\nu(\mathbf{v}) \mathrm{d}\mathbf{v},$$

which we will denote below by $\boldsymbol{\mathcal{A}}\tilde{\mathbf{m}}$. Again, $\tilde{\mathbf{m}}$ does not depend on $\mathbf{A}$. Hence, the asymptotic covariance matrix $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{I}}_\nu^{-1} - \left(1 + \frac{1}{\nu}\right) \boldsymbol{\mathcal{I}}_\nu^{-1} \mathrm{E}(P_\nu \mathbf{g}) \mathrm{E}(P_\nu \mathbf{g})^T \boldsymbol{\mathcal{I}}_\nu^{-1},$$

in Theorem 3 is for the ICA model with orthogonal mixing matrix $\mathbf{A}$ given by

$$\boldsymbol{\Sigma}_{\text{ortICA}} = \boldsymbol{\mathcal{A}}\left[\tilde{\boldsymbol{\mathcal{I}}}_{\nu}^{-1} - \left(1+\frac{1}{\nu}\right)\tilde{\boldsymbol{\mathcal{I}}}_{\nu}^{-1}\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T\tilde{\boldsymbol{\mathcal{I}}}_{\nu}^{-1}\right]\boldsymbol{\mathcal{A}}^T.$$

The block matrix $\boldsymbol{\mathcal{A}}$ is orthogonal since $\mathbf{A}$ is orthogonal. The asymptotic variance, that is the trace of $\boldsymbol{\Sigma}_{\text{ortICA}}$, does hence not depend on $\mathbf{A}$.

B.2 Natural Images, Section 5.3: Optimal Stimuli

We show here that the optimal stimulus, namely the image which yields the largest feature output for feature $\mathbf{w}$ while satisfying the sphere constraints in Equation (22), is proportional to $\mathbf{V}^{-}(\mathbf{w}-\langle\mathbf{w}\rangle)$. The term $\langle\mathbf{w}\rangle$ denotes the average value of the elements in the vector $\mathbf{w}$.

Each coordinate vector $\mathbf{x}$ defines an image $\mathbf{i} = \mathbf{V}^{-}\mathbf{x}$, see Equation (23). The optimal image is thus $\mathbf{i}^* = \mathbf{V}^{-}\mathbf{x}^*$ where $\mathbf{x}^*$ is the solution to the optimization problem

$$\max_{\mathbf{x}} \mathbf{w}^T\mathbf{x}$$

subject to $\sum_{k=1}^n \mathbf{x}(k) = 0$ and $1/(n-1)\sum_{k=1}^n \mathbf{x}(k)^2 = 1$, which are the constraints in Equation (22). The Lagrangian associated with this constrained optimization problem is

$$L(\mathbf{x},\lambda,\omega) = \mathbf{w}^T\mathbf{x} - \lambda\left(\frac{1}{n-1}\sum_{k=1}^n \mathbf{x}(k)^2 - 1\right) - \omega\sum_{k=1}^n \mathbf{x}(k)$$

The maximizing $\mathbf{x}^*$ is $\mathbf{x}^* = (n-1)/(2\lambda)(\mathbf{w}-\omega)$. Taking ω such that the constraint $\sum_{k=1}^n \mathbf{x}^*(k) = 0$ is fulfilled gives

$$\mathbf{x}^* = \frac{n-1}{2\lambda}(\mathbf{w}-\langle\mathbf{w}\rangle).$$

Hence, the optimal image $\mathbf{i}^*$ is proportional to $\mathbf{V}^{-}(\mathbf{w}-\langle\mathbf{w}\rangle)$.

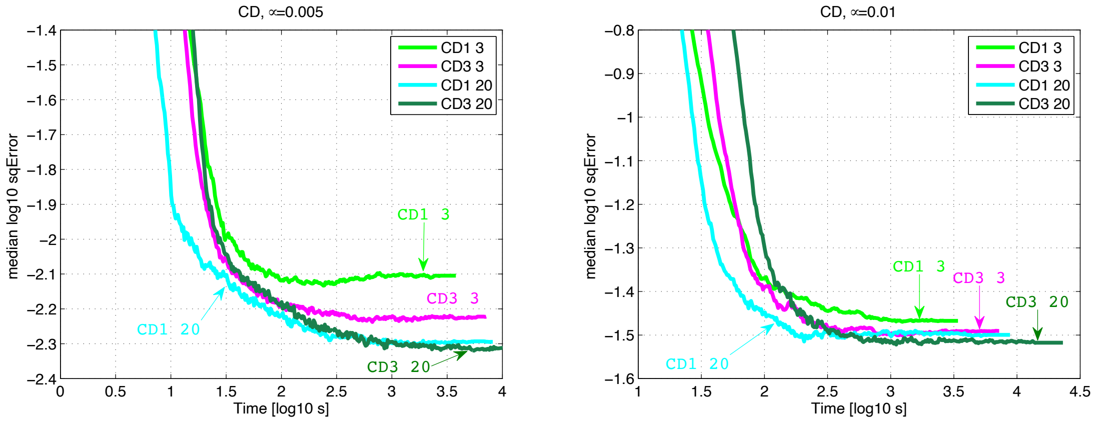
Note that if we had a norm constraint on $\mathbf{i}$ instead of the constraints in Equation (22), the Lagrangian would be

$$\tilde{L}(\mathbf{x},\lambda) = \mathbf{w}^T\mathbf{x} - \lambda\left(\sum_{k=1}^n \mathbf{x}(k)^2 d_k - 1\right)$$

where we have used that $\mathbf{i}^T\mathbf{i} = \mathbf{x}^T\mathbf{V}^{-T}\mathbf{V}^{-}\mathbf{x} = \mathbf{x}^T\mathbf{D}\mathbf{x}$. The $n\times n$ matrix $\mathbf{D}$ is diagonal with the eigenvalue d_k of the covariance matrix of the natural image patches as k-th element. The optimal $\mathbf{x}$ would thus be $\tilde{\mathbf{x}}^* = 1/(2\lambda)\mathbf{D}^{-1}\mathbf{w}$ so that the optimal image $\tilde{\mathbf{i}}^*$ would be proportional to $\mathbf{V}^{-}\mathbf{D}^{-1}\mathbf{w} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{w} = \mathbf{V}^T\mathbf{w}$, for which we have used the notation $\tilde{\mathbf{w}}$ in Section 5.3. Since the eigenvalues d_k fall off with the spatial frequency f (like $1/f^2$, see for example Hyvärinen et al., 2009, Chapter 5.6) the norm constraint on $\mathbf{i}$ punishes low frequencies more heavily than the constraints in Equation (22). As a consequence, the $\tilde{\mathbf{w}}$, which are shown in Figure 11(a), are tuned to high frequencies while the optimal stimuli $\mathbf{i}^*$, shown in Figure 11(b), contain more low frequency components.

Appendix C. Further Simulation Results

The following sections contain additional simulation results related to Section 4 and Section 5.



(a) Laplacian sources

(b) Logistic sources

Figure 18: Trade-off between statistical and computational performance for contrastive divergence (CD).While the algorithms were running, measurements of the estimation error at a given time were made. The time variable indicates thus the time since the algorithm was started. Note the difference to Figure 6 where the time indicates the time-till-convergence. The plots show the median performance over the 100 estimation problems. CDx y refers to contrastive divergence with x Monte Carlo steps, each using y leapfrog steps.

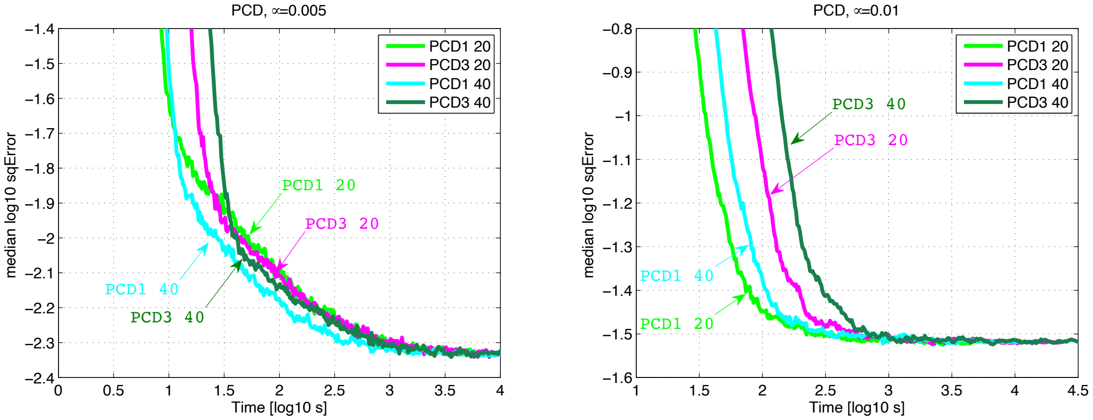
C.1 Trade-Off, Section 4: Comparison of the Different Settings of Contrastive and Persistent Contrastive Divergence

We compare here the different settings of contrastive and persistent contrastive divergence. Since the two estimations methods do not have an objective function, and given the randomness that is introduced by the minibatches, choosing a reliable stopping criterion is difficult. Hence, we did not impose any stopping criterion but the maximal number of iterations. The algorithms had always converged before this maximal number of iterations was reached, in the sense that the estimation error did not visibly decrease any more. In real applications, where the true parameters are not known, assessing convergence based on the estimation error is, however, clearly not possible.

C.1.1 RESULTS

Figure 18 shows that for contrastive divergence, using 20 leapfrog steps gives better results than using only three leapfrog steps. A trade-off between computation time and accuracy is visible: running the Markov chains for three Markov steps (CD3 20, in dark green) yields more accurate estimates than running them for one Markov step (CD1 20, in cyan) but the computations take also longer.

Figure 19 shows that for the tested schemes of persistent contrastive divergence, using one Markov step together with 40 leapfrog steps (PCD1 40, in cyan) is the preferred choice for Laplacian sources; for logistic sources, it is PCD1 20 (shown in light green).



(a) Laplacian sources

(b) Logistic sources

Figure 19: Trade-off between statistical and computational performance for persistent contrastive divergence (PCD). The results are plotted in the same way as for contrastive divergence in Figure 18.

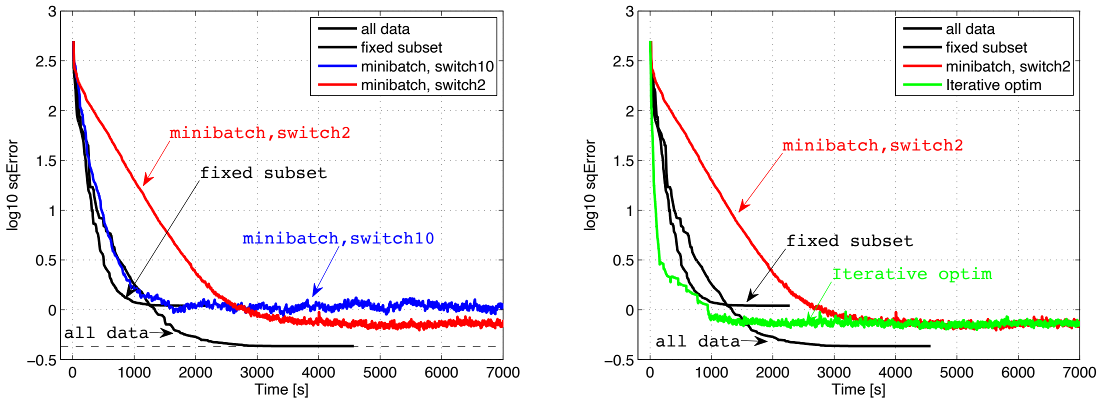
C.2 Natural Images, Section 5: Reducing Computation Time in the Optimization

The objective function J_T in Equation (8) is defined through an sample average. In an iterative optimization scheme, not all the data may be used to compute the average. The reason for using a smaller subset of the data can lie in memory considerations or in the desire to speed up the computations. We analyze here what statistical cost (reduction of estimation accuracy) such a optimization scheme implies. Furthermore, we show that optimizing J_T for increasingly larger values of ν reduces computation time without affecting estimation accuracy. The presented results were obtained by using the the nonlinear conjugate gradient algorithm of Rasmussen (2006) for the optimization.

As working example, we consider the unnormalized Gaussian distribution of Section 3.1 for $n = 40$. Estimating the precision matrix and the normalizing parameter means estimating 821 parameters. We use $T_d = 50000$, and $\nu = 10$. We assume further that, for whatever reason, it is not feasible to work with all the data points at the same time but only with $\tilde{T}_d = 25000$ samples (although for the present example, it is of course possible to use all the data).

C.2.1 RESULTS

The lower black curve in Figure 20(a) shows the performance for the hypothetical situation where we could use all the data. The mean squared error (MSE) reaches the level which Corollary 4 predicts (dashed horizontal line). This is the smallest error which can be obtained with noise-contrastive estimation for $\nu = 10$ and $T_d = 50000$. The upper black curve in the same figure shows the MSE when only a fixed subset with $\tilde{T}_d = 25000$ data points is used in the optimization. This clearly leads to less precise estimates. The performance can, however, be improved by randomly choosing a new subset of size $\tilde{T}_d$ after two updates of the parameters (red curve). The improved performance comes, however, at the cost of slowing down convergence. If the resampling of the



(a) Increasing accuracy

(b) Increasing accuracy and speed of convergence

Figure 20: Analysis of the optimization strategy in Section 5. See Section C.2 for details.

subset is switched at a lower rate, for example, after 10 updates, the speed of convergence stays the same but the accuracy does not improve (blue curve).

Figure 20(b) shows the proposed optimization strategy, which we also use in Section 5 for the simulations with natural image data: We iteratively optimize J_T for increasingly larger values of ν. Whenever we increase ν to $\nu+1$, we also take a new subset. When ν reaches its maximal value, which is here $\nu = 10$, we switch the subset after two parameter updates. For the other values of ν, we switch the subsets at a lower rate of 50 iterations. The results for this optimization strategy are shown in green (curve labelled "iterative optim"). It speeds up convergence while achieving the same precision as in the optimization with resampled subsets of size $\tilde{T}_d$ alone (red curve in Figures (a) and (b)). By resampling new subsets, all the data are actually used in the optimization. However, the estimation accuracy is clearly worse than when all the data are used at once (as in the lower black curve). Hence, there is room for improvement in the way the optimization is performed.

C.3 Natural Images, Section 5.4: Details for the Spline-Based One-Layer Model

The one-layer model that we consider here is

$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(\mathbf{w}_k^T\mathbf{x}; a_1, a_2, \ldots) + c,$$

where the nonlinearity f is a cubic spline. While the two-layer models in Section 5.3 and Section 5.4 were hardcoded to assign the same value to $\mathbf{x}$ and $-\mathbf{x}$, here, no symmetry assumption is made. The parameters are the feature weights $\mathbf{w}_k \in \mathbb{R}^n$, $c \in \mathbb{R}$ for the normalization of the pdf, as well as the $a_i \in \mathbb{R}$ for the parameterization of the nonlinearity f. For the modeling of the nonlinearity, its domain needs to be defined. Its domain is related to the range of its arguments $\mathbf{w}_k^T\mathbf{x}$. To avoid ambiguities in the model specification, we constrain the vectors as in Equation (26). Defining f as a cubic spline on the whole real line is impossible since the number of parameters a_i would become intractable. With the constraint in Equation (26), it is enough to define f only on the interval $[-10\ 10]$ as a cubic spline. For that, we use a knot sequence with an equal spacing of 0.1. Outside the interval, we define f to stay constant. With these specifications, we can write f in terms of B-spline basis functions with 203 coefficients $a_1, \dots, a_{203}$.

C.3.1 RESULTS

The learned features are "Gabor-like" (results not shown). We observed, however, a smaller number of feature detectors that are tuned to low frequencies. Figure 16(a) in Section 5.4 shows the learned nonlinearity f (black solid curve) and the random initialization (blue dashed curve). The dashed vertical lines indicate the interval where 99% of the feature outputs occur for natural image input. The learned nonlinearity should thus only be considered valid on that interval. The nonlinearity has two striking properties: First, it is an even function. Note that no such constraint was imposed, so the symmetry of the nonlinearity is due to the symmetry in the natural images. This result validates the symmetry assumption inherent in the two-layer models. It also updates a previous result of ours where we have searched for f in a more restrictive space of functions and no symmetric nonlinearity emerged (Gutmann and Hyvärinen, 2009). Second, f is not monotonic. The shape of f is closely related to the sparsity of the feature outputs $\mathbf{w}_k^T\mathbf{x}$. Since the absolute values of the feature outputs are often very large or very small in natural images, f tends to map natural images to larger numbers than the noise input. This means that the model assigns more often a higher probability density to natural images than to the noise.

C.4 Natural Images, Section 5.5: Refinement of the Thresholding Model

We are taking here a simple approach to the estimation of a two-layer model with spline nonlinearity f: We leave the feature extraction layers that were obtained for the thresholding model in Section 5.3 fixed, and learn only the cubic spline f. The model is thus

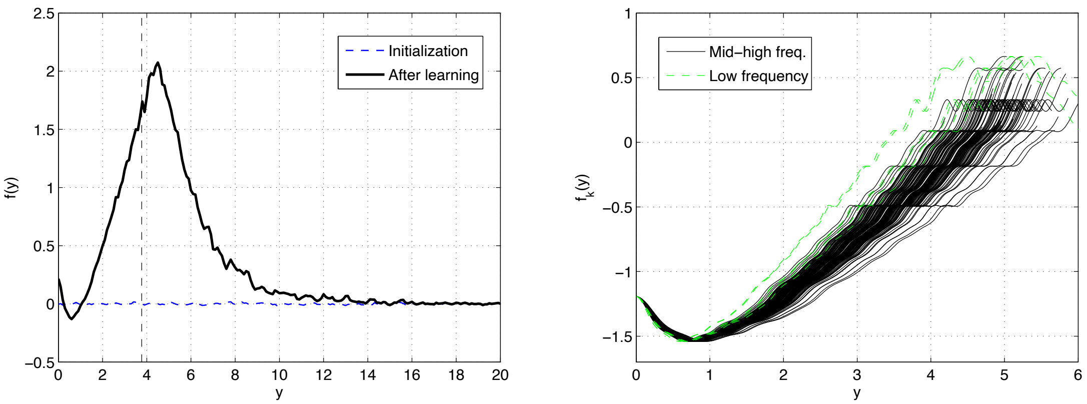
$$\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{n} f(y_k; a_1, a_2, \dots) + c, \qquad\qquad y_k = \sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2,$$

where the vector $\boldsymbol{\theta}$ contains the parameters a_i for f and the normalizing parameter c. The knots of the spline are set to have an equal spacing of 0.1 on the interval $[0\ 20]$. Outside that interval, we define f to stay constant. With that specification, we can write f in terms of 203 B-spline basis functions. The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{204}$ contains then the 203 coefficients for the basis functions and the parameter c.

C.4.1 RESULTS

Figure 21(a) shows the learned nonlinearity (black solid curve) and its random initialization (blue dashed curve). The dashed vertical line around $y = 4$ indicates the border of validity of the nonlinearity since 99% of the y_k fall, for natural image input, to the left of the dashed line. The salient property of the emerging nonlinearity is the "dip" after zero which makes f non-monotonic, as the nonlinearity which emerged in Section 5.4. Figure 21(b) shows the effective nonlinearities f_k when the different scales of the second layer outputs y_k and the normalizing parameter c are taken into account, as we have done in Figure 14(a). We calculated the scale σ_k by taking the average value of y_k over the natural images. The different scales σ_k then define different nonlinearities. Incorporating the normalizing parameter c into the nonlinearity, we obtain the set of effective nonlinearities $f_k(y)$,

$$f_k(y) = f(\sigma_k y) + c/n,\ k = 1, \dots n. \tag{36}$$



(a) Learned nonlinearity

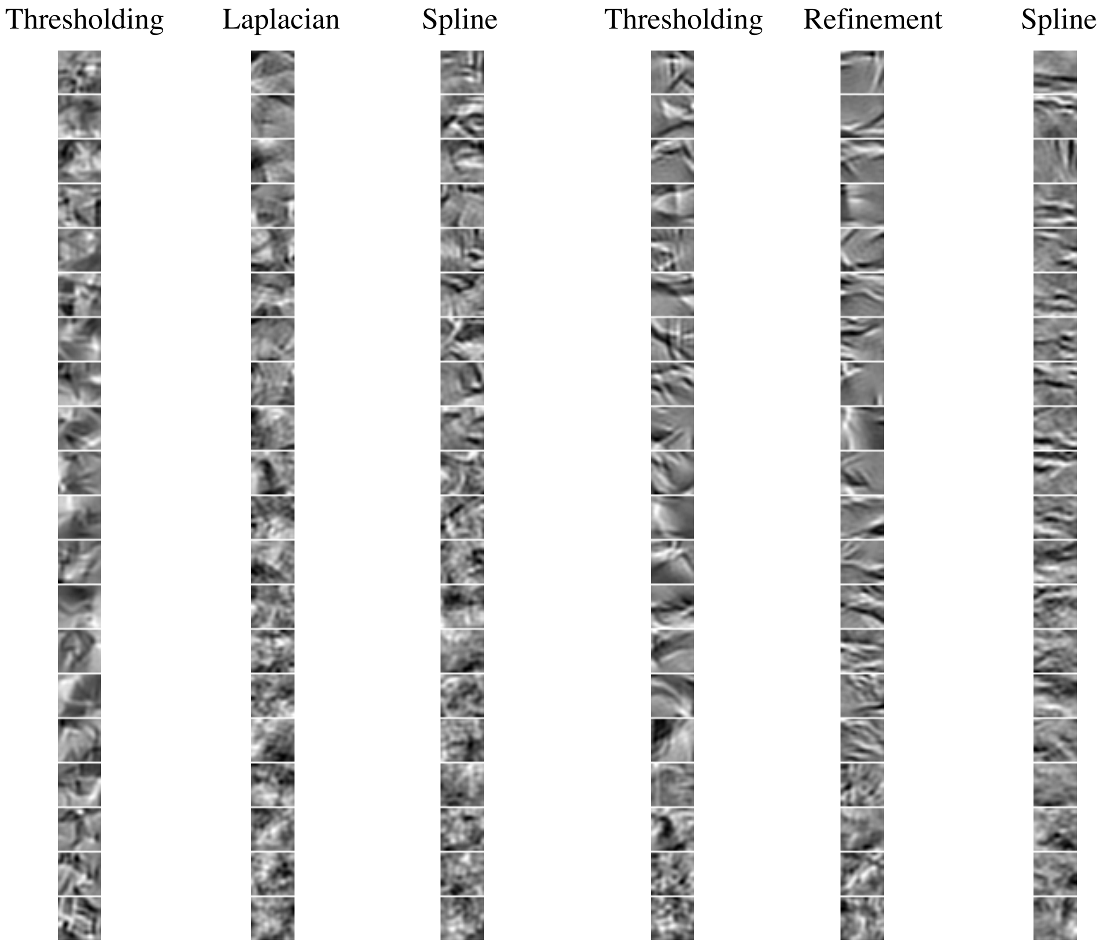
(b) Learned effective nonlinearities

Figure 21: Refinement of the thresholding model of Section 5.3. Only the nonlinearity was learned, the features were kept fixed. The features are shown in Figures 11 to 13. (a) Learned spline (black solid curve) and the initialization (blue dashed curve). The dashed vertical line indicates the border of validity of the learned nonlinearity since 99% of the y_k fall, for natural image input, to the left of it. (b) The different scales of the y_k give rise to a set of effective nonlinearities f_k, as defined in Equation (36). Nonlinearities acting on low-frequency feature detectors are shown in green (dashed lines), the others in black (solid lines), as in Figure 14(a).

For the nonlinearities f_k, the dip occurs between zero and two. Inspection of Figure 14(b) shows that the optimal nonlinearities f_k take, unlike the thresholding nonlinearities, the distribution of the second-layer outputs y_k fully into account. The region where the dip occurs is just the region where noise input is more likely than natural image input. This means that the model is assigning more often a higher probability density to natural images than to the noise.

C.5 Natural Images, Section 5.5: Samples from the Different Models

In Figure 17, we compared images which are considered likely by the different models. In Figure 22, we show samples that we drew from the models using Markov chains (Hamiltonian Monte Carlo). Since the models are defined on a sphere, we constrained the Hamiltonian dynamics by projecting the states after each leapfrog step back onto the sphere. The number of leapfrog steps was set to 100, and the rejection rate to 0.35 (Neal, 2010, Section 4.4, p.30). The top row shows the most likely samples while the bottom row show the least likely ones. The least likely samples appear similar for all models. For the more probable ones, however, the two-layer models lead to more structured samples than the one-layer models.



(a) One-layer models

(b) Two-layer models

Figure 22: Sampling from the learned models of natural images. Figure (a) shows samples from the one-layer models, Figure (b) shows samples from the two-layer models. The samples are sorted so that the top ones are the most likely ones while those at the bottom are the least probable ones. See caption of Table 1 in Section 5.5 for information on the models used. Samples of the training data and the noise are shown in Figure 9 in Section 5.1.

References

C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

C.J. Geyer. On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society, Series B (Methodological)*, 56(1):261–274, 1994.

M. Gutmann and A. Hyvärinen. Learning features by contrasting natural images with noise. In *Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN)*, volume 5769 of *Lecture Notes in Computer Science*, pages 623–632. Springer Berlin / Heidelberg, 2009.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR W&CP*, pages 297–304, 2010.

T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

A. Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Optimal approximation of signal priors. *Neural Computation*, 20:3087–3110, 2008.

A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001a.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001b.

A. Hyvärinen, J. Hurri, and P.O. Hoyer. *Natural Image Statistics*. Springer, 2009.

Y. Karklin and M. Lewicki. A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Computation*, 17:397–423, 2005.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

U. Köster and A. Hyvärinen. A two-layer model of natural stimuli estimated with score matching. *Neural Computation*, 22(9):2308–2333, 2010.

J. Lücke and M. Sahani. Maximal causes for non-linear component extraction. *Journal of Machine Learning Research*, 9:1227–1267, 2008.

R.M. Neal. *Handbook of Markov Chain Monte Carlo*, chapter MCMC using Hamiltonian Dynamics. Chapman & Hall /CRC Press, 2010.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

S. Osindero and G. Hinton. Modeling image patches with a directed hierarchy of Markov random fields. In *Advances in Neural Information Processing Systems 20*, pages 1121–1128. MIT Press, 2008.

S. Osindero, M. Welling, and G. E. Hinton. Topographic product models applied to natural scene statistics. *Neural Computation*, 18 (2):381–414, 2006.

M. Pihlaja, M. Gutmann, and A. Hyvärinen. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 442–449. AUAI Press, 2010.

M.A. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2551–2558, 2010.

C.E. Rasmussen. Conjugate gradient algorithm, Matlab code version 2006-09-08. Downloaded from http://learning.eng.cam.ac.uk/carl/code/minimize/minimize.m. 2006.

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

N.N. Schraudolph and T. Graepel. Towards stochastic conjugate gradient methods. In *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, volume 2, pages 853–856, 2002.

W. Sun and Y. Yuan. *Optimization Theory and Methods: Nonlinear Programming*. Springer, 2006.

Y. Teh, M. Welling, S. Osindero, and G. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, 2004.

T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071, 2008.

J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1394):359–366, 1998.

Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

L. Wasserman. *All of Statistics*. Springer, 2004.

L. Younes. Parametric inference for imperfectly observed Gibbsian fields. *Probability Theory and Related Fields*, 82(4):625–645, 1989.

Journal of Machine Learning Research 13 (2012) 363-387 Submitted 5/10; Revised 8/11; Published 2/12

Bounding the Probability of Error for High Precision Optical Character Recognition

Gary B. Huang GBHUANG@CS.UMASS.EDU
Andrew Kae AKAE@CS.UMASS.EDU
Department of Computer Science
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003

Carl Doersch CDOERSCH@CS.CMU.EDU
Department of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Erik Learned-Miller ELM@CS.UMASS.EDU
Department of Computer Science
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003

Editor: Leon Bottou

Abstract

We consider a model for which it is important, early in processing, to estimate some variables with high precision, but perhaps at relatively low recall. If some variables can be identified with near certainty, they can be conditioned upon, allowing further inference to be done efficiently. Specifically, we consider optical character recognition (OCR) systems that can be bootstrapped by identifying a subset of correctly translated document words with very high precision. This "clean set" is subsequently used as document-specific training data. While OCR systems produce confidence measures for the identity of each letter or word, thresholding these values still produces a significant number of errors.

We introduce a novel technique for identifying a set of correct words with very high precision. Rather than estimating posterior probabilities, we **bound** the probability that any given word is incorrect using an approximate worst case analysis. We give empirical results on a data set of difficult historical newspaper scans, demonstrating that our method for identifying correct words makes only two errors in 56 documents. Using document-specific character models generated from this data, we are able to reduce the error over properly segmented characters by 34.1% from an initial OCR system's translation.[1]

Keywords: optical character recognition, probability bounding, document-specific modeling, computer vision

[1] This work is an expanded and revised version of Kae et al. (2010). Supported by NSF Grant IIS-0916555.

1. Introduction

A long-standing desire in classification has been the ability to adapt a model specifically to a given test instance. For instance, if one had a reliable method for gauging the probability of correctness of an initial set of predictions, then one could iteratively use the predictions likely to be correct to refine the classification model and adapt to the specific test distribution. This general strategy has been considered in the past (Ho, 1998; Hong and Hull, 1995b,c), but particularly within the domain of computer vision, it has not had much success. We believe that this lack of success stems from the difficulty of reliably estimating probabilities in the high dimensional vector spaces common in computer vision. Rather than attempting to reliably estimate probabilities for all predictions, we instead propose a shift in perspective, focusing on identifying cases where we can reliably bound probabilities.

We show that this old idea of using a first pass system to identify some reliable samples, which are then used in turn to train a second pass system, can be quite powerful when the method of selecting reliable samples is appropriate. In particular, by formally upper bounding the probability of error in a first pass system, we can select results whose probability of error is not greater than some very small threshold, leading to the automatic selection of a subset of results of the first pass system with very low error rate. These results can be considered highly reliable "training data", specific to the test distribution, for a second pass system. Using this test-specific training data, we demonstrate significant error reductions on some difficult OCR problems. Thus, the main contribution of our paper is the combination of standard bounding techniques with the idea of multi-pass test-specific classification systems. To our knowledge, there is no preceding work which does this.

We first describe why adapting a model to a specific test distribution is an important goal, and in Section 2, discuss our rationale for bounding rather than estimating probabilities.

1.1 Adapting to the Test Distribution

In supervised learning, we are given training data $\{(x_i, y_i)\}$ to learn a model capable of predicting variables y from observations x, and apply this model at test time to new, previously unseen observations x'. An important implicit assumption in this framework is that the training instances (x_i, y_i) are drawn from the same distribution as the test instances (x', y'). Unfortunately, however, this is often not the case, and when this assumption is violated, the performance of supervised learning techniques can decay rapidly.

One natural setting in which this scenario arises is text recognition. In everyday life, we encounter a variety of fonts and character appearances that differ widely from each other and may be entirely new to us, such as in outdoor signs, graffiti, and handwritten messages. Despite not having appropriate labeled training examples, as humans we would be able to quickly adapt and recognize such text, whereas a machine learning algorithm would not.

There are several methods of addressing this problem. We may attempt to leverage knowledge from a closely related task and apply that knowledge to solve the new test problem, as in transfer learning. Alternatively, we may attempt to explicitly parameterize and model the manner in which the data varies, as in a hierarchical Bayes model. Instead, we argue for a third, non-parametric option, inspired by human behavior.

When presented with text in an unusual font, or with a new situation in general, we argue that humans will first identify elements that they are very confident in their understanding of, based on previous experience. For instance, they may be able to identify a particular letter based on similarity to previously seen fonts or by the occurrence statistics. Once they have done this, they will condition on this information and use it as an aid to understanding the remaining elements.

Similarly, we argue that a machine learning algorithm could benefit by first understanding, with very low probability of being incorrect, some subset of the new test instance, and conditioning on this information as training data specific to the test case.

Ideally, rather than making a hard decision and potentially throwing away useful information, we would like to maintain a distribution over the possible interpretations, such as a distribution over the possible characters a particular letter could be. We could then reason probabilistically over the different joint labelings of all characters to determine a maximum a posteriori estimate. In practice, however, we believe this has two pitfalls. The first is the difficulty in obtaining accurate distributions over labels, especially when dealing with very high dimensional data, as we describe later. These initial errors can then propagate as we do further reasoning. The second is the computational complexity of performing learning and inference on such distributions over labelings. Instead, by making a hard decision, we can make learning and inference much more efficient, and make use of the conditioned information as test-case specific training data with minor modifications to standard algorithms. In essence, by making hard decisions only where we are very confident of the labeling, we gain the computational efficiencies associated with making such decisions without the common risk of making unrecoverable errors.

1.2 Document-Specific OCR

In this paper, we focus on the problem of improving optical character recognition (OCR) performance on difficult test cases. In these instances, the non-stationarity between the distribution in the training examples and distribution in the test cases arises due to factors such as non-standard fonts and corruption from noise and low resolution.

Applying the reasoning above, we would like to obtain training data from the test documents themselves. In this paper, we use the output from an OCR program and identify a list of words which the program got correct. We can then use these correct words to build new, *document-specific* OCR models.

While identifying correct words in OCR program output may seem like an easy thing to do, to our knowledge, there are no existing techniques to perform this task with very high accuracy. There are many methods that could be used to produce lists of words that are mostly correct, but contain some errors. Unfortunately, such lists are not much good as training data for document-specific models since they contain errors, and these errors in training propagate to create more errors later.

Although some classifiers may be robust to errors in the training data, this will be very dependent on the number of training examples available. For characters such as `'j'` that appear less frequently, having even a few errors may mean that more than half of the training examples are incorrect. While we can tolerate some errors in character sets such as `'e'`, we cannot tolerate them everywhere.

Thus, it is essential that our error rate be very low in the list of words we choose as correct. As described below, our error rate is less than 0.002, as predicted by our theoretical bounds, making our generated lists appropriate for training document-specific models.

We first give some background on why we believe this problem of bounding probabilities to achieve high-precision, document-specific training data is interesting. In Section 3, we present the specifics of our method for creating nearly error-free training sets, and give theoretical bounds on the probability of error in these sets in Section 4. We then describe how we use the document-specific model to reduce the error rate in Section 5, and give experimental set-up and results in Section 6. Finally, we conclude with directions for future research in OCR, as well as potential applications of our method to other problem domains, in Section 8.

2. Background

Humans and machines both make lots of errors in recognition problems. However, one of the most interesting differences between people and machines is that, for some inputs, humans are extremely confident of their results and appear to be well-justified in this confidence. Machines, on the other hand, while producing numbers such as posterior probabilities, which are supposed to represent confidences, are often wrong even when posterior probabilities are extremely close to 1.

This is a particularly vexing problem when using generative models in areas like computer vision and pattern recognition. For example, consider a two class problem in which we are discriminating between two similar image classes, A and B. Because images are so high-dimensional, likelihood exponents are frequently very small, and small percentage errors in these exponents can render the posteriors meaningless. For example, suppose that $Pr(\texttt{image}|\texttt{A}) = \exp(-\texttt{1000} + \varepsilon_{\texttt{A}})$ and $Pr(\texttt{image}|\texttt{B}) = \exp(-\texttt{1005} + \varepsilon_{\texttt{B}})$, where ε_A and ε_B represent errors in the estimates of the image distributions.[2] Assuming a roughly equal prior on A and B, if ε_A and ε_B are Gaussian distributed with standard deviation a small proportion (for instance, around 1%) of the magnitude of the exponents, the estimate of the posterior will be extremely sensitive to the error. In particular, we will frequently conclude, incorrectly, that $Pr(B|\texttt{image}) \approx \texttt{1}$ and $Pr(A|\texttt{image}) \approx \texttt{0}$. This phenomenon, which is quite common in computer vision, makes it quite difficult to assess confidence values in recognition problems.

Rather than estimating posterior probabilities very accurately in order to be sure of certain results, we suggest an alternative. We formulate our confidence estimate as an hypothesis test that a certain result is *incorrect*, and if there is sufficient evidence, we reject the hypothesis that the result is incorrect. As we shall see, this comes closer to *bounding* the probabilities of certain results, which can be done with greater confidence, than *estimating* the probability of results, which is much more difficult. A critical aspect of our approach is that if there is insufficient evidence to reject a hypothesis, then we make no judgment on the correctness of the result. Our process only makes decisions when there is enough evidence, and avoids making decisions when there is not.

One interesting aspect of our work is that we make use of our bounding result as an important intermediate step in our overall system. In general, bounds given in machine learning are used to give theoretical justification for pursuing a particular algorithm and to gain insights on why they work. For instance, variational mean field inference can be viewed as optimizing a lower bound on the log partition function (Koller and Friedman, 2009).

In contrast, we make active use of our bound to guarantee that our document-specific training data will be nearly error-free. In this way, our bound plays in an integral role in the system itself, rather than as an analysis of the system.

[2] Such errors are extremely difficult to avoid in high-dimensional estimation problems, since there is simply not enough data to estimate the exponents accurately.

2.1 OCR and Document-Specific Modeling

Despite claims to the contrary, getting OCR systems to obtain very high accuracy rates on moderately degraded documents continues to be a challenging problem (Nagy, 2000). One promising approach to achieving very high OCR accuracy rates is to incorporate *document-specific modeling* (Ho, 1998; Hong and Hull, 1995b,c). This set of approaches attempts to refine OCR models to specifically model the document currently being processed by adapting to the fonts in the document, adapting to the noise model in the document, or adapting to the lexicon in the document.

If one had some method for finding a sample of words in a document that were known to be correct with high confidence, one could effectively use the characters in such words as training data with which to build document-specific models of the fonts in a document. Resolving this circular-dependency problem is not easy, however.

To tackle this problem of producing "clean word lists" for document-specific modeling, we consider a somewhat different approach. Rather than trying to estimate the probability that an intermediate output of an OCR system (like an HMM or CRF) is correct and then thresholding this probability, we instead form a set of hypotheses about each word in the document. Each hypothesis poses that one particular word of the first-pass OCR system is incorrect. We then search for hypotheses that we can reject with high confidence. More formally, we treat a third party OCR system (in this case, the open source OCR program Tesseract (`http://code.google.com/p/tesseract-ocr/`) as a null hypothesis generator, in which each attempted transcription T produced by the OCR system is treated as the basis for a separate null hypothesis. The null hypothesis for word T is simply "*Transcription T is incorrect*.". Letting W be the true identity of a transcription T, we notate this as

$$T \neq W.$$

Our goal is to find as many hypotheses as possible that can be rejected *with high confidence*. In this paper, we take high confidence to mean with fewer than 1 error in a thousand rejected hypotheses. As we mention later, we only make 2 errors in 4465 words in our clean word lists, even when they come from quite challenging documents.

Before proceeding, we stress that the following are **not** goals of this paper:

- to present a complete end-to-end system for OCR,
- to produce accurate estimates of the probability of error of particular words in OCR.

Once again, our goal is to produce large lists of clean words from OCR output and demonstrate how they can be used for document-specific modeling. After presenting our method for producing clean word lists, we provide a formal analysis of the bounds on the probability of incorrectly including a word in our clean word list, under certain assumptions. When our assumptions hold, our error bound is very loose, meaning our true probability of error is much lower. However, some documents do in fact violate our assumptions.

We analyze this approach, and find that, with modest assumptions, we can bound the probability that our method produces an error at less than 0.002. Moreover, as a first-step validation of our general approach, we give a simple method for building a model from the document-specific data that significantly reduces the character error on a difficult, real-world data set.

We also compare our method with using the built-in confidence measure of a public domain OCR system, and thresholding this value to produce document-specific training data. We find that this method produces results that are less consistent and worse at reducing character error than our method.

2.2 Related Work

Our approach has ties with both prior work in OCR as well as methods outside of OCR, such as in image retrieval. We give a survey of related work below.

2.2.1 IN OCR

There has been significant work done in making use of the output of OCR in an iterative fashion, although all different from the work we present here. Kukich (1992) surveyed various methods to correct words, either in isolation or with context, using natural language processing techniques. Isolated-word error correction methods analyze spelling error patterns, for example, by deriving heuristics for common errors or by examining phonetic errors, and attempting to fix these errors through techniques such as minimum edit distance, *n*-gram statistics, and neural networks. Context-dependent word correction methods include using statistical language models such as word *n*-gram probabilities to correct errors using neighboring words.

Kolak (2003) developed a generative model to estimate the true word sequence from noisy OCR output. They assume a generative process that produces words, characters, and word boundaries, in order to model segmentation and character recognition errors of an OCR system. The model can be trained on OCR output paired with ground truth and then used to post-process and correct additional OCR output by finding the set of words, characters, and word boundaries that maximize the probability of the observed labeling.

Our work is distinguished from the above mentioned methods in that we examine the document images themselves to build document-specific models of the characters. A similar idea was used by Hong and Hull (1995a), who examined the inter-word relationships of character patches to help constrain possible interpretations. Specifically, they cluster whole word images and use majority voting of the associated OCR labels to decide on the correct output and create character image prototypes. This information is then used to correct additional errors by examining sub-patterns (e.g., a word is a prefix of another word) and decompositions of unknown words into known word patterns using the document images. Our work extends these ideas to produce clean, document-specific training data that can then be used in other methods, rather than only using potentially noisy labels through sub-pattern and decomposition analysis.

Our work is also related to a variety of approaches that leverage inter-character similarity in documents in order to reduce the dependence upon a priori character models. One method for making use of such information is to treat OCR as a cryptogram decoding problem, which dates back to Casey (1986) and Nagy (1986). After performing character clustering, decoding can be performed by a lexicon-based method (Ho and Nagy, 2000) or using hidden Markov models (Lee, 2002); however, such methods are limited by the assumption that characters can be clustered cleanly into pure clusters consisting of only one character. This particular problem can be overcome by solving the decoding problem iteratively, using word and character statistics to first decode least ambiguous characters, then to iteratively decode progressively more difficult characters (Kae and Learned-Miller, 2009).

An alternative approach to obtaining document-specific character models is presented by Edwards and Forsyth (2005), using an iterative algorithm to extract character templates from high confidence regions. One major difference is that we provide a theoretical bound on the number of errors expected using our algorithm to identify highly confident words. Another significant difference is that the authors provide a small amount of manually defined training data in their application, whereas we provide none.

Another method for leveraging inter-character similarity is to perform some type of character clustering. Hobby and Ho (1997) perform clustering in order to replace individual, potentially degraded character images, with a smoothed image over the cluster. Breuel (2003) learns a probabilistic similarity function to perform nearest-neighbor classification of characters.

The inability to attain high confidence in either the identity or equivalence of characters in these papers has hindered their use in subsequent OCR developments. We hope that the high confidence values we obtain will spur the use of these techniques for document-specific modeling.

2.2.2 OTHER WORK

Outside of OCR, our work is similar to Leisink and Kappen (2003), which deals with inference in graphical models for which exact inference is intractable. As an alternative to approximate inference techniques (which may bound a different quantity, the log partition function), they directly bound the marginal probabilities at each node in an iterative process called bound propagation. Each iteration consists of solving a linear program, where some of the constraints are due to bounds computed by previous iterations.

The end product of bound propagation is an upper and lower bound for each of the marginal probabilities of the nodes in the graphical model, with no guarantee on the tightness of any particular bound. In contrast, our work focuses on finding the subset of words for which we can put a very tight bound on the probability of error, and thus is a different approach under the general idea of bounding probabilities.

Our work is also related to the problem of covariate shift (Shimodaira, 2000), in which it is assumed that the conditional distributions $p(y|x)$ remain the same for both the training and test distributions, but the distribution on the observations $p(x)$ may differ. In this case, letting $p_0(x)$ be the distribution for the training set, and $p_1(x)$ be the distribution for a test set, one can reweight the log likelihood of the training instances with $\frac{p_1(x)}{p_0(x)}$. The principal difficulty is estimating this ratio. In particular, in OCR, test documents may have a range of degradation and noise, and potentially unseen font models, and thus the support of $p_0(x)$ may potentially not contain the support of $p_1(x)$, in which case a re-weighting approach could not be applied. Moreover, noise and font appearance specific to the test document may also lead to a change in $p(y|x)$ for ambiguous or noisy x. Instead, our work attempts to identify highly confident labelings (x',y') in order to characterize the test-specific distribution over appearance and labels.

Another area closely related to the method presented in this paper is the meta-recognition work of Scheirer et al. (2011). They consider the problem of multiclass recognition, such as object or face recognition. A given test image produces a set of scores indicating how well the test image matched each class. Since the test image can belong to at most one class, all but the highest returned score can be used to model the distribution of non-matching scores, specific to the single test image. The authors use some fraction of the top non-matching scores produced for a test image to model the tail of the non-matching distribution using extreme value theory, and then use this distribution to normalize the top matching score.

Similar to our work, the tail distribution that is modeled can be used to attempt to reject the null hypothesis that the top matching score belongs to the non-matching distribution. Our work differs in that we specifically focus only on cases where we can reject this null hypothesis with very high confidence. To do so, we leverage the appearance of the entire document, which allows us to be more robust to cases where the test distribution differs substantially from the training distribution.

The idea of identifying objects which can confidently be given a particular label is also an important component of query expansion in the information retrieval field. Query expansion is a technique used to add terms to an initial query based on the highly ranking documents of the initial query. In Chum et al. (2007), query expansion is used for image retrieval where the initial results of an image query are processed to find resulting images that the system is confident match the initial query. The confidence in a particular match is evaluated using a spatial verification scheme that is similar to our consistency check presented below. This verification is critical to query expansion, as false positives can lead to drift, causing irrelevant features to be added to the expanded query. Later, we propose a possible extension to see whether our bound analysis can be applied to give a bound on the probability of a false match passing the spatial verification.

Building models specific to a test image has also been applied in other areas of computer vision. In work by Nilsback and Zisserman (2007), an initial, general flower model is applied to an image to segment a flower from the background. This initial segmentation is used to build an image-specific color model of the foreground flower, and this process of segmentation and color estimation is iterated until convergence.

Berg et al. (2007) follow a similar approach to image parsing, first extracting a per pixel segmentation of the image, then using pixels with high confidence to learn an image-specific color model of sky and building. Ramanan (2006) uses an initial edge model to infer the pose of a person in the image, then uses this to build an image-specific color model, and iterates until convergence. These methods can be sensitive to the initial steps, underscoring the need for high precision in constructing image-specific models. Sapp et al. (2010) take a slightly different approach by using similarity between a test image and a set of training exemplars and kernel regression to learn image-specific model parameters, and then performing inference with the image-specific model.

3. Method for Producing Clean Word Lists

In this section, we present our method for examining a document bitmap and the output of an OCR system for that document to produce a so-called *clean word list*, that is, a list of words which we believe to be correct, with high confidence. Our success will be measured by the number of words that can be produced, and whether we achieve a very low error rate in the clean list. Ideally, we must produce a clean word list which is large enough to provide sufficient training data for document-specific modeling.

We assume the following setup.

- We are provided with a document D in the form of a grayscale image.

- We are provided with an OCR system.

- We further assume that the OCR system provides an *attempted* segmentation of the document D into words, and that the words are segmented into characters. It is not necessary that the segmentation be entirely correct, but merely that the system produces an attempted segmentation.

- In addition to a segmentation of words and letters, the system should produce a best guess for every character it has segmented, and hence, by extension, of every word (or string) it has segmented. Of course, we do not expect all of the characters or words to be correct, as that would make our exercise pointless.

- Using the segmentations provided by the OCR system, we assume we can extract the gray-valued bitmaps representing each guessed character from the original document image.
- Finally, we assume we are given a lexicon. Our method is relatively robust to the choice of lexicon, and assumes there will be a significant number of non-lexicon words in the document.

We define a few terms before proceeding. The *Hamming distance* between two strings of the same number of characters is the number of character substitutions necessary to convert one string to the other. The *Hamming ball* of radius r for a word W, $H_r(W)$, is the set of strings whose Hamming distance to W is less than or equal to r. Later, after defining certain equivalence relationships among highly confusable characters such as 'o' and 'c', we define a *pseudo-Hamming distance* which is equivalent to the Hamming distance except that it ignores substitutions among characters in the same equivalence class. We also use the notions of edit distance, which extends Hamming distance by including joins and splits of characters, and pseudo-edit distance, which is edit distance using the aforementioned equivalence classes.

Our method for identifying words in the clean list has three basic steps. We consider each word T output by the initial OCR system.

1. If T is not in the lexicon, we discard it and make no attempt to classify whether it is correct. That is, we do not put it on the clean word list.[3]
2. Given that T is a lexicon word, we evaluate whether $H_1(T)$ is non-empty, that is, whether there are any lexicon words for which a single change of a letter can produce T. If $H_1(T)$ is non-empty, we discard T and again make no attempt to classify whether it is correct.
3. Assuming we have passed the first two tests, we now perform a *consistency check* (described below) of each character in the word. If the consistency check is passed, we declare the word to be correctly recognized and include it in the clean list.

[3] Why is it not trivial to simply declare any output of an OCR system that is a lexicon word to be highly confident? The reason is that OCR systems frequently use language models to project uncertain words onto nearby lexicon words. For example, suppose the original string was “Rumpledpigskin”, and the OCR system, confused by its initial interpretation, projected “Rumpledpigskin” onto the nearest lexicon word “Rumplestiltskin”. A declaration that this word is correct would then be wrong. However, our method will not fail in this way because if the true string were in fact “Rumpledpigskin”, the character consistency check would never pass. It is for this reason that our method is highly non-trivial, and represents a significant advance in the creation of highly accurate clean word lists.

We could potentially restrict our attention to OCR systems that did not project onto lexicon words, or for which it is possible to access intermediate results prior to such projection. For such results, it is much more likely that a word labeled as a lexicon word with an empty Hamming ball of some radius is, in fact, correctly labeled. We choose not to make such a restriction, both so that our method is more general, and because projecting uncertain words to nearby lexicon words can often substantially increase the labeling accuracy. In other words, by only considering labelings obtained without such projection, we may find far fewer words that we can confidently classify as being correctly labeled, due to the lower accuracy of the initial OCR system. The benefit of using a lexicon is evident in the scene text recognition work of Weinman et al. (2009). In this work, simply forcing all predicted words to be lexicon words led to a 3 percentage point increase in word accuracy, and incorporating factors with lexicon information into the probability model led to an additional 5 percentage point increase in word accuracy. By performing a more robust analysis than accepting lexicon words, our method is equally applicable to sophisticated OCR systems that make use of lexicon information.

3.1 Consistency Check

In the following discussion, we use the term *glyph* to refer to a rectangular portion of an image that is likely to be a single character, but may be only a portion of a character, multiple characters, or a stray mark. Let W_j be the true character class of the jth glyph of a word W, and let T_j be the initial OCR system's interpretation of the same glyph. The goal of a consistency check is to ensure that the OCR system's interpretation of a glyph is reliable. We will assess reliability by checking whether other similar-looking glyphs are usually interpreted the same by the OCR system.

To understand the purpose of the consistency check, consider the following situation. Imagine that a document contains a stray mark that does not look like any character at all, but was interpreted by the initial OCR system as a character. If the OCR system thought that the stray mark was a character, it would have to assign it to a character class like 't'. We would like to detect that this character is unreliable. Our scheme for doing this is to find other characters that are similar to this glyph, and to check the identity assigned to those characters by the initial OCR system. If a large majority of those characters are given the same interpretation by the OCR system, then we consider the original character to be reliable. Since it is unlikely that the characters closest to the stray mark are clustered tightly around the true character 't', we hope to detect that the stray mark is atypical, and hence unreliable.

More formally, to test a glyph g for reliability, we first find the M glyphs in the document that are most similar to g (using normalized correlation as the similarity measure). If a fraction θ of the M glyphs most similar to g have the character label c, then we say that the glyph g is θ-dominated by c. More precisely, we run the following procedure:

```
// $n$ :  vector storing the counts for each character $c$.
// $L$ :  set of character labels.
// $M$ :  number of glyphs to compare to.
$n[c] \leftarrow 0, \forall c \in L$
for $i \leftarrow 1$ to $M$ do
    $c \leftarrow$ label of character $i$th most similar to $g$
    $n[c] = n[c] + 1$
    if $\frac{n[c]}{i+1} > \theta$ then
        return $g$ is $\theta$-dominated by $c$
    end
end
return $g$ is undominated
```

Algorithm 1: Consistency check algorithm.

There are three possible outcomes of the consistency check. The first is that the glyph g is dominated by the same class c as the OCR system's interpretation of g, namely T_j. The second outcome is that g is dominated by some other class that does not match T_j. The third outcome is that g is undominated, meaning that the neighbors of g are relatively inconsistent. In the latter two cases, we declare the glyph g to be *unreliable*. The interpretation of glyph g is reliable only if g is dominated by the same class as the original OCR system. Furthermore, a word is included in the clean list only if all of the characters in the word are reliable.

The constants used in our experiments were $M = 20$ and $\theta = 0.66$. That is, we compared each glyph against a maximum of 20 other glyphs in our reliability check, and we insisted that a "smoothed" estimate of the number of similarly interpreted glyphs was at least 0.66 before declaring a character to be reliable. We now analyze the probability of making an error in the clean set, under a specific set of assumptions.

4. Theoretical Bound

For a word in a document, let W be the ground truth label of the word and T be the initial OCR system's labeling of the word. Consider the problem of trying to estimate the probability that the labeling was correct, $P(W = w_t | T = w_t)$. It is difficult to formulate a bound or performance guarantee on such an estimate, due to the non-stationarity in the sequence of words. The distribution and appearance of words is dependent on the topics and fonts present in the document containing the words, and any noise in the document, which may range from local, such as stray marks, to global, such as low contrast. Therefore, we would not be able to rely on a general *i.i.d.* assumption on the words.

Rather than attempting to estimate the probability that the labeling was correct, we circumvent the above problems by focusing on bounding the probability for a subset of words. Let C be a binary indicator equal to 1 if the word passed the consistency check. We want to upper bound the probability $\Pr(W \neq w_t | T = w_t, C = 1)$ when w_t is a lexicon word and has an empty Hamming ball of size 1. We decompose the probability into three terms:

$$\begin{aligned}
\Pr(W \neq w_t | T = w_t, C = 1) &= \sum_{w \neq w_t} \Pr(W = w | T = w_t, C = 1) \\
&= \sum_{w \neq w_t, w \in \text{Lex}} \Pr(W = w | T = w_t, C = 1) \\
&\quad + \sum_{w \neq w_t, w \notin \text{Lex}} \Pr(W = w | T = w_t, C = 1) \\
&= \sum_{w \neq w_t, w \in \text{Lex}, |w| = |w_t|} \Pr(W = w | T = w_t, C = 1) \\
&\quad + \sum_{w \neq w_t, w \in \text{Lex}, |w| \neq |w_t|} \Pr(W = w | T = w_t, C = 1) \\
&\quad + \sum_{w \neq w_t, w \notin \text{Lex}} \Pr(W = w | T = w_t, C = 1).
\end{aligned} \tag{1}$$

Our approach for bounding this probability will be to individually bound the three terms in Equation 1. The first term considers all words in the lexicon with the same length as w_t, and accounts for the most likely type of error. The second term considers all words in the lexicon, but with a different length from w_t, and so considers many more possible words resulting from segmentation errors, but each of which is much less likely to occur. Finally the third term considers words not in the lexicon, each of which occurs even less frequently.

To bound the first term, we can consider all words of a given Hamming distance i from w_t. Our strategy will then be to bound the contribution to the error from all words of Hamming distance i, enabling us to bound the total error as the sum of a geometric series. We can then follow the same approach to bound the next two terms, where we will instead need to consider edit distance rather than Hamming distance.

In order to bound these terms using a geometric series, we will need two initial steps. We will need an upper bound on the probability of the consistency check passing for a specific character when the label is incorrect (2ε), and a lower bound on the probability of the consistency check passing when the label is correct (δ). We explain these quantities in the next section. Next, we will need to relate these bounds on the character consistency check, $\Pr(C = 1|T = w_t, W)$, to the terms in the geometric series, $\Pr(W = w|T = w_t, C = 1)$, which we do in Section 4.2.

4.1 Bounding the Character Consistency Check

We will rewrite the $\Pr(W = w|T = w_t, C = 1)$ terms as bounds involving $\Pr(C = 1|T = w_t, W = w)$ using Bayes' rule. We will make the assumption that the individual character consistency checks are independent, although this is not exactly true, since there may be local noise that degrades characters in a word in the same way.

Assume that each character is formed on the page by taking a single true, latent appearance based on the font and the particular character class and adding some amount of noise. Let ε be an upper bound on the probability that noise has caused a character of any given class to look like it belongs to another specific class other than its own class. More formally, letting $p_c(a)$ be the probability of a character appearance a for a given class c under the noise model, ε satisfies, for all character classes $c_1, c_2, c_1 \neq c_2$,

$$\varepsilon > \int_{a|p_{c_1}(a)<p_{c_2}(a)} p_{c_1}(a)da. \qquad (2)$$

In order to obtain a small value for ε, and hence later a small probability of error, we revise Equation 2 to be a bound only on *non-confusable* character classes. In other words, since some character classes are highly confusable, such as 'o', 'c', and 'e', we ignore such substitutions when computing Hamming and edit distance. We'll refer to these distances as pseudo distances, so "mode" and "mere" have a true Hamming distance of 2 but a pseudo-Hamming distance of 1.

This is similar to defining an equivalence relation where confusable characters belong to the same equivalence class, and computing distance over the quotient set, but without transitivity, as, for example, 'h' may be confusable with 'n', and 'n' may be confusable with 'u', but 'h' may not necessarily be confusable with 'u'.

For a character to pass a consistency check with the label c_2 when the true underlying label is c_1, roughly one of two things must happen: (a) either the character was corrupted and looked more like c_2 than c_1, or (b) some number of other characters with label c_2 were corrupted and looked like c_1's.

The probability (a) is clearly upper bounded by ε, since it requires both the corruption and most of its neighbors to have the same label c_2. Since $\varepsilon \ll 1$ and (b) requires several other characters with label c_2 to be corrupted to look like c_1, the probability of (b) should be bounded by (a), and thus ε, as well. Therefore the probability of the consistency check giving a label c_2 when the true underlying label is c_1 is less than 2ε, for any classes c_1, c_2.

We will also need a lower bound on the probability that a character consistency check will succeed if the OCR system's label of the character matches the ground truth label. Let δ be a lower bound on this quantity, which is dependent on both the amount of noise in the document and the length of the document. (The latter condition is due to the fact that the character consistency check requires a character to match to at least a certain number of other similarly labeled characters, so, for example, if that number is not present in the document to begin with, then the check will fail with certainty.)

4.2 Bounding One Term

Consider bounding $\Pr(W = w|T = w_t, C = 1)$:

$$\begin{aligned}
&\Pr(W = w|T = w_t, C = 1) \\
&\qquad = \frac{\Pr(C = 1|T = w_t, W = w)\Pr(W = w|T = w_t)}{\sum_{w'}\Pr(C = 1|T = w_t, W = w')\Pr(W = w'|T = w_t)} \\
&\qquad = \frac{\Pr(C = 1|T = w_t, W = w)\Pr(T = w_t|W = w)\Pr(W = w)}{\sum_{w'}\Pr(C = 1|T = w_t, W = w')\Pr(T = w_t|W = w')\Pr(W = w')} \\
&\qquad \leq \frac{\Pr(C = 1|T = w_t, W = w)\Pr(T = w_t|W = w)\Pr(W = w)}{\Pr(C = 1|T = w_t, W = w_t)\Pr(T = w_t|W = w_t)\Pr(W = w_t)}. \qquad (3)
\end{aligned}$$

Here the inequality follows from the fact that w_t is one of the words being summed over in the denominator, and hence replacing the sum with only the w_t component will make the denominator less than or equal to the sum.

4.3 Bounding the Probability of Lexicon Words

Recall that w_t, the initial OCR system's word labeling, is a lexicon word with empty pseudo-Hamming ball of size 1. For lexicon words w, we will assume that

$$\frac{\Pr(T = w_t|W = w)\Pr(W = w)}{\Pr(T = w_t|W = w_t)\Pr(W = w_t)} \quad < \quad 1,$$

or, equivalently,

$$\frac{\Pr(W = w|T = w_t)}{\Pr(W = w_t|T = w_t)} \quad < \quad 1. \qquad (4)$$

One way to view this is to think of $T = w_t$ as a feature. Then, for a reasonable classifier, this assumption should hold for any document in the training set, as this is simply the Bayes decision rule. (If the assumption did not hold, then we could increase the training accuracy by predicting w whenever we saw the feature $T = w_t$.)

Thus, we are assuming that a test document does not differ from the training documents used to train the initial OCR system so much as to change the most probable word conditioned on the feature $T = w_t$, as suggested by Equation 4.[4] Note that w_t has an empty Hamming ball of size 1, so w differs from w_t by at least two letters. For this assumption to be violated, either the document must be such that at least one letter is consistently interpreted as another, or has an extremely different prior distribution on words than that of the training set, both of which are unlikely. As we discuss later, the first case is also problematic for the character consistency check as well, and so falls outside the scope of documents for which our method will be applicable.

It is important to note that this does not imply that the word accuracy need be particularly high, for example, if all the words have the same prior probability of occurring, then the assumption could hold for a classifier with accuracy simply better than the chance accuracy of $\frac{1}{|\text{Lex}|}$, where $|\text{Lex}|$ is the size of the lexicon.

Applying this to Equation 3, we get

$$\Pr(W = w|T = w_t, C = 1) \quad \leq \quad \frac{\Pr(C = 1|T = w_t, W = w)}{\Pr(C = 1|T = w_t, W = w_t)}. \qquad (5)$$

[4] This assumption is similar to, but slightly weaker than, the assumption made under covariate shift (Shimodaira, 2000).

4.3.1 BOUNDING THE PROBABILITY OF LEXICON HAMMING WORDS

Consider a lexicon word w that is a pseudo-Hamming distance i from w_t. We can then simplify Equation 5 to

$$\Pr(W = w|T = w_t, C = 1) \quad \leq \quad \frac{(2\varepsilon)^i}{\delta^i}$$

by making use of the assumption that the character consistency checks are independent, and that w and w_t only differ in i characters. For those i characters, w does not match the OCR system's label and w_t does match the OCR system's label, so we use the bounds 2ε and δ.

Now let D_i be the number of lexicon words of pseudo-Hamming distance i away from w_t. Let r_D be the rate of growth of D_i as a function of i, that is, $D_{i+2} \leq r_D^i D_2$. Assume, since $\varepsilon \ll 1$, that $r_D(\frac{2\varepsilon}{\delta}) < \frac{1}{2}$.[5] (To produce a final number for our theoretical bound, we later assume that $\varepsilon < 10^{-3}$ and $\delta^2 > 10^{-1}$. Given these numbers, our assumption becomes $r_d < 79$, which is a very conservative bound as experiments on the lexicon used in our main experiments showed that r_D is generally bounded by 5.)

To get the total contribution to the error from all lexicon Hamming words, we sum over D_i for all $i > 1$,

$$\begin{aligned}\sum_{w \neq w_t, w \in \text{Lex}, |w| = |w_t|} \Pr(W = w|T = w_t, C = 1) \quad &\leq \quad \sum_{i=2} D_i \frac{(2\varepsilon)^i}{\delta^i} \\ &= \quad D_2 \frac{(2\varepsilon)^2}{\delta^2} + D_2 \frac{(2\varepsilon)^2}{\delta^2} \sum_{i=1} (2r_D \frac{\varepsilon}{\delta})^i \\ &\leq \quad 8D_2 \frac{\varepsilon^2}{\delta^2}.\end{aligned}$$

4.3.2 BOUNDING LEXICON EDIT WORDS

Traditionally, edit distance is computed in terms of number of substitutions, insertions, and deletions necessary to convert one string to another string. In our context, a more natural notion may be splits and joins rather than insertions and deletions. For example, the interpretation of an 'm' may be split into an 'r' and an 'n', or vice-versa for a join.

The probability that a split or a join passes the consistency check is upper bounded by $(2\varepsilon)^2$. We can see this from two perspectives. First, a split or join has traditional edit distance of 2, since it requires an insertion or deletion and a substitution ("m" to "mn" insertion followed by "mn" to "rn" substitution).

A more intuitive explanation is that, for a split, one character must be corrupted to look like the left hand side of the resulting character and another character corrupted to look like the right hand side, and for a join, the left hand side of a character must be corrupted to look like one character and the right hand side corrupted to look like another.

Similar to the case of confusable characters for substitutions, we also ignore confusable characters for splits and joins, namely 'r' followed by 'n' with 'm', and 'v' followed by 'v' with 'w'. Thus, "corn" and "comb" have an edit distance of 2 but a pseudo-edit distance of 1.

[5] Recall that δ, as defined earlier, is a lower bound on the probability that a character consistency check will succeed if the OCR system's label of the character is correct.

Consider a lexicon word w with pseudo-edit distance i from w_t, and involving at least one insertion or deletion (so $|w| \neq |w_t|$). Similar to the lexicon Hamming words, we can simplify Equation 5 for w as

$$\Pr(W = w | T = w_t, C = 1) \quad \leq \quad \frac{(2\varepsilon)^{i+1}}{\delta^i},$$

since each substitution contributes a $\frac{2\varepsilon}{\delta}$ and each insertion or deletion, of which there is at least one, contributes a $\frac{(2\varepsilon)^2}{\delta}$.

Let E_i be the number of lexicon words w with a pseudo-edit distance i away from w_t and $|w| \neq |w_t|$. Again, also assume that r_E, the rate of growth of E_i, satisfies $r_E(\frac{2\varepsilon}{\delta}) < \frac{1}{2}$. Summing the total contribution to the error from lexicon edit words,

$$\begin{aligned}
\sum_{w \neq w_t, w \in \text{Lex}, |w| \neq |w_t|} \Pr(W = w | T = w_t, C = 1) \quad &\leq \quad \sum_{i=1} E_i \frac{(2\varepsilon)^{i+1}}{\delta^i} \\
&= \quad E_1 \frac{(2\varepsilon)^2}{\delta} + E_1 \frac{(2\varepsilon)^2}{\delta} \sum_{i=1} (2r_E \frac{\varepsilon}{\delta})^i \\
&\leq \quad 8E_1 \frac{\varepsilon^2}{\delta} \\
&\leq \quad 8E_1 \frac{\varepsilon^2}{\delta^2}.
\end{aligned}$$

4.4 Bounding Non-Lexicon Words

Let N_i be the set of non-lexicon words with a pseudo-edit distance i from w_t, and let $p_i = \frac{\Pr(T=w_t|W\in N_i)\Pr(W\in N_i)}{\Pr(T=w_t|W=w_t)\Pr(W=w_t)}$. Assume the rate of growth of r_N of p_i satisfies $r_N(\frac{2\varepsilon}{\delta}) < \frac{1}{2}$.

Rearranging Equation 3 and summing over all non-lexicon words:

$$\begin{aligned}
&\sum_{w \neq w_t, w \notin \text{Lex}} \Pr(W = w | T = w_t, C = 1) \\
&\leq \sum_{i=1} \sum_{w \in N_i} \frac{\Pr(C = 1 | T = w_t, W = w) \Pr(W = w | T = w_t)}{\Pr(C = 1 | T = w_t, W = w_t) \Pr(W = w_t | T = w_t)} \\
&\leq \sum_{i=1} \sum_{w \in N_i} \frac{(2\varepsilon)^i}{\delta^i} \frac{\Pr(W = w | T = w_t)}{\Pr(W = w_t | T = w_t)} \\
&= \sum_{i=1} \frac{(2\varepsilon)^i}{\delta^i} \frac{\Pr(W \in N_i | T = w_t)}{\Pr(W = w_t | T = w_t)} \\
&= \sum_{i=1} \frac{(2\varepsilon)^i}{\delta^i} p_i \\
&\leq p_1 \frac{2\varepsilon}{\delta} + p_1 \frac{2\varepsilon}{\delta} \sum_{i=1} (2r_N \frac{\varepsilon}{\delta})^i \\
&\leq 4p_1 \frac{\varepsilon}{\delta^2}.
\end{aligned}$$

4.5 Final Bound

Combining each of the individual bounds derived above, we have

$$\Pr(W \neq w_t | T = w_t, C = 1) \quad \leq \quad \frac{(8D_2 + 8E_1)\varepsilon^2 + 4p_1\varepsilon}{\delta^2}.$$

To use this in practice, we need to set some realistic (but conservative) values for the remaining constants. For $\varepsilon < 10^{-3}, 8D_2 + 8E_1 < 10^2, 4p_1 < 10^{-1}, \delta^2 > 10^{-1}$,

$$\Pr(W \neq w_t | T = w_t, C = 1) \quad \leq \quad 2 \cdot 10^{-3}.$$

The bounds for the constants chosen above were selected conservatively to hold for a large range of documents, from very clean to moderately noisy. Not all documents will necessarily satisfy these bounds. In a sense, these inequalities define the set of documents for which our algorithm is expected to work, and for heavily degraded documents that fall outside this set, the character consistency checks may no longer be robust enough to guarantee a very low probability of error.

Our final bound on the probability of error, 0.002, is the result of a *worst case analysis* under our assumptions. If our assumptions hold, the probability of error will likely be much lower for the following reasons. For most pairs of letters, $\varepsilon = 10^{-3}$ is not a tight upper bound. The quantity on the right of Equation 4 is typically much lower than 1. The rate of growths r_D, r_E, r_N are typically much lower than assumed. The bound on p_1, the non-lexicon word probabilities, is not a tight upper bound, as non-lexicon words mislabeled as lexicon words are rare. Finally, the number of Hamming and edit distance neighbors D_2 and E_1 will typically be less than assumed.

On the other hand, for sufficiently noisy documents, and certain types of errors, our assumptions do not hold. Some of the problematic cases include the following. As discussed, the assumption that the individual character consistency checks are independent is not true. If a document is degraded or has a font such that one letter is consistently interpreted as another,[6] then that error will likely pass the consistency check (i.e., ε will be very large). If a document is degraded or is very short, then δ may be much smaller than $10^{-\frac{1}{2}}$. (The character consistency check requires a character to match to at least a certain number of other similarly labeled characters, so, for example, if that number isn't present in the document to begin with, then the check will fail with certainty.) Finally, if the lexicon is not appropriate for the document then $4p_1 < 10^{-1}$ may not hold. This problem is compounded if the OCR system projects to lexicon words. Still these assumptions appear to hold for a wide range of documents.

5. Character Recognition

To validate the utility of our clean word lists, we implemented a simple technique for constructing document-specific character appearance models, using SIFT features (Lowe, 2004),[7] and demonstrated that this model can be used to significantly reduce character error in the remainder of the document. We refer to our algorithm as SIFT_Align. In the future, we believe these clean word lists can be incorporated into more sophisticated document-specific OCR models to obtain further improvements in recognition accuracy, as we discuss in future work.

[6] It should be noted that the probability of such an error (consistently interpreting one letter as another) is substantially reduced by using a language model, for example, projecting uncertain words to nearby lexicon words.

[7] A SIFT feature is essentially computed by dividing the image into a set of non-overlapping patches, and computing a histogram over edge orientations weighted by edge strength, for each patch.

We use the traditional SIFT descriptor without applying the Gaussian weighting because we did not want to weight the center of an image more highly than the rest. In addition, we fix the scale to be 1 and orientation to be 0 at all times. The SIFT_Align procedure is presented below:

1. Compute the SIFT descriptor for each character image in the clean list, at the center of the image.

2. Compute the component-wise arithmetic mean of all SIFT descriptors for each character class in the clean list. These mean descriptors are the “representations” (or character models) of the respective classes.

3. For each character image in the clean list, compute a SIFT descriptor for each point in a window in the center of the image (we use a 5x5 window) and select the descriptor with smallest L2 distance to the mean SIFT descriptor for this character class. This aligns each character’s descriptor to the mean class descriptor.

4. Test images are defined as follows. We start by collecting all character images that are *not* in the clean list (since we do not want to test on images we trained on). We also filter the test images as follows. If Tesseract gives a label to an image that is *not* a label for any of the clean set characters, then we do not include this character in our test set. The rationale for this is the following. Since our method will only assign to a character a label that appears in the clean set, then if the character was originally correct, we will definitely introduce an error by attempting to correct it. Furthermore, we have no direct information about the appearance of characters whose labels are *not* in the clean set, so it is relatively difficult to assess if the original label is unreasonable. For these reasons, we only attempt to correct characters whose Tesseract label appears as one of the clean set labels.

5. For each test image, again compute a 5x5 window of 25 SIFT descriptors, and select the descriptor which has minimum L2 distance to *any* of the mean descriptors. This aligned descriptor is the final descriptor for the test image.

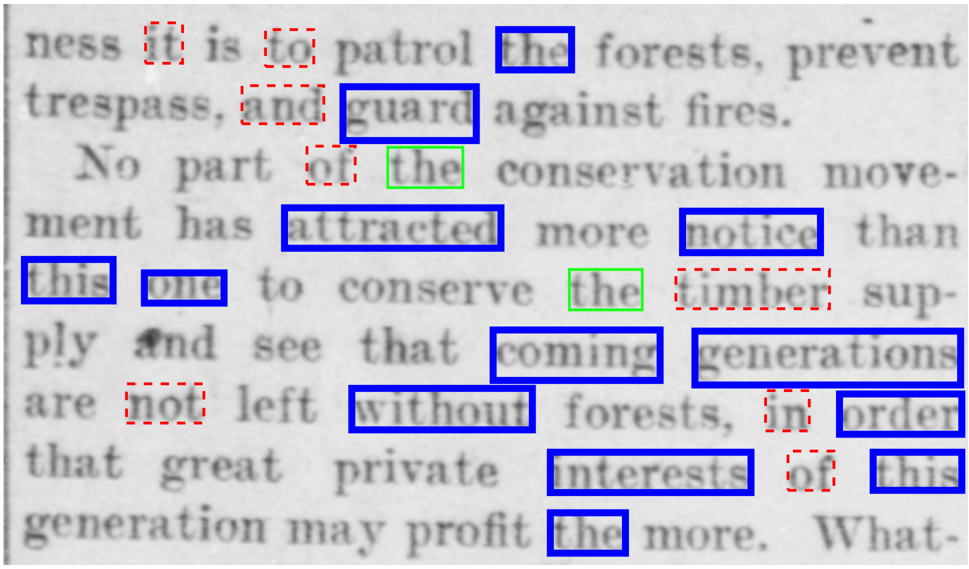
6. Pass the SIFT descriptors for the training/test images found in the previous steps to a multi-class SVM.

In summary, this classifier can be described as simply using an SVM with SIFT descriptors, except that care is taken to align characters as well as possible for both training and testing. We use the $SVM^{multiclass}$ implementation[8] of multiclass SVM (Tsochantaridis et al., 2004) and use a high C value of 5,000,000, which was selected through cross-validation. This makes sense since we generally do not have many instances of each character class in the clean list, and so we want a minimum of slack, which a high C value enforces.

6. Experiments

In this section, we describe three types of experiments. First, we show that our procedure for generating clean sets achieves the very low error rate predicted by our bounds. Next, we show that for a collection of 56 documents, using the clean sets to train new, document-specific classifiers

[8] SVM implementation can be found at `http://svmlight.joachims.org/`.



ness it is to patrol the forests, prevent trespass, and guard against fires.

No part of the conservation movement has attracted more notice than this one to conserve the timber supply and see that coming generations are not left without forests, in order that great private interests of this generation may profit the more. What-

Figure 1: Thick blue boxes indicate clean list words. Dashed red boxes indicate Tesseract's confident word list. Thin green boxes indicate words in both lists. Despite being in Tesseract's list of high confidence words, "timber" is misrecognized by Tesseract as "timhcr". All other words in boxes were correctly translated by Tesseract. (Best viewed in color.)

significantly reduces OCR errors over the initial OCR system used. Finally, we show what happens if a traditional measure of confidence is used to select a document-specific training set. In particular, we show that our clean sets have far fewer errors and result in document-specific models that can correct a much larger number of errors in the original OCR output.

6.1 Initial Clean Set Experiments

We experimented with two sets of documents. The first set consists of 10 documents from the JSTOR archive[9] and Project Gutenberg.[10] This initial set of documents was used to evaluate our clean list generation algorithm and develop our algorithm for producing character models from the clean lists (Kae et al., 2009). In this work, our clean lists selected an average of 6% of the words from each document. *These clean lists did not contain a single error*, that is, the precision of our clean lists was 100%. This strongly supports our theoretical bounds established in Section 4.

[9] JSTOR can be found at `http://www.jstor.org`.

[10] Project Gutenberg can be found at `http://www.gutenberg.org/`.

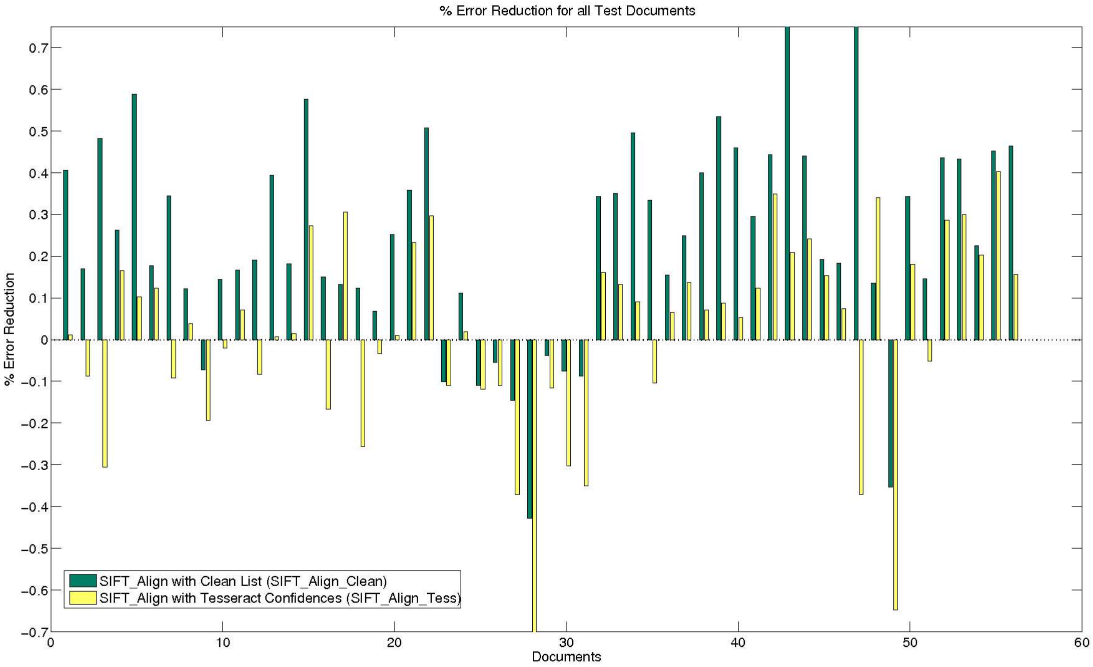


Figure 2: Character error reduction rates for SIFT_Align using the clean list (SIFT_Align_Clean) and Tesseract's confident word list (SIFT_Align_Tess) on the test sets of 56 documents. SIFT_Align_Clean increases the error rate in 10 documents whereas SIFT_Align_Tess increases the error rate in 21 documents.

6.2 Correcting OCR Errors

After establishing the basic viability of the clean set procedure, we selected another set of documents on which to test our end-to-end system of generating clean sets, using them to build document-specific models, and using these models, in turn, to correct errors made by the original OCR system.

The second set of documents, used for performance evaluation of the SIFT_Align algorithm, are 56 documents taken from the Chronicling America[11] archive of historical newspapers. Since our initial OCR system (Tesseract) can only accept blocks of text and does not perform layout analysis, we manually cropped out single columns of text from these newspaper pages. Other than cropping and converting to the TIFF image format for Tesseract, the documents were not modified in any way. There are on average 1204 words per document. The clean list contains 2 errors out of a total of 4465 words, within the theoretical bound of 0.002 mentioned earlier.

In an effort to increase the size of the clean lists beyond 6% per document, we experimented with relaxing some of the criteria used to select the clean lists. In particular, we allowed the Hamming ball of radius 1 for a word to be non-empty as long as the words within the ball did not appear within the original OCR system's translation. By making this small change, we were able to increase the size of the clean lists to an average of 18% per document while introducing at most one error per document. We refer to the original clean lists as *conservative clean lists* and to the modified, larger,

[11] Documents can be found at `http://chroniclingamerica.loc.gov/`.

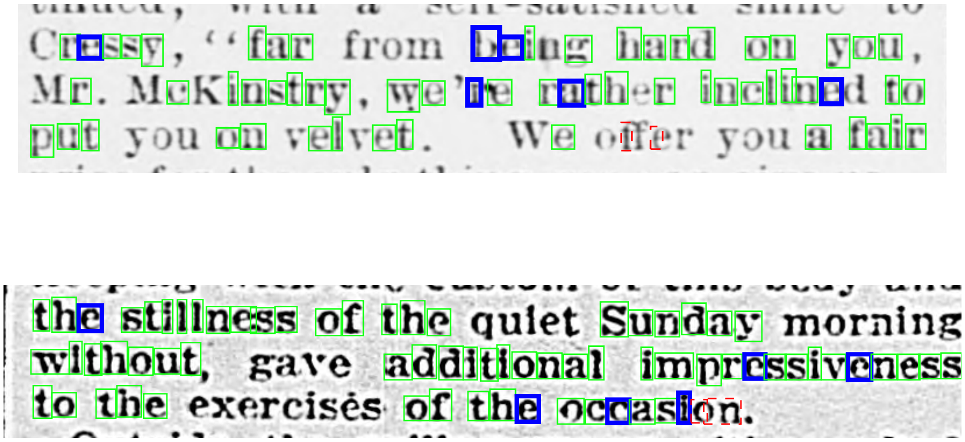

Cressy, "far from being hard on you,
Mr. McKinstry, we're rather inclined to
put you on velvet. We offer you a fair
the stillness of the quiet Sunday morning
without, gave additional impressiveness
to the exercises of the occasion.

Figure 3: Sample of results from two documents. A thin green box indicates both the initial OCR system (Tesseract) and SIFT_Align correctly classified the character. A dashed red box indicates both systems misclassified the character, and a thick blue box indicates that SIFT_Align classified the character correctly and Tesseract misclassified it. In this example, there are no cases shown where Tesseract correctly classified a character and SIFT_Align misclassifies it. (Best viewed in color.)

and slightly less accurate clean lists as *aggressive clean lists*. We decided to use the aggressive clean lists for our experiments because they contain few errors and there are more character instances.[12] From this point, our use of "clean list" refers to the aggressive clean list.

We then ran Tesseract on all documents, obtaining character bounding boxes[13] and guesses for each character. Next, we used Mechanical Turk[14] to label all character bounding boxes to produce a ground truth labeling. We instructed annotators to only label bounding boxes for which a single character is clearly visible. Other cases (multiple characters in the bounding box or a partial character) were discarded.

After the initial OCR system was used to make an initial pass at each document, the clean list for that document was extracted. Character recognition was then performed as described in Section 5. Even though many of the characters were already recognized correctly by the original

[12] To account for the looser criteria of the aggressive set, we would need to add a term to the theoretical bound that considers the probability of error due to the true labeling of the word being a neighbor of Hamming distance 1 from the OCR system's interpretation. This term would be $D_1 q^{\frac{2\varepsilon}{\delta}}$, where D_1 is the number of neighbors of Hamming distance 1, and q is the probability that a word that was not detected anywhere in the document actually appears in the document. Given our assumed bounds of $\varepsilon < 10^{-3}$, $\delta^2 > 10^{-1}$, if we further assume $D_1 q$ to be conservatively bounded by 0.3, then using the aggressive criteria doubles the probability of error to 0.004.

We note that the assumptions we make to produce the theoretical bound are very conservative. This leaves room for some experimentation to find the optimal balance between probability of error and clean set size, while still maintaining an empirical error close to the predicted bound of 0.002.

[13] This feature is not available out of the box; we edited the source code.

[14] Mechanical Turk can be found at `https://www.mturk.com/mturk/welcome`.

OCR system, our approach improves the recognition to produce an even higher accuracy than the original OCR system's accuracy, on average. As shown in the next section, in most cases, this resulted in correcting a significant portion of the characters in the documents.

6.3 Comparison to Another Confidence Measure

In order to judge the effectiveness of using our clean list, we also generated another confident word list using Tesseract's own measure of confidence.[15] To generate the confident word list, we sort Tesseract's recognized words by their measure of confidence and take the top *n* words that result in the same number of characters as our clean list.

In Figure 1, we show a portion of a document and the corresponding subset of clean list words (generated by our process) and highly confident Tesseract words. All of the words in our clean list were, in fact, correctly labeled by the initial Tesseract pass. In other words, our clean list for this example was error free. But Tesseract's high confidence word list includes "timber" which was mistranslated by Tesseract.

We refer to the SIFT_Align algorithm using our clean list as SIFT_Align_Clean and the SIFT_Align algorithm using Tesseract's confidences as SIFT_Align_Tess. In Figure 2, we show the character error reduction rates for both SIFT_Align_Clean and SIFT_Align_Tess. In 46 of the 56 documents, SIFT_Align_Clean results in a reduction of errors whereas SIFT_Align_Tess reduces error in 35 documents. Note this figure shows percent error reduction, not the raw number of errors. SIFT_Align_Clean made a total of 2487 character errors (44.4 errors per document) on the test set compared to 7745 errors (138.3 errors per document) originally made by Tesseract on those same characters. For the 10 cases where SIFT_Align_Clean increased error, SIFT_Align_Clean made 356 character errors and Tesseract made 263 errors. Thus, overall, the error reductions achieved by SIFT_Align_Clean were much greater than the errors introduced.

SIFT_Align_Clean outperforms Sift_Align_Tess. Average error reduction for SIFT_Align_Clean is 34.1% compared to 9.5% for Sift_Align_Tess. Error reduction is calculated as $(TT - ST)/TD$ where TT is # Tesseract errors in the test set, ST is # SIFT_Align errors in the test set and TD is # Tesseract errors in the document. SIFT_Align_Clean also reduces the character error in more documents than does Sift_Align_Tess.

Our test cases only consider properly segmented characters which account for about half of all the errors in these documents. The error reduction for SIFT_Align_Clean over all characters (segmented properly or not) is 20.3%.

Our experiments have shown that, on two separate sets of documents, our conservative clean sets have very low error rates, meeting the theoretical bounds presented, and that by relaxing the criteria slightly, we can get significantly larger sets while maintaining a low error rate. We have shown that using these clean sets to build document-specific models can significantly reduce OCR errors, and that traditional confidence measures do not result in the same benefits.

7. Applications to Other Domains

We believe that our method of identifying subsets of results for which we can achieve a very high bound of being correct can also be applied to other domains outside of OCR.

[15] There are two measures of word confidence in Tesseract, described in the Tesseract documentation as "rating" and "certainty". We use "certainty".

One such domain is speech recognition. Here, our consistency check would be over acoustic signals rather than a patch of pixels from a scanned document. This check would be similar to Algorithm 1 except that we now apply the check to a segment of speech signal. In this speech recognition context, the segment is now the "glyph" g and we want to check whether the label assigned to g is reliable by comparing g with other segments from the same recording that are most similar to g. Acoustic segment g should be given the same label (in our terminology, dominated by that label) as its most similar segments. We can then form equivalence classes of easily confusable phonemes, and perform consistency checks on segments of speech that have been labeled as a word with an empty pseudo-Hamming distance of 1. We could then follow a procedure similar to the proof presented in Section 4 to bound the probability that segments of speech that pass such a consistency check were incorrectly labeled.

By using this framework for speech recognition, we can potentially obtain the equivalent of clean lists: portions of speech for which we are very confident the initial labeling was correct. This may allow us to refine the speech recognition model to be specific to the recording, for instance, allowing for a model that is specific to a particular individual's accent.

The application of our idea to speech recognition may involve a slightly different set of difficulties than when applied to OCR. For instance, identifying word segmentations may be more difficult. However, when training speech recognition systems, we may have access to an additional source of information in the form of closed captions. We can model the closed captions as a noisy signal of the ground truth, independent of the speech recognizer. Taking a conservative estimate of the closed captioning error rate, we can use the closed captions to reduce our bound on the probability of error by requiring that the closed captioning match the labeling given by the speech recognition system.

In Lamel et al. (2002), audio with closed captions is used to generate additional labeled training data for a speech recognizer, by aligning the speech recognizer output to the closed captioning and accepting segments where the two agree. Given the large amount of closed captioning data available, this scenario is particularly amenable to our method of generating high precision training data (at some cost to recall). Additionally, using a consistency check approach as presented in Algorithm 1 can yield advantages over using an ad-hoc check such as directly accepting segments where the speech recognizer and closed captioning agree. For instance, we may find it necessary to throw out words where the two agree if the word has many nearby phonemic neighbors with which it may be confused, and thereby likely reduce the error rate of the labelings used as training data.

Another potential application for our bound analysis is in the area of information retrieval using query expansion. As mentioned earlier in Section 2.2.2, query expansion is a technique used in information retrieval to add terms to an initial query based on the highly ranked documents of the initial query. One issue when performing query expansion is that matching with false positives can quickly lead to errors due to drift in the query. For image retrieval, Chum et al. (2007) give a method of spatial verification to eliminate false matches. In this context, queries are objects in images and images are represented using a bag-of-visual-words.

Let the original query image be Q, and let the set of images returned initially by the search engine be $\mathbf{R}$. The goal is to identify returned images $\{R_i\}$ that we believe contain the same object as Q with very high confidence. We can then add these images $\{R_i\}$ to the query and repeat the search procedure with this expanded query set, ideally increasing the number of relevant images returned by the search engine.

To reduce the possibility of adding a false match R_i to the query set, Chum et al. (2007) apply a spatial verification procedure as follows. A feature point in R_i matching a feature point in Q

generates a hypothesized transformation that would put the object in R_i in correspondence with the object in Q. If this hypothesis leads to at least a certain number of matching feature points in R_i and Q (same visual word and location after transformation), then R_i is spatially verified and added to the query set.

We could estimate a bound on the probability of a false match passing spatial verification by assuming a feature in a random, non-matching returned image matches a feature in the query image Q with probability p. If we further assume that the probability of two features matching is independent of the other features in Q and returned image R_i, then the number of features in correspondence between the Q and non-matching result R_i is a binomial distribution with parameters n, the number of features in the query image, and p.

A result image R_i passes the spatial verification if, for at least one hypothesized transformation for putting Q in correspondence with R_i, at least 20 features are in correspondence. With the number of hypotheses being approximately 10^3, n also being approximately 10^3, and a conservatively high estimate of p as 10^{-3} (given a visual dictionary of size 10^6), we find that even with a requirement of just at least 12 features being in correspondence, the probability of a false match being spatially verified is less than $10^3 \cdot (1 - \sum_{i=0}^{11} \binom{n}{i} p^i (1-p)^{n-i}) < 10^{-6}$.

However, spatial verification will likely result in more errors than predicted by this analysis, due to the overly restrictive assumption that the probability of features matching in a result image R_i is independent of the other features. One potential method for removing this assumption is to analyze the error in terms of common substructures: features belonging to similar substructures are more likely to match, but the probability of one feature matching is independent of the features in different substructures of the same image. This analysis may suggest ways of improving the spatial verification, such as requiring that the matching features not be closely clustered in only one section of the image.

8. Conclusions and Future Work

In this paper, we advocate dealing with the problem of non-stationarity between the training and test distributions by identifying a subset of information whose interpretation we can be confident of, and using this information as test-specific training data. We have applied this approach to the problem of OCR, demonstrating that we can produce high-precision document-specific training data. Under modest assumptions, we show the error rate of this labeled data to be bounded by 0.002, and give empirical results consistent with this theoretical bound.

By combining this document-specific training data with simple appearance-based character recognition techniques, we are able to achieve significant reductions in average character error. We believe that further improvements can be achieved by using the clean lists in conjunction with more sophisticated models, such as document-specific language models, as suggested by Wick et al. (2007). In addition, while our work has taken the character segmentations produced by the initial OCR system as fixed, we believe that the clean lists can also be used to re-segment and fix the large percentage of initial errors that result from incorrect character segmentation.

Lastly, we also show potential applications to problems in other domains such as speech recognition and query expansion. While many of the techniques used in this work are specific to an OCR application, we believe that the principles are quite general, and that the search for more formal bounds on probabilities of error will lead toward a variety of new applications.

References

A. C. Berg, F. Grabler, and J. Malik. Parsing images of architectural scenes. In *International Conference on Computer Vision*, 2007.

T. M. Breuel. Character recognition by adaptive statistical similarity. In *International Conference on Document Analysis and Recognition*, 2003.

R. G. Casey. Text OCR by solving a cryptogram. In *International Conference on Pattern Recognition*, pages 349–351, 1986.

O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *International Conference on Computer Vision*, 2007.

J. Edwards and D. Forsyth. Searching for character models. In *Neural Information Processing Systems*, 2005.

T. K. Ho. Bootstrapping text recognition from stop words. In *International Conferece on Pattern Recognition*, pages 605–609, 1998.

T. K. Ho and G. Nagy. OCR with no shape training. In *International Conference on Pattern Recognition*, pages 27–30, 2000.

J. D. Hobby and T. K. Ho. Enhancing degraded document images via bitmap clustering and averaging. In *International Conference on Document Analysis and Recognition*, pages 394 – 400, 1997.

T. Hong and J. J. Hull. Visual inter-word relations and their use in OCR post-processing. In *International Conference on Document Analysis and Recognition*, pages 14–16, 1995a.

T. Hong and J. J. Hull. Improving OCR performance with word image equivalence. In *Symposium on Document Analysis and Information Retrieval*, pages 177–190, 1995b.

T. Hong and J. J. Hull. Character segmentation using visual inter-word constraints in a text page. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 15–25, 1995c.

A. Kae and E. Learned-Miller. Learning on the fly: Font free approaches to difficult OCR problems. In *International Conference on Document Analysis and Recognition*, 2009.

A. Kae, G. B. Huang, and E. Learned-Miller. Bounding the probability of error for high precision recognition. Technical Report UM-CS-2009-031, University of Massachusetts Amherst, 2009.

A. Kae, G. B. Huang, C. Doersch, and E. Learned-Miller. Improving state-of-the-art OCR through high-precision document-specific modeling. In *Computer Vision and Pattern Recognition*, 2010.

O. Kolak. A generative probabilistic OCR model for NLP applications. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 55–62, 2003.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

K. Kukich. Techniques for automatically correcting words in text. *Association for Computing Machinery Computing Surveys*, 24(4):377–439, 1992. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/146370.146380.

L. Lamel, J. Gauvain, and G. Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech and Language*, 16(1):115–129, 2002.

D. Lee. Substitution deciphering based on HMMs with applications to compressed document processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1661–1666, 2002.

M. Leisink and B. Kappen. Bound propagation. *Journal of Artificial Intelligence Research*, 19(1): 139–154, 2003.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

G. Nagy. Efficient algorithms to decode substitution ciphers with applications to OCR. In *International Conference on Pattern Recognition*, pages 352–355, 1986.

G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

M.-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *British Machine Vision Conference*, 2007.

D. Ramanan. Learning to parse images of articulated bodies. In *Neural Information Processing Systems*, 2006.

B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Computer Vision and Pattern Recognition*, 2010.

W. J. Scheirer, A Rocha, R. J. Micheals, and T. E. Boult. Meta-recognition: The theory and practice of recognition score analysis. *Pattern Analysis and Machine Intelligence*, 33(8):1689–1695, 2011.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, 2004.

J. Weinman, E. Learned-Miller, and A. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1733–1746, Oct 2009.

M. Wick, M. Ross, and E. Learned-Miller. Context-sensitive error correction: Using topic models to improve OCR. In *International Conference on Document Analysis and Recognition*, 2007.

Journal of Machine Learning Research 13 (2012) 389-427 Submitted 8/10; Revised 12/11; Published 2/12

Minimax-Optimal Rates For Sparse Additive Models Over Kernel Classes Via Convex Programming

Garvesh Raskutti GARVESHR@STAT.BERKELEY.EDU
Martin J. Wainwright[*] WAINWRIG@STAT.BERKELEY.EDU
Bin Yu[*] BINYU@STAT.BERKELEY.EDU
Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Editor: Nicolas Vayatis

Abstract

Sparse additive models are families of d-variate functions with the additive decomposition $f^* = \sum_{j \in S} f_j^*$, where S is an unknown subset of cardinality $s \ll d$. In this paper, we consider the case where each univariate component function f_j^* lies in a reproducing kernel Hilbert space (RKHS), and analyze a method for estimating the unknown function f^* based on kernels combined with ℓ_1-type convex regularization. Working within a high-dimensional framework that allows both the dimension d and sparsity s to increase with n, we derive convergence rates in the $L^2(\mathbb{P})$ and $L^2(\mathbb{P}_n)$ norms over the class $\mathcal{F}_{d,s,\mathcal{H}}$ of sparse additive models with each univariate function f_j^* in the unit ball of a univariate RKHS with bounded kernel function. We complement our upper bounds by deriving minimax lower bounds on the $L^2(\mathbb{P})$ error, thereby showing the optimality of our method. Thus, we obtain optimal minimax rates for many interesting classes of sparse additive models, including polynomials, splines, and Sobolev classes. We also show that if, in contrast to our univariate conditions, the d-variate function class is assumed to be globally bounded, then much faster estimation rates are possible for any sparsity $s = \Omega(\sqrt{n})$, showing that global boundedness is a significant restriction in the high-dimensional setting.

Keywords: sparsity, kernel, non-parametric, convex, minimax

1. Introduction

The past decade has witnessed a flurry of research on sparsity constraints in statistical models. Sparsity is an attractive assumption for both practical and theoretical reasons: it leads to more interpretable models, reduces computational cost, and allows for model identifiability even under high-dimensional scaling, where the dimension d exceeds the sample size n. While a large body of work has focused on sparse linear models, many applications call for the additional flexibility provided by non-parametric models. In the general setting, a non-parametric regression model takes the form $y = f^*(x_1, \ldots, x_d) + w$, where $f^* : \mathbb{R}^d \to \mathbb{R}$ is the unknown regression function, and w is scalar observation noise. Unfortunately, this general non-parametric model is known to suffer severely from the so-called "curse of dimensionality", in that for most natural function classes (e.g., twice differentiable functions), the sample size n required to achieve any given error grows exponentially in the dimension d. Given this curse of dimensionality, it is essential to further constrain the com-

[*] Also in the department of Elecrical Engineering & Computer Science.

plexity of possible functions f^*. One attractive candidate is the class of *additive non-parametric models* (see Hastie and Tibshirani, 1986), in which the function f^* has an additive decomposition of the form

$$f^*(x_1, x_2, \ldots, x_d) = \sum_{j=1}^{d} f_j^*(x_j), \tag{1}$$

where each component function f_j^* is univariate. Given this additive form, this function class no longer suffers from the exponential explosion in sample size of the general non-parametric model. Nonetheless, one still requires a sample size $n \gg d$ for consistent estimation; note that this is true even for the linear model, which is a special case of Equation (1).

A natural extension of sparse linear models is the class of *sparse additive models*, in which the unknown regression function is assumed to have a decomposition of the form

$$f^*(x_1, x_2 \ldots, x_d) = \sum_{j \in S} f_j^*(x_j), \tag{2}$$

where $S \subseteq \{1, 2, \ldots, d\}$ is some unknown subset of cardinality $|S| = s$. Of primary interest is the case when the decomposition is genuinely sparse, so that $s \ll d$. To the best of our knowledge, this model class was first introduced by Lin and Zhang (2006), and has since been studied by various researchers (e.g., Koltchinskii and Yuan, 2010; Meier et al., 2009; Ravikumar et al., 2009; Yuan, 2007). Note that the sparse additive model (2) is a natural generalization of the sparse linear model, to which it reduces when each univariate function is constrained to be linear.

In past work, several groups have proposed computationally efficient methods for estimating sparse additive models (2). Just as ℓ_1-based relaxations such as the Lasso have desirable properties for sparse parametric models, more general ℓ_1-based approaches have proven to be successful in this setting. Lin and Zhang (2006) proposed the COSSO method, which extends the Lasso to cases where the component functions f_j^* lie in a reproducing kernel Hilbert space (RKHS); see also Yuan (2007) for a similar extension of the non-negative garrote (Breiman, 1995). Bach (2008) analyzes a closely related method for the RKHS setting, in which least-squares loss is penalized by an ℓ_1-sum of Hilbert norms, and establishes consistency results in the classical (fixed d) setting. Other related ℓ_1-based methods have been proposed in independent work by Koltchinskii and Yuan (2008), Ravikumar et al. (2009) and Meier et al. (2009), and analyzed under high-dimensional scaling ($d \gg n$). As we describe in more detail in Section 3.4, each of the above papers establish consistency and convergence rates for the prediction error under certain conditions on the covariates as well as the sparsity s and dimension d. However, it is not clear whether the rates obtained in these papers are sharp for the given methods, nor whether the rates are minimax-optimal. Past work by Koltchinskii and Yuan (2010) establishes rates for sparse additive models with an additional global boundedness condition, but as will be discussed at more length in the sequel, these rates are not minimax optimal in general.

This paper makes three main contributions to this line of research. Our first contribution is to analyze a simple polynomial-time method for estimating sparse additive models and provide upper bounds on the error in the $L^2(\mathbb{P})$ and $L^2(\mathbb{P}_n)$ norms. The estimator[1] that we analyze is based on a combination of least-squares loss with two ℓ_1-based sparsity penalty terms, one corresponding to

[1] The same estimator was proposed concurrently by Koltchinskii and Yuan (2010); we discuss connections to this work in the sequel.

an $\ell_1/L^2(\mathbb{P}_n)$ norm and the other an $\ell_1/\|\cdot\|_{\mathcal{H}}$ norm. Our first main result (Theorem 1) shows that with high probability, if we assume the univariate functions are bounded and independent, the error of our procedure in the squared $L^2(\mathbb{P}_n)$ and $L^2(\mathbb{P})$ norms is bounded by $O\big(\frac{s\log d}{n} + s\nu_n^2\big)$, where the quantity ν_n^2 corresponds to the optimal rate for estimating a single univariate function. Importantly, our analysis does *not* require a global boundedness condition on the class $\mathcal{F}_{d,s,\mathcal{H}}$ of all s-sparse models, an assumption that is often imposed in classical non-parametric analysis. Indeed, as we discuss below, when such a condition is imposed, then significantly faster rates of estimation are possible. The proof of Theorem 1 involves a combination of techniques for analyzing M-estimators with decomposable regularizers (Negahban et al., 2009), combined with various techniques in empirical process theory for analyzing kernel classes (e.g., Bartlett et al., 2005; Mendelson, 2002; van de Geer, 2000). Our second contribution is complementary in nature, in that it establishes algorithm-independent minimax lower bounds on $L^2(\mathbb{P})$ error. These minimax lower bounds, stated in Theorem 2, are specified in terms of the metric entropy of the underlying univariate function classes. For both finite-rank kernel classes and Sobolev-type classes, these lower bounds match our achievable results, as stated in Corollaries 1 and 2, up to constant factors in the regime of sub-linear sparsity ($s = o(d)$). Thus, for these function classes, we have a sharp characterization of the associated minimax rates. The lower bounds derived in this paper initially appeared in the Proceedings of the NIPS Conference (December 2009). The proofs of Theorem 2 is based on characterizing the packing entropies of the class of sparse additive models, combined with classical information theoretic techniques involving Fano's inequality and variants (e.g., Has'minskii, 1978; Yang and Barron, 1999; Yu, 1996).

Our third contribution is to determine upper bounds on minimax $L^2(\mathbb{P})$ and $L^2(\mathbb{P}_n)$ error when we impose a global boundedness assumption on the class $\mathcal{F}_{d,s,\mathcal{H}}$. More precisely, a global boundedness condition means that the quantity $B(\mathcal{F}_{d,s,\mathcal{H}}) = \sup_{f\in\mathcal{F}_{d,s,\mathcal{H}}} \sup_x |\sum_{j=1}^d f_j(x_j)|$ is assumed to be bounded independently of (s,d). As mentioned earlier, our upper bound in Theorem 1 does *not* impose a global boundedness condition, whereas in contrast, the analysis of Koltchinskii and Yuan (2010), or KY for short, does impose such a global boundedness condition. Under global boundedness, their work provides rates on the $L^2(\mathbb{P})$ and $L^2(\mathbb{P}_n)$ norm that are of the same order as the results presented here. It is natural to wonder whether or not this difference is actually significant—that is, do the minimax rates for the class of sparse additive models depend on whether or not global boundedness is imposed? In Section 3.5, we answer this question in the affirmative. In particular, Theorem 3 and Corollary 3 provide upper bounds on the minimax rates, as measured in either the $L^2(\mathbb{P})$ and $L^2(\mathbb{P}_n)$-norms, under a global boundedness condition. These rates are faster than those of Theorem 3 in the KY paper, in particular showing that the KY rates are not optimal for problems with $s = \Omega(\sqrt{n})$. In this way, we see that the assumption of global boundedness, though relatively innocuous for classical (low-dimensional) non-parametric problems, can be quite limiting in high dimensions.

The remainder of the paper is organized as follows. In Section 2, we provide background on kernel spaces and the class of sparse additive models considered in this paper. Section 3 is devoted to the statement of our main results and discussion of their consequences; it includes description of our method, the upper bounds on the convergence rate that it achieves, and a matching set of minimax lower bounds. Section 3.5 investigates the restrictiveness of the global uniform boundedness assumption and in particular, Theorem 3 and Corollary 3 demonstrate that there are classes of globally bounded functions for which faster rates are possible. Section 4 is devoted to the proofs of our three main theorems, with the more technical details deferred to the Appendices. We conclude with a discussion in Section 5.

2. Background and Problem Set-up

We begin with some background on reproducing kernel Hilbert spaces, before providing a precise definition of the class of sparse additive models studied in this paper.

2.1 Reproducing Kernel Hilbert Spaces

Given a subset $\mathcal{X} \subset \mathbb{R}$ and a probability measure $\mathbb{Q}$ on $\mathcal{X}$, we consider a Hilbert space $\mathcal{H} \subset L^2(\mathbb{Q})$, meaning a family of functions $g : \mathcal{X} \to \mathbb{R}$, with $\|g\|_{L^2(\mathbb{Q})} < \infty$, and an associated inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ under which $\mathcal{H}$ is complete. The space $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) if there exists a symmetric function $\mathbb{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ such that for each $x \in \mathcal{X}$: (a) the function $\mathbb{K}(\cdot, x)$ belongs to the Hilbert space $\mathcal{H}$, and (b) we have the reproducing relation $f(x) = \langle f, \mathbb{K}(\cdot, x) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$. Any such kernel function must be positive semidefinite; under suitable regularity conditions, Mercer's theorem (1909) guarantees that the kernel has an eigen-expansion of the form

$$\mathbb{K}(x, x') = \sum_{k=1}^{\infty} \mu_k \phi_k(x) \phi_\ell(x'), \tag{3}$$

where $\mu_1 \geq \mu_2 \geq \mu_3 \geq \ldots \geq 0$ are a non-negative sequence of eigenvalues, and $\{\phi_k\}_{k=1}^{\infty}$ are the associated eigenfunctions, taken to be orthonormal in $L^2(\mathbb{Q})$. The decay rate of these eigenvalues will play a crucial role in our analysis, since they ultimately determine the rate ν_n for the univariate RKHS's in our function classes.

Since the eigenfunctions $\{\phi_k\}_{k=1}^{\infty}$ form an orthonormal basis, any function $f \in \mathcal{H}$ has an expansion of the $f(x) = \sum_{k=1}^{\infty} a_k \phi_k(x)$, where $a_k = \langle f, \phi_k \rangle_{L^2(\mathbb{Q})} = \int_{\mathcal{X}} f(x) \phi_k(x) \, d\mathbb{Q}(x)$ are (generalized) Fourier coefficients. Associated with any two functions in $\mathcal{H}$—say $f = \sum_{k=1}^{\infty} a_k \phi_k$ and $g = \sum_{k=1}^{\infty} b_k \phi_k$—are two distinct inner products. The first is the usual inner product in $L^2(\mathbb{Q})$, $\langle f, g \rangle_{L^2(\mathbb{Q})} := \int_{\mathcal{X}} f(x) g(x) \, d\mathbb{Q}(x)$. By Parseval's theorem, it has an equivalent representation in terms of the expansion coefficients—namely

$$\langle f, g \rangle_{L^2(\mathbb{Q})} = \sum_{k=1}^{\infty} a_k b_k.$$

The second inner product, denoted $\langle f, g \rangle_{\mathcal{H}}$, is the one that defines the Hilbert space; it can be written in terms of the kernel eigenvalues and generalized Fourier coefficients as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{k=1}^{\infty} \frac{a_k b_k}{\mu_k}.$$

Using this definition, the Hilbert ball of unit radius for a kernel with eigenvalues $\{\mu_k\}_{k=1}^{\infty}$ and eigenfunctions $\{\phi_k\}_{k=1}^{\infty}$ is given by

$$\mathbb{B}_{\mathcal{H}}(1) := \big\{ f = \sum_{k=1}^{\infty} a_k \phi_k \mid \sum_{k=1}^{\infty} \frac{a_k^2}{\mu_k} \leq 1 \big\}.$$

For more background on reproducing kernel Hilbert spaces, we refer the reader to various standard references (e.g., Aronszajn, 1950; Saitoh, 1988; Schölkopf and Smola, 2002; Wahba, 1990).

2.2 Sparse Additive Models Over RKHS

For each $j = 1,\ldots,d$, let $\mathcal{H}_j \subset L^2(\mathbb{Q})$ be a reproducing kernel Hilbert space of univariate functions on the domain $\mathcal{X} \subset \mathbb{R}$. We assume that

$$\mathbb{E}[f_j(x)] = \int_{\mathcal{X}} f_j(x) d\,\mathbb{Q}(x) \;=\; 0 \qquad \text{for all } f_j \in \mathcal{H}_j, \text{and for each } j = 1,2,\ldots,d.$$

As will be clarified momentarily, our observation model (5) allows for the possibility of a non-zero mean $\bar{f}$, so that there is no loss of generality in this assumption. For a given subset $S \subset \{1,2,\ldots,d\}$, we define

$$\mathcal{H}(S) := \big\{ f = \sum_{j \in S} f_j \mid f_j \in \mathcal{H}_j, \text{ and } f_j \in \mathbb{B}_{\mathcal{H}_j}(1)\ \forall\, j \in S \big\},$$

corresponding to the class of functions $f : \mathcal{X}^d \to \mathbb{R}$ that decompose as sums of univariate functions on co-ordinates lying within the set S. Note that $\mathcal{H}(S)$ is also (a subset of) a reproducing kernel Hilbert space, in particular with the norm

$$\|f\|^2_{\mathcal{H}(S)} = \sum_{j \in S} \|f_j\|^2_{\mathcal{H}_j},$$

where $\|\cdot\|_{\mathcal{H}_j}$ denotes the norm on the univariate Hilbert space $\mathcal{H}_j$. Finally, for $s \in \{1,2,\ldots,\lfloor d/2 \rfloor\}$, we define the function class

$$\mathcal{F}_{d,s,\mathcal{H}} := \bigcup_{\substack{S \subset \{1,2,\ldots,d\} \\ |S| = s}} \mathcal{H}(S). \tag{4}$$

To ease notation, we frequently adopt the shorthand $\mathcal{F} = \mathcal{F}_{d,s,\mathcal{H}}$, but the reader should recall that $\mathcal{F}$ depends on the choice of Hilbert spaces $\{\mathcal{H}_j\}_{j=1}^d$, and moreover, that we are actually studying a *sequence of function classes* indexed by (d,s).

Now let $\mathbb{P} = \mathbb{Q}^d$ denote the product measure on the space $\mathcal{X}^d \subseteq \mathbb{R}^d$. Given an arbitrary $f^* \in \mathcal{F}$, we consider the observation model

$$y_i = \bar{f} + f^*(x_i) + w_i, \quad \text{for } i = 1,2,\ldots,n, \tag{5}$$

where $\{w_i\}_{i=1}^n$ is an i.i.d. sequence of standard normal variates, and $\{x_i\}_{i=1}^n$ is a sequence of design points in $\mathbb{R}^d$, sampled in an i.i.d. manner from $\mathbb{P}$.

Given an estimate $\widehat{f}$, our goal is to bound the error $\widehat{f} - f^*$ under two norms. The first is the *usual* $L^2(\mathbb{P})$ *norm* on the space $\mathcal{F}$; given the product structure of $\mathbb{P}$ and the additive nature of any $f \in \mathcal{F}$, it has the additive decomposition $\|f\|^2_{L^2(\mathbb{P})} = \sum_{j=1}^d \|f_j\|^2_{L^2(\mathbb{Q})}$. In addition, we consider the error in the *empirical* $L^2(\mathbb{P}_n)$*-norm* defined by the sample $\{x_i\}_{i=1}^n$, defined as

$$\|f\|^2_{L^2(\mathbb{P}_n)} := \frac{1}{n} \sum_{i=1}^n f^2(x_i).$$

Unlike the $L^2(\mathbb{P})$ norm, this norm does not decouple across the dimensions, but part of our analysis will establish an approximate form of such decoupling. For shorthand, we frequently use the notation $\|f\|_2 = \|f\|_{L^2(\mathbb{P})}$ and $\|f\|_n = \|f\|_{L^2(\mathbb{P}_n)}$ for a d-variate function $f \in \mathcal{F}$. With a minor abuse of notation, for a univariate function $f_j \in \mathcal{H}_j$, we also use the shorthands $\|f_j\|_2 = \|f_j\|_{L^2(\mathbb{Q})}$ and $\|f_j\|_n = \|f_j\|_{L^2(\mathbb{Q}_n)}$.

3. Main Results and Their Consequences

This section is devoted to the statement of our three main results, and discussion of some of their consequences. We begin in Section 3.1 by describing a regularized M-estimator for sparse additive models, and we state our upper bounds for this estimator in Section 3.2. We illustrate our upper bounds for various concrete instances of kernel classes. In Section 3.3, we state minimax lower bounds on the $L^2(\mathbb{P})$ error over the class $\mathcal{F}_{d,s,\mathcal{H}}$, which establish the optimality of our procedure. In Section 3.4, we provide a detailed comparison between our results to past work, and in Section 3.5 we discuss the effect of global boundedness conditions on optimal rates.

3.1 A Regularized M-Estimator For Sparse Additive Models

For any function of the form $f = \sum_{j=1}^d f_j$, the $(L^2(\mathbb{Q}_n),1)$ and $(\mathcal{H},1)$-norms are given by

$$\|f\|_{n,1} := \sum_{j=1}^d \|f_j\|_n, \quad \text{and} \quad \|f\|_{\mathcal{H},1} := \sum_{j=1}^d \|f_j\|_{\mathcal{H}},$$

respectively. Using this notation and defining the sample mean $\bar{y}_n = \frac{1}{n}\sum_{i=1}^n y_i$, we define the cost functional

$$\mathcal{L}(f) = \frac{1}{2n}\sum_{i=1}^n \big(y_i - \bar{y}_n - f(x_i)\big)^2 + \lambda_n \|f\|_{n,1} + \rho_n \|f\|_{\mathcal{H},1}.$$

The cost functional $\mathcal{L}(f)$ is least-squares loss with a sparsity penalty $\|f\|_{n,1}$ and a smoothness penalty $\|f\|_{\mathcal{H},1}$. Here (λ_n, ρ_n) are a pair of positive regularization parameters whose choice will be specified by our theory. Given this cost functional, we then consider the M-estimator

$$\widehat{f} \in \arg\min_f \mathcal{L}(f) \quad \text{subject to } f = \textstyle\sum_{j=1}^d f_j \text{ and } \|f_j\|_{\mathcal{H}} \le 1 \text{ for all } j = 1,2,\ldots,d. \tag{6}$$

In this formulation (6), the problem is infinite-dimensional in nature, since it involves optimization over Hilbert spaces. However, an attractive feature of this M-estimator is that, as a consequence of the representer theorem (Kimeldorf and Wahba, 1971), it can be reduced to an equivalent convex program in $\mathbb{R}^n \times \mathbb{R}^d$. In particular, for each $j = 1,2,\ldots,d$, let $\mathbb{K}^j$ denote the kernel function for co-ordinate j. Using the notation $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ for the i^{th} sample, we define the collection of empirical kernel matrices $K^j \in \mathbb{R}^{n\times n}$, with entries $K^j_{i\ell} = \mathbb{K}^j(x_{ij}, x_{\ell j})$. By the representer theorem, any solution $\widehat{f}$ to the variational problem (6) can be written in the form

$$\widehat{f}(z_1,\ldots,z_d) = \sum_{i=1}^n \sum_{j=1}^d \widehat{\alpha}_{ij}\mathbb{K}^j(z_j, x_{ij}),$$

for a collection of weights $\big\{\widehat{\alpha}_j \in \mathbb{R}^n,\ j = 1,\ldots,d\big\}$. The optimal weights $(\widehat{\alpha}_1,\ldots,\widehat{\alpha}_d)$ are any minimizer to the following convex program:

$$\arg \min_{\substack{\alpha_j \in \mathbb{R}^n \\ \alpha_j^T K^j \alpha_j \le 1}} \left\{ \frac{1}{2n}\|y - \bar{y}_n - \sum_{j=1}^d K^j \alpha_j\|_2^2 + \lambda_n \sum_{j=1}^d \sqrt{\frac{1}{n}\|K^j\alpha_j\|_2^2} + \rho_n \sum_{j=1}^d \sqrt{\alpha_j^T K^j \alpha_j} \right\}. \tag{7}$$

This problem is a second-order cone program (SOCP), and there are various algorithms for finding a solution to arbitrary accuracy in time polynomial in (n,d), among them interior point methods (e.g., see §11 in Boyd and Vandenberghe 2004).

Various combinations of sparsity and smoothness penalties have been used in past work on sparse additive models. For instance, the method of Ravikumar et al. (2009) is based on least-squares loss regularized with single sparsity constraint, and separate smoothness constraints for each univariate function. They solve the resulting optimization problem using a back-fitting procedure. Koltchinskii and Yuan (2008) develop a method based on least-squares loss combined with a single penalty term $\sum_{j=1}^{d} \|f_j\|_{\mathcal{H}}$. Their method also leads to an SOCP if $\mathcal{H}$ is a reproducing kernel Hilbert space, but differs from the program (7) in lacking the additional sparsity penalties. Meier et al. (2009) analyzed least-squares regularized with a penalty term of the form $\sum_{j=1}^{d} \sqrt{\lambda_1 \|f_j\|_n^2 + \lambda_2 \|f_j\|_{\mathcal{H}}^2}$, where λ_1 and λ_2 are a pair of regularization parameters. In their method, λ_1 controls the sparsity while λ_2 controls the smoothness. If $\mathcal{H}$ is an RKHS, the method in Meier et al. (2009) reduces to an ordinary group Lasso problem on a different set of variables, which can be cast as a quadratic program. The more recent work of Koltchinskii and Yuan (2010) is based on essentially the same estimator as problem (6), except that we allow for a non-zero mean for the function, and estimate it as well. In addition, the KY analysis involves a stronger condition of global boundedness. We provide a more in-depth comparison of our analysis and results with the past work listed above in Sections 3.4 and 3.5.

3.2 Upper Bound

We now state a result that provides upper bounds on the estimation error achieved by the estimator (6), or equivalently (7). To simplify presentation, we state our result in the special case that the univariate Hilbert space $\mathcal{H}_j$, $j = 1, \ldots, d$ are all identical, denoted by $\mathcal{H}$. However, the analysis and results extend in a straightforward manner to the general setting of distinct univariate Hilbert spaces, as we discuss following the statement of Theorem 1.

Let $\mu_1 \geq \mu_2 \geq \ldots \geq 0$ denote the non-negative eigenvalues of the kernel operator defining the univariate Hilbert space $\mathcal{H}$, as defined in Equation (3), and define the function

$$Q_{\sigma,n}(t) := \frac{1}{\sqrt{n}} \big[\sum_{\ell=1}^{\infty} \min\{t^2, \mu_\ell\} \big]^{1/2}.$$

Let $\nu_n > 0$ be the smallest positive solution to the inequality

$$40\nu_n^2 \geq Q_{\sigma,n}(\nu_n), \tag{8}$$

where the 40 is simply used for technical convenience. We refer to ν_n as the *critical univariate rate*, as it is the minimax-optimal rate for $L^2(\mathbb{P})$-estimation of a single univariate function in the Hilbert space $\mathcal{H}$ (e.g., Mendelson, 2002; van de Geer, 2000). This quantity will be referred to throughout the remainder of the paper.

Our choices of regularization parameters are specified in terms of the quantity

$$\gamma_n := \kappa \max\big\{\nu_n, \sqrt{\frac{\log d}{n}}\big\}, \tag{9}$$

where κ is a fixed constant that we choose later. We assume that each function within the unit ball of the univariate Hilbert space is uniformly bounded by a constant multiple of its Hilbert norm—that is, for each $j = 1, \ldots, d$ and each $f_j \in \mathcal{H}$,

$$\|f_j\|_\infty := \sup_{x_j} |f_j(x_j)| \leq c \, \|f_j\|_{\mathcal{H}}. \tag{10}$$

This condition is satisfied for many kernel classes including Sobolev spaces, and any univariate RKHS in which the kernel function[2] bounded uniformly by c. Such a condition is routinely imposed for proving upper bounds on rates of convergence for non-parametric least squares in the univariate case $d = 1$ (see, e.g., Stone, 1985; van de Geer, 2000). Note that this univariate boundedness does not imply that the multivariate functions $f = \sum_{j \in S} f_j$ in $\mathcal{F}$ are uniformly bounded independently of (d, s); rather, since such functions are the sum of s terms, they can take on values of the order $\sqrt{s}$.

The following result applies to any class $\mathcal{F}_{d,s,\mathcal{H}}$ of sparse additive models based on a univariate Hilbert space satisfying condition (10), and to the estimator (6) based on n i.i.d. samples $(x_i, y_i)_{i=1}^n$ from the observation model (5).

Theorem 1 *Let $\widehat{f}$ be any minimizer of the convex program* (6) *with regularization parameters $\lambda_n \geq 16\gamma_n$ and $\rho_n \geq 16\gamma_n^2$. Then provided that $n\gamma_n^2 = \Omega(\log(1/\gamma_n))$, there are universal constants (C, c_1, c_2) such that*

$$\mathbb{P}\bigg[\max\{\|\widehat{f} - f^*\|_2^2, \, \|\widehat{f} - f^*\|_n^2\} \geq C\big\{s\lambda_n^2 + s\rho_n\big\}\bigg] \leq c_1 \exp(-c_2 n\gamma_n^2).$$

We provide the proof of Theorem 1 in Section 4.1.

3.2.1 REMARKS

First, the technical condition $n\gamma_n^2 = \Omega(\log(1/\gamma_n))$ is quite mild, and satisfied in most cases of interest, among them the kernels considered below in Corollaries 1 and 2.

Second, note that setting $\lambda_n = c\gamma_n$ and $\rho_n = c\gamma_n^2$ for some constant $c \in [16, \infty)$ yields the rate $\Theta(s\gamma_n^2 + s\rho_n) = \Theta(\frac{s \log d}{n} + s\nu_n^2)$. This rate may be interpreted as the sum of a subset selection term $(\frac{s \log d}{n})$ and an s-dimensional estimation term $(s\nu_n^2)$. Note that the subset selection term $(\frac{s \log d}{n})$ is independent of the choice of Hilbert space $\mathcal{H}$, whereas the s-dimensional estimation term is independent of the ambient dimension d. Depending on the scaling of the triple (n, d, s) and the smoothness of the univariate RKHS $\mathcal{H}$, either the subset selection term or function estimation term may dominate. In general, if $\frac{\log d}{n} = o(\nu_n^2)$, the s-dimensional estimation term dominates, and vice versa otherwise. At the boundary, the scalings of the two terms are equivalent.

Finally, for clarity, we have stated our result in the case where the univariate Hilbert space $\mathcal{H}$ is identical across all co-ordinates. However, our proof extends with only notational changes to the general setting, in which each co-ordinate j is endowed with a (possibly distinct) Hilbert space $\mathcal{H}_j$. In this case, the M-estimator returns a function $\widehat{f}$ such that (with high probability)

$$\max\big\{\|\widehat{f} - f^*\|_n^2, \, \|\widehat{f} - f^*\|_2^2\big\} \;\leq\; C\bigg\{\frac{s \log d}{n} + \sum_{j \in S} \nu_{n,j}^2\bigg\},$$

[2] Indeed, we have

$$\sup_{x_j} |f_j(x_j)| = \sup_{x_j} |\langle f_j(.), \mathbb{K}(., x_j)\rangle_{\mathcal{H}}| \;\leq\; \sup_{x_j} \sqrt{\mathbb{K}(x_j, x_j)} \|f_j\|_{\mathcal{H}}.$$

where $\nu_{n,j}$ is the critical univariate rate associated with the Hilbert space $\mathcal{H}_j$, and S is the subset on which f^* is supported.

Theorem 1 has a number of corollaries, obtained by specifying particular choices of kernels. First, we discuss m-rank operators, meaning that the kernel function $\mathbb{K}$ can be expanded in terms of m eigenfunctions. This class includes linear functions, polynomial functions, as well as any function class based on finite dictionary expansions. First we present a corollary for finite-rank kernel classes.

Corollary 1 *Under the same conditions as Theorem 1, consider an univariate kernel with finite rank m. Then any solution $\widehat{f}$ to the problem* (6) *with $\lambda_n = c\gamma_n$ and $\rho_n = c\gamma_n^2$ with $16 \leq c < \infty$ satisfies*

$$\mathbb{P}\Big[\max\big\{\|\widehat{f}-f^*\|_n^2, \|\widehat{f}-f^*\|_2^2\big\} \geq C\big\{\frac{s\log d}{n} + s\frac{m}{n}\big\}\Big] \leq c_1 \exp\big(-c_2(m+\log d)\big).$$

Proof : It suffices to show that the critical univariate rate (8) satisfies the scaling $\nu_n^2 = O(m/n)$. For a finite-rank kernel and any $t > 0$, we have

$$Q_{\sigma,n}(t) = \frac{1}{\sqrt{n}}\sqrt{\sum_{j=1}^{m}\min\{t^2, \mu_j\}} \;\leq\; t\sqrt{\frac{m}{n}},$$

from which the claim follows by the definition (8). ■

Next, we present a result for the RKHS's with infinitely many eigenvalues, but whose eigenvalues decay at a rate $\mu_k \simeq (1/k)^{2\alpha}$ for some parameter $\alpha > 1/2$. Among other examples, this type of scaling covers the case of Sobolev spaces, say consisting of functions with α derivatives (e.g., Birman and Solomjak, 1967; Gu, 2002).

Corollary 2 *Under the same conditions as Theorem 1, consider an univariate kernel with eigenvalue decay $\mu_k \simeq (1/k)^{2\alpha}$ for some $\alpha > 1/2$. Then the kernel estimator defined in* (6) *with $\lambda_n = c\gamma_n$ and $\rho_n = c\gamma_n^2$ with $16 \leq c < \infty$ satisfies*

$$\mathbb{P}\Big[\max\big\{\|\widehat{f}-f^*\|_n^2, \|\widehat{f}-f^*\|_2^2\big\} \geq C\big\{\frac{s\log d}{n} + s\big(\frac{1}{n}\big)^{\frac{2\alpha}{2\alpha+1}}\big\}\Big] \leq c_1 \exp\big(-c_2(n^{\frac{1}{2\alpha+1}}+\log d)\big).$$

Proof : As in the previous corollary, we need to compute the critical univariate rate ν_n. Given the assumption of polynomial eigenvalue decay, a truncation argument shows that $Q_{\sigma,n}(t) = O\big(\frac{t^{1-\frac{1}{2\alpha}}}{\sqrt{n}}\big)$. Consequently, the critical univariate rate (8) satisfies the scaling $\nu_n^2 \asymp \nu_n^{1-\frac{1}{2\alpha}}/\sqrt{n}$, or equivalently, $\nu_n^2 \asymp n^{-\frac{2\alpha}{2\alpha+1}}$. ■

3.3 Minimax Lower Bounds

In this section, we derive lower bounds on the minimax error in the $L^2(\mathbb{P})$-norm that complement the achievability results derived in Theorem 1. Given the function class $\mathcal{F}$, we define the minimax $L^2(\mathbb{P})$-error $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})$ to be the largest quantity such that

$$\inf_{\widehat{f}_n}\sup_{f^*\in\mathcal{F}} \mathbb{P}_{f^*}[\|\widehat{f}_n - f^*\|_2^2 \geq \mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})] \geq 1/2, \tag{11}$$

where the infimum is taken over all measurable functions of the n samples $\{(x_i, y_i)\}_{i=1}^n$, and $\mathbb{P}_{f^*}$ denotes the data distribution when the unknown function is f^*. Given this definition, note that Markov's inequality implies that

$$\inf_{\widehat{f}_n} \sup_{f^* \in \mathcal{F}} \mathbb{E}\|\widehat{f}_n - f^*\|_2^2 \geq \frac{\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})}{2}.$$

Central to our proof of the lower bounds is the metric entropy structure of the univariate reproducing kernel Hilbert spaces. More precisely, our lower bounds depend on the *packing entropy,* defined as follows. Let $(\mathcal{G}, \rho)$ be a totally bounded metric space, consisting of a set $\mathcal{G}$ and a metric $\rho : \mathcal{G} \times \mathcal{G} \to \mathbb{R}_+$. An ε-packing of $\mathcal{G}$ is a collection $\{f^1, \ldots, f^M\} \subset \mathcal{G}$ such that $\rho(f^i, f^j) \geq \varepsilon$ for all $i \neq j$. The ε-packing number $M(\varepsilon; \mathcal{G}, \rho)$ is the cardinality of the largest ε-packing. The packing entropy is the simply the logarithm of the packing number, namely the quantity $\log M(\varepsilon; \mathcal{G}, \rho)$, to which we also refer as the metric entropy. In this paper, we derive explicit minimax lower bounds for two different scalings of the univariate metric entropy.

3.3.1 LOGARITHMIC METRIC ENTROPY

There exists some $m > 0$ such that

$$\log M(\varepsilon; \mathbb{B}_{\mathcal{H}}(1), L^2(\mathbb{P})) \simeq m \log(1/\varepsilon) \qquad \text{for all } \varepsilon \in (0,1). \tag{12}$$

Function classes with metric entropy of this type include linear functions (for which $m = k$), univariate polynomials of degree k (for which $m = k+1$), and more generally, any function space with finite VC-dimension (van der Vaart and Wellner, 1996). This type of scaling also holds for any RKHS based on a kernel with rank m (e.g., see Carl and Triebel, 1980), and these finite-rank kernels include both linear and polynomial functions as special cases.

3.3.2 POLYNOMIAL METRIC ENTROPY

There exists some $\alpha > 0$ such that

$$\log M(\varepsilon; \mathbb{B}_{\mathcal{H}}(1), L^2(\mathbb{P})) \simeq (1/\varepsilon)^{1/\alpha} \qquad \text{for all } \varepsilon \in (0,1). \tag{13}$$

Various types of Sobolev/Besov classes exhibit this type of metric entropy decay (e.g., Birman and Solomjak, 1967; Gu, 2002). In fact, any RKHS in which the kernel eigenvalues decay at a rate $k^{-2\alpha}$ have a metric entropy with this scaling (Carl and Stephani, 1990; Carl and Triebel, 1980).

We are now equipped to state our lower bounds on the minimax risk (11):

Theorem 2 *Given n i.i.d. samples from the sparse additive model* (5) *with sparsity* $s \leq d/4$*, there is an universal constant* $C > 0$ *such that:*

(a) For a univariate class $\mathcal{H}$ *with logarithmic metric entropy* (12) *indexed by parameter m, we have*

$$\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}}) \;\geq\; C\left\{\frac{s\log(d/s)}{n} + s\,\frac{m}{n}\right\}.$$

(b) *For a univariate class $\mathcal{H}$ with polynomial metric entropy* (13) *indexed by* α*, we have*

$$\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}}) \;\geq\; C\left\{\frac{s\log(d/s)}{n} + s\big(\frac{1}{n}\big)^{\frac{2\alpha}{2\alpha+1}}\right\}.$$

The proof of Theorem 2 is provided in Section 4.2. The most important consequence of Theorem 2 is in establishing the minimax-optimality of the results given in Corollary 1 and 2; in particular, in the regime of sub-linear sparsity (i.e., for which $\log d = O(\log(d/s))$), the combination of Theorem 2 with these corollaries identifies the minimax rates up to constant factors.

3.4 Comparison With Other Estimators

It is interesting to compare these convergence rates in $L^2(\mathbb{P}_n)$ error with those established in the past work. Ravikumar et al. (2009) show that any solution to their back-fitting method is consistent in terms of mean-squared error risk (see Theorem 3 in their paper), but their analysis does not allow $s \to \infty$. The method of Koltchinskii and Yuan (2008) is based regularizing the least-squares loss with the $(\mathcal{H},1)$-norm penalty—that is, the regularizer $\sum_{j=1}^{d}\|f_j\|_{\mathcal{H}}$; Theorem 2 in their paper provides a rate that holds for the triple (n,d,s) tending to infinity. In quantitative terms, however, their rates are looser than those given here; in particular, their bound includes a term of the order $\frac{s^3\log d}{n}$, which is larger than the bound in Theorem 1. Meier et al. (2009) analyze a different M-estimator to the one we analyze in this paper. Rather than adding two separate $(\mathcal{H},1)$-norm and an $(\|.\|_n,1)$-norm penalties, they combine the two terms into a single sparsity and smoothness penalty. For their estimator, Meier et al. (2009) establish a convergence rate of the form $O(s(\frac{\log d}{n})^{\frac{2\alpha}{2\alpha+1}})$ in the case of α-smooth Sobolev spaces (see Theorem 1 in their paper). Note that relative to optimal rates given here in Theorem 2(b), this scaling is sub-optimal: more precisely, we either have $\frac{\log d}{n} < (\frac{\log d}{n})^{\frac{2\alpha}{2\alpha+1}}$, when the subset selection term dominates, or $(\frac{1}{n})^{\frac{2\alpha}{2\alpha+1}} < (\frac{\log d}{n})^{\frac{2\alpha}{2\alpha+1}}$, when the s-dimensional estimation term dominates. In all of the above-mentioned methods, it is unclear whether or not a sharper analysis would yield better rates. Finally, Koltchinskii and Yuan (2010) analyze the same estimator as the M-estimator (6), and for the case of polynomial metric entropy, establish the same rates Theorem 1, albeit under a global boundedness condition. In the following section, we study the implications of this assumption.

3.5 Upper Bounds Under A Global Boundedness Assumption

As discussed previously in the introduction, the paper of Koltchinskii and Yuan (2010), referred to as KY for short, is based on the M-estimator (6). In terms of rates obtained, they establish a convergence rate based on two terms as in Theorem 1, but with a pre-factor that depends on the global quantity

$$B = \sup_{f\in\mathcal{F}_{d,s,\mathcal{H}}} \|f\|_\infty \;=\; \sup_{f\in\mathcal{F}_{d,s,\mathcal{H}}}\sup_{x}|f(x)|,$$

assumed to be bounded independently of dimension and sparsity. Such types of global boundedness conditions are fairly standard in classical non-parametric estimation, where they have no effect on minimax rates. In sharp contrast, the analysis of this section shows that for sparse additive models in the regime $s = \Omega(\sqrt{n})$, such global boundedness can *substantially speed up* minimax rates, showing that the rates proven in KY are not minimax optimal for these classes. The underlying insight is as follows: when the sparsity grows, imposing global boundedness over s-variate functions substantially reduces the effective dimension from its original size s to a lower dimensional quantity, which we denote by $sK_B(s,n)$, and moreover, the quantity $K_B(s,n) \to 0$ when $s = \Omega(\sqrt{n})$ as described below.

Recall the definition (4) of the function class $\mathcal{F}_{d,s,\mathcal{H}}$. The model considered in the KY paper is the smaller function class

$$\mathcal{F}^*_{d,s,\mathcal{H}}(B) := \bigcup_{\substack{S \subset \{1,2,\ldots,d\} \\ |S|=s}} \mathcal{H}(S,B),$$

where $\mathcal{H}(S,B) := \big\{ f = \sum_{j \in S} f_j \mid f_j \in \mathcal{H}, \text{ and } f_j \in \mathbb{B}_{\mathcal{H}}(1) \ \forall\, j \in S \text{ and } \|f\|_\infty \leq B \big\}$.

The following theorem provides sharper rates for the Sobolev case, in which each univariate Hilbert space has eigenvalues decaying as $\mu_k \simeq k^{-2\alpha}$ for some smoothness parameter $\alpha > 1/2$. Our probabilistic bounds involve the quantity

$$\delta_n := \max\Big(\sqrt{\frac{s \log(d/s)}{n}}, B\big(\frac{s^{\frac{1}{\alpha}} \log s}{n}\big)^{1/4} \Big), \tag{14}$$

and our rates are stated in terms of the function

$$K_B(s,n) := B\sqrt{\log s}\big(s^{-1/2\alpha} n^{1/(4\alpha+2)}\big)^{2\alpha-1},$$

where it should be noted that $K_B(s,n) \to 0$ if $s = \Omega(\sqrt{n})$.

With this notation, we have the following *upper bound* on the minimax risk over the function class $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$.

Theorem 3 *Consider any RKHS $\mathcal{H}$ with eigenvalue decay $k^{-2\alpha}$, and uniformly bounded eigenfunctions (i.e., $\|\phi_k\|_\infty \leq C < \infty$ for all k). Then there are universal constants (c_1, c_2, κ) such that with probability greater than $1 - 2\exp\big(-c_1 n\delta_n^2\big)$, we have*

$$\min_{\hat{f}} \max_{f^* \in \mathcal{F}^*_{d,s,\mathcal{H}}(B)} \|\widehat{f} - f^*\|_2^2 \leq \underbrace{\kappa^2(1+B)Csn^{-\frac{2\alpha}{2\alpha+1}}\Bigg(K_B(s,n) + n^{-1/(2\alpha+1)}\log(d/s)\Bigg)}_{\mathfrak{M}_{\mathbb{P}}(\mathcal{F}^*_{d,s,\mathcal{H}}(B))}, \tag{15}$$

as long as $n\delta_n^2 = \Omega(\log(1/\delta_n))$.

We provide the proof of Theorem 3 in Section 4.3; it is based on analyzing directly the least-squares estimator over $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$. The assumption that $\|\phi_k\|_\infty \leq C < \infty$ for all k includes the usual Sobolev spaces in which ϕ_k are (rescaled) Fourier basis functions. An immediate consequence of Theorem 3 is that the minimax rates over the function class $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$ can be strictly faster than minimax rates for the class $\mathcal{F}_{d,s,\mathcal{H}}$, which does not impose global boundedness. Recall that the minimax lower bound from Theorem 2 (b) is based on the quantity

$$\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}}) \ := \ C_1\big\{s\big(\frac{1}{n}\big)^{\frac{2\alpha}{2\alpha+1}} + \frac{s\log(d/s)}{n}\big\} \ = \ C_1 sn^{-\frac{2\alpha}{2\alpha+1}}\Bigg(1 + n^{-1/(2\alpha+1)}\log(d/s)\Bigg),$$

for a universal constant C_1. Note that up to constant factors, the achievable rate (15) from Theorem 3 is the same except that the term 1 is replaced by the function $K_B(s,n)$. Consequently, for scalings of (s,n) such that $K_B(s,n) \to 0$, global boundedness conditions lead to strictly faster rates.

Corollary 3 *Under the conditions of Theorem 3, we have*

$$\frac{\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})}{\mathfrak{M}_{\mathbb{P}}(\mathcal{F}^*_{d,s,\mathcal{H}}(B))} \geq \frac{C_1(1+n^{-1/(2\alpha+1)}\log(d/s))}{C\kappa^2(1+B)\,(K_B(s,n)+n^{-1/(2\alpha+1)}\log(d/s))} \to +\infty$$

whenever $B = O(1)$ and $K_B(s,n) \to 0$.

3.5.1 REMARKS

The quantity $K_B(s,n)$ is guaranteed to decay to zero as long as the sparsity index s grows in a non-trivial way with the sample size. For instance, if we have $s = \Omega(\sqrt{n})$ for a problem of dimension $d = O(n^\beta)$ for any $\beta \geq 1/2$, then it can be verified that $K_B(s,n) = o(1)$. As an alternative view of the differences, it can be noted that there are scalings of (n,s,d) for which the minimax rate $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})$ over $\mathcal{F}_{d,s,\mathcal{H}}$ is constant—that is, does not vanish as $n \to +\infty$—while the minimax rate $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}^*_{d,s,\mathcal{H}}(B))$ does vanish. As an example, consider the Sobolev class with smoothness $\alpha = 2$, corresponding to twice-differentiable functions. For a sparsity index $s = \Theta(n^{4/5})$, then Theorem 2(b) implies that $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}}) = \Omega(1)$, so that the minimax rate over $\mathcal{F}_{d,s,\mathcal{H}}$ is strictly bounded away from zero for all sample sizes. In contrast, under a global boundedness condition, Theorem 3 shows that the minimax rate is upper bounded as $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}^*_{d,s,\mathcal{H}}(B)) = O\big(n^{-1/5}\sqrt{\log n}\big)$, which tends to zero.

In summary, Theorem 3 and Theorem 2(b) together show that the minimax rates over $\mathcal{F}_{d,s,\mathcal{H}}$ and $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$ can be drastically different. Thus, global boundedness is a stringent condition in the high-dimensional setting; in particular, the rates given in Theorem 3 of Koltchinskii and Yuan (2010) are not minimax optimal when $s = \Omega(\sqrt{n})$.

4. Proofs

In this section, we provide the proofs of our three main theorems. For clarity in presentation, we split the proofs up into a series of lemmas, with the bulk of the more technical arguments deferred to the appendices. This splitting allows our presentation in Section 4 to be relatively streamlined.

4.1 Proof of Theorem 1

At a high-level, Theorem 1 is based on an appropriate adaptation to the non-parametric setting of various techniques that have been developed for sparse linear regression (e.g., Bickel et al., 2009; Negahban et al., 2009). In contrast to the parametric setting where classical tail bounds are sufficient, controlling the error terms in the non-parametric case requires more advanced techniques from empirical process theory. In particular, we make use of various concentration theorems for Gaussian and empirical processes (e.g., Ledoux, 2001; Massart, 2000; Pisier, 1989; van de Geer, 2000), as well as results on the Rademacher complexity of kernel classes (Bartlett et al., 2005; Mendelson, 2002).

At the core of the proof are three technical lemmas. First, Lemma 1 provides an upper bound on the Gaussian complexity of any function of the form $f = \sum_{j=1}^d f_j$ in terms of the norms $\|\cdot\|_{\mathcal{H},1}$ and $\|\cdot\|_{n,1}$ previously defined. Lemma 2 exploits the notion of decomposability (Negahban et al., 2009), as applied to these norms, in order to show that the error function belongs to a particular cone-shaped set. Finally, Lemma 3 establishes an upper bound on the $L^2(\mathbb{P})$ error of our estimator in terms of the $L^2(\mathbb{P}_n)$ error. The latter lemma can be interpreted as proving that our problem satisfies non-parametric analog of a restricted eigenvalue condition (Bickel et al., 2009), or more generally, of a restricted strong convexity condition (Negahban et al., 2009). The proof of Lemma 3 involves a new approach that combines the Sudakov minoration (Pisier, 1989) with a one-sided tail bound for non-negative random variables (Chung and Lu, 2006; Einmahl and Mason, 1996).

Throughout the proof, we use C and c_i, $i = 1,2,3,4$ to denote universal constants, independent of (n,d,s). Note that the precise numerical values of these constants may change from line to line. The reader should recall the definitions of ν_n and γ_n from Equations (8) and (9) respectively. For a subset $A \subseteq \{1,2,\ldots,d\}$ and a function of the form $f = \sum_{j=1}^{d} f_j$, we adopt the convenient notation

$$\|f_A\|_{n,1} := \sum_{j\in A} \|f_j\|_n, \quad \text{and} \quad \|f_A\|_{\mathcal{H},1} := \sum_{j\in A} \|f_j\|_{\mathcal{H}}. \tag{16}$$

We begin by establishing an inequality on the error function $\widehat{\Delta} := \widehat{f} - f^*$. Since $\widehat{f}$ and f^* are, respectively, optimal and feasible for the problem (6), we are guaranteed that $\mathcal{L}(\widehat{f}) \leq \mathcal{L}(f^*)$, and hence that the error function $\widehat{\Delta}$ satisfies the bound

$$\frac{1}{2n}\sum_{i=1}^{n}(w_i + \bar{f} - \bar{y}_n - \widehat{\Delta}(x_i))^2 + \lambda_n\|\widehat{f}\|_{n,1} + \rho_n\|\widehat{f}\|_{\mathcal{H},1} \leq \frac{1}{2n}\sum_{i=1}^{n}(w_i + \bar{f} - \bar{y}_n)^2 + \lambda_n\|f^*\|_{n,1} + \rho_n\|f^*\|_{\mathcal{H},1}.$$

Some simple algebra yields the bound

$$\frac{1}{2}\|\widehat{\Delta}\|_n^2 \leq \Big|\frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)\Big| + |\bar{y}_n - \bar{f}|\Big|\frac{1}{n}\sum_{i=1}^{n}\widehat{\Delta}(x_i)\Big| + \lambda_n\|\widehat{\Delta}\|_{n,1} + \rho_n\|\widehat{\Delta}\|_{\mathcal{H},1}. \tag{17}$$

Following the terminology of van de Geer (2000), we refer to this bound as our *basic inequality*.

4.1.1 CONTROLLING DEVIATION FROM THE MEAN

Our next step is to control the error due to estimating the mean $|\bar{y}_n - \bar{f}|$. We begin by observing that this error term can be written as $\bar{y}_n - \bar{f} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{f})$. Next we observe that $y_i - \bar{f} = \sum_{j\in S} f_j^*(x_{ij}) + w_i$ is the sum of the s independent random variables $f_j^*(x_{ij})$, each bounded in absolute value by one, along with the independent sub-Gaussian noise term w_i; consequently, the variable $y_i - \bar{f}$ is sub-Gaussian with parameter at most $\sqrt{s+1}$. (See, for instance, Lemma 1.4 in Buldygin and Kozachenko 2000). By applying standard sub-Gaussian tail bounds, we have $\mathbb{P}(|\bar{y}_n - \bar{f}| > t) \leq 2\exp(-\frac{nt^2}{2(s+1)})$, and hence, if we define the event $\mathcal{C}(\gamma_n) = \{|\bar{y}_n - \bar{f}| \leq \sqrt{s}\gamma_n\}$, we are guaranteed

$$\mathbb{P}[\mathcal{C}(\gamma_n)] \geq 1 - 2\exp(-\frac{n\gamma_n^2}{4}).$$

For the remainder of the proof, we condition on the event $\mathcal{C}(\gamma_n)$. Under this conditioning, the bound (17) simplifies to:

$$\frac{1}{2}\|\widehat{\Delta}\|_n^2 \leq \Big|\frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)\Big| + \sqrt{s}\gamma_n\|\widehat{\Delta}\|_n + \lambda_n\|\widehat{\Delta}\|_{n,1} + \rho_n\|\widehat{\Delta}\|_{\mathcal{H},1},$$

where we have applied the Cauchy-Schwarz inequality to write $\big|\frac{1}{n}\sum_{i=1}^{n}\widehat{\Delta}(x_i)\big| \leq \|\widehat{\Delta}\|_n$.

4.1.2 CONTROLLING THE GAUSSIAN COMPLEXITY TERM

The following lemma provides control the Gaussian complexity term on the right-hand side of inequality (17) by bounding the Gaussian complexity for the univariate functions $\widehat{\Delta}_j$, $j = 1, 2, \ldots, d$ in terms of their $\|\cdot\|_n$ and $\|\cdot\|_{\mathcal{H}}$ norms. In particular, recalling that $\gamma_n = \kappa \max\{\sqrt{\frac{\log d}{n}}, \nu_n\}$, we have the following lemma.

Lemma 1 *Define the event*

$$\mathcal{T}(\gamma_n) := \Big\{ \forall\ j = 1, 2, \ldots, d,\ \big|\frac{1}{n}\sum_{i=1}^{n} w_i \widehat{\Delta}_j(x_{ij})\big| \leq 8\gamma_n^2\ \|\widehat{\Delta}_j\|_{\mathcal{H}} + 8\gamma_n\ \|\widehat{\Delta}_j\|_n \Big\}.$$

Then under the condition $n\gamma_n^2 = \Omega(\log(1/\gamma_n))$, we have

$$\mathbb{P}(\mathcal{T}(\gamma_n)) \geq 1 - c_1 \exp(-c_2 n \gamma_n^2).$$

The proof of this lemma, provided in Appendix B, uses concentration of measure for Lipschitz functions of Gaussian random variables (e.g., Ledoux, 2001), combined with peeling and weighting arguments from empirical process theory (Alexander, 1987; van de Geer, 2000). In particular, the subset selection term $(\frac{s \log d}{n})$ in Theorem 1 arises from taking the maximum over all d components.

The remainder of our analysis involves conditioning on the event $\mathcal{T}(\gamma_n) \cap \mathcal{C}(\gamma_n)$. Using Lemma 1, when conditioned on the event $\mathcal{T}(\gamma_n) \cap \mathcal{C}(\gamma_n)$ we have:

$$\|\widehat{\Delta}\|_n^2 \quad \leq \quad 2\sqrt{s}\gamma_n \|\widehat{\Delta}\|_n + (16\gamma_n + 2\lambda_n)\|\widehat{\Delta}\|_{n,1} + (16\gamma_n^2 + 2\rho_n)\|\widehat{\Delta}\|_{\mathcal{H},1}. \tag{18}$$

4.1.3 EXPLOITING DECOMPOSABILITY

Recall that S denotes the true support of the unknown function f^*. By the definition (16), we can write $\|\widehat{\Delta}\|_{n,1} = \|\widehat{\Delta}_S\|_{n,1} + \|\widehat{\Delta}_{S^c}\|_{n,1}$, where $\widehat{\Delta}_S := \sum_{j \in S} \widehat{\Delta}_j$ and $\widehat{\Delta}_{S^c} := \sum_{j \in S^c} \widehat{\Delta}_j$. Similarly, we have an analogous representation for $\|\widehat{\Delta}\|_{\mathcal{H},1}$. The next lemma shows that conditioned on the event $\mathcal{T}(\gamma_n)$, the quantities $\|\widehat{\Delta}\|_{\mathcal{H},1}$ and $\|\widehat{\Delta}\|_{n,1}$ are not significantly larger than the corresponding norms as applied to the function $\widehat{\Delta}_S$.

Lemma 2 *Conditioned on the events $\mathcal{T}(\gamma_n)$ and $\mathcal{C}(\gamma_n)$, and with the choices $\lambda_n \geq 16\gamma_n$ and $\rho_n \geq 16\gamma_n^2$, we have*

$$\lambda_n \|\widehat{\Delta}\|_{n,1} + \rho_n \|\widehat{\Delta}\|_{\mathcal{H},1} \leq 4\lambda_n \|\widehat{\Delta}_S\|_{n,1} + 4\rho_n \|\widehat{\Delta}_S\|_{\mathcal{H},1} + \frac{1}{2} s \gamma_n^2. \tag{19}$$

The proof of this lemma, provided in Appendix C, is based on the decomposability (see Negahban et al. 2009) of the $\|\cdot\|_{\mathcal{H},1}$ and $\|\cdot\|_{n,1}$ norms. This lemma allows us to exploit the sparsity assumption, since in conjunction with Lemma 1, we have now bounded the right-hand side of the inequality (18) by terms involving only $\widehat{\Delta}_S$.

For the remainder of the proof of Theorem 1, we assume $\lambda_n \geq 16\gamma_n$ and $\rho_n \geq 16\gamma_n^2$. In particular, still conditioning on $\mathcal{C}(\gamma_n) \cap \mathcal{T}(\gamma_n)$ and applying Lemma 2 to inequality (18), we obtain

$$\begin{aligned} \|\widehat{\Delta}\|_n^2 \quad &\leq \quad 2\sqrt{s}\gamma_n \|\widehat{\Delta}\|_n + 3\lambda_n \|\widehat{\Delta}\|_{n,1} + 3\rho_n \|\widehat{\Delta}\|_{\mathcal{H},1} \\ &\leq \quad 2\sqrt{s}\lambda_n \|\widehat{\Delta}\|_n + 12\lambda_n \|\widehat{\Delta}_S\|_{n,1} + 12\rho_n \|\widehat{\Delta}_S\|_{\mathcal{H},1} + \frac{3}{32} s \rho_n, \end{aligned}$$

Finally, since both $\widehat{f}_j$ and f_j^* belong to $\mathbb{B}_{\mathcal{H}}(1)$, we have $\|\widehat{\Delta}_j\|_{\mathcal{H}} \leq \|\widehat{f}_j\|_{\mathcal{H}} + \|f_j^*\|_{\mathcal{H}} \leq 2$, which implies that $\|\widehat{\Delta}_S\|_{\mathcal{H},1} \leq 2s$, and hence

$$\|\widehat{\Delta}\|_n^2 \leq 2\sqrt{s}\lambda_n \|\widehat{\Delta}\|_n + 12\lambda_n \|\widehat{\Delta}_S\|_{n,1} + 25 s\rho_n. \tag{20}$$

4.1.4 UPPER BOUNDING $\|\widehat{\Delta}_S\|_{n,1}$

The next step is to control the term $\|\widehat{\Delta}_S\|_{n,1} = \sum_{j\in S}\|\widehat{\Delta}_j\|_n$ that appears in the upper bound (20). Ideally, we would like to upper bound it by a quantity of the order $\sqrt{s}\|\widehat{\Delta}_S\|_2 = \sqrt{s}\sqrt{\sum_{j\in S}\|\widehat{\Delta}_j\|_2^2}$. Such an upper bound would follow immediately if it were phrased in terms of the population $\|\cdot\|_2$-norm rather than the empirical-$\|\cdot\|_n$ norm, but there are additional cross-terms with the empirical norm. Accordingly, a somewhat more delicate argument is required, which we provide here. First define the events

$$\mathcal{A}_j(\lambda_n) := \{\|\widehat{\Delta}_j\|_n \leq 2\|\widehat{\Delta}_j\|_2 + \lambda_n\},$$

and $\mathcal{A}(\lambda_n) = \cap_{j=1}^d \mathcal{A}_j(\lambda_n)$. By applying Lemma 7 from Appendix A with $t = \lambda_n \geq 16\gamma_n$ and $b = 2$, we conclude that $\|\widehat{\Delta}_j\|_n \leq 2\|\widehat{\Delta}_j\|_2 + \lambda_n$ with probability greater than $1 - c_1\exp(-c_2 n\lambda_n^2)$. Consequently, if we define the event $\mathcal{A}(\lambda_n) = \cap_{j\in S}\mathcal{A}_j(\lambda_n)$, then this tail bound together with the union bound implies that

$$\mathbb{P}[\mathcal{A}^c(\lambda_n)] \leq s\, c_1\exp(-c_2 n\lambda_n^2) \;\leq\; c_1\exp(-c_2' n\lambda_n^2), \tag{21}$$

where we have used the fact that $\lambda_n = \Omega(\sqrt{\frac{\log s}{n}})$. Now, conditioned on the event $\mathcal{A}(\lambda_n)$, we have

$$\begin{aligned}\|\widehat{\Delta}_S\|_{n,1} = \sum_{j\in S}\|\widehat{\Delta}_j\|_n \;&\leq\; 2\sum_{j\in S}\|\widehat{\Delta}_j\|_2 + s\lambda_n \\ &\leq\; 2\sqrt{s}\|\widehat{\Delta}_S\|_2 + s\lambda_n \leq 2\sqrt{s}\|\widehat{\Delta}\|_2 + s\lambda_n.\end{aligned} \tag{22}$$

Substituting this upper bound (22) on $\|\widehat{\Delta}_S\|_{n,1}$ into our earlier inequality (20) yields

$$\|\widehat{\Delta}\|_n^2 \;\leq\; 2\sqrt{s}\lambda_n\|\widehat{\Delta}\|_n + 24\sqrt{s}\lambda_n\|\widehat{\Delta}\|_2 + 12 s\lambda_n^2 + 25 s\rho_n. \tag{23}$$

At this point, we encounter a challenge due to the unbounded nature of our function class. In particular, if $\|\widehat{\Delta}\|_2$ were upper bounded by $C\max(\|\widehat{\Delta}\|_n, \sqrt{s}\lambda_n, \sqrt{s\rho_n})$, then the upper bound (23) would immediately imply the claim of Theorem 1. If one were to assume global boundedness of the multivariate functions $\widehat{f}$ and f^*, as done in past work of Koltchinskii and Yuan (2010), then an upper bound on $\|\widehat{\Delta}\|_2$ of this form would directly follow from known results (e.g., Theorem 2.1 in Bartlett et al. 2005.) However, since we do not impose global boundedness, we need to develop a novel approach to this final hurdle.

4.1.5 CONTROLLING $\|\widehat{\Delta}\|_2$ FOR UNBOUNDED CLASSES

For the remainder of the proof, we condition on the event $\mathcal{A}(\lambda_n) \cap \mathcal{T}(\gamma_n) \cap \mathcal{C}(\gamma_n)$. We split our analysis into three cases. Throughout the proof, we make use of the quantity

$$\tilde{\delta}_n := B\max(\sqrt{s}\lambda_n, \sqrt{s\rho_n}), \tag{24}$$

where $B \in (1, \infty)$ is a constant to be chosen later in the argument.

Case 1: If $\|\widehat{\Delta}\|_2 < \|\widehat{\Delta}\|_n$, then combined with inequality (23), we conclude that

$$\|\widehat{\Delta}\|_n^2 \leq 2\sqrt{s}\lambda_n \|\widehat{\Delta}\|_n + 24\sqrt{s}\lambda_n \|\widehat{\Delta}\|_n + 12 s\lambda_n^2 + 25 s\rho_n.$$

This is a quadratic inequality in terms of the quantity $\|\widehat{\Delta}\|_n$, and some algebra shows that it implies the bound $\|\widehat{\Delta}\|_n \leq 15 \max(\sqrt{s}\lambda_n, \sqrt{s\rho_n})$. By assumption, we then have $\|\widehat{\Delta}\|_2 \leq 15 \max(\sqrt{s}\lambda_n, \sqrt{s\rho_n})$ as well, thereby completing the proof of Theorem 1.

Case 2: If $\|\widehat{\Delta}\|_2 < \tilde{\delta}_n$, then together with the bound (23), we conclude that

$$\|\widehat{\Delta}\|_n^2 \leq 2\sqrt{s}\lambda_n \|\widehat{\Delta}\|_n + 24\sqrt{s}\lambda_n \tilde{\delta}_n + 12 s\lambda_n^2 + 25 s\rho_n.$$

This inequality is again a quadratic in $\|\widehat{\Delta}\|_n$; moreover, note that by definition (24) of $\tilde{\delta}_n$, we have $s\lambda_n^2 + s\rho_n = O(\tilde{\delta}_n^2)$. Consequently, this inequality implies that $\|\widehat{\Delta}\|_n \leq C\tilde{\delta}_n$ for some constant C. Our starting assumption implies that $\|\widehat{\Delta}\|_2 \leq \tilde{\delta}_n$, so that the claim of Theorem 1 follows in this case.

Case 3: Otherwise, we may assume that $\|\widehat{\Delta}\|_2 \geq \tilde{\delta}_n$ and $\|\widehat{\Delta}\|_2 \geq \|\widehat{\Delta}\|_n$. In this case, the inequality (23) together with the bound $\|\widehat{\Delta}\|_2 \geq \|\widehat{\Delta}\|_n$ implies that

$$\|\widehat{\Delta}\|_n^2 \leq 2\sqrt{s}\lambda_n \|\widehat{\Delta}\|_2 + 24\sqrt{s}\lambda_n \|\widehat{\Delta}\|_2 + 12 s\lambda_n^2 + 25 s\rho_n. \tag{25}$$

Our goal is to establish a lower bound on the left-hand-side—namely, the quantity $\|\widehat{\Delta}\|_n^2$—in terms of $\|\widehat{\Delta}\|_2^2$. In order to do so, we consider the function class $\mathcal{G}(\lambda_n, \rho_n)$ defined by functions of the form $g = \sum_{j=1}^d g_j$, and such that

$$\lambda_n \|g\|_{n,1} + \rho_n \|g\|_{\mathcal{H},1} \leq\; 4\lambda_n \|g_S\|_{n,1} + 4\rho_n \|g_S\|_{\mathcal{H},1} + \frac{1}{32} s\rho_n, \tag{26}$$

$$\|g_S\|_{1,n} \leq\; 2\sqrt{s}\|g_S\|_2 + s\lambda_n \qquad \text{and} \tag{27}$$

$$\|g\|_n \leq \|g\|_2. \tag{28}$$

Conditioned on the events $\mathcal{A}(\gamma_n)$, $\mathcal{T}(\gamma_n)$ and $\mathcal{C}(\gamma_n)$, and with our choices of regularization parameter, we are guaranteed that the error function $\widehat{\Delta}$ satisfies all three of these constraints, and hence that $\widehat{\Delta} \in \mathcal{G}(\lambda_n, \rho_n)$. Consequently, it suffices to establish a lower bound on $\|g\|_n$ that holds uniformly over the class $\mathcal{G}(\lambda_n, \rho_n)$. In particular, define the event

$$\mathcal{B}(\lambda_n, \rho_n) := \left\{ \|g\|_n^2 \geq \|g\|_2^2/2 \quad \text{for all } g \in \mathcal{G}(\lambda_n, \rho_n) \quad \text{such that} \quad \|g\|_2 \geq \tilde{\delta}_n \right\}.$$

The following lemma shows that this event holds with high probability.

Lemma 3 *Under the conditions of Theorem 1, there are universal constants c_i such that*

$$\mathbb{P}[\mathcal{B}(\lambda_n, \rho_n)] \geq 1 - c_1 \exp(-c_2 n \gamma_n^2).$$

We note that this lemma can be interpreted as guaranteeing a version of restricted strong convexity (see Negahban et al., 2009) for the least-squares loss function, suitably adapted to the non-parametric setting. Since we do not assume global boundedness, the proof of this lemma requires a novel technical argument, one which combines a one-sided tail bound for non-negative random variables (Chung and Lu, 2006; Einmahl and Mason, 1996) with the Sudakov minoration (Pisier, 1989) for the Gaussian complexity. We refer the reader to Appendix D for the details of the proof.

Using Lemma 3 and conditioning on the event $\mathcal{B}(\lambda_n, \rho_n)$, we are guaranteed that $\|\widehat{\Delta}\|_n^2 \geq \|\widehat{\Delta}\|_2^2/2$, and hence, combined with our earlier bound (25), we conclude that

$$\|\widehat{\Delta}\|_2^2 \leq 4\sqrt{s}\lambda_n\|\widehat{\Delta}\|_2 + 48\sqrt{s}\lambda_n\|\widehat{\Delta}\|_2 + 24s\lambda_n^2 + 50s\rho_n.$$

Hence $\|\widehat{\Delta}\|_n \leq \|\widehat{\Delta}\|_2 \leq C\max(\sqrt{s}\lambda_n, \sqrt{s\rho_n})$, completing the proof of the claim in the third case.

In summary, the entire proof is based on conditioning on the three events $\mathcal{T}(\gamma_n)$, $\mathcal{A}(\lambda_n)$ and $\mathcal{B}(\lambda_n, \rho_n)$. From the bound (21) as well as Lemmas 1 and 3, we have

$$\mathbb{P}\big[\mathcal{T}(\gamma_n) \cap \mathcal{A}(\lambda_n) \cap \mathcal{B}(\lambda_n, \rho_n) \cap \mathcal{C}(\gamma_n)\big] \geq 1 - c_1 \exp\big(-c_2 n\gamma_n^2\big),$$

thereby showing that $\max\{\|\widehat{f} - f^*\|_n^2, \|\widehat{f} - f^*\|_2^2\} \leq C\max(s\lambda_n^2, s\rho_n)$ with the claimed probability. This completes the proof of Theorem 1.

4.2 Proof of Theorem 2

We now turn to the proof of the minimax lower bounds stated in Theorem 2. For both parts (a) and (b), the first step is to follow a standard reduction to testing (see, e.g., Has'minskii, 1978; Yang and Barron, 1999; Yu, 1996) so as to obtain a lower bound on the minimax error $\mathfrak{M}_{\mathbb{P}}(\mathcal{F}_{d,s,\mathcal{H}})$ in terms of the probability of error in a multi-way hypothesis testing. We then apply different forms of the Fano inequality (see Yang and Barron, 1999; Yu, 1996) in order to lower bound the probability of error in this testing problem. Obtaining useful bounds requires a precise characterization of the metric entropy structure of $\mathcal{F}_{d,s,\mathcal{H}}$, as stated in Lemma 4.

4.2.1 REDUCTION TO TESTING

We begin with the reduction to a testing problem. Let $\{f^1, \ldots, f^M\}$ be a δ_n-packing of $\mathcal{F}$ in the $\|\cdot\|_2$-norm, and let Θ be a random variable uniformly distributed over the index set $[M] := \{1, 2, \ldots, M\}$. Note that we are using M as a shorthand for the packing number $M(\delta_n; \mathcal{F}, \|\cdot\|_2)$. A standard argument (e.g., Has'minskii, 1978; Yang and Barron, 1999; Yu, 1996) then yields the lower bound

$$\inf_{\widehat{f}} \sup_{f^* \in \mathcal{F}} \mathbb{P}\big[\|\widehat{f} - f^*\|_2^2 \geq \delta_n^2/2\big] \geq \inf_{\widehat{\Theta}} \mathbb{P}[\widehat{\Theta} \neq \Theta],$$

where the infimum on the right-hand side is taken over all estimators $\widehat{\Theta}$ that are measurable functions of the data, and take values in the index set $[M]$.

Note that $\mathbb{P}[\widehat{\Theta} \neq \Theta]$ corresponds to the error probability in a multi-way hypothesis test, where the probability is taken over the random choice of Θ, the randomness of the design points $X_1^n := \{x_i\}_{i=1}^n$, and the randomness of the observations $Y_1^n := \{y_i\}_{i=1}^n$. Our initial analysis is performed conditionally on the design points, so that the only remaining randomness in the observations Y_1^n comes from the observation noise $\{w_i\}_{i=1}^n$. From Fano's inequality (Cover and Thomas, 1991), for any estimator $\widehat{\Theta}$, we have $\mathbb{P}\big[\widehat{\Theta} \neq \Theta \mid X_1^n\big] \geq 1 - \frac{I_{X_1^n}(\Theta; Y_1^n) + \log 2}{\log M}$, where $I_{X_1^n}(\Theta; Y_1^n)$ denotes the mutual information between Θ and Y_1^n with X_1^n fixed. Taking expectations over X_1^n, we obtain the lower bound

$$\mathbb{P}\big[\widehat{\Theta} \neq \Theta\big] \geq 1 - \frac{\mathbb{E}_{X_1^n}\big[I_{X_1^n}(\Theta; Y_1^n)\big] + \log 2}{\log M}. \tag{29}$$

The remainder of the proof consists of constructing appropriate packing sets of $\mathcal{F}$, and obtaining good upper bounds on the mutual information term in the lower bound (29).

4.2.2 CONSTRUCTING APPROPRIATE PACKINGS

We begin with results on packing numbers. Recall that $\log M(\delta;\mathcal{F},\|\cdot\|_2)$ denotes the δ-packing entropy of $\mathcal{F}$ in the $\|\cdot\|_2$ norm.

Lemma 4 *(a) For all* $\delta \in (0,1)$ *and* $s \leq d/4$*, we have*

$$\log M(\delta;\mathcal{F},\|\cdot\|_2) = O\big(s\ \log M(\frac{\delta}{\sqrt{s}};\mathbb{B}_{\mathcal{H}}(1),\|\cdot\|_2) + s\log\frac{d}{s}\big).$$

(b) For a Hilbert class with logarithmic metric entropy (12) *and such that* $\|f\|_2 \leq \|f\|_{\mathcal{H}}$*, there exists set* $\{f^1,\ldots,f^M\}$ *with* $\log M\ \geq\ C\left\{s\log(d/s)+sm\right\}$*, and*

$$\delta\ \leq\ \|f^k - f^\ell\|_2\ \leq\ 8\delta \qquad \textit{for all } k\neq\ell\in\{1,2,\ldots,M\}.$$

The proof, provided in Appendix E, is combinatorial in nature. We now turn to the proofs of parts (a) and (b) of Theorem 2.

4.2.3 PROOF OF THEOREM 2(A)

In order to prove this claim, it remains to exploit Lemma 4 in an appropriate way, and to upper bound the resulting mutual information. For the latter step, we make use of the generalized Fano approach (e.g., Yu, 1996).

From Lemma 4, we can find a set $\{f^1,\ldots,f^M\}$ that is a δ-packing of $\mathcal{F}$ in ℓ_2-norm, and such that $\|f^k - f^\ell\|_2 \leq 8\delta$ for all $k,\ell\in[M]$. For $k=1,\ldots,M$, let $\mathbb{Q}^k$ denote the conditional distribution of Y_1^n conditioned on X_1^n and the event $\{\Theta = k\}$, and let $D(\mathbb{Q}^k\,\|\,\mathbb{Q}^\ell)$ denote the Kullback-Leibler divergence. From the convexity of mutual information (Cover and Thomas, 1991), we have the upper bound $I_{X_1^n}(\Theta;Y_1^n) \leq \frac{1}{\binom{M}{2}}\sum_{k,\ell=1}^M D(\mathbb{Q}^k\,\|\,\mathbb{Q}^\ell)$. Given our linear observation model (5), we have

$$D(\mathbb{Q}^k\,\|\,\mathbb{Q}^\ell) = \frac{1}{2\sigma^2}\sum_{i=1}^n \big(f^k(x_i) - f^\ell(x_i)\big)^2\ =\ \frac{n\,\|f^k - f^\ell\|_n^2}{2},$$

and hence

$$\mathbb{E}_{X_1^n}\big[I_{X_1^n}(Y_1^n;\Theta)\big] \leq \frac{n}{2}\frac{1}{\binom{M}{2}}\sum_{k\neq\ell}\mathbb{E}_{X_1^n}[\|f^k - f^\ell\|_n^2]\ =\ \frac{n}{2}\frac{1}{\binom{M}{2}}\sum_{k\neq\ell}\|f^k - f^\ell\|_2^2.$$

Since our packing satisfies $\|f^k - f^\ell\|_2^2 \leq 64\delta^2$, we conclude that

$$\mathbb{E}_{X_1^n}\big[I_{X_1^n}(Y_1^n;\Theta)\big] \leq 32n\delta^2.$$

From the Fano bound (29), for any $\delta > 0$ such that $\frac{32n\delta^2+\log 2}{\log M} < \frac{1}{4}$, then we are guaranteed that $\mathbb{P}[\widehat{\Theta}\neq\Theta] \geq \frac{3}{4}$. From Lemma 4(b), our packing set satisfies $\log M \geq C\big\{sm + s\log(d/s)\big\}$, so that so that the choice $\delta^2 = C'\left\{\frac{sm}{n} + \frac{s\log(d/s)}{n}\right\}$, for a suitably small $C'>0$, can be used to guarantee the error bound $\mathbb{P}[\widehat{\Theta}\neq\Theta] \geq \frac{3}{4}$.

4.2.4 PROOF OF THEOREM 2(B)

In this case, we use an upper bounding technique due to Yang and Barron (1999) in order to upper bound the mutual information. Although the argument is essentially the same, it does not follow verbatim from their claims—in particular, there are some slight differences due to our initial conditioning—so that we provide the details here. By definition of the mutual information, we have

$$I_{X_1^n}(\Theta; Y_1^n) = \frac{1}{M}\sum_{k=1}^{M} D(\mathbb{Q}^k \,\|\, \mathbb{P}_Y),$$

where $\mathbb{Q}^k$ denotes the conditional distribution of Y_1^n given $\Theta = k$ and still with X_1^n fixed, whereas $\mathbb{P}_Y$ denotes the marginal distribution of $\mathbb{P}_Y$.

Let us define the notion of a covering number, in particular for a totally bounded metric space $(\mathcal{G}, \rho)$, consisting of a set $\mathcal{G}$ and a metric $\rho : \mathcal{G} \times \mathcal{G} \to \mathbb{R}_+$. An ε-covering set of $\mathcal{G}$ is a collection $\{f^1, \ldots, f^N\}$ of functions such that for all $f \in \mathcal{G}$ there exists $k \in \{1, 2, ..., N\}$ such that $\rho(f, f^k) \leq \varepsilon$. The ε-covering number $N(\varepsilon; \mathcal{G}, \rho)$ is the cardinality of the smallest ε-covering set.

Now let $\{g^1, \ldots, g^N\}$ be an ε-cover of $\mathcal{F}$ in the $\|\cdot\|_2$ norm, for a tolerance ε to be chosen. As argued in Yang and Barron (1999), we have

$$I_{X_1^n}(\Theta; Y_1^n) \;=\; \frac{1}{M}\sum_{j=1}^{M} D(\mathbb{Q}^j \,\|\, \mathbb{P}_Y) \leq D(\mathbb{Q}^k \,\|\, \frac{1}{N}\sum_{k=1}^{N} \mathbb{P}^k),$$

where $\mathbb{P}^\ell$ denotes the conditional distribution of Y_1^n given g^ℓ and X_1^n. For each ℓ, let us choose $g^{\ell^*(k)}$ as follows: $\ell^*(k) \in \arg\min_{\ell=1,\ldots,N} \|g^\ell - f^k\|_2$. We then have the upper bound

$$I_{X_1^n}(\Theta; Y_1^n) \leq \frac{1}{M}\sum_{k=1}^{M} \big\{\log N + \frac{n}{2}\|g^{\ell^*(k)} - f^k\|_n^2\big\}.$$

Taking expectations over X_1^n, we obtain

$$\begin{aligned}\mathbb{E}_{X_1^n}[I_{X_1^n}(\Theta; Y_1^n)] &\leq \frac{1}{M}\sum_{k=1}^{M} \big\{\log N + \frac{n}{2}\mathbb{E}_{X_1^n}[\|g^{\ell^*(k)} - f^k\|_n^2]\big\} \\ &\leq \log N + \frac{n}{2}\,\varepsilon^2,\end{aligned}$$

where the final inequality follows from the choice of our covering set.

From this point, we can follow the same steps as Yang and Barron (1999). The polynomial scaling (13) of the metric entropy guarantees that their conditions are satisfied, and we conclude that the minimax error is lower bounded by any $\delta_n > 0$ such that $n\delta_n^2 \;\geq\; C \log N(\delta_n; \mathcal{F}, \|\cdot\|_2)$. From Lemma 4 and the assumed scaling (13), it is equivalent to solve the equation

$$n\delta_n^2 \;\geq\; C\bigg\{s\log(d/s) + s(\sqrt{s}/\delta_n)^{1/\alpha}\bigg\},$$

from which some algebra yields $\delta_n^2 = C\big\{\frac{s\log(d/s)}{n} + s\big(\frac{1}{n}\big)^{\frac{2\alpha}{2\alpha+1}}\big\}$ as a suitable choice.

4.3 Proof of Theorem 3

Recall the definition of $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$ and $\mathcal{H}(S,B)$ from Section 3.5; note that it guarantees that $\|f^*\|_\infty \leq B$. In order to establish upper bounds on the minimax rate in $L^2(\mathbb{P})$-error over $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$, we analyze a least-squares estimator—albeit *not* the same as the original M-estimator (6)—constrained to $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$, namely

$$\widehat{f} \in \arg \min_{f \in \mathcal{F}^*_{d,s,\mathcal{H}}(B)} \sum_{i=1}^n (y_i - \bar{y}_n - f(x_i))^2. \tag{30}$$

Since our goal is to upper bound the minimax rate in $L^2(\mathbb{P})$ error, it is sufficient to upper bound the $L^2(\mathbb{P})$-norm of $\widehat{f} - f^*$ where $\widehat{f}$ is any solution to (30). The proof shares many steps with the proof of Theorem 1. First, the same reasoning shows that the error $\widehat{\Delta} := \widehat{f} - f^*$ satisfies the basic inequality

$$\frac{1}{n}\sum_{i=1}^n \widehat{\Delta}^2(x_i) \leq \frac{2}{n}\Big|\sum_{i=1}^n w_i \widehat{\Delta}(x_i)\Big| + |\bar{y}_n - \bar{f}|\Big|\frac{1}{n}\sum_{i=1}^n \widehat{\Delta}(x_i)\Big|.$$

Recall the definition (14) of the critical rate δ_n. Once again, we first control the term error due to estimating the mean $|\bar{y}_n - \bar{f}| = |\frac{1}{n}\sum_{i=1}^n (y_i - \bar{f})|$. Since $|f^*(x_i)|$ is at most B and w_i is standard Gaussian and independent, the random variable $y_i - \bar{f} = f^*(x_i) + w_i$ is sub-Gaussian with parameter $\sqrt{B^2+1}$. The samples are all i.i.d., so that by standard sub-Gaussian tail bounds, we have

$$\mathbb{P}[|\bar{y}_n - \bar{f}| > t] \leq 2\exp(-\frac{nt^2}{2(B^2+1)}).$$

Setting $\mathcal{A}(\delta_n) = \{|\bar{y}_n - \bar{f}| \leq B\delta_n\}$, it is clear that

$$\mathbb{P}[\mathcal{A}(\delta_n)] \geq 1 - 2\exp(-\frac{n\delta_n^2}{4}).$$

For the remainder of the proof, we condition on the event $\mathcal{A}(\delta_n)$, in which case Equation (17) simplifies to

$$\frac{1}{2}\|\widehat{\Delta}\|_n^2 \leq \Big|\frac{1}{n}\sum_{i=1}^n w_i \widehat{\Delta}(x_i)\Big| + B\delta_n \|\widehat{\Delta}\|_n. \tag{31}$$

Here we have used the fact that $\left|\frac{1}{n}\sum_{i=1}^n \widehat{\Delta}(x_i)\right| \leq \|\widehat{\Delta}\|_n$, by the Cauchy-Schwartz inequality.

Now we control the Gaussian complexity term $\left|\frac{1}{n}\sum_{i=1}^n w_i \widehat{\Delta}(x_i)\right|$. For any fixed subset S, define the random variable

$$\widehat{Z}_n(w,t;\mathcal{H}(S,2B)) := \sup_{\substack{\Delta \in \mathcal{H}(S,2B) \\ \|\Delta\|_n \leq t}} \Big|\frac{1}{n}\sum_{i=1}^n w_i \Delta(x_i)\Big|. \tag{32}$$

We first bound this random variable for a fixed subset S of size $2s$, and then take the union bound over all $\binom{d}{2s}$ possible subsets.

Lemma 5 *Assume that the RKHS $\mathcal{H}$ has eigenvalues $(\mu_k)_{k=1}^{\infty}$ that satisfy $\mu_k \simeq k^{-2\alpha}$ and eigenfunctions such that $\|\phi_k\|_\infty \leq C$. Then we have*

$$\mathbb{P}\big[\exists t > 0 \text{ such that } \widehat{Z}_n(w,t;\mathcal{H}(S,2B)) \geq 16BC\sqrt{\frac{s^{1/\alpha}\log s}{n}} + 3t\delta_n\big] \leq c_1\exp(-9n\delta_n^2).$$

The proof of Lemma 5 is provided Appendix F.1. Returning to inequality (31), we note that by definition,

$$\frac{2}{n}|\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)| \leq \max_{|S|=2s}\widehat{Z}_n(w,\|\widehat{\Delta}\|_n;\mathcal{H}(S,2B)).$$

Lemma 5 combined with the union bound implies that

$$\max_{|S|=2s}\widehat{Z}_n(w,\|\widehat{\Delta}\|_n;\mathcal{H}(S,2B)) \leq 16BC\sqrt{\frac{s^{1/\alpha}\log s}{n}} + 3\delta_n\|\widehat{\Delta}\|_n$$

with probability at least $1 - c_1\binom{d}{2s}\exp(-3n\delta_n^2)$. Our choice (14) of δ_n ensures that this probability is at least $1 - c_1\exp(-c_2 n\delta_n^2)$. Combined with the basic inequality (31), we conclude that

$$\|\widehat{\Delta}\|_n^2 \leq 32BC\sqrt{\frac{s^{1/\alpha}\log s}{n}} + 7B\delta_n\|\widehat{\Delta}\|_n \tag{33}$$

with probability $1 - c_1\exp(-c_2 n\delta_n^2)$.

By definition (14) of δ_n, the bound (33) implies that $\|\widehat{\Delta}\|_n = O(\delta_n)$ with high probability. In order to translate this claim into a bound on $\|\widehat{\Delta}\|_2$, we require the following result:

Lemma 6 *There exist universal constants (c, c_1, c_2) such that for all $t \geq c\delta_n$, we have*

$$\frac{\|g\|_2}{2} \leq \|g\|_n \leq \frac{3}{2}\|g\|_2 \qquad \text{for all } g \in \mathcal{H}(S,2B) \text{ with } \|g\|_2 \geq t \tag{34}$$

with probability at least $1 - c_1\exp(-c_2 n t^2)$.

Proof The bound (34) follows by applying Lemma 7 in Appendix A with $\mathcal{G} = \mathcal{H}(S,2B)$ and $b = 2B$. The critical radius from equation (35) needs to satisfy the relation $Q_{w,n}(\varepsilon_n;\mathcal{H}(S,2B)) \leq \frac{\varepsilon_n^2}{40}$. From Lemma 11, the choice $\varepsilon_n^2 = 320BC\sqrt{\frac{s^{1/\alpha}\log s}{n}}$ satisfies this relation. By definition (14) of δ_n, we have $\delta_n \geq c\varepsilon_n$ for some universal constant c, which completes the proof. ■

This lemma implies that with probability at least $1 - c_1\exp(-c_2 Bn\delta_n^2)$, we have $\|\widehat{\Delta}\|_2 \leq 2\|\widehat{\Delta}\|_n + C\delta_n$. Combined with our earlier upper bound on $\|\widehat{\Delta}\|_n$, this completes the proof of Theorem 3.

5. Discussion

In this paper, we have studied estimation in the class of sparse additive models in which each univariate function lies within a reproducing kernel Hilbert space. In conjunction, Theorems 1 and 2 provide a precise characterization of the minimax-optimal rates for estimating f^* in the $L^2(\mathbb{P})$-norm for various kernel classes with bounded univariate functions. These classes include finite-rank kernels (with logarithmic metric entropy), as well as kernels with polynomially decaying eigenvalues (and hence polynomial metric entropy). In order to establish achievable rates, we analyzed a simple M-estimator based on regularizing the least-squares loss with two kinds of ℓ_1-based norms, one defined by the univariate Hilbert norm and the other by the univariate empirical norm. On the other hand, we obtained our lower bounds by a combination of approximation-theoretic and information-theoretic techniques.

An important feature of our analysis is we assume only that each univariate function is bounded, but do not assume that the multivariate function class is bounded. As discussed in Section 3.5, imposing a global boundedness condition in the high-dimensional setting can lead to a substantially smaller function classes; for instance, for Sobolev classes and sparsity $s = \Omega(\sqrt{n})$, Theorem 3 shows that it is possible to obtain much faster rates than the optimal rates for the class of sparse additive models with univariate functions bounded. Theorem 3 in our paper shows that the rates obtained under global boundedness conditions are not minimax optimal for Sobolev spaces in the regime $s = \Omega(\sqrt{n})$.

There are a number of ways in which this work could be extended. Our work considered only a hard sparsity model, in which at most s co-ordinate functions were non-zero, whereas it could be realistic to use a "soft" sparsity model involving ℓ_q-norms. Some recent work by Suzuki and Sugiyama (2012) has studied some extensions of this type. In addition, the analysis here was based on assuming independence of the covariates x_j, $j = 1, 2, \ldots d$; it would be interesting to investigate the case when the random variables are endowed with some correlation structure. One might expect some changes in the optimal rates, particularly if many of the variables are strongly dependent. Finally, this work considered only the function class consisting of sums of co-ordinate functions, whereas a natural extension would be to consider nested non-parametric classes formed of sums over hierarchies of subsets of variables.

Acknowledgments

This work was partially supported by NSF grants DMS-0605165 and DMS-0907632 to MJW and BY. In addition, BY was partially supported by the NSF grant SES-0835531 (CDI), NSF grant 0939370 (CCF), and the ARO grant (W911NF-11-1-0114). MJW was also partially supported AFOSR Grant FA9550-09-1-0466. During this work, GR was financially supported by a Berkeley Graduate Fellowship. The authors would like to thank Arash Amini and Sara van de Geer for helpful discussions.

Appendix A. A General Result On Equivalence Of $L^2(\mathbb{P})$ And $L^2(\mathbb{P}_n)$ Norms

Since it is required in a number of our proofs, we begin by stating and proving a general result that provides uniform control on the difference between the empirical $\|\cdot\|_n$ and population $\|\cdot\|_2$ norms over a uniformly bounded function class $\mathcal{G}$. We impose two conditions on this class:

(a) it is uniformly bounded, meaning that there is some $b \geq 1$ such that $\|g\|_\infty \leq b$ for all $g \in \mathcal{G}$.

(b) it is star-shaped, meaning that if $g \in \mathcal{G}$, then $\lambda g \in \mathcal{G}$ for all $\lambda \in [0, 1]$.

For each co-ordinate, the Hilbert ball $\mathbb{B}_{\mathcal{H}}(2)$ satisfies both of these conditions; we use $\mathcal{G} = \mathbb{B}_{\mathcal{H}}(2)$. (To be clear, we cannot apply this result to the multivariate function class $\mathcal{F}_{d,s,\mathcal{H}}$, since it is not uniformly bounded.)

Let $\{\sigma_i\}_{i=1}^n$ be an i.i.d. sequence of Rademacher variables, and let $\{x_i\}_{i=1}^n$ be an i.i.d. sequence of variables from $\mathcal{X}$, drawn according to some distribution $\mathbb{Q}$. For each $t > 0$, we define the local Rademacher complexity

$$\mathcal{Q}_{\sigma,n}(t, \mathcal{G}) \;:= \mathbb{E}_{x,\sigma}\big[\sup_{\substack{\|g\|_2 \le t \\ g \in \mathcal{G}}} \frac{1}{n}\sum_{i=1}^n \sigma_i g(x_i)\big]$$

We let ε_n denote the smallest solution (of size at least $1/\sqrt{n}$) to the inequality

$$\mathcal{Q}_{\sigma,n}(\varepsilon_n, \mathcal{G}) = \frac{\varepsilon_n^2}{40}, \tag{35}$$

where our scaling by the constant 40 is for later theoretical convenience. Such an ε_n exists, because the star-shaped property implies that the function $\mathcal{Q}_{\sigma,n}(t, \mathcal{G})/t$ is non-increasing in t. This quantity corresponds to the critical rate associated with the population Rademacher complexity. For any $t \ge \varepsilon_n$, we define the event $\mathcal{E}(t) := \big\{ \sup_{\substack{g \in \mathcal{G} \\ \|g\|_2 \le t}} \big|\|g\|_n - \|g\|_2\big| \ge \frac{bt}{2}\big\}$.

Lemma 7 *Suppose that $\|g\|_\infty \le b$ for all $g \in \mathcal{G}$. Then there exist universal constants (c_1, c_2) such that for any $t \ge \varepsilon_n$,*

$$\mathbb{P}\big[\mathcal{E}(t)\big] \le c_1 \exp(-c_2 n t^2).$$

In addition, for any $g \in \mathcal{G}$ with $\|g\|_2 \ge t$, we have $\|g\|_n \;\le\; \|g\|_2(1 + \frac{b}{2})$, and moreover, for all $g \in \mathcal{G}$ with $\|g\|_2 \ge bt$, we have

$$\frac{1}{2}\|g\|_2 \;\le\; \|g\|_n \;\le\; \frac{3}{2}\|g\|_2, \tag{36}$$

both with probability at least $1 - c_1 \exp(-c_2 n t^2)$.

Lemma 7 follows from a relatively straightforward adaptation of known results (e.g., Lemma 5.16 in van de Geer, 2000 and Theorem 2.1 in Bartlett et al., 2005), so we omit the proof details here.

Appendix B. Proof of Lemma 1

The proof of this lemma is based on peeling and weighting techniques from empirical process theory (Alexander, 1987; van de Geer, 2000) combined with results on the local Rademacher and Gaussian complexities of kernel classes (Bartlett et al., 2005; Mendelson, 2002). For each univariate Hilbert space $\mathcal{H}_j = \mathcal{H}$, let us introduce the random variables

$$\widehat{Z}_n(w,t;\mathcal{H}) := \sup_{\substack{\|g_j\|_{\mathcal{H}} \le 1 \\ \|g_j\|_n \le t}} \big|\frac{1}{n}\sum_{i=1}^n w_i g_j(x_{ij})\big|, \quad \text{and} \quad Z_n(w,t;\mathcal{H}) := \mathbb{E}_x\Big[\sup_{\substack{\|g_j\|_{\mathcal{H}} \le 1 \\ \|g_j\|_2 \le t}} \big|\frac{1}{n}\sum_{i=1}^n w_i g_j(x_{ij})\big|\Big], \tag{37}$$

where $w_i \sim N(0,1)$ are i.i.d. standard normal. The empirical and population Gaussian complexities are given by

$$\widehat{\mathcal{Q}}_{w,n}(t,\mathcal{H}) \;:= \mathbb{E}_w\big[\widehat{Z}_n(w;t,\mathcal{H})\big] \quad \text{and} \quad \mathcal{Q}_{w,n}(t,\mathcal{H}) \;:= \mathbb{E}_w\big[Z_n(w;t,\mathcal{H})\big].$$

For future reference, we note that in the case of a univariate Hilbert space $\mathcal{H}$ with eigenvalues $\{\mu_k\}_{k=1}^{\infty}$, results in Mendelson (2002) imply that there are universal constants $c_\ell \leq c_u$ such that for all $t^2 \geq 1/n$, we have

$$\frac{c_\ell}{\sqrt{n}}\big[\sum_{k=1}^{\infty}\min\{t^2,\mu_k\}\big]^{1/2} \;\leq\; Q_{w,n}(t,\mathcal{H}) \;\leq\; \frac{c_u}{\sqrt{n}}\big[\sum_{k=1}^{\infty}\min\{t^2,\mu_k\}\big]^{1/2}, \tag{38}$$

for all j. The same bounds hold for the local Rademacher complexity in our special case of reproducing kernel Hilbert spaces.

Let $\widehat{\nu}_{n,j} > 0$ denote the smallest positive solution r of the inequality

$$\widehat{Q}_{w,n}(r,\mathcal{H}) \leq 4r^2. \tag{39}$$

The function $\widehat{Q}_{w,n}(r,\mathcal{H})$ defines the local Gaussian complexity of the kernel class in co-ordinate j. Recall the bounds (38) that apply to both the empirical and population Gaussian complexities. Recall that the critical univariate rate ν_n is defined in terms of the population Gaussian complexity (see Equation (8)).

B.1 Some Auxiliary Results

In order to prove Lemma 1, we also need some auxiliary results, stated below as Lemmas 8 and 9.

Lemma 8 *For any function class $\mathcal{G}$ and all $\delta \geq 0$, we have*

$$\mathbb{P}\big[|\widehat{Z}_n(w,t,\mathcal{G}) - \widehat{Q}_{w,n}(t,\mathcal{G})| \geq \delta t\big] \leq 2\exp\big(-\frac{n\delta^2}{2}\big), \quad \textit{and} \tag{40}$$

$$\mathbb{P}\big[|Z_n(w,t,\mathcal{G}) - Q_{w,n}(t,\mathcal{G})| \geq \delta t\big] \leq 2\exp\Big(-\frac{n\delta^2}{2}\Big). \tag{41}$$

Proof We have

$$|\widehat{Z}_n(w,t,\mathcal{G}) - \widehat{Z}_n(w',t,\mathcal{G})| \;\leq\; \sup_{\substack{g\in\mathcal{G}\\ \|g\|_n\leq t}} \frac{1}{n}\big|\sum_{i=1}^{n}(w_i-w_i')g(x_i)\big| \;\leq\; \frac{t}{\sqrt{n}}\|w-w'\|_2,$$

showing that $\widehat{Z}_n(w,t,\mathcal{G})$ is $\frac{t}{\sqrt{n}}$-Lipschitz with respect to the ℓ_2 norm. Consequently, concentration for Lipschitz functions of Gaussian random variables (see Ledoux, 2001) yields the tail bound (40). Turning to the quantity $Z_n(w,t,\mathcal{H})$, a similar argument yields that

$$\begin{aligned}|Z_n(w,t,\mathcal{G}) - Z_n(w',t,\mathcal{G})| &\leq \mathbb{E}_x\big[\sup_{\substack{g\in\mathcal{G}\\ \|g\|_2\leq t}} \frac{1}{n}\big|\sum_{i=1}^{n}(w_i-w_i')g(x_i)\big|\big] \\ &\leq \sup_{\substack{g\in\mathcal{G}\\ \|g\|_2\leq t}} \mathbb{E}_x\big[\big(\frac{1}{n}\sum_{i=1}^{n}g^2(x_i)\big)^{1/2}\big]\;\|w-w'\|_2 \;\leq\; \frac{t}{\sqrt{n}}\|w-w'\|_2,\end{aligned}$$

where the final step uses Jensen's inequality and the fact that $\mathbb{E}_x[g^2(x_i)] \leq t^2$ for all $i = 1,\ldots,n$. The same reasoning then yields the tail bound (41). ■

Our second lemma involves the event $\mathcal{D}(\gamma_n) := \big\{\widehat{\nu}_{n,j} \leq \gamma_n, \quad \text{for all } j = 1, 2, \ldots, d\big\}$, where we recall the definition (39) of $\widehat{\nu}_{n,j}$, and that $\gamma_n := \kappa \max\big\{\nu_n, \sqrt{\frac{\log d}{n}}\big\}$.

Lemma 9 *For all* $1 \leq j \leq d$*, we have*

$$\mathbb{P}\big[\widehat{\nu}_{n,j} \leq \gamma_n\big] \geq 1 - c_1 \exp(-c_2 n \gamma_n^2).$$

Proof

We first bound the probability of the event $\{\widehat{\nu}_{n,j} > \gamma_n\}$ for a fixed $\mathcal{H}_j$. Let $g \in \mathbb{B}_{\mathcal{H}_j}(1)$ be any function such that $\|g\|_2 > t \geq \nu_n$. Then conditioned on the sandwich relation (36) with $b = 1$, we are guaranteed that $\|g\|_n > \frac{t}{2}$. Taking the contrapositive, we conclude that $\|g\|_n \leq \frac{t}{2}$ implies $\|g\|_2 \leq t$, and hence that $\widehat{Z}_n(w, t/2, \mathcal{H}) \leq Z_n(w, t, \mathcal{H})$ for all $t \geq \nu_n$, under the stated conditioning.

For any $t \geq \nu_n$, the inequalities (36), (40) and (41) hold with probability at least $1 - c_1 \exp(-c_2 n t^2)$. Conditioning on these inequalities, we can set $t = \gamma_n > \nu_n$, and thereby obtain

$$\begin{aligned}
\widehat{Q}_{w,n}(\gamma_n, \mathcal{H}) &\overset{(a)}{\leq} \widehat{Z}_n(w, \gamma_n, \mathcal{H}) + \gamma_n^2 \\
&\overset{(b)}{\leq} Z_n(w, 2\gamma_n, \mathcal{H}) + \gamma_n^2 \\
&\overset{(c)}{\leq} Q_{w,n}(2\gamma_n, \mathcal{H}) + 2\gamma_n^2 \\
&\overset{(d)}{\leq} 4\gamma_n^2,
\end{aligned}$$

where inequality (a) follows from the bound (40), inequality (b) follows the initial argument, inequality (c) follows from the bound (41), and inequality (d) follows since $2\gamma_n > \varepsilon_n$ and the definition of ε_n.

By the definition of $\widehat{\nu}_{n,j}$ as the minimal t such that $\widehat{Q}_{w,n}(t, \mathcal{H}) \leq 4t^2$, we conclude that for each fixed $j = 1, \ldots, n$, we have $\widehat{\nu}_{n,j} \leq \gamma_n$ with probability at least $1 - c_1 \exp(-c_2 n \gamma_n^2)$. Finally, the uniformity over $j = 1, 2, \ldots, d$ follows from the union bound and our choice of $\gamma_n \geq \kappa\sqrt{\frac{\log d}{n}}$.

■

B.2 Main Argument To Prove Lemma 1

We can now proceed with the proof of Lemma 1. Combining Lemma 9 with the union bound over $j = 1, 2, \ldots, d$, we conclude that that

$$\mathbb{P}[\mathcal{D}(\gamma_n)] \geq 1 - c_1 \exp(-c_2 n \gamma_n^2),$$

as long as $c_2 \geq 1$. For the remainder of our proofs, we condition on the event $\mathcal{D}(\gamma_n)$. In particular, our goal is to prove that

$$\Big|\frac{1}{n}\sum_{i=1}^{n} w_i f_j(x_{ij})\Big| \leq C\,\big\{\gamma_n^2\,\|f_j\|_{\mathcal{H}} + \gamma_n\,\|f_j\|_n\big\} \qquad \text{for all } f_j \in \mathcal{H} \tag{42}$$

with probability greater than $1 - c_1 \exp(-c_2 n \gamma_n^2)$. By combining this result with our choice of γ_n and the union bound, the claimed bound then follows on $\mathbb{P}[\mathcal{T}(\gamma_n)]$.

If $f_j = 0$, then the claim (42) is trivial. Otherwise we renormalize f_j by defining $g_j := f_j / \|f_j\|_{\mathcal{H}}$, and we write

$$\frac{1}{n}\sum_{i=1}^{n} w_i f_j(x_{ij}) = \|f_j\|_{\mathcal{H}}\, \frac{1}{n}\sum_{i=1}^{n} w_i g_j(x_{ij}) \;\leq\; \|f_j\|_{\mathcal{H}}\, \widehat{Z}_n\big(w; \|g_j\|_n, \mathcal{H}\big),$$

where the final inequality uses the definition (37), and the fact that $\|g_j\|_{\mathcal{H}} = 1$. We now split the analysis into two cases: (1) $\|g_j\|_n \leq \gamma_n$, and (2) $\|g_j\|_n > \gamma_n$.

Case 1: $\|g_j\|_n \leq \gamma_n$. In this case, it suffices to upper bound the quantity $\widehat{Z}_n(w; \gamma_n, \mathcal{H})$. Note that $\|g_j\|_{\mathcal{H}} = 1$ and recall definition (37) of the random variable $\widehat{Z}_n$. On one hand, since $\gamma_n \geq \widehat{\nu}_{n,j}$ by Lemma 9, the definition of $\widehat{\nu}_{n,j}$ implies that $\widehat{Q}_{w,n}(\gamma_n, \mathcal{H}) \; \leq 4\gamma_n^2$, and hence

$$\mathbb{E}[\widehat{Z}_n(w; \gamma_n; \mathcal{H})] = \widehat{Q}_{w,n}(\gamma_n; \mathcal{H}) \leq 4\gamma_n^2.$$

Applying the bound (40) from Lemma 8 with $\delta = \gamma_n = t$, we conclude that $\widehat{Z}_n(w; \gamma_n; \mathcal{H}) \leq C\, \gamma_n^2$ with probability at least $1 - c_1 \exp\big\{-c_2 n \gamma_n^2\big\}$, which completes the proof in the case where $\|g\|_n \leq \gamma_n$.

Case 2: $\|g_j\|_n > \gamma_n$. In this case, we study the random variable $\widehat{Z}_n(w; r_j; \mathcal{H})$ for some $r_j > \gamma_n$. Our intermediate goal is to prove the bound

$$\mathbb{P}\Big[\widehat{Z}_n(w; r_j; \mathcal{H}) \geq C\, r_j \gamma_n\Big] \leq c_1 \exp\big\{-c_2 n \gamma_n^2\big\}. \tag{43}$$

Applying the bound (40) with $t = r_j$ and $\delta = \gamma_n$, we are guaranteed an upper bound of the form $\widehat{Z}_n(w; r_j; \mathcal{H}) \leq \widehat{Q}_{w,n}(r_j, \mathcal{H}) + r_j \gamma_n$ with probability at least $1 - c_1 \exp\big(-c_2 n \gamma_n^2\big)$. In order to complete the proof, we need to show that $\widehat{Q}_{w,n}(r_j, \mathcal{H}) \leq r_j \gamma_n$. Since $r_j > \gamma_n > \widehat{\nu}_{n,j}$, we have

$$\widehat{Q}_{w,n}(r_j, \mathcal{H}) = \frac{r_j}{\widehat{\nu}_{n,j}} \mathbb{E}_w\Big[\sup_{\substack{\|g_j\|_n \leq \widehat{\nu}_{n,j} \\ \|g_j\|_{\mathcal{H}} \leq \frac{\widehat{\nu}_{n,j}}{r_j}}} \Big|\frac{1}{n}\sum_{i=1}^{n} w_i g_j(x_{ij})\Big|\Big] \;\leq\; \frac{r_j}{\widehat{\nu}_{n,j}} \widehat{Q}_{w,n}(\widehat{\nu}_{n,j}, \mathcal{H}) \;\leq 4 r_j \widehat{\nu}_{n,j},$$

where the final inequality uses the fact that $\widehat{Q}_{w,n}(\widehat{\nu}_{n,j}, \mathcal{H}) \leq 4\widehat{\nu}_{n,j}^2$. On the event $\mathcal{D}(\gamma_n)$ from Lemma 9, we have $\widehat{\nu}_{n,j} \leq \gamma_n$, from which the claim (43) follows.

We now use the bound (43) to prove the bound (42), in particular via a "peeling" operation over all choices of $r_j = \|f_j\|_n / \|f_j\|_{\mathcal{H}}$. (See van de Geer, 2000 for more details on these peeling arguments.) We claim that it suffices to consider $r_j \leq 1$. It is equivalent to show that $\|g_j\|_n \leq 1$ for any $g_j \in \mathbb{B}_{\mathcal{H}}(1)$. Since $\|g_j\|_\infty \leq \|g_j\|_{\mathcal{H}} \leq 1$, we have $\|g_j\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} g_j^2(x_{ij}) \; \leq 1$, as required. Now define the event

$$\mathcal{T}_j(\gamma_n) := \left\{ \exists f_j \in \mathbb{B}_{\mathcal{H}}(1) \mid \Big|\frac{1}{n}\sum_{i=1}^{n} w_i f_j(x_{ij})\Big| > 8\, \|f_j\|_{\mathcal{H}}\, \gamma_n \, \frac{\|f_j\|_n}{\|f_j\|_{\mathcal{H}}}, \text{ and } \frac{\|f_j\|_n}{\|f_j\|_{\mathcal{H}}} \in (\gamma_n, 1] \right\}.$$

and the sets $S_m := \big\{2^{m-1}\gamma_n \leq \frac{\|f_j\|_n}{\|f_j\|_{\mathcal{H}}} \leq 2^m \gamma_n\big\}$ for $m = 1, 2, \ldots, M$. By choosing $M = 2\log_2(1/\gamma_n)$, we ensure that $2^M \gamma_n \geq 1$, and hence that if the event $\mathcal{T}_j(\gamma_n)$ occurs, then it must occur for function

f_j belonging to some S_m, so that we have a function f_j such that $\frac{\|f_j\|_n}{\|f_j\|_{\mathcal{H}}} \leq t_m := 2^m \gamma_n$, and

$$\Big|\frac{1}{n}\sum_{i=1}^{n} w_i f_j(x_{ij})\Big| > 8\, \|f_j\|_{\mathcal{H}} \gamma_n \, \frac{\|f_j\|_n}{\|f_j\|_{\mathcal{H}}} \, \geq C\, \|f_j\|_{\mathcal{H}} t_m,$$

which implies that $\widehat{Z}_n(w; t_m, \mathcal{H}) \geq 4t_m$. Consequently, by union bound and the tail bound (43), we have

$$\mathbb{P}[\mathcal{T}_j(\gamma_n)] \leq M\, c_1 \exp\big\{-c_2 n\gamma_n^2\big\} \;\leq\; c_1 \exp\big\{-c_2' n\gamma_n^2\big\}$$

by the condition $n\gamma_n^2 = \Omega(\log(1/\gamma_n))$, which completes the proof.

Appendix C. Proof of Lemma 2

Define the function

$$\widetilde{\mathcal{L}}(\Delta) := \frac{1}{2n}\sum_{i=1}^{n}\big(w_i + \bar{f} + \bar{y}_n - \Delta(x_i)\big)^2 + \lambda_n \|f^* + \Delta\|_{n,1} + \rho_n \|f^* + \Delta\|_{\mathcal{H},1}$$

and note that by definition of our M-estimator, the error function $\widehat{\Delta} := \widehat{f} - f^*$ minimizes $\widetilde{\mathcal{L}}$. From the inequality $\widetilde{\mathcal{L}}(\widehat{\Delta}) \leq \widetilde{\mathcal{L}}(0)$, we obtain the upper bound $\frac{1}{2}\|\widehat{\Delta}\|_n^2 \leq T_1 + T_2$, where

$$T_1 := \Big|\frac{1}{n}\sum_{i=1}^{n} w_i \widehat{\Delta}(x_i)\Big| + |\bar{y}_n - \bar{f}|\Big|\frac{1}{n}\sum_{i=1}^{n}\widehat{\Delta}(x_i)\Big|, \quad \text{and}$$

$$T_2 := \lambda_n \sum_{j=1}^{d}\big\{\|f_j^*\|_n - \|f_j^* + \widehat{\Delta}_j\|_n\big\} + \rho_n \sum_{j=1}^{d}\big\{\|f_j^*\|_{\mathcal{H}} - \|f_j^* + \widehat{\Delta}_j\|_{\mathcal{H}}\big\}.$$

Conditioned on the event $\mathcal{C}(\gamma_n)$, we have the bound $|\bar{y}_n - \bar{f}|\big|\frac{1}{n}\sum_{i=1}^{n}\widehat{\Delta}(x_i)\big| \leq \sqrt{s}\gamma_n\|\widehat{\Delta}\|_n$, and hence $\frac{1}{2}\|\widehat{\Delta}\|_n^2 \leq T_2 + \big|\frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)\big| + \sqrt{s}\gamma_n\|\widehat{\Delta}\|_n$, or equivalently

$$0 \;\leq\; \frac{1}{2}\big(\|\widehat{\Delta}\|_n - \sqrt{s}\gamma_n\big)^2 \leq T_2 + \Big|\frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)\Big| + \frac{1}{2}s\gamma_n^2. \tag{44}$$

It remains to control the term T_2. On one hand, for any $j \in S^c$, we have

$$\|f_j^*\|_n - \|f_j^* + \widehat{\Delta}_j\|_n \;=\; -\|\widehat{\Delta}_j\|_n, \quad \text{and} \quad \|f_j^*\|_{\mathcal{H}} - \|f_j^* + \widehat{\Delta}_j\|_{\mathcal{H}} \;=\; -\|\widehat{\Delta}_j\|_{\mathcal{H}}.$$

On the other hand, for any $j \in S$, the triangle inequality yields $\|f_j^*\|_n - \|f_j^* + \widehat{\Delta}_j\|_n \leq \|\widehat{\Delta}_j\|_n$, with a similar inequality for the terms involving $\|\cdot\|_{\mathcal{H}}$. Combined with the bound (44), we conclude that

$$0 \leq \frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i) + \lambda_n\big\{\|\widehat{\Delta}_S\|_{n,1} - \|\widehat{\Delta}_{S^c}\|_{n,1}\big\} + \rho_n\big\{\|\widehat{\Delta}_S\|_{\mathcal{H},1} - \|\widehat{\Delta}_{S^c}\|_{\mathcal{H},1}\big\} + \frac{1}{2}s\gamma_n^2. \tag{45}$$

Recalling our conditioning on the event $\mathcal{T}(\gamma_n)$, by Lemma 1, we have the upper bound

$$\Big|\frac{1}{n}\sum_{i=1}^{n} w_i\widehat{\Delta}(x_i)\Big| \leq 8\,\big\{\gamma_n\|\widehat{\Delta}\|_{n,1} + \gamma_n^2\|\widehat{\Delta}\|_{\mathcal{H},1}\big\}.$$

Combining with the inequality (45) yields

$$0 \le 8\,\{\gamma_n\|\widehat{\Delta}\|_{n,1} + \gamma_n^2\|\widehat{\Delta}\|_{\mathcal{H},1}\} + \lambda_n\{\|\widehat{\Delta}_S\|_{n,1} - \|\widehat{\Delta}_{S^c}\|_{n,1}\} + \rho_n\{\|\widehat{\Delta}_S\|_{\mathcal{H},1} - \|\widehat{\Delta}_{S^c}\|_{\mathcal{H},1}\} + \frac{1}{2}s\gamma_n^2$$
$$\le \frac{\lambda_n}{2}\|\widehat{\Delta}\|_{n,1} + \frac{\rho_n}{2}\|\widehat{\Delta}\|_{\mathcal{H},1} + \lambda_n\{\|\widehat{\Delta}_S\|_{n,1} - \|\widehat{\Delta}_{S^c}\|_{n,1}\} + \rho_n\{\|\widehat{\Delta}_S\|_{\mathcal{H},1} - \|\widehat{\Delta}_{S^c}\|_{\mathcal{H},1}\} + \frac{1}{2}s\gamma_n^2,$$

where we have recalled our choices of (λ_n, ρ_n). Finally, re-arranging terms yields the claim (19).

Appendix D. Proof of Lemma 3

Recalling the definitions (26), (27) and (28) of the function class $\mathcal{G}(\lambda_n, \rho_n)$ and the critical radius $\tilde{\delta}_n$ from Equation (24), we define the function class $\mathcal{G}'(\lambda_n, \rho_n, \tilde{\delta}_n) := \{h \in \mathcal{G}(\lambda_n, \rho_n) \mid \|h\|_2 = \tilde{\delta}_n\}$, and the alternative event

$$\mathcal{B}'(\lambda_n, \rho_n) := \{\|h\|_n^2 \ge \tilde{\delta}_n^2/2 \quad \text{for all } h \in \mathcal{G}'(\lambda_n, \rho_n, \tilde{\delta}_n)\}.$$

We claim that it suffices to show that $\mathcal{B}'(\lambda_n, \rho_n)$ holds with probability at least $1 - c_1\exp(-c_2 n\gamma_n^2)$. Indeed, given an arbitrary non-zero function $g \in \mathcal{G}(\lambda_n, \rho_n)$, consider the rescaled function $h = \frac{\tilde{\delta}_n}{\|g\|_2}g$. Since $g \in \mathcal{G}(\lambda_n, \rho_n)$ and $\mathcal{G}(\lambda_n, \rho_n)$ is star-shaped, we have $h \in \mathcal{G}(\lambda_n, \rho_n)$, and also $\|h\|_2 = \tilde{\delta}_n$ by construction. Consequently, when the event $\mathcal{B}'(\lambda_n, \rho_n)$ holds, we have $\|h\|_n^2 \ge \tilde{\delta}_n^2/2$, or equivalently $\|g\|_n^2 \ge \|g\|_2^2/2$, showing that $\mathcal{B}(\lambda_n, \rho_n)$ holds. Accordingly, the remainder of the proof is devoted to showing that $\mathcal{B}'(\lambda_n, \rho_n)$ holds with probability greater than $1 - c_1\exp(-c_2 n\gamma_n^2)$. Alternatively, if we define the random variable $Z_n(\mathcal{G}') := \sup_{f \in \mathcal{G}'}\{\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n f^2(x_i)\}$, then it suffices to show that $Z_n(\mathcal{G}') \le \tilde{\delta}_n^2/2$ with high probability.

Recall from Section 4.2.4 the definition of a covering set; here we use the notion of a proper covering, which restricts the covering to use only members of the set $\mathcal{G}$. Letting $N_{\text{pr}}(\varepsilon; \mathcal{G}, \rho)$ denote the propert covering number, it can be shown that $N_{\text{pr}}(\varepsilon; \mathcal{G}, \rho) \le N(\varepsilon; \mathcal{G}, \rho) \le N_{\text{pr}}(\varepsilon/2; \mathcal{G}, \rho)$. Now let $g^1, \ldots, g^N$ be a minimal $\tilde{\delta}_n/8$-proper covering of $\mathcal{G}'$ in the $L^2(\mathbb{P}_n)$-norm, so that for all $f \in \mathcal{G}'$, there exists $g = g^k \in \mathcal{G}'$ such that $\|f - g\|_n \le \tilde{\delta}_n/8$. We can then write

$$\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n f^2(x_i) = \{\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n g^2(x_i)\} + \{\frac{1}{n}\sum_{i=1}^n (g^2(x_i) - f^2(x_i))\}.$$

By the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n (g^2(x_i) - f^2(x_i)) &= \frac{1}{n}\sum_{i=1}^n (g(x_i) - f(x_i))(g(x_i) + f(x_i)) \\ &\le \sqrt{\frac{1}{n}\sum_{i=1}^n (g(x_i) - f(x_i))^2}\sqrt{\frac{1}{n}\sum_{i=1}^n (f(x_i) + g(x_i))^2} \\ &= \|g - f\|_n\sqrt{\frac{1}{n}\sum_{i=1}^n (f(x_i) + g(x_i))^2}.\end{aligned}$$

By our choice of the covering, we have $\|g - f\|_n \le \tilde{\delta}_n/8$. On the other hand, we have

$$\sqrt{\frac{1}{n}\sum_{i=1}^n (f(x_i) + g(x_i))^2} \le \sqrt{2\|f\|_n^2 + 2\|g\|_n^2} \;\le\; \sqrt{4\tilde{\delta}_n^2} \;=\; 2\tilde{\delta}_n,$$

where the final inequality follows since $\|f\|_n = \|g\|_n = \tilde{\delta}_n$. Overall, we have established the upper bound $\frac{1}{n}\sum_{i=1}^n (g^2(x_i) - f^2(x_i)) \leq \frac{\tilde{\delta}_n^2}{4}$, and hence shown that

$$Z_n(\mathcal{G}') \leq \max_{g^1,g^2,\ldots,g^N} \{\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n (g^k(x_i))\} + \frac{\tilde{\delta}_n^2}{4},$$

where $N = N_{\text{pr}}(\tilde{\delta}_n/8, \mathcal{G}', \|\cdot\|_n)$. For any g in our covering set, since $g^2(x_i) \geq 0$, we may apply a one-sided tail bound (e.g., Theorem 3.5 from Chung and Lu, 2006, or Lemma 2.1 in Einmahl and Mason, 1996) with $t = \tilde{\delta}_n^2/4$ to obtain the one-sided tail bound

$$\mathbb{P}[\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n g^2(x_i) \geq \frac{\tilde{\delta}_n^2}{4}] \leq \exp\big(-\frac{n\tilde{\delta}_n^4}{32\mathbb{E}[g^4(x)]}\big), \tag{46}$$

where we used the upper bound $\text{var}(g^2(x)) \leq \mathbb{E}[g^4(x)]$. Next using the fact that the variables $\{g_j(x_j)\}_{j=1}^d$ are independent and zero-mean, we have

$$\begin{aligned}
\mathbb{E}[g^4(x)] &= \sum_{j=1}^d \mathbb{E}[g_j^4(x_j)] + \binom{4}{2}\sum_{j\neq k}\mathbb{E}[[g_j^2(x_j)]\mathbb{E}[g_k^2(x_k)] \\
&\leq 4\sum_{j=1}^d \mathbb{E}[g_j^2(x_j)] + 6\sum_{j=1}^d \mathbb{E}[g_j^2(x_j)]\sum_{k=1}^d \mathbb{E}[g_k^2(x_k)] \\
&\leq 4\tilde{\delta}_n^2 + 6\tilde{\delta}_n^4 \\
&\leq 10\tilde{\delta}_n^2,
\end{aligned}$$

where the second inequality follows since $\|g_j\|_\infty \leq \|g_j\|_{\mathcal{H}} \leq 2$ for each j. Combining this upper bound on $\mathbb{E}[g^4(x)]$ with the earlier tail bound (46) and applying union bound yields

$$\mathbb{P}[\max_{k=1,2,\ldots,N}\{\tilde{\delta}_n^2 - \frac{1}{n}\sum_{i=1}^n g^2(x_i)\} \geq \frac{\tilde{\delta}_n^2}{4}] \leq \exp\big(\log N_{\text{pr}}(\tilde{\delta}_n/8, \mathcal{G}', \|\cdot\|_n) - \frac{n\tilde{\delta}_n^2}{320}\big). \tag{47}$$

It remains to bound the covering entropy $\log N_{\text{pr}}(\tilde{\delta}_n/8, \mathcal{G}', \|\cdot\|_n)$. Since the proper covering entropy $\log N_{\text{pr}}(\tilde{\delta}_n/8, \mathcal{G}', \|\cdot\|_n)$ is at most $\log N(\tilde{\delta}_n/16, \mathcal{G}', \|\cdot\|_n)$, it suffices to upper bound the usual covering entropy. Viewing the samples $(x_1, x_2, \ldots, x_n)$ as fixed, let us define the zero-mean Gaussian process $\{W_g, g \in \mathcal{G}'\}$ via $W_g := \frac{1}{\sqrt{n}}\sum_{i=1}^n \varepsilon_i g(x_i)$, where the variables $\{\varepsilon_i\}_{i=1}^n$ are i.i.d. standard Gaussian variates. By construction, we have $\text{var}[(W_g - W_f))] = \|g - f\|_n^2$. Consequently, by the Sudakov minoration (see Pisier, 1989), for all $\varepsilon > 0$, we have $\varepsilon\sqrt{\log N(\varepsilon; \mathcal{G}', \|\cdot\|_n)} \leq 4\mathbb{E}_\varepsilon[\sup_{g\in\mathcal{G}'} W_g]$. Setting $\varepsilon = \tilde{\delta}_n/16$ and performing some algebra, we obtain the upper bound

$$\frac{1}{\sqrt{n}}\sqrt{\log N(\tilde{\delta}_n/16; \mathcal{G}', \|\cdot\|_n)} \leq \frac{64}{\tilde{\delta}_n}\mathbb{E}_\varepsilon[\sup_{g\in\mathcal{G}'}\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)]. \tag{48}$$

The final step is to upper bound the Gaussian complexity $\mathbb{E}_\varepsilon[\sup\limits_{g\in\mathcal{G}'}\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)]$. In the proof of Lemma 1, we showed that for any co-ordinate $j \in \{1, 2, \ldots, d\}$, the univariate Gaussian complexity is upper bounded as

$$\mathbb{E}\big[\sup_{\substack{\|g_j\|_n \leq r_j \\ \|g_j\|_{\mathcal{H}} \leq R_j}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g_j(x_{ij})\big] \leq C\{\gamma_n r_j + \gamma_n^2 R_j\}.$$

Summing across co-ordinates and recalling the fact that the constant C may change from line to line, we obtain the upper bound

$$\begin{aligned}\mathbb{E}_\varepsilon[\sup_{g\in\mathcal{G}'}\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)] &\leq C\sup_{g\in\mathcal{G}'}\big\{\gamma_n\,\|g\|_{1,n}+\gamma_n^2\|g\|_{1,\mathcal{H}}\big\}\\ &\overset{(a)}{\leq} C\sup_{g\in\mathcal{G}'}\big\{4\gamma_n\,\|g_S\|_{1,n}+4\gamma_n^2\|g_S\|_{1,\mathcal{H}}+\frac{1}{32}s\rho_n\big\}\\ &\overset{(b)}{\leq} C\sup_{g\in\mathcal{G}'}\big\{\gamma_n\,\|g_S\|_{1,n}+s\rho_n\big\}\\ &\overset{(c)}{\leq} C\sup_{g\in\mathcal{G}'}\Big\{\gamma_n\,[2\sqrt{s}\|g\|_2+s\gamma_n]+s\rho_n\Big\},\end{aligned}$$

where step (a) uses inequality (26) in the definition of $\mathcal{G}'$; step (b) uses the inequality $\|g_j\|_{\mathcal{H}}\leq 2$ for each co-ordinate and hence $\|g_S\|_{1,\mathcal{H}}\leq 2s$, and our choice of regularization parameter $\rho_n\geq\gamma_n^2$; and step (c) uses inequality (27) in the definition of $\mathcal{G}'$. Since $\|g\|_2=\tilde{\delta}_n$ for all $g\in\mathcal{G}'$, we have shown that

$$\mathbb{E}_\varepsilon[\sup_{g\in\mathcal{G}'}\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)]\leq C\big\{s\gamma_n^2+\sqrt{s}\gamma_n\tilde{\delta}_n+s\rho_n\big\}\ \overset{(d)}{\leq}\ C\big\{\frac{\tilde{\delta}_n^2}{B^2}+\frac{\tilde{\delta}_n^2}{B}\big\}, \tag{49}$$

where inequality (d) follows from our choice (24) of $\tilde{\delta}_n$, and the constant B can be chosen as large as we please. In particular, by choosing B sufficiently large, and combining the bound (49) with the Sudakov bound (48), we can ensure that

$$\frac{1}{n}\ \log N(\tilde{\delta}_n/16;\mathcal{G}',\|\cdot\|_n)\leq\frac{\tilde{\delta}_n^2}{640}.$$

Combined with the earlier tail bound (47), we conclude that

$$\mathbb{P}[\max_{k=1,2,\ldots,N}\big\{\tilde{\delta}_n^2-\frac{1}{n}\sum_{i=1}^n g^2(x_i)\big\}\geq\frac{\tilde{\delta}_n^2}{4}]\leq\exp\big(-\frac{n\tilde{\delta}_n^2}{640}\big),$$

which completes the proof of Lemma 3.

Appendix E. Proof of Lemma 4

In this section, we present the proofs of Lemma 4 (a) and (b).

E.1 Proof of Part (a)

Let $N=M(\frac{\delta}{\sqrt{s}};\mathbb{B}_{\mathcal{H}}(1),\|\cdot\|_2)-1$, and define $I=\{0,1,\ldots,N\}$. Consider the set

$$\mathfrak{S}:=\big\{u\in I^d\mid\|u\|_0:=\sum_{j=1}^d\mathbb{I}[u_j\neq 0]=s\big\}. \tag{50}$$

Note that this set has cardinality $|\mathfrak{S}| = \binom{d}{s} N^s$, since any element is defined by first choosing s co-ordinates are non-zero, and then for each co-ordinate, choosing non-zero entry from a total of N possible symbols.

For each $j = 1, \ldots, d$, let $\{0, f_j^1, f_j^2, \ldots, f_j^N\}$ be a $\delta/\sqrt{s}$-packing of $\mathbb{B}_{\mathcal{H}}(1)$. Based on these packings of the univariate function classes, we can use $\mathfrak{S}$ to index a collection of functions contained inside $\mathcal{F}$. In particular, any $u \in \mathfrak{S}$ uniquely defines a function $g^u = \sum_{j=1}^d g_j^{u_j} \in \mathcal{F}$, with elements

$$g_j^{u_j} = \begin{cases} f_j^{u_j} & \text{if } u_j \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Since $\|u\|_0 = s$, we are guaranteed that at most s co-ordinates of g are non-zero, so that $g \in \mathcal{F}$.

Now consider two functions g^u and h^v contained within the class $\{g^u, u \in \mathfrak{S}\}$. By definition, we have

$$\|g^u - h^v\|_2^2 = \sum_{j=1}^d \|f_j^{u_j} - f_j^{v_j}\|_2^2 \;\geq\; \frac{\delta^2}{s} \sum_{j=1}^d \mathbb{I}[u_j \neq v_j], \tag{51}$$

Consequently, it suffices to establish the existence of a "large" subset $\mathcal{A} \subset \mathfrak{S}$ such that the Hamming metric $\rho_H(u,v) := \sum_{j=1}^d \mathbb{I}[u_j \neq v_j]$ is at least $s/2$ for all pairs $u, v \in \mathcal{A}$, in which case we are guaranteed that $\|g - h\|_2^2 \geq \delta^2$. For any $u \in \mathfrak{S}$, we observe that

$$\left| \left\{ v \in \mathfrak{S} \mid \rho_H(u,v) \leq \frac{s}{2} \right\} \right| \leq \binom{d}{\frac{s}{2}} (N+1)^{\frac{s}{2}}.$$

This bound follows because we simply need to choose a subset of size $s/2$ where u and v agree, and the remaining $s/2$ co-ordinates can be chosen arbitrarily in $(N+1)^{\frac{s}{2}}$ ways. For a given set $\mathcal{A}$, we write $\rho_H(u, \mathcal{A}) \leq \frac{s}{2}$ if there exists some $v \in \mathcal{A}$ such that $\rho_H(u,v) \leq \frac{s}{2}$. Using this notation, we have

$$\left| \left\{ u \in \mathfrak{S} \mid \rho_H(u, \mathcal{A}) \leq \frac{s}{2} \right\} \right| \leq |\mathcal{A}| \binom{d}{\frac{s}{2}} (N+1)^{\frac{s}{2}} \overset{(a)}{<} |\mathfrak{S}|,$$

where inequality (a) follows as long as

$$|\mathcal{A}| \leq N^* \;:= \frac{1}{2} \frac{\binom{d}{s}}{\binom{d}{\frac{s}{2}}} \; \frac{N^s}{(N+1)^{s/2}}.$$

Thus, as long as $|\mathcal{A}| \leq N^*$, there must exist some element $u \in \mathfrak{S}$ such that $\rho_H(u, \mathcal{A}) > \frac{s}{2}$, in which case we can form the augmented set $\mathcal{A} \cup \{u\}$. Iterating this procedure, we can form a set with N^* elements such that $\rho_H(u,v) \geq \frac{s}{2}$ for all $u, v \in \mathcal{A}$.

Finally, we lower bound N^*. We have

$$\begin{aligned} N^* &\overset{(i)}{\geq} \frac{1}{2} \left(\frac{d-s}{s/2}\right)^{\frac{s}{2}} \frac{(N)^s}{(N+1)^{s/2}} \\ &= \frac{1}{2} \left(\frac{d-s}{s/2}\right)^{\frac{s}{2}} N^{s/2} \left(\frac{N}{N+1}\right)^{s/2} \\ &\geq \frac{1}{2} \left(\frac{d-s}{s/2}\right)^{\frac{s}{2}} N^{s/2}, \end{aligned}$$

where inequality (i) follows by elementary combinatorics (see Lemma 5 in Raskutti et al., 2011 for details). We conclude that for $s \le d/4$, we have

$$\log N^* = \Omega\big(s \log \frac{d}{s} + s \log M(\frac{\delta}{\sqrt{s}}; \mathbb{B}_{\mathcal{H}}(1), \|\cdot\|_2)\big),$$

thereby completing the proof of Lemma 4(a).

E.2 Proof of Part (b)

In order to prove part (b), we instead let $N = M(\frac{1}{2}; \mathbb{B}_{\mathcal{H}}(1), \|\cdot\|_2) - 1$, and then follow the same steps. Since $\log N = \Omega(m)$, we have the modified lower bound

$$\log N^* = \Omega\big(s \log \frac{d}{s} + sm\big),$$

Moreover, instead of the lower bound (51), we have

$$\|g^u - h^v\|_2^2 = \sum_{j=1}^{d} \|f_j^{u_j} - f_j^{v_j}\|_2^2 \;\ge\; \frac{1}{4} \sum_{j=1}^{d} \mathbb{I}[u_j \neq v_j] \;\ge\; \frac{s}{8},$$

using our previous result on the Hamming separation. Furthermore, since $\|f_j\|_2 \le \|f_j\|_{\mathcal{H}}$ for any univariate function, we have the upper bound

$$\|g^u - h^v\|_2^2 = \sum_{j=1}^{d} \|f_j^{u_j} - f_j^{v_j}\|_2^2 \;\le\; \sum_{j=1}^{d} \|f_j^{u_j} - f_j^{v_j}\|_{\mathcal{H}}^2.$$

By the definition (50) of $\mathfrak{S}$, at most $2s$ of the terms $f_j^{u_j} - f_j^{v_j}$ can be non-zero. Moreover, by construction we have $\|f_j^{u_j} - f_j^{v_j}\|_{\mathcal{H}} \le 2$, and hence

$$\|g^u - h^v\|_2^2 \le 8s.$$

Finally, by rescaling the functions by $\sqrt{8}\,\delta/\sqrt{s}$, we obtain a class of N^* rescaled functions $\{\widetilde{g}^u, u \in I\}$ such that

$$\|\widetilde{g}^u - \widetilde{h}^v\|_2^2 \ge \delta^2, \quad \text{and} \quad \|\widetilde{g}^u - \widetilde{h}^v\|_2^2 \le 64\delta^2,$$

as claimed.

Appendix F. Results For Proof Of Theorem 3

The reader should recall from Section 3.5 the definitions of the function classes $\mathcal{F}^*_{d,s,\mathcal{H}}(B)$ and $\mathcal{H}(S,B)$. The function class $\mathcal{H}(S,B)$ can be parameterized by the two-dimensional sequence $(a_{j,k})_{j\in S, k\in\mathbb{N}}$ of co-efficients, and expressed in terms of two-dimensional sequence of basis functions $(\phi_{j,k})_{j\in S, k\in\mathbb{N}}$ and the sequence of eigenvalues $(\mu_k)_{k\in\mathbb{N}}$ for the univariate RKHS $\mathcal{H}$ as follows:

$$\mathcal{H}(S,B) := \big\{f = \sum_{j\in S}\sum_{k=1}^{\infty} a_{j,k}\phi_{j,k} \mid \sum_{k=1}^{\infty} \frac{a_{j,k}^2}{\mu_k} \le 1 \;\forall\, j \in S \text{ and } \|f\|_\infty \le B\big\}.$$

For any integer $M \ge 1$, we also consider the truncated function class

$$\mathcal{H}(S,B,M) := \big\{f = \sum_{j\in S}\sum_{k=1}^{M} a_{j,k}\phi_{j,k} \mid \sum_{k=1}^{\infty} \frac{a_{j,k}^2}{\mu_k} \le 1 \;\forall\, j \in S \text{ and } \|f\|_\infty \le B\big\}.$$

Lemma 10 *We have the inclusion* $\mathcal{H}(S,B,M) \subseteq \big\{f \in \mathcal{H}(S) \mid \sum_{j\in S}\sum_{k=1}^{M}|a_{j,k}| \le B\sqrt{M}\big\}$.

Proof Without loss of generality, let us assume that $S=\{1,2,...,s\}$, and consider a function $f = \sum_{j=1}^{s} f_j \in \mathcal{H}(S,B,M)$. Since each f_j acts on a different co-ordinate, we are guaranteed that $\|f\|_\infty = \sum_{j=1}^{s}\|f_j\|_\infty$. Consider any univariate function $f_j = \sum_{k=1}^{M} a_{j,k}\phi_{j,k}$. We have

$$\sum_{k=1}^{M}|a_{j,k}| \le \sqrt{M}\Big(\sum_{k=1}^{M} a_{j,k}^2\Big)^{1/2} \overset{(a)}{\le} \sqrt{M}\,\big[\mathbb{E}[f_j^2(X_j)]\big]^{1/2} \le \sqrt{M}\|f_j\|_\infty,$$

where step (a) uses the fact that $\mathbb{E}[f_j^2(X_j)] = \sum_{k=1}^{\infty} a_{j,k}^2 \ge \sum_{k=1}^{M} a_{j,k}^2$ for any $M \ge 1$. Adding up the bounds over all co-ordinates, we obtain

$$\|a\|_1 \;=\; \sum_{j=1}^{s}\sum_{k=1}^{M}|a_{j,k}| \le \sqrt{M}\sum_{j=1}^{s}\|f_j\|_\infty \;=\; \sqrt{M}\|f\|_\infty \;\le\; \sqrt{M}B,$$

where the final step uses the uniform boundedness condition. ■

F.1 Proof of Lemma 5

Recalling the definition of $\widehat{Z}_n(w;t,\mathcal{H}(S,2B))$ stated from (32), let us view it as a function of the standard Gaussian random vector $(w_1,\ldots,w_n)$. It is straightforward to verify that this variable is Lipschitz (with respect to the Euclidean norm) with parameter at most $t/\sqrt{n}$. Consequently, by concentration for Lipschitz functions (see Ledoux, 2001), we have

$$\mathbb{P}\big[\widehat{Z}_n(w;t,\mathcal{H}(S,2B)) \ge \mathbb{E}[\widehat{Z}_n(w;t,\mathcal{H}(S,2B))] + 3t\delta_n\big] \le \exp\big(-\frac{9n\delta_n^2}{2}\big).$$

Next we prove an upper bound on the expectations

$$\widehat{Q}_{w,n}(t;\mathcal{H}(S,2B)) := \mathbb{E}_w\big[\sup_{\substack{g\in\mathcal{H}(S,2B)\\ \|g\|_n\le t}} \frac{1}{n}\sum_{i=1}^{n} w_i g(x_i)\big], \quad \text{and}$$

$$Q_{w,n}(t;\mathcal{H}(S,2B)) := \mathbb{E}_{x,w}\big[\sup_{\substack{g\in\mathcal{H}(S,2B)\\ \|g\|_2\le t}} \frac{1}{n}\sum_{i=1}^{n} w_i g(x_i)\big].$$

Lemma 11 *Under the conditions of Theorem 3, we have*

$$\max\big\{\widehat{Q}_{w,n}(t;\mathcal{H}(S,2B)),\; Q_{w,n}(t;\mathcal{H}(S,2B))\big\} \;\le\; 8BC\sqrt{\frac{s^{1/\alpha}\log s}{n}}.$$

Proof By definition, any function $g \in \mathcal{H}(S,2B)$ has support at most $2s$, and without loss of generality (re-indexing as necessary), we assume that $S = \{1,2,...,2s\}$. We can thus view functions in $\mathcal{H}(S,2B)$ as having domain $\mathbb{R}^{2s}$, and we can an operator Φ that maps from $\mathbb{R}^{2s}$ to $[\ell^2(\mathbb{N})]^{2s}$, via

$$x \mapsto \Phi_{j,k}(x) = \phi_{j,k}(x_j), \qquad \text{for } j=1,\ldots,2s, \text{ and } k \in \mathbb{N}.$$

Any function in $g \in \mathcal{H}(S,2B)$ can be expressed in terms of two-dimensional sequence $(a_{j,k})$ and the functions $(\Phi_{j,k})$ as $g(x) = g(x_1,x_2,\ldots,x_{2s}) = \sum_{j=1}^{2s}\sum_{k=1}^{\infty}\Phi_{j,k}(x)a_{j,k} \;=\; \langle\!\langle \Phi(x),\, a\rangle\!\rangle$, where $\langle\!\langle \cdot,\, \cdot\rangle\!\rangle$ is a convenient shorthand for the inner product between the two arrays.

For any function $g \in \mathcal{H}(S,2B)$, triangle inequality yields the upper bound

$$\sup_{g\in 2\mathcal{H}(S,2B)} \frac{1}{n}\Big|\sum_{i=1}^{n} w_i \langle\!\langle \Phi(x_i),\, a\rangle\!\rangle\Big| \le \underbrace{\sup_{g\in 2\mathcal{H}(S,2B)} \frac{1}{n}\Big|\sum_{i=1}^{n} w_i \langle\!\langle \Phi_{\cdot,1:M}(x_i),\, a_{\cdot,1:M}\rangle\!\rangle}_{A_1} + A_2 \tag{52}$$

where $A_2 := \sup_{g\in 2\mathcal{H}(S,2B)} \frac{1}{n}\big|\sum_{i=1}^{n} w_i \langle\!\langle \Phi_{\cdot,M+1:\infty}(x_i),\, a_{\cdot,M+1:\infty}\rangle\!\rangle\big|$.

F.1.1 BOUNDING THE QUANTITIES $\mathbb{E}_{x,w}[A_1]$ AND $\mathbb{E}_{w}[A_1]$

By Hölder's inequality and Lemma 10, we have

$$A_1 \le \frac{1}{\sqrt{n}} \sup_{g\in 2\mathcal{H}(S,2B)} \|a_{\cdot,1:M}\|_{1,1} \max_{j,k}\Big|\sum_{i=1}^{n}\frac{w_i}{\sqrt{n}}\Phi_{j,k}(x_i)\Big| \;\le\; \frac{2\,B\,\sqrt{M}}{\sqrt{n}} \max_{j,k}\Big|\sum_{i=1}^{n}\frac{w_i}{\sqrt{n}}\Phi_{j,k}(x_i)\Big|.$$

By assumption, we have $|\Phi_{j,k}(x_i)| \le C$ for all indices (i,j,k), implying that $\sum_{i=1}^{n}\frac{w_i}{\sqrt{n}}\Phi_{j,k}(x_i)$ is zero-mean with sub-Gaussian parameter bounded by C and we are taking the maximum of $2s \times M$ such terms. Consequently, we conclude that

$$\mathbb{E}_w[A_1] \le 4BC\sqrt{\frac{M\log(2Ms)}{n}}. \tag{53}$$

Note that the same bound holds for $\mathbb{E}_{x,w}[A_1]$.

F.1.2 BOUNDING THE QUANTITIES $\mathbb{E}_{x,w}[A_2]$ AND $\mathbb{E}_{w}[A_2]$

In order to control this term, we simply recognize that it corresponds to the usual Gaussian complexity of the sum of $2s$ univariate Hilbert spaces, each of which is an RKHS truncated to the eigenfunctions $\{\mu_k\}_{k\ge M+1}$. In particular, we have

$$\begin{aligned}
\frac{1}{n}\Big|\sum_{i=1}^{n} w_i \langle\!\langle \Phi_{\cdot,M+1:\infty}(x_i),\, a_{\cdot,M+1:\infty}\rangle\!\rangle\Big| &\le \frac{1}{\sqrt{n}}\sum_{j=1}^{2s}\Big|\sum_{k\ge M+1} a_{j,k}\underbrace{\sum_{i=1}^{n}\Phi_{j,k}(x_i)\frac{w_i}{\sqrt{n}}\Big|}_{b_{j,k}} \\
&= \frac{1}{\sqrt{n}}\sum_{j=1}^{2s}\Big|\sum_{k\ge M+1}\frac{a_{j,k}}{\sqrt{\mu_k}}\,\sqrt{\mu_k}b_{j,k}\Big| \\
&\overset{(i)}{\le} \frac{1}{\sqrt{n}}\sum_{j=1}^{2s}\sqrt{\sum_{k\ge M+1}\frac{a_{j,k}^2}{\mu_k}}\,\sqrt{\sum_{k\ge M+1}\mu_k b_{j,k}^2} \\
&\overset{(ii)}{\le} \frac{1}{\sqrt{n}}\sum_{j=1}^{2s}\,\sqrt{\sum_{k\ge M+1}\mu_k b_{j,k}^2},
\end{aligned}$$

where step (i) follows by applying the Cauchy-Schwarz inequality, and step (ii) exploits the fact that $\sum_{k\ge M+1}\frac{a_{j,k}^2}{\mu_k} \le 1$ for all j.

This bound no longer depends on the coefficients a (or equivalently, the function g), so that we have shown that

$$\mathbb{E}_w[A_2] \leq \frac{1}{\sqrt{n}} \sum_{j=1}^{2s} \mathbb{E}_w\Big[\sqrt{\sum_{k \geq M+1} \mu_k b_{j,k}^2}\Big] \;\leq\; \frac{1}{\sqrt{n}} \sum_{j=1}^{2s} \sqrt{\sum_{k \geq M+1} \mu_k \mathbb{E}_w[b_{j,k}^2]},$$

where the second step uses Jensen's inequality to move the expectation inside the square root. Recalling that $b_{j,k}^2 = \big(\sum_{i=1}^n \Phi_{j,k}(x_i)\frac{w_i}{\sqrt{n}}\big)^2$ and using the independence of the noise variable $\{w_i\}_{i=1}^n$, we have

$$\mathbb{E}_w[b_{j,k}^2] = \frac{1}{n}\sum_{i=1}^n \Phi_{j,k}^2(x_i)\mathbb{E}_w[w_i^2] \;\leq\; C^2.$$

Putting together the pieces, we conclude that

$$\mathbb{E}_w[A_2] \leq \frac{C}{\sqrt{n}} \sum_{j=1}^{2s} \sqrt{\sum_{k \geq M+1} \mu_k} \;=\; \frac{2Cs}{\sqrt{n}} \sqrt{\sum_{k \geq M+1} \mu_k}. \tag{54}$$

Once again, a similar bound holds for $\mathbb{E}_{x,w}[A_2]$.

Substituting the bounds (53) and (54) into the inequality (52), we conclude that

$$\begin{aligned} Q_{w,n}(2\mathcal{H}(S,2B)) &\leq 4BC\sqrt{\frac{M\log(2Ms)}{n}} + 2Cs\sqrt{\frac{\sum_{k\geq M+1}\mu_k}{n}} \\ &\leq 4BC\sqrt{\frac{M\log(2Ms)}{n}} + 2Cs\sqrt{\frac{M^{1-2\alpha}}{n}}, \end{aligned}$$

where the second inequality follows from the relation $\mu_k \simeq k^{-2\alpha}$. Finally, setting $M = s^{\frac{1}{\alpha}}$ yields the claim. Note that the same argument works for the Rademacher complexity, since we only exploited the sub-Gaussianity of the variables w_i. This completes the proof of Lemma 11. ■

Returning to the proof of Lemma 5, combining Lemma 11 with the bound (40) in Lemma 8:

$$\mathbb{P}\Big[\widehat{Z}_n(w;t,\mathcal{H}(S,2B)) \geq 8BC\sqrt{\frac{s^{1/\alpha}\log s}{n}} + 3t\delta_n\Big] \leq \exp\big(-\frac{9n\delta_n^2}{2}\big).$$

Since $\|g\|_n \leq 2B$ for any function $g \in \mathcal{H}(S,2B)$, the proof Lemma 5 is completed using a peeling argument over the radius, analogous to the proof of Lemma 1 (see Appendix B).

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.

N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

F. Bach. Consistency of the Group Lasso and Multiple Kernel Learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

P. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher Complexities. *Annals of Statistics*, 33:1497–1537, 2005.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous Analysis of Lasso and Dantzig Selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

M. S. Birman and M. Z. Solomjak. Piecewise-polynomial Approximations of Functions of the Classes W_p^α. *Math. USSR-Sbornik*, 2(3):295–317, 1967.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

L. Breiman. Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, 37:373–384, 1995.

V. V. Buldygin and Y. V. Kozachenko. *Metric Characterization of Random Variables and Random Processes*. American Mathematical Society, Providence, RI, 2000.

B. Carl and I. Stephani. *Entropy, Compactness and the Approximation of Operators*. Cambridge Tracts in Mathematics. Cambridge University Press, Cambridge, UK, 1990.

B. Carl and H. Triebel. Inequalities Between Eigenvalues, Entropy Numbers and Related Quantities of Compact Operators in Banach Spaces. *Annals of Mathematics*, 251:129–133, 1980.

F. Chung and L. Lu. Concentration Inequalities and Martingale Inequalities. *Internet Mathematics*, 3:79–127, 2006.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

U. Einmahl and D. M. Mason. Some Universal Results on the Behavior of the Increments of Partial Sums. *Annals of Probability*, 24:1388–1407, 1996.

C. Gu. *Smoothing Spline ANOVA Models*. Springer Series in Statistics. Springer, New York, NY, 2002.

R. Z. Has'minskii. A Lower Bound on the Risks of Nonparametric Estimates of Densities in the Uniform Metric. *Theory Prob. Appl.*, 23:794–798, 1978.

T. Hastie and R. Tibshirani. Generalized Additive Models. *Statistical Science*, 1(3):297–310, 1986.

G. Kimeldorf and G. Wahba. Some Results on Tchebycheffian Spline Functions. *Jour. Math. Anal. Appl.*, 33:82–95, 1971.

V. Koltchinskii and M. Yuan. Sparse Recovery in Large Ensembles of Kernel Machines. In *Proceedings of COLT*, 2008.

V. Koltchinskii and M. Yuan. Sparsity in Multiple Kernel Learning. *Annals of Statistics*, 38:3660–3695, 2010.

M. Ledoux. *The Concentration of Measure Phenomenon*. Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI, 2001.

Y. Lin and H. H. Zhang. Component Selection and Smoothing in Multivariate Nonparametric Regression. *Annals of Statistics*, 34:2272–2297, 2006.

P. Massart. About the Constants in Talagrand's Concentration Inequalities for Empirical Processes. *Annals of Probability*, 28(2):863–884, 2000.

L. Meier, S. van de Geer, and P. Buhlmann. High-dimensional Additive Modeling. *Annals of Statistics*, 37:3779–3821, 2009.

S. Mendelson. Geometric Parameters of Kernel Machines. In *Proceedings of COLT*, pages 29–43, 2002.

J. Mercer. Functions of Positive and Negative Type and Their Connection With the Theory of Integral Equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.

S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A Unified Framework for High-dimensional Analysis of M-estimators with Decomposable Regularizers. In *NIPS Conference*, 2009.

G. Pisier. *The Volume of Convex Bodies and Banach Space Geometry*, volume 94 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, UK, 1989.

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax Rates of Estimation for High-dimensional Linear Regression Over ℓ_q-balls. *IEEE Trans. Information Theory*, 57(10):6976—6994, October 2011.

P. Ravikumar, H. Liu, J. Lafferty, and L. Wasserman. SpAM: Sparse Additive Models. *Journal of the Royal Statistical Society, Series B*, 71(5):1009–1030, 2009.

S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, UK, 1988.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

C. J. Stone. Additive Regression and Other Nonparametric Models. *Annals of Statistics*, 13(2): 689–705, 1985.

T. Suzuki and M. Sugiyama. Fast Learning Rate of Multiple Kernel Learning: Trade-off Between Sparsity and Smoothness. In *AISTATS Conference*, 2012.

S. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000.

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

G. Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, PN, 1990.

Y. Yang and A. Barron. Information-theoretic Determination of Minimax Rates of Convergence. *Annals of Statistics*, 27(5):1564–1599, 1999.

B. Yu. Assouad, Fano and Le Cam. *Research Papers in Probability and Statistics: Festschrift in Honor of Lucien Le Cam*, pages 423–435, 1996.

M. Yuan. Nonnegative Garrote Component Selection in Functional ANOVA Models. In *Conference on Artificial Intelligence and Statistics*, pages 660–666, 2007.

Journal of Machine Learning Research 13 (2012) 429-458 Submitted 6/11; Revised 11/11; Published 2/12

Online Learning in the Embedded Manifold of Low-rank Matrices

Uri Shalit[*] URI.SHALIT@MAIL.HUJI.AC.IL
Computer Science Department and ICNC/ELSC
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Daphna Weinshall DAPHNA@CS.HUJI.AC.IL
Computer Science Department
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Gal Chechik[†] GAL@GOOGLE.COM
The Gonda Brain Research Center
Bar Ilan University
52900 Ramat-Gan, Israel

Editor: Léon Bottou

Abstract

When learning models that are represented in matrix forms, enforcing a low-rank constraint can dramatically improve the memory and run time complexity, while providing a natural regularization of the model. However, naive approaches to minimizing functions over the set of low-rank matrices are either prohibitively time consuming (repeated singular value decomposition of the matrix) or numerically unstable (optimizing a factored representation of the low-rank matrix). We build on recent advances in optimization over manifolds, and describe an iterative online learning procedure, consisting of a gradient step, followed by a *second-order retraction* back to the manifold. While the ideal retraction is costly to compute, and so is the projection operator that approximates it, we describe another retraction that can be computed efficiently. It has run time and memory complexity of $O((n+m)k)$ for a rank-k matrix of dimension $m \times n$, when using an online procedure with rank-one gradients. We use this algorithm, LORETA, to learn a matrix-form similarity measure over pairs of documents represented as high dimensional vectors. LORETA improves the mean average precision over a passive-aggressive approach in a factorized model, and also improves over a full model trained on pre-selected features using the same memory requirements. We further adapt LORETA to learn positive semi-definite low-rank matrices, providing an online algorithm for *low-rank metric learning*. LORETA also shows consistent improvement over standard weakly supervised methods in a large (1600 classes and 1 million images, using *ImageNet*) multi-label image classification task.

Keywords: low rank, Riemannian manifolds, metric learning, retractions, multitask learning, online learning

1. Introduction

Many learning problems involve models represented in matrix form. These include metric learning, collaborative filtering, and multi-task learning where all tasks operate over the same set of features.

[*] Also at The Gonda Brain Research Center, Bar Ilan University, 52900 Ramat-Gan, Israel.
[†] Also at Google Research, 1600 Amphitheatre Parkway, Mountain View CA, 94043.

In many of these tasks, a natural way to regularize the model is to limit the rank of the corresponding matrix. In metric learning, a low-rank constraint allows to learn a low dimensional representation of the data in a discriminative way. In multi-task problems, low-rank constraints provide a way to tie together different tasks. In all cases, low-rank matrices can be represented in a factorized form that dramatically reduces the memory and run-time complexity of learning and inference with that model. Low-rank matrix models could therefore scale to handle substantially many more features and classes than models with full rank dense matrices.

Unfortunately, the rank constraint is non-convex, and in the general case, minimizing a convex function subject to a rank constraint is NP-hard (Natarajan, 1995).[1] As a result of these issues, two main approaches have been commonly used to address the problem of learning under a low-rank constraint. Sometimes, a matrix $W \in \mathbb{R}^{n\times m}$ of rank k is represented as a product of two low dimension matrices $W = AB^T, A \in \mathbb{R}^{n\times k}, B \in \mathbb{R}^{m\times k}$ and simple gradient descent techniques are applied to each of the product terms separately (Bai et al., 2009). Second, projected gradient algorithms can be applied by repeatedly taking a gradient step and projecting back to the manifold of low-rank matrices. Unfortunately, computing the projection to that manifold becomes prohibitively costly for large matrices and cannot be computed after every gradient step.

Work in the field has focused mostly on two realms. First, learning low-rank positive semi-definite (PSD) models (as opposed to general low-rank models), as in the works of Kulis et al. (2009) and Meyer et al. (2011). Second, approximating a noisy matrix of observations by a low-rank matrix, as in the work of Negahban and Wainwright (2010). This task is commonly addressed in the field of recommender systems. Importantly, the current paper does not address the problem of low-rank *approximation to a given data matrix*, but rather addresses the problem of learning a *low-rank parametric model* in the context of ranking and classification.

In this paper we propose new algorithms for online learning on the manifold of low-rank matrices. It is based on an operation called *retraction*, which is an operator that maps from a vector space that is tangent to the manifold, into the manifold (Do Carmo, 1992; Absil et al., 2008). Retractions include the projection operator as a special case, but also include other operators that can be computed substantially more efficiently. We use second order retractions to develop LORETA —an online algorithm for learning low-rank matrices. LORETA has a memory and run time complexity of $O\left((n+m)k\right)$ per update when the gradients have rank one. We show below that the case of rank-one gradients is relevant to numerous online learning problems.

We test LORETA in two different domains and learning tasks. First, we learn a bilinear similarity measure among pairs of text documents, where the number of features (text terms) representing each document could become very large. LORETA performed better than other techniques that operate on a factorized model, and also improves retrieval precision by 33% as compared with training a full rank model over pre-selected most informative features, using comparable memory footprint. Second, we applied LORETA to image multi-label ranking, a problem in which the number of classes could grow to millions. LORETA significantly improved over full rank models, using a fraction of the memory required. These two experiments suggest that low-rank optimization could become very useful for learning in high-dimensional problems.

[1] Some special cases are solvable (notably, PCA), relying mainly on singular value decomposition (Fazel et al., 2005) and semi-definite programming techniques. For SDP of rank $k \geq 2$ it is not known whether this problem is NP-hard. For $k = 1$ it is equivalent to the MAX-CUT problem (Briët et al., 2010). Both SDP and SVD scale poorly to large scale tasks.

This paper is organized as follows. We start with an introduction to optimization on manifolds, describing the notion of retractions. We then derive our low-rank online learning algorithm in three variants: one which learns a general low-rank matrix, one which learns a low-rank PSD matrix, and one which concentrates most of the learning in a reduced dimensional space. Finally we test our algorithms in two applications: learning similarity of text documents, and multi-label ranking on a set of one million images.

This paper extends a shorter version published in Advances in Neural Information Systems (Shalit et al., 2010), by adding a new PSD version of the algorithm, much larger-scale and wider experiments, giving a full mathematical discussion and proofs, and adding thorough complexity analysis.

2. Optimization on Riemannian Manifolds

The field of numerical optimization on smooth manifolds has advanced significantly in the past few years. For a recent exposition on this subject see Absil et al. (2008). We start with a short introduction to embedded manifolds, which are the focus of this paper.

An *embedded manifold* is a smooth subset of an ambient space $\mathbb{R}^n$. For instance, the set $\{\mathbf{x} : ||\mathbf{x}||_2 = 1, \mathbf{x} \in \mathbb{R}^n\}$, the unit sphere, is an $n-1$ dimensional manifold embedded in n-dimensional space $\mathbb{R}^n$. As another example, the *orthogonal group* O_n, which comprises of the set of orthogonal $n \times n$ matrices, is an $\frac{n(n-1)}{2}$ dimensional manifold embedded in $\mathbb{R}^{n\times n}$. Here we focus on the manifold of *low-rank* matrices, namely, the set of $n \times m$ matrices of rank k where $k < m,n$. It is an $(n+m)k - k^2$ dimensional manifold embedded in $\mathbb{R}^{n\times m}$, which we denote $\mathcal{M}_k^{n,m}$, or plainly $\mathcal{M}$. Embedded manifolds inherit many properties from the ambient space, a fact which simplifies their analysis. For example, the natural Riemannian metric for embedded manifolds is simply the Euclidean metric restricted to the manifold.

Motivated by online learning, we focus here on developing a stochastic gradient descent procedure to minimize a loss function $\mathcal{L}$ over the manifold of low-rank matrices $\mathcal{M}_k^{n,m}$,

$$\min_W \quad \mathcal{L}(W) \qquad \text{s.t.} \quad W \in \mathcal{M}_k^{n,m} \quad .$$

To illustrate the challenge in this problem, consider a simple stochastic gradient descent algorithm (Figure 1). At every step t of the algorithm, a gradient step update $W^t - \tilde{\nabla}\mathcal{L}(W^t)$ takes the model outside of the manifold $\mathcal{M}$ and has to be mapped back onto the manifold. The most common mapping operation is the *projection* operation, which, given a point $W^t - \tilde{\nabla}\mathcal{L}(W^t)$ outside the manifold, would find the closest point in $\mathcal{M}$. Unfortunately, the projection operation is very expensive to compute for the manifold of low-rank matrices, since it basically involves a singular value decomposition. Here we describe a wider class of operations called *retractions*, that serve a similar purpose: they find a point on the manifold that is in the direction of the gradient. To explain how retractions are computed, we first describe the notion of a *tangent space* and the *Riemannian gradient* of a function on a manifold.

2.1 Riemannian Gradient and the Tangent Space

Each point W in an embedded manifold $\mathcal{M}$ has a tangent space associated with it, denoted $T_{\mathbf{W}}\mathcal{M}$, as shown in Figure 2 (for a formal definition of the tangent space, see Appendix A). The tangent space is a vector space of the same dimension as the manifold that can be identified in a natural way

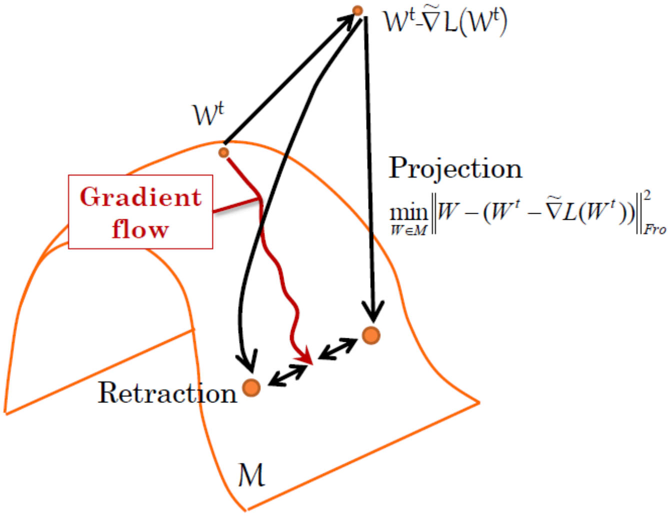

W^t-∇̃L(W^t)
W^t
Projection
$\min_{W\in M}\left\|W-(W^t-\tilde\nabla L(W^t))\right\|_{Fro}^2$
Gradient flow
Retraction
M

Figure 1: Projection onto the manifold is just a particular case of a retraction. Retractions are defined as operators that approximate the geodesic gradient flow on the manifold.

with a linear subspace of the ambient space. It is usually simple to compute the linear projection P_W of any point in the ambient space onto the tangent space $T_{\mathbf{W}}\mathcal{M}$.

Given a manifold $\mathcal{M}$ and a differentiable function $\mathcal{L}:\mathcal{M}\to\mathbb{R}$, the *Riemannian gradient* $\nabla\mathcal{L}(W)$ of $\mathcal{L}$ on $\mathcal{M}$ at a point $\mathbf{W}$ is a vector in the tangent space $T_{\mathbf{W}}\mathcal{M}$. A very useful property of embedded manifolds is the following: given a differentiable function f defined on the ambient space (and thus on the manifold), the Riemannian gradient of f at point W is simply the linear projection P_W of the Euclidean gradient of f onto the tangent space $T_W\mathcal{M}$.

Thus, if we denote the Euclidean gradient of $\mathcal{L}$ in $\mathbb{R}^{n\times m}$ by $\tilde\nabla\mathcal{L}$, we have $\nabla\mathcal{L}(W)=P_W(\tilde\nabla\mathcal{L})$. An important consequence follows in case the manifold represents the set of points obeying a certain constraint. In this case the Riemannian gradient of f is equivalent to the Euclidean gradient of f minus the component which is normal to the constraint. Indeed this normal component is exactly the component which is irrelevant when performing constrained optimization.

The Riemannian gradient allows us to compute $W^{t+\frac{1}{2}}=W^t-\eta^t\nabla\mathcal{L}(W)$, for a given iterate point W^t and step size η^t. We now examine how $W^{t+\frac{1}{2}}$ can be mapped back onto the manifold.

2.2 Retractions

Intuitively, *retractions* capture the notion of "going along a straight line" on the manifold. The mathematically ideal retraction is called the *exponential mapping* (Do Carmo, 1992, Chapter 3): it maps the tangent vector $\xi\in T_W\mathcal{M}$ to a point along a geodesic curve which goes through W in the direction of ξ Figure 1. Unfortunately, for many manifolds (including the low-rank manifold considered here) calculating the geodesic curve is computationally expensive (Vandereycken et al., 2009). A major insight from the field of Riemannian manifold optimization is that one can use retractions which merely approximate the exponential mapping. Using such retractions maintains the convergence properties obtained with the exponential mapping, but is much cheaper computationally for a suitable choice of mapping.

Definition 1 *Given a point W in an embedded manifold $\mathcal{M}$, a retraction is any function $R_W : T_{\mathbf{W}}\mathcal{M} \to \mathcal{M}$ which satisfies the following two conditions (Absil et al., 2008, Chapter 4):*

1. *Centering: $R_W(0) = W$.*

2. *Local rigidity: The curve $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}$ defined by $\gamma_\xi(\tau) = R_W(\tau\xi)$ satisfies $\dot{\gamma_\xi}(0) = \xi$, where $\dot{\gamma}$ is the derivative of γ by τ.*

It can be shown that any such retraction approximates the exponential mapping to a first order (Absil et al., 2008). *Second-order retractions*, which approximate the exponential mapping to second order around W, have to satisfy in addition the following stricter condition:

$$P_W \left(\frac{\mathrm{d}R_W(\tau\xi)}{\mathrm{d}\tau^2}|_{\tau=0} \right) = 0,$$

for all $\xi \in T_W\mathcal{M}$, where P_W is the *linear* projection from the ambient space onto the tangent space $T_{\mathbf{W}}\mathcal{M}$. When viewed intrinsically, the curve $R_W(\tau\xi)$ defined by a second-order retraction has zero acceleration at point W, namely, its second order derivatives are all normal to the manifold. The best known example of a second-order retraction onto embedded manifolds is the projection operation (Absil and Malick, 2010), which maps a point X to the point $Y \in \mathcal{M}$ which is closest to it in the Frobenius norm. That is, the projection of X onto $\mathcal{M}$ is simply:

$$Proj_{\mathcal{M}}(X) = \underset{Y \in \mathcal{M}}{\operatorname{argmin}} \|X - Y\|_{Fro}$$

Importantly, such projections are viewed here as one type of a second order approximation to the exponential mapping, which can be replaced by any other second-order retractions, when computing the projection is too costly (see Figure 1).

Given the tangent space and a retraction, we now define a Riemannian gradient descent procedure for the loss $\mathcal{L}$ at point $W^t \in \mathcal{M}$. Conceptually, the procedure has three steps (Figure 2):

1. **Step 1: Ambient gradient:** Obtain the Euclidean gradient $\tilde{\nabla}\mathcal{L}(W^t)$ in the ambient space.

2. **Step 2: Riemannian gradient:** Linearly project the ambient gradient onto the tangent space $T_{\mathbf{W}}\mathcal{M}$. Compute $\xi^t = P_{W^t}(\tilde{\nabla}\mathcal{L}(W^t))$.

3. **Step 3: Retraction:** Retract the Riemannian gradient ξ^t back to the manifold: $W^{t+1} = R_{W^t}(\xi^t)$.

With a proper choice of step size, this procedure can be proved to have local convergence for any retraction (Absil et al., 2008). In practice, the algorithm merges these three steps for efficiency, as discussed in the next section.

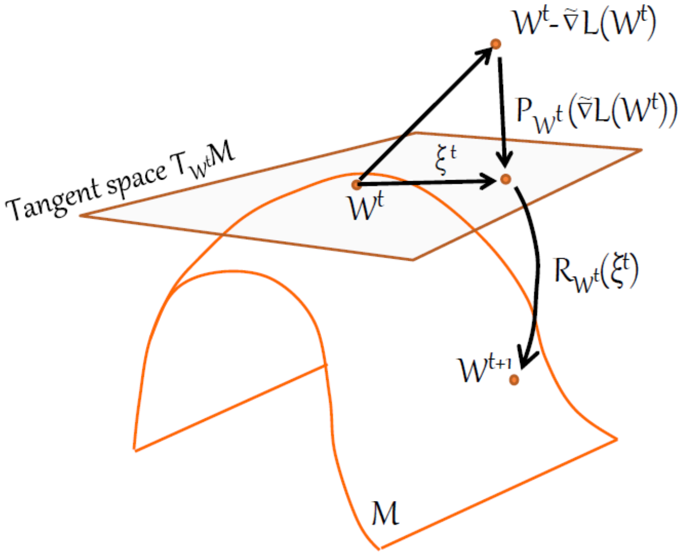


Figure 2: A three step procedure for computing a retracted gradient at point W^t. Step 1: standard (Euclidean) gradient step. Step 2: linearly project ambient gradient onto tangent space $T_{\mathbf{W}}\mathcal{M}$ in order to get the Riemannian gradient ξ^t. Step 3: retract the Riemannian gradient ξ^t back to the manifold.

3. Online Learning on the Low-rank Manifold

Based on the retractions described above, we now present an online algorithm for learning low-rank matrices, by performing stochastic gradient descent on the manifold of low-rank matrices. We name the algorithm LORETA (for a *LOw rank RETraction Algorithm*). At every iteration the algorithm suffers some loss, and performs a Riemannian gradient step followed by a retraction to the manifold $\mathcal{M}_k^{n,m}$. Section 3.1 discusses general online updates. Section 3.2 discusses the very common case where the online updates induce a gradient of rank $r = 1$.

Algorithm 1 : Online algorithm for learning in the manifold of low-rank matrices

Input: Initial low-rank model matrix $W^0 \in \mathcal{M}_k^{n,m}$. Examples $(x_0, x_1, \ldots)$. Loss function $\mathcal{L}$. Gradient descent step sizes $(\eta^0, \eta^1, \ldots)$.
Output: Final low-rank model matrix $W^{final} \in \mathcal{M}_k^{n,m}$.

```
repeat:
    Get example $x_t$
    Calculate the stochastic loss gradient: $\tilde{\nabla}\mathcal{L}(W^t; x_t)$
    Linearly project onto the tangent space: $\xi^t = P_{W^t}(\tilde{\nabla}\mathcal{L}(W^t; x_t))$
    Retract back to the manifold: $W^{t+1} = R_{W^t}(-\eta^t \xi^t)$
until stopping condition is satisfied
```

In what follows, lowercase Greek letters like ξ denote an abstract tangent vector, and uppercase Roman letters like A denote concrete matrix representations as kept in memory (taking $n \times m$ float numbers to store). We intermix the two notations, as in $\xi = AZ$, when the meaning is clear from the context. The set of $n \times k$ matrices of rank k is denoted $\mathbb{R}_*^{n\times k}$.

3.1 The General-Rank LORETA Algorithm

In online learning we are repeatedly given a rank-r gradient matrix $Z = \tilde{\nabla}\mathcal{L}W$, and want to compute a step on $\mathcal{M}_k^{n,m}$ in the direction of Z. As a first step we find its linear projection onto the tangent space $\hat{Z} = P_W(Z)$.

We start with a lemma that gives a representation of the tangent space $T_{\mathbf{W}}\mathcal{M}$ (Figure 2), extending the constructions given by Vandereycken and Vandewalle (2010) to the general manifold of low-rank matrices.

Lemma 2 *Let $W \in \mathcal{M}_k^{n,m}$ have a (non-unique) factorization $W = AB^T$, where $A \in \mathbb{R}_*^{n\times k}$, $B \in \mathbb{R}_*^{m\times k}$. Let $A_\perp \in \mathbb{R}^{n\times(n-k)}$ and $B_\perp \in \mathbb{R}^{m\times(m-k)}$ be the orthogonal complements of A and B respectively, such that $A_\perp^T A = 0$, $B_\perp^T B = 0$, $A_\perp^T A_\perp = I_{n-k}$, $B_\perp^T B_\perp = I_{m-k}$. The tangent space to $\mathcal{M}_k^{n,m}$ at W is:*

$$T_{\mathbf{W}}\mathcal{M} = \left\{ \begin{bmatrix} A & A_\perp \end{bmatrix} \begin{bmatrix} M & N_1^T \\ N_2 & 0 \end{bmatrix} \begin{bmatrix} B^T \\ B_\perp^T \end{bmatrix} : M \in \mathbb{R}^{k\times k}, N_1 \in \mathbb{R}^{(m-k)\times k}, N_2 \in \mathbb{R}^{(n-k)\times k} \right\}.$$

Proof The proof is given in Appendix A. ■

We note that the assumption that A and B are both of full column rank is tantamount to assuming that the model W is exactly of rank k, and no less. Let $\xi \in T_{\mathbf{W}}\mathcal{M}$ be a tangent vector to $W = AB^T$. From the characterization above it follows that ξ can be decomposed in a unique manner into three orthogonal components: $\xi = \xi^{AB} + \xi^{AB_\perp} + \xi^{A_\perp B}$, where:

$$\xi^{AB} = AMB^T, \quad \xi^{AB_\perp} = AN_1^T B_\perp^T, \quad \xi^{A_\perp B} = A_\perp N_2 B^T. \tag{1}$$

It is easy to verify that each pair is orthogonal, following from the relations $A_\perp^T A = 0$, $B_\perp^T B = 0$.

We wish to find the three matrices M, N_1 and N_2 associated with $\hat{Z} = P_W(Z)$, such that $\hat{Z} = AMB^T + AN_1^T B_\perp^T + A_\perp N_2 B^T$. We can find each of the matrices M, N_1 and N_2 separately, because each belongs to a space orthogonal to the other two. Thus we solve the following three problems:

$$\begin{aligned} &\underset{M\in\mathbb{R}^{k\times k}}{\arg\min} && \|Z - AMB^T\|_{Fro}^2, \\ &\underset{N_1\in\mathbb{R}^{(m-k)\times k}}{\arg\min} && \|Z - AN_1^T B_\perp^T\|_{Fro}^2, \\ &\underset{N_2\in\mathbb{R}^{(n-k)\times k}}{\arg\min} && \|Z - A_\perp N_2 B^T\|_{Fro}^2. \end{aligned}$$

To find the minimum of each of these three equations, we compute the derivatives and set them to zero. The solutions involve the pseudoinverses of A and B. Since A and B are of full column rank, their pseudoinverses are $A^\dagger = (A^TA)^{-1}A^T$, $B^\dagger = (B^TB)^{-1}B^T$.

$$
\begin{aligned}
M &= (A^TA)^{-1}A^TZB(B^TB)^{-1} = A^{\dagger}ZB^{\dagger T}, \qquad (2)\\
N_1 &= B_{\perp}^T Z^T A(A^TA)^{-1} = B_{\perp}^T Z^T A^{\dagger},\\
N_2 &= A_{\perp}^T ZB(B^TB)^{-1} = A_{\perp}^T ZB^{\dagger T}.
\end{aligned}
$$

The matrix $AA^{\dagger}$ is the matrix projecting onto the column space of A, and similarly for B. We will denote these matrices by P_A, P_B, etc. For the matrices projecting onto $A_{\perp}$ and $B_{\perp}$'s columns we actually have $P_{A_{\perp}} = A_{\perp}A_{\perp}^T$ because the columns of $A_{\perp}$ are orthogonal, and likewise for $P_{B_{\perp}}$. Substituting the expressions in *Equation* (2) into expressions of the components of the Riemannian gradient vector in *Equation* (1), we obtain:

$$\xi^{AB} = P_AZP_B, \quad \xi^{AB_{\perp}} = P_AZP_{B_{\perp}}, \quad \xi^{A_{\perp}B} = P_{A_{\perp}}ZP_B.$$

We can now define the retraction. The following theorem presents the retraction we will apply.

Theorem 3 *Let $W \in \mathcal{M}_k^{n,m}$, $W = AB^T$, and $W^{\dagger} = B^{\dagger T}A^{\dagger}$. Let $\xi \in T_W\mathcal{M}_k^{n,m}$, $\xi = \xi^{AB} + \xi^{AB_{\perp}} + \xi^{A_{\perp}B}$, as in Equation (1), and let:*

$$
\begin{aligned}
V_1 &= W + \frac{1}{2}\xi^{AB} + \xi^{A_{\perp}B} - \frac{1}{8}\xi^{AB}W^{\dagger}\xi^{AB} - \frac{1}{2}\xi^{A_{\perp}B}W^{\dagger}\xi^{AB} \quad ,\\
V_2 &= W + \frac{1}{2}\xi^{AB} + \xi^{AB_{\perp}} - \frac{1}{8}\xi^{AB}W^{\dagger}\xi^{AB} - \frac{1}{2}\xi^{AB}W^{\dagger}\xi^{AB_{\perp}} \quad .
\end{aligned}
$$

The mapping

$$R_W(\xi) = V_1W^{\dagger}V_2$$

is a second order retraction from a neighborhood $\Theta_W \subset T_W\mathcal{M}_k^{n,m}$ to $\mathcal{M}_k^{n,m}$.

Proof The proof is given in Appendix B. ■

A more succinct representation of this retraction is the following:

Lemma 4 *The retraction $R_W(\xi)$ can be presented as:*

$$
\begin{aligned}
R_W(\xi) = &\left[A\left(I_k + \frac{1}{2}M - \frac{1}{8}M^2\right) + A_{\perp}N_2\left(I_k - \frac{1}{2}M\right)\right]\cdot\\
&\left[B\left(I_k + \frac{1}{2}M^T - \frac{1}{8}\left(M^T\right)^2\right) + B_{\perp}N_1\left(I_k - \frac{1}{2}M^T\right)\right]^T.
\end{aligned}
$$

Proof The proof is given in Appendix C. ■

As a result from Lemma 4, we can calculate the retraction as the product of two low-rank factors: the first is an $n \times k$ matrix, the second a $k \times m$ matrix. Given a gradient $\tilde{\nabla}\mathcal{L}(x)$ in the ambient space, we can calculate the matrices M, N_1 and N_2 which allow us to represent its projection onto the tangent space, and furthermore allow us to calculate the retraction. We now have all the ingredients

```
Algorithm 2 : Naive Riemannian stochastic gradient descent

Input: Matrices $A \in \mathbb{R}_*^{n\times k}$, $B \in \mathbb{R}_*^{m\times k}$ s.t. $W = AB^T$. Gradient matrix $G \in \mathbb{R}^{n\times m}$ s.t.
$G = -\eta\tilde{\nabla}\mathcal{L}(W) \in \mathbb{R}^{n\times m}$, where $\tilde{\nabla}\mathcal{L}(W)$ is the gradient in the ambient space and $\eta > 0$ is
the step size.

Output: Matrices $Z_1 \in \mathbb{R}_*^{n\times k}, Z_2 \in \mathbb{R}_*^{m\times k}$ such that $Z_1 Z_2^T = R_W(-\eta\nabla\mathcal{L}(W))$.

Compute:                                                                              matrix dimension
    $A^\dagger = (A^T A)^{-1} A^T, B^\dagger = (B^T B)^{-1} B^T$                                $k \times n$,    $k \times m$
    $A_\perp, B_\perp$= orthogonal complements of $A, B$                          $n \times (n-k)$,    $m \times (m-k)$
    $M = A^\dagger G B^{\dagger T}$                                                                        $k \times k$
    $N_1 = B_\perp^T G^T A^{\dagger T}$                                                            $(m-k) \times k$
    $N_2 = A_\perp^T G B^{\dagger T}$                                                                $(n-k) \times k$
    $Z_1 = A\left(I_k + \frac{1}{2}M - \frac{1}{8}M^2\right) + A_\perp N_2 \left(I_k - \frac{1}{2}M\right)$                     $n \times k$
    $Z_2 = B\left(I_k + \frac{1}{2}M^T - \frac{1}{8}(M^T)^2\right) + B_\perp N_1 \left(I_k - \frac{1}{2}M^T\right)$             $m \times k$
```

necessary for a Riemannian stochastic gradient descent algorithm. The procedure is outlined in Algorithm 2.

Algorithm 2 explicitly computes and stores the orthogonal complement matrices $A_\perp$ and $B_\perp$, which in the low rank case $k \ll m, n$, have size $O(mn)$, the same as the full sized W. To improve the memory complexity, we use the fact that the matrices $A_\perp$ and $B_\perp$ always operate with their transpose. Since $A_\perp$ and $B_\perp$ have orthogonal columns, the matrix $A_\perp A_\perp^T$ is actually the projection matrix that we denoted earlier by $P_{A_\perp}$, and likewise for $B_\perp$. Because of orthogonal complementarity, these projection matrices are equal to $I_n - P_A$ and $I_m - P_B$ respectively. Thus we can write $A_\perp N_2 = \left(I - AA^\dagger\right) Z B^{\dagger^T}$, and a similar identity for $B_\perp N_1$.

Consider now the case where the gradient matrix is of rank-r and is available in a factorized form $Z = G_1 G_2^T$, with $G_1 \in \mathbb{R}^{n\times r}, G_2 \in \mathbb{R}^{m\times r}$. Using the factorized gradient we can reformulate the algorithm to keep in memory only matrices of size at most $\max(n,m) \times k$ or $\max(n,m) \times r$. Optimizing the order of matrix operations so that the number of operations is minimized gives Algorithm 3. The runtime complexity of Algorithm 3 is readily computed based on matrix multiplications complexity,[2] and is $O\left((n+m)(k+r)^2\right)$.

3.2 LORETA With Rank-one Gradients

In many learning problems, the gradient matrix $\tilde{\nabla}\mathcal{L}(W)$ required for a gradient step update has a rank of one. This is the case for example, when the matrix model W acts as a bilinear form on two vectors, p and q, and the loss is a piecewise linear function of $\mathbf{p}^T W \mathbf{q}$ (as in Grangier and Bengio, 2008; Chechik et al., 2010; Weinberger and Saul, 2009; Shalev-Shwartz et al., 2004 and Section 7.1 below). In that case, the gradient is the rank-one outer product matrix $\mathbf{p}\mathbf{q}^T$. As another example, consider the case of multitask learning, where the matrix model W operates on a vector input $\mathbf{p}$, and the loss is the squared loss $\|W\mathbf{p} - \mathbf{q}\|^2$ between the multiple predictions $W\mathbf{p}$ and the true labels $\mathbf{q}$. The gradient of this loss is $(W\mathbf{p} - \mathbf{q})\,\mathbf{p}^T$, which is again a rank-one matrix. We now show how to

[2] We assume throughout this paper the use of ordinary (schoolbook) matrix multiplication.

Algorithm 3 : **LORETA-r** - General-rank Riemannian stochastic gradient descent

Input: Matrices $A \in \mathbb{R}_*^{n\times k}$, $B \in \mathbb{R}_*^{m\times k}$ s.t. $W = AB^T$. Matrices $G_1 \in \mathbb{R}^{n\times r}$, $G_2 \in \mathbb{R}^{m\times r}$ s.t. $G_1 G_2^T = -\eta\tilde{\nabla}\mathcal{L}(W) \in \mathbb{R}^{n\times m}$, where $\tilde{\nabla}\mathcal{L}(W)$ is the gradient in the ambient space and $\eta > 0$ is the step size.

Output: Matrices $Z_1 \in \mathbb{R}_*^{n\times k}, Z_2 \in \mathbb{R}_*^{m\times k}$ such that $Z_1 Z_2^T = R_W(-\eta\nabla\mathcal{L}(W))$.

Compute:	matrix dimension	runtime complexity
$A^\dagger = (A^T A)^{-1} A^T, \quad B^\dagger = (B^T B)^{-1} B^T$	$k\times n, \quad k\times m$	$O((n+m)k^2)$
$\mathbf{a_1} = A^\dagger \cdot G_1, \quad \mathbf{b_1} = B^\dagger \cdot G_2$	$k\times r, \quad k\times r$	$O((n+m)kr)$
$\mathbf{a_2} = A\cdot\mathbf{a_1}$	$n\times r$	$O(nkr)$
$Q = \mathbf{b_1}^T \cdot \mathbf{a_1}$	$r\times r$	$O(kr^2)$
$\mathbf{a_3} = -\frac{1}{2}\mathbf{a_2} + \frac{3}{8}\mathbf{a_2}\cdot Q + G_1 - \frac{1}{2}G_1\cdot Q$	$n\times r$	$O(nr^2)$
$Z_1 = A + \mathbf{a_3}\cdot\mathbf{b_1}^T$	$n\times k$	$O(nkr)$
$\mathbf{b_2} = \left(G_2^T B\right)\cdot B^\dagger$	$r\times m$	$O(mkr)$
$\mathbf{b_3} = -\frac{1}{2}\mathbf{b_2} + \frac{3}{8}Q\cdot\mathbf{b_2} + G_2^T - \frac{1}{2}Q\cdot G_2^T$	$r\times m$	$O(mr^2)$
$Z_2^T = B^T + \mathbf{a_1}\cdot\mathbf{b_3}$	$k\times m$	$O(mkr)$

reduce the complexity of each iteration to be linear in the model rank k when the rank of the gradient matrix is $r = 1$.

Algorithm 4 : **LORETA-1** - Rank-one Riemannian stochastic gradient descent

Input: Matrices $A \in \mathbb{R}_*^{n\times k}, B \in \mathbb{R}_*^{m\times k}$ s.t. $W = AB^T$. Matrices $A^\dagger$ and $B^\dagger$, the pseudo-inverses of A and B respectively. Vectors $\mathbf{p} \in \mathbb{R}^{n\times 1}$, $\mathbf{q} \in \mathbb{R}^{m\times 1}$ s.t. $\mathbf{pq}^T = -\eta\tilde{\nabla}\mathcal{L}(W) \in \mathbb{R}^{n\times m}$, where $\tilde{\nabla}\mathcal{L}(W)$ is the gradient in the ambient space and $\eta > 0$ is the step size.

Output: Matrices $Z_1 \in \mathbb{R}_*^{n\times k}$, $Z_2 \in \mathbb{R}_*^{m\times k}$ s.t. $Z_1 Z_2^T = R_W(-\eta\nabla\mathcal{L}(W))$. Matrices $Z_1^\dagger$ and $Z_2^\dagger$, the pseudo-inverses of Z_1 and Z_2 respectively.

Compute:	matrix dimension	runtime complexity
$\mathbf{a_1} = A^\dagger\cdot\mathbf{p}, \mathbf{b_1} = B^\dagger\cdot\mathbf{q}$	$k\times 1$	$O((n+m)k)$
$\mathbf{a_2} = A\cdot\mathbf{a_1}$	$n\times 1$	$O(nk)$
$s = \mathbf{b_1}^T\cdot\mathbf{a_1}$	1×1	$O(k)$
$\mathbf{a_3} = \mathbf{a_2}\left(-\frac{1}{2} + \frac{3}{8}s\right) + \mathbf{p}(1 - \frac{1}{2}s)$	$n\times 1$	$O(n)$
$Z_1 = A + \mathbf{a_3}\cdot\mathbf{b_1}^T$	$n\times k$	$O(nk)$
$\mathbf{b_2} = \left(\mathbf{q}^T B\right)\cdot B^\dagger$	$1\times m$	$O(mk)$
$\mathbf{b_3} = \mathbf{b_2}\left(-\frac{1}{2} + \frac{3}{8}s\right) + \mathbf{q}^T(1 - \frac{1}{2}s)$	$1\times m$	$O(m)$
$Z_2^T = B^T + \mathbf{a_1}\cdot\mathbf{b_3}$	$k\times m$	$O(mk)$
$Z_1^\dagger = rank\_one\_pseudoinverse\_update(A, A^\dagger, \mathbf{a_3}, \mathbf{b_1})$	$k\times n$	$O(nk)$
$Z_2^\dagger = rank\_one\_pseudoinverse\_update(B, B^\dagger, \mathbf{b_3}, \mathbf{a_1})$	$k\times m$	$O(mk)$

Given rank-one gradients, the most computationally demanding step in Algorithm 3 is the computation of the pseudo-inverse of the matrices A and B, taking $O(nk^2)$ and $O(mk^2)$ operations. All other operations are $O(\max(n,m)\cdot k)$ at most. To speed up calculations we use the fact that for

$r = 1$ the outputs Z_1 and Z_2 become rank-one updates of the input matrices A and B. This enables us to keep the pseudo-inverses $A^\dagger$ and $B^\dagger$ from the previous round, and perform a rank-one update to them, following a procedure developed by Meyer (1973). The full procedure is included in Appendix D. This procedure is similar to the better known Sherman-Morrison formula for the inverse of a rank-one perturbed matrix, and its computational complexity for an $n \times k$ matrix is $O(nk)$ operations. Using that procedure, we derive our final algorithm, LORETA-*1*, the rank-one Riemannian stochastic gradient descent. Its overall time and space complexity are both $O((n+m)k)$ per gradient step. It can be seen that the LORETA-*1* algorithm uses only basic matrix operations, with the most expensive ones being low-rank matrix-vector multiplication and low-rank matrix-matrix addition. The memory requirement of LORETA-1 is about $4nk$ (assuming $m = n$), since it receives four input matrices of size nk $(A, B, A^\dagger, B^\dagger)$ and assuming it can compute the four outputs $(Z_1, Z_2, Z_1^\dagger, Z_2^\dagger)$, in-place while destroying previously computed terms.

4. Online Learning of Low-rank Positive Semidefinite Matrices

In this section we adapt the derivation above to the special case of positive semidefinite (PSD) matrices. PSD matrices are of special interest because they encode a true Euclidean metric. An n-by-n PSD matrix W of rank-k can be factored as $W = YY^T$, with $Y \in \mathbb{R}^{n\times k}$. Thus, the bilinear form $x^T Wz$ is equal to $(Yx)^T(Yz)$, which is a Euclidean inner product in the space spanned by Y's columns. These properties have led to an extensive use of PSD matrix models in metric and similarity learning, see, for example, Xing et al. (2002), Goldberger et al. (2005), Globerson and Roweis (2006), Bar-Hillel et al. (2006) and Jain et al. (2008). The set of n-by-n PSD matrices of rank-k forms a manifold of dimension $nk - \frac{k(k-1)}{2}$, embedded in the Euclidean space $\mathbb{R}^{n\times n}$ (Vandereycken et al., 2009). We denote this manifold by $\mathcal{S}_+(k,n)$.

We now give a characterization of the tangent space of this manifold, due to Vandereycken and Vandewalle (2010).

Lemma 5 *Let $W \in \mathcal{S}_+(k,n)$ have a (non-unique) factorization $W = YY^T$, where $Y \in \mathbb{R}_*^{n\times k}$. Let $Y_\perp \in \mathbb{R}^{n\times(n-k)}$ be the orthogonal complement of Y such that $Y_\perp^T Y = 0$, $Y_\perp^T Y_\perp = I_{n-k}$. The tangent space to $\mathcal{S}_+(k,n)$ at W is:*

$$T_W\mathcal{S}_+(k,n) = \left\{ \begin{bmatrix} Y & Y_\perp \end{bmatrix} \begin{bmatrix} S & N^T \\ N & 0 \end{bmatrix} \begin{bmatrix} Y^T \\ Y_\perp^T \end{bmatrix} : S \in \mathbb{R}^{k\times k}, N \in \mathbb{R}^{(n-k)\times k}, S = S^T \right\}.$$

Proof See Vandereycken and Vandewalle (2010), Proposition 5.2. ■

Let $\xi \in T_W\mathcal{S}_+(k,n)$ be a tangent vector to $W = YY^T$. As shown by Vandereycken and Vandewalle (2010), ξ can be decomposed into two orthogonal components, $\xi = \xi^S + \xi^P$. Given a rank-r gradient matrix Z, and using the projection matrices P_Y and $P_{Y_\perp}$ they show that:

$$\xi^S = P_Y \frac{Z+Z^T}{2} P_Y,$$
$$\xi^P = P_{Y_\perp} \frac{Z+Z^T}{2} P_Y + P_Y \frac{Z+Z^T}{2} P_{Y_\perp}.$$

Using this characterization of the tangent vector when given an ambient gradient Z, one can define a retraction analogous to that defined in Section 3. This retraction is referred to as $R_W^{(2)}$ in Vandereycken and Vandewalle (2010).

Theorem 6 *Let $W \in \mathcal{S}_+(k,n)$, $W = YY^T$, and $W^\dagger$ be its pseudo-inverse. Let $\xi \in T_W\mathcal{S}_+(k,n)$, $\xi = \xi^S + \xi^P$, as described above, and let*

$$V = W + \frac{1}{2}\xi^S + \xi^P - \frac{1}{8}\xi^S W^\dagger \xi^S - \frac{1}{2}\xi^P W^\dagger \xi^S.$$

The mapping $R_W^{PSD}(\xi) = VW^\dagger V$ is a second order retraction from a neighborhood $\Theta_W \subset T_W\mathcal{S}_+(k,n)$ to $\mathcal{S}_+(k,n)$.

Proof See Vandereycken and Vandewalle (2010), Proposition 5.10. ■

Algorithm 5 : **LORETA-1-PSD** - Rank-one Riemannian PSD stochastic gradient descent

Input: A matrix $Y \in \mathbb{R}_*^{n\times k}$, s.t. $W = YY^T$. The matrix $Y^\dagger$, the pseudoinverse of Y. Vectors $\mathbf{p} \in \mathbb{R}^{n\times 1}$, $\mathbf{q} \in \mathbb{R}^{n\times 1}$ s.t. $pq^T = -\eta\tilde{\nabla}\mathcal{L}(W) \in \mathbb{R}^{n\times m}$, where $\tilde{\nabla}\mathcal{L}(W)$ is the gradient in the ambient space and $\eta > 0$ is the step size.

Output: Matrix $Z \in \mathbb{R}_*^{n\times k}$, s.t. $ZZ^T = R_W^{PSD}(-\eta\nabla\mathcal{L}(W))$. Matrix $Z^\dagger$, the pseudo-inverse of Z.

Compute:	matrix dimension	runtime complexity
$\mathbf{h_1} = Y^\dagger\mathbf{p}$	$k\times 1$	$O(nk)$
$\mathbf{h_2} = Y^\dagger\mathbf{q}$	$k\times 1$	$O(nk)$
$n_1 = \mathbf{h_1}^T\mathbf{h_1}$	1×1	$O(k)$
$n_2 = \mathbf{h_2}^T\mathbf{h_2}$	1×1	$O(k)$
$\hat{\mathbf{h_1}} = Y\mathbf{h_1}$	$n\times 1$	$O(nk)$
$\hat{\mathbf{h_2}} = Y\mathbf{h_2}$	$n\times 1$	$O(nk)$
$s = \mathbf{h_1}^T\mathbf{h_2}$	1×1	$O(k)$
$\mathbf{l_1} = (-\frac{1}{4} + \frac{3}{32}s)\hat{\mathbf{h_1}} + (\frac{1}{2} - \frac{1}{8}s)\mathbf{p} + \frac{3}{32}n_1\hat{\mathbf{h_2}} - \frac{1}{8}n_1\mathbf{q}$	$n\times 1$	$O(n)$
$\mathbf{l_2} = (-\frac{1}{4} + \frac{3}{32}s)\hat{\mathbf{h_2}} + (\frac{1}{2} - \frac{1}{8}s)\mathbf{q} + \frac{3}{32}n_2\hat{\mathbf{h_1}} - \frac{1}{8}n_2\mathbf{p}$	$n\times 1$	$O(n)$
$P_1 = \mathbf{l_1}\mathbf{h_2}^T$	$n\times k$	$O(nk)$
$P_2 = \mathbf{l_2}\mathbf{h_1}^T$	$n\times k$	$O(nk)$
$Z = Y + P_1 + P_2$	$n\times k$	$O(nk)$
$Z_{temp}^\dagger = rank\_one\_pseudoinverse\_update(Y, Y^\dagger, \mathbf{l_1}, \mathbf{h_2})$	$k\times n$	$O(nk)$
$Z^\dagger = rank\_one\_pseudoinverse\_update(Y + P_1, Z_{temp}^\dagger, \mathbf{l_2}, \mathbf{h_1})$	$k\times n$	$O(nk)$

Following the derivation of algorithms 2-4, and after some rearrangement, we obtain a PSD version of the LORETA-*1* algorithm. This PSD version is given in Algorithm (5). The algorithm is very similar to LORETA-*1* , but instead of learning a general rank-k matrix it learns a positive semidefinite rank-k matrix. The computational complexity and memory complexity of a gradient step for this algorithm is $O(nk)$, namely, it is linear in the reduced number of model parameters.

5. Manifold Identification

Until now, we formalized the problem of learning a low-rank matrix based on a factorization $W = AB$. At test time, computing the bilinear score using the model can be even faster when the data is sparse. For instance, given two vectors x_1 and x_2 with c_1 and c_2 non-zero values, computing the bilinear form $x_1^T AB^T x_2$ requires $O(c_1k+k+kc_2) = O((c_1+c_2)k)$ operations, and can be significantly faster than the dense case. However, at training time, the LORETA-1 algorithm still has a complexity of $O((m+n)k)$ for each iteration even when the data is sparse.

The current section describes an attempt to adapt LORETA-1 such that it treats sparse data more efficiently. The empirical evaluation of this adaptation showed mixed results, but we include the derivation for completeness. The main idea is to separate the low-rank projection into two steps. First, a projection to a low dimensional space Ax that can be computed efficiently when x is sparse. Then, learning a second matrix, whose role is to tune the representation in the k-dimensional space.

To explain the idea, we focus on the case of learning a low-rank model which parametrizes a similarity function. The model is $W = AB^T$, $A \in \mathbb{R}^{n\times k}$, $B \in \mathbb{R}^{n\times k}$. The similarity between two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$ is then given by

$$Sim(\mathbf{p},\mathbf{q}) = \mathbf{p}^T W \mathbf{q} = (A^T\mathbf{p})^T \cdot (B^T\mathbf{q}). \tag{3}$$

This similarity measure can be viewed as the cosine similarity in $\mathbb{R}^k$ between the projected vectors $B^T\mathbf{q}$ and $A^T\mathbf{p}$. We now introduce another similarity model which operates directly in the projected space. Formally, we have $M \in \mathbb{R}^{k\times k}$, and the similarity model is

$$Sim(\mathbf{p},\mathbf{q}) = (A^T\mathbf{p})^T M (B^T\mathbf{q}) = \mathbf{p}^T AMB^T\mathbf{q}. \tag{4}$$

Clearly, since the model in Equation (4) involves only linear matrix multiplications, its descriptive power is equivalent to that of the model Equation (3). However, it has the potential to be learned faster. To speed the training we can iterate between learning the outer projections A,B using LORETA , and learning the inner low-dimensional similarity model M using standard methods operating in the low-dimensional space. Specifically, the idea is to execute s update steps of M for every update step of A,B (Algorithm 6). After s update steps to M, it is decomposed using SVD to obtain $M = USV^T$, and these factors are used to update the outer projections using $A \leftarrow AUsqrt(S)$, $B \leftarrow BVsqrt(S)$.

Consider the computational complexity: Given two sparse vectors x_1, x_2 with c_1 and c_2 non-zero values respectively, projecting them using A and B to the low dimensional space takes $O(k(c_1+c_2))$, and an update step of M takes $O(k^2)$. Decomposing M using SVD takes $O(k^3)$, so the overall complexity for s updates is $O\left(k\cdot\left(s(k+c_1+c_2)+k^2\right)\right)$. When $s \geq k$ the cost of decomposition is amortized across many M updates and does not increase the overall complexity. The update of A, B takes $O(k(n+m))$ as before. This approach is related to the idea of manifold identification (Oberlin and Wright, 2007), where the learning of A, B "identifies" a manifold of rank k and the inner steps operate to tune the representation within that subspace.

This iterative procedure could be a significant speed up compared to the original $O((m+n)k)$. Unfortunately, when we tested this algorithm in a similarity learning task (as in Section 7.1), its performance was not as good as that of LORETA-1. The main reason was numerical instability: The matrix M typically collapsed to match few directions in A, and decomposing it has amplified the same A directions. This approach awaits deeper investigation which is outside the scope of the current paper.

6. Related Work

A recent summary of many advances in the field of optimization on manifolds is given by Absil et al. (2008). Advances in this field have lately been applied to matrix completion (Keshavan et al.,

Algorithm 6 : Manifold identification meta-algorithm

Input: Initial model matrices $A \in \mathbb{R}_*^{n\times k}$, $B \in \mathbb{R}_*^{m\times k}$ s.t. $W = AB^T$. Matrices $A^\dagger$ and $B^\dagger$, the pseudo-inverses of A and B respectively. Loss function $\mathcal{L}$.

Output: Matrices $A \in \mathbb{R}_*^{n\times k}, B \in \mathbb{R}_*^{m\times k}$ s.t. $W = AB^T$.

Parameters: η_1: LORETA step size . η_2: low-dimensional similarity learning step size. s: number of low-dimensional learning steps per round

```
repeat:
    $[g_1, g_2] = \nabla\mathcal{L}(AB^T)$
    $[A, B, A^\dagger, B^\dagger] = \text{LORETA}\ \left(A, B, A^\dagger, B^\dagger, g_1, g_2, \eta_1\right)$
    initialize $M = I_k$
    for i=1:s
        $[g_1, g_2] = \nabla\mathcal{L}(AMB^T)$
        $M = full - rank - metric - learning\left(M, A^T g_1, B^T g_2, \eta_2\right)$
    endfor
    $[U, S, V] = svd(M)$
    $A = A \cdot U \cdot sqrt(S)$
    $B = B \cdot V \cdot sqrt(S)$
until stopping condition is satisfied
```

2010), tensor-rank estimation (Eldén and Savas, 2009; Ishteva et al., 2011) and sparse PCA (Journée et al., 2010b).

Broadly speaking, there are two kinds of manifolds used in optimization. The first are *embedded manifolds*, manifolds that form a subset of Euclidean space, and are the ones we employ in this work. The second kind are *quotient manifolds*, which are formed by defining an equivalence relation on a Euclidean space, and endowing the resulting equivalence classes with an appropriate Riemannian metric. For example, the equivalence relation on $\mathbb{R}^n$ defined by $x \sim y \iff \exists\lambda > 0, x = \lambda y$, yields a quotient space called the *real projective space* when given a proper Riemannian metric.

More specific to the field of low-rank matrix manifolds, work has been done on the general problem of optimization with low-rank positive semi-definite (PSD) matrices. The latest and most relevant is the work of Meyer et al. (2011). In this work, Meyer and colleagues develop a framework for Riemannian stochastic gradient descent on the manifold of PSD matrices, and apply it to the problem of kernel learning and the learning of Mahalanobis distances. Their main technical tool is that of quotient manifolds mentioned above, as opposed to the embedded manifold we use in this work. Another paper which uses a quotient manifold representation is that of Journée et al. (2010a), which introduces a method for optimizing over low-rank PSD matrices.

In their 2010 paper (Vandereycken and Vandewalle, 2010), Vandereycken et al. introduced a retraction for PSD matrices in the context of modeling systems of partial differential equations. We build on this work in order to construct our methods of learning general and PSD low-rank matrices.

In general, the problem of minimizing a convex function over the set of low-rank matrices was addressed by several authors, including Fazel (2002). Recht et al. (2010) and more recently Jain et al. (2011) also consider the same problem, with additional affine constraints, and its connection to recent advances in compressed sensing. The main tools used in these papers are the trace norm (sum of singular values) and semi-definite programming. See also Fazel et al. (2005) for a short introduction to these methods.

More closely related to the current paper are the papers by Kulis et al. (2009) and Meka et al. (2008). Kulis et al. (2009) deal with learning low-rank PSD matrices, and use the rank-preserving log-det divergence and clever factorization and optimization in order to derive an update rule with runtime complexity of $O(nk^2)$ for an $n \times n$ matrix of rank k. Meka et al. (2008) use online learning in order to find a minimal rank square matrix under approximate affine constraints. The algorithm does not directly allow a factorized representation, and depends on an "oracle" component, which typically requires to compute an SVD.

Multi-class ranking with a large number of features was studied by Bai et al. (2009), and in the context of factored representations, by Weston et al. (2011) (WSABIE). WSABIE combines projected gradient updates with a novel sampling scheme which is designed to minimize a ranking loss named WARP. WARP is shown to outperform simpler triplet sampling approaches. Since WARP yields rank-1 gradients, it can easily be adapted for Riemannian SGD, but we leave experiments with such sampling schemes to future work.

7. Experiments

We tested LORETA in two learning tasks: learning a similarity measure between pairs of text documents using the 20-newsgroups data collected by Lang (1995), and learning to rank image label annotations based on a multi-label annotated set, using the *ImageNet* data set (Deng et al., 2009). Matlab code for LORETA-1 is available online at *http://chechiklab.biu.ac.il/research/LORETA*.

7.1 Learning Similarity on the 20 Newsgroups Data Set

In our first set of experiments, we looked at the problem of learning a similarity measure between pairs of text documents. Similarity learning is a well studied problem, closely related to metric learning (see Yang 2007 for a review). It has numerous applications in information retrieval such as *query by example*, and finding related content on the web.

One approach to learn pairwise relations is to measure the similarity of two documents $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$ using a bilinear form parametrized by a model $W \in \mathbb{R}^{n \times n}$:

$$S_W(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T W \mathbf{q}.$$

Such models can be learned online (Chechik et al., 2010) and were shown to achieve high precision. Sometimes the matrix W is required to be symmetric and positive definite, which means it actually encodes a metric, also known as a Mahalanobis distance. Unfortunately, since the number of parameters grows as n^2, storing the matrix W in memory is only feasible for limited feature dimensionality. To handle larger vocabularies, like those containing all textual terms found in a corpus, a common approach is to pre-select a subset of the features and train a model over the low dimensional data. However, such preprocessing may remove crucial signals in the data even if features are selected in a discriminative way.

To overcome this difficulty, we used LORETA-1 and LORETA-1-PSD to learn a rank-k parametrization of the model W. This model can be factorized as $W = AB^T$, where $A, B \in \mathbb{R}^{n \times k}$ for the general case, or as $W = AA^T$ for the PSD case. In each of our experiments, we selected a subset of n features, and trained a rank k model. We varied the number of features n and the rank of the matrix k

so as to use a fixed amount of memory. For example, we used a rank-10 model with $50K$ features, and a rank-50 model with $10K$ features.

7.1.1 SIMILARITY LEARNING WITH LORETA-1

We use an online procedure similar to that in Grangier and Bengio (2008) and Chechik et al. (2010). At each round, three instances are sampled: a query document $\mathbf{q} \in \mathbb{R}^n$, and two documents $\mathbf{p}_+$, $\mathbf{p}_- \in \mathbb{R}^n$ such that $\mathbf{p}_+$ is known to be more similar to $\mathbf{q}$ than $\mathbf{p}_-$. We wish that the model assigns a higher similarity score to the pair $(\mathbf{q}, \mathbf{p}_+)$ than the pair $(\mathbf{q}, \mathbf{p}_-)$, and hence use the online ranking hinge loss defined as $l_W(\mathbf{q}, \mathbf{p}_+, \mathbf{p}_-) = [1 - S_W(\mathbf{q}, \mathbf{p}_+) + S_W(\mathbf{q}, \mathbf{p}_-)]_+$, where $[z]_+ = max(z, 0)$.

We initialized the model to be a truncated identity matrix, with only the first k ones along the diagonal. This corresponds in our case to choosing the k most informative terms as the initial data projection.

7.1.2 DATA PREPROCESSING AND FEATURE SELECTION

We used the 20 newsgroups data set (people.csail.mit.edu/jrennie/20Newsgroups), containing 20 classes with approximately 1000 documents each. We removed stop words but did not apply stemming. The document terms form a vocabulary of 50,000 terms, and we selected a subset of these features that conveyed high information about the identity of the class (over the training set) using the *infogain* criterion (Yang and Pedersen, 1997). This is a discriminative criterion,which measures the number of bits gained for category prediction by knowing the presence or absence of a term in a document. The selected features were normalized using *tf-idf*, and then represented each document as a bag of words. Two documents were considered similar if they shared the same class label, out of the possible 20 labels.

7.1.3 EXPERIMENTAL PROCEDURE AND EVALUATION PROTOCOL

The 20 newsgroups site proposes a split of the data into train and test sets. We repeated splitting 5 times based on the sizes of the proposed splits (a train / test ratio of 65% / 35%). We evaluated the learned similarity measures using a ranking criterion. We view every document $\mathbf{q}$ in the test set as a query, and rank the remaining test documents $\mathbf{p}$ by their similarity scores $\mathbf{q}^T W \mathbf{p}$. We then compute the precision (fraction of positives) at the top r ranked documents. We then average the precision over all positions r such that there exists a positive example in the top r. This final measure is called *mean average precision*, and is commonly used in the information retrieval community (Manning et al., 2008, Chapter 8).

7.1.4 COMPARISONS

We compared LORETA with the following approaches.

1. **Naive gradient descent** (GD): similar to Bai et al. (2009). The model is represented as a product of two matrices $W = AB^T$. Stochastic gradient descent steps are computed over the factors A and B, for the same loss used by LORETA $l_W(\mathbf{q}, \mathbf{p}_+, \mathbf{p}_-)$. The GD steps are:

$$A_{new} = A - \eta\, \mathbf{q}(\mathbf{p}_- - \mathbf{p}_+)^T B,$$
$$B_{new} = B - \eta\, (\mathbf{p}_- - \mathbf{p}_+)\mathbf{q}^T A.$$

   We found this approach to be very unstable, and thus its results are not presented.

2. **Naive PSD gradient descent**: similar to the method above, except that now the model is constrained to be PSD. The model is represented as a product $W = AA^T$. Stochastic gradient descent steps are computed over the factor A for the same loss used by LORETA : $l_W(\mathbf{q}, \mathbf{p}_+, \mathbf{p}_-)$. As shown by Meyer et al. (2011), this is in fact equivalent to Riemannian stochastic GD in the manifold of PSD matrices when this manifold is endowed with a certain metric the authors call the *flat metric*.

   The GD step is:

   $$A_{new} = A - \eta \left(\mathbf{q}(\mathbf{p}_- - \mathbf{p}_+)^T + (\mathbf{p}_- - \mathbf{p}_+)\mathbf{q}^T\right) A.$$

   The step size η was chosen by cross validation. This approach was more stable in the PSD case than in the general case, probably because the invariant space here is only the group of orthogonal matrices (which are well-conditioned), as opposed to the group of invertible matrices which might be ill-conditioned.

3. **Iterative Passive-Aggressive (PA)**: since we found the above general GD procedure **(1)** to be very unstable, we experimented with a related online algorithm from the family of passive-aggressive algorithms (Crammer et al., 2006). We iteratively optimize over A given a fixed B and vice versa. The optimization is a tradeoff between minimizing the loss l_W, and limiting how much the models change at each iteration. The steps sizes for updating A and B are computed to be:

   $$\begin{aligned}\eta_A &= \min\left(\frac{l_W(\mathbf{q}, \mathbf{p}_+, \mathbf{p}_-)}{\|\mathbf{q}\|^2 \cdot \|B^T(\mathbf{p}_+ - \mathbf{p}_-)\|^2}, C\right),\\ \eta_B &= \min\left(\frac{l_W(\mathbf{q}, \mathbf{p}_+, \mathbf{p}_-)}{\|(\mathbf{p}_+ - \mathbf{p}_-)\|^2 \cdot \|A^T\mathbf{q}\|^2}, C\right).\end{aligned}$$

   C is a predefined parameter controlling the maximum magnitude of the step size, chosen by cross-validation. This procedure is numerically more stable because of the normalization by the norms of the matrices multiplied by the gradient factors.

4. **Full rank similarity learning models.** We compared with two full rank online metric learning methods, LEGO (Jain et al., 2008) and OASIS (Chechik et al., 2010). Both algorithms learn a full (non-factorized) model, and were run with $n = 1000$, in order to be consistent with the memory constraint of LORETA-1. We have also compared with both full-rank models using rank 2000, that is, 4 times the memory constraint. We have not compared with batch approaches such as Kulis et al. (2009), since they are not expected to scale to very large data sets such as those our work is ultimately aiming towards.

In addition, we have experimented with the method for learning PSD matrices using a polar geometry characterization of the quotient manifold, due to Meyer et al. (2011). This method's runtime complexity is $O((n+m)k^2)$, and we have found that its performance was not in line with the methods described above.

7.1.5 RESULTS

Figure 3c shows the mean average precision obtained with all the above methods. LORETA outperforms the other approaches across all ranks. LORETA-PSD achieves slightly higher precision than LORETA. The reason may be that similarity was defined based on two samples belonging to a common class, and this relation is symmetric and transitive, two relations which are respected by PSD matrices but not by general similarity matrices. Moreover, LORETA-PSD learned faster along the training iterations when compared with LORETA - see Figure 3a for a comparison of the learning curves. Interestingly, for both LORETA algorithms learning a low-rank model of rank 30, using the best 16660 features, was significantly more precise than learning a much fuller model of rank 100 and 5000 features, or a model using the full 50000 word vocabulary but with rank 10 . The intuition is that LORETA can be viewed as adaptively learning a linear projection of the data into low dimensional space, which is tailored to the pairwise similarity task.

7.2 Image Multilabel Ranking

Our second set of experiments tackled the problem of learning to rank labels for images taken from a large number of classes ($L = 1660$) with multiple labels per image.

In our approach, we learn a linear classifier over n features for each label $c \in \mathcal{C} = \{1, \dots, L\}$, and stack all models together to a single matrix $W \in \mathbb{R}^{L \times n}$. At test time, given an image $\mathbf{p} \in \mathbb{R}^n$, the product $W\mathbf{p}$ provides scores for every label for that image $\mathbf{p}$. Given ground truth labeling, a good model would rank the true labels higher than the false ones. Each row of the matrix model can be thought of as a sub-model for the corresponding label. Imposing a low-rank constraint on the model implies that these sub-models are linear combinations of a smaller number of latent models. Alternatively, we can view learning a factored rank-k model $W = AB^T$ as learning a projection and classifier in the projected space concurrently. The matrix B^T projects the data onto a k dimensional space, and the matrix A consists of L classifiers operating in the low-dimensional space. The data we used for the experiment had ∼1500 labels, but the full ImageNet data set currently has ∼15000 labels, and is growing.

7.2.1 ONLINE LEARNING OF LABEL RANKINGS WITH LORETA-1

At each iteration, an image $\mathbf{p}$ is sampled, and using the current model W the scores for all its labels are computed, $W\mathbf{p}$. These scores are compared with the ground truth labeling $\mathbf{y} = \{y_1, \dots, y_r\} \subset \mathcal{C}$. We wish for all the scores of the true labels to be higher than the scores for the other labels by a margin of 1. Thus, the learner suffers a multilabel multiclass hinge loss as follows. Let $\bar{y} = \text{argmax}_{s \notin \mathbf{y}} (W\mathbf{p})_s$, be the negative label which obtained the highest score, where $(W\mathbf{p})_s$ is the s^{th} component of the score vector $W\mathbf{p}$.

The loss is then $\mathcal{L}(W, \mathbf{p}, \mathbf{y}) = \sum_{i=1}^{r} \left[(W\mathbf{p})_{\bar{y}} - (W\mathbf{p})_{y_i} + 1 \right]_+$, which is the sum of the margins between the top-ranked false label and all the positive labels which violated the margin of one from it. We used the subgradient G of this loss for LORETA: for the set of indices $i_1, i_2, \dots i_d \subset \mathbf{y}$ which incurred a non zero hinge loss, the i_j row of G is $\mathbf{p}$, and for the row $\bar{y}$ G is $-d \cdot \mathbf{p}$. The matrix G is rank one, unless no loss was suffered in which case it is 0.

The non-convex and stochastic nature of the learning procedure has lead us to try several initial conditions:

- **Zero matrix**: in this initialization we begin with a low-rank matrix composed entirely of zeros. This matrix is not included in the low-rank manifold $\mathcal{M}_k^{n,m}$, since its rank is less than k. We therefore perform a simple pre-training session in which we add up subgradients until a matrix of rank k is obtained. In practice we added the first $2k$ subgradients (each such subgradient being of rank one), and then performed an SVD to obtain the best rank-k model.

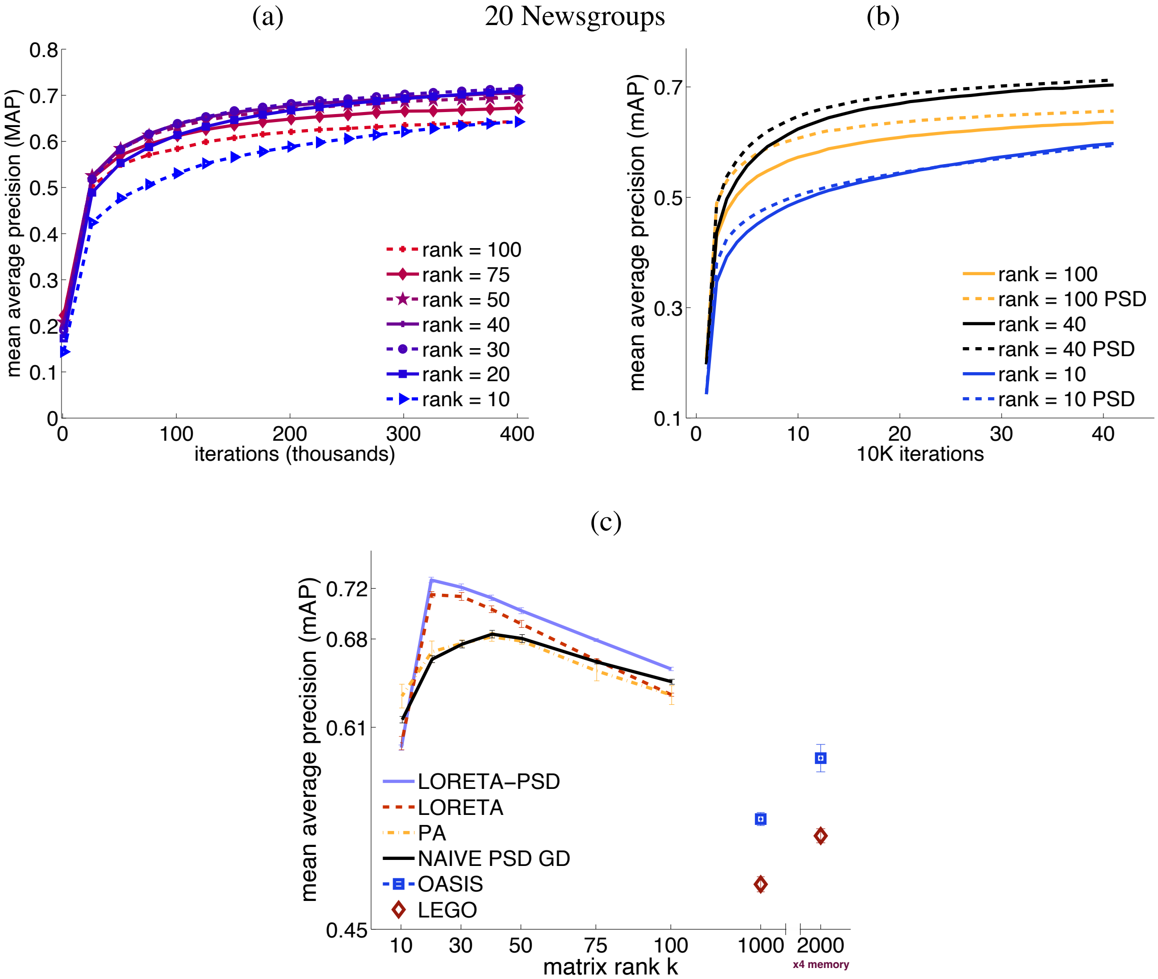


Figure 3: (a) Mean average precision (mAP) over 20 newsgroups test set as traced along LORETA learning for various ranks. Curve values are averages over 5 train-test splits. (b) Comparison of the learning curves of LORETA and LORETA-PSD. LORETA-PSD learns faster than LORETA across all ranks (shown are results for ranks 10, 40 and 100). (c) mAP of different models with varying rank. For each rank, a different number of features was selected using an information gain criterion, such that the total memory requirement is kept fixed (number of features × rank is constant). 50000 features were used for rank = 10. LEGO and OASIS were trained with the same memory (using 1000 features and rank=1000), as well as with 4 times the same memory (rank=2000). Error bars denote the standard error of the mean over 5 train-test splits.

We chose $2k$ because we wanted to ensure that the matrix we obtain has rank greater or equal to k.

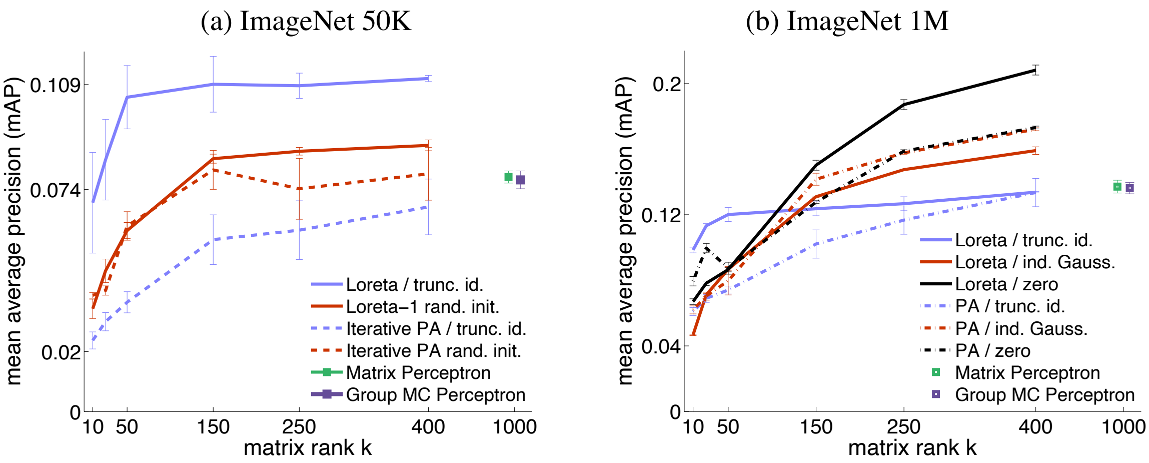


Figure 4: ImageNet data. Mean average precision (mAP) as a function of the rank k. Curves are means over five train-test splits. Error bars denote the standard error of the mean. Note the different scale of the left and right figure. All hyper parameters were selected using cross validation. Three different initializations were used: the zero matrix, a zero padded $k \times k$ identity matrix, and a product of two i.i.d. Gaussian matrices. See Section 7.2.1 for details.

- **Zero-padded identity**: we begin with a matrix composed of the $k \times k$ identity matrix I_k on the top left corner, padded with zeros so as to form an $L \times n$ matrix. This is guaranteed to be of rank k. The choice of the location of the identity matrix block is arbitrary.

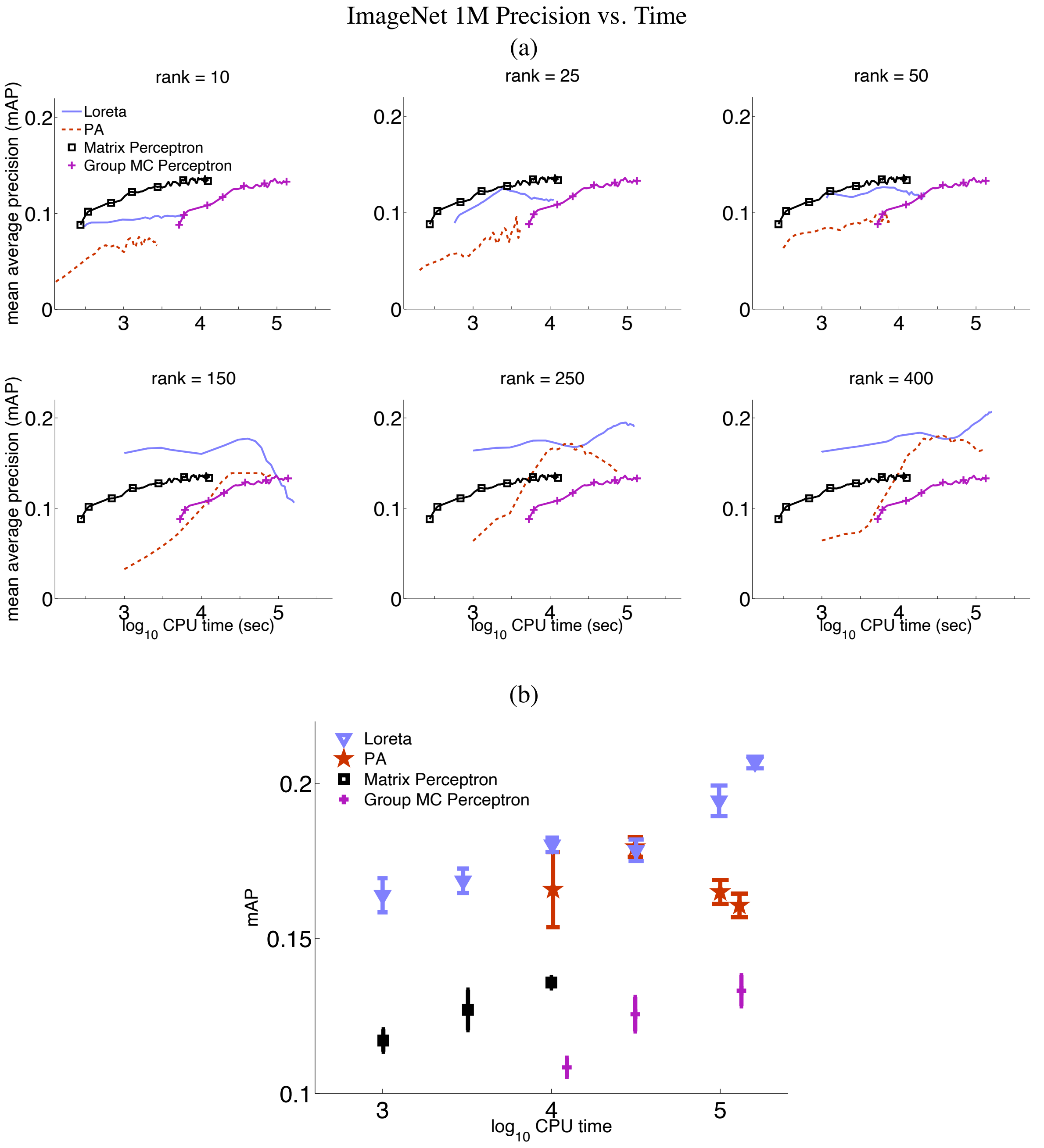
- **Independent Gaussian**: we sample independently the entries of the two factor matrices $A \in \mathbb{R}^{n\times k}$, $B\mathbb{R}^{m\times k}$ from a standard normal distribution. This model is thus initialized as a product of two random Gaussian matrices.

7.2.2 Data Set and Preprocessing

We used data from the ImageNet 2010 Challenge (www.imagenet.org/challenges/LSVRC/2010/) containing images labeled with respect to the WordNet hierarchy. Each image was manually labeled with a single class label (for a total of 1000 classes). We added labels for each image, using classes along the path to the root of the hierarchy (adding 676 classes in total). We discarded ancestor labels covering more than 10% of the images, leaving 1660 labels (5.2 labels per image on average). We used ImageNet's bag of words representation, based on vector quantizing SIFT features with a vocabulary of 1000 words, followed by *tf-idf* normalization.

7.2.3 Experimental Procedure and Evaluation Protocol

We trained on two data sets. A medium scale one of 50000 images, and a large data set consisting of 908210 images. We tested on 20000 images for the medium scale, and 252284 images for the large scale. The quality of the learned label ranking was evaluated using the *mean average precision* (mAP) criterion mentioned in 7.1.3 above (Manning et al., 2008, Chapter 8).


ImageNet 1M Precision vs. Time

(a)

(b)

Figure 5: (a) Mean average precision (mAP) as function of single CPU processing time in seconds for different algorithms and model ranks, presented on a log-scale. Matrix Perceptron (black squares) and Group Multi-Class Perceptron (purple crosses) are both full rank (rank=1000), and their curves are reproduced on all six panels for comparison. For each rank and algorithm (LORETA and PA), we used the best performing initialization scheme. (b) mAP of the best performing model for different algorithms and time points. Error bars represent standard deviation across 5 train-test splits.

7.2.4 COMPARISONS

We compared the performance of LORETA on this task with three other approaches:

1. **PA - Iterative Passive-Aggressive**: same as described in Section 7.1.4 above for the 20 Newsgroups experiment.

2. **Matrix Perceptron**: a full rank stochastic subgradient descent. The model is initialized as a zero matrix of size 1660 × 1000, and in each round the loss subgradient is subtracted from it. After a sufficient number of rounds, the model is typically full rank and dense.

3. **Group Multi-Class Perceptron**: a mixed (2,1) norm online mirror descent algorithm (Kakade et al., 2010). This algorithm encourages a group-sparsity pattern within the learned matrix model, thus presenting an alternative form of regularization when compared with low-rank models.

LORETA and PA were run using a range of model ranks. For all three methods, the step size (or the C parameter for PA) was chosen by 5-fold cross validation on a validation set.

7.2.5 RESULTS

Figure 4 plots the mAP precision of LORETA and PA for different model ranks, while showing on the right the mAP of the full rank 1000 Matrix Perceptron and $(2,1)$ norm algorithms. LORETA significantly improves over all other methods across all ranks. However, we note that LORETA, being a non-convex algorithm, does depend significantly on the method of initialization, with the zero-padded identity matrix being the best initialization for lower rank models, and the zero matrix the best initialization for higher rank models (rank ≥ 150).

In Figure 5 we show the accuracy as a function of CPU tim on a single CPU for the different algorithms and model ranks. We ran Matlab R2011a on an Intel Xeon 2.27 GHz machine, and used Matlab's `-singlethread` flag to control multithreading. The higher-rank LORETA models outperform all others both in the short time scale (~ 1000 sec.) and the long time scale ($\sim 100{,}000$ sec.). For some of the higher-rank models there is evident overtraining at some point, but this overtraining could be avoided by adopting an early-stopping procedure.

8. Discussion

We presented LORETA, an algorithm which learns a low-rank matrix based on stochastic Riemannian gradient descent and efficient retraction to the manifold of low-rank matrices. LORETA achieves superior precision in a task of learning similarity in high dimensional feature spaces, and in multi-label annotation, where it scales well with the number of classes. A PSD variant of LORETA can be used efficiently for low-rank metric learning.

There are many ways to tie together different classifiers in a multi-class setting. We have seen here that a low-rank assumption coupled with a Riemannian SGD procedure outperformed the (2,1) mixed norm. Other approaches leverage the hierarchical structure inherent in many of these tasks. For example, Deng et al. (2011) use the label hierarchy of ImageNet to compute a similarity measure between images.

For similarity learning, the approach we take in this paper uses a weak supervision based on ranking similar pairs: one only knows that the pair $(\mathbf{q}, \mathbf{p}_+)$ is more similar than the pair $(\mathbf{q}, \mathbf{p}_-)$. In some cases, a stronger supervision signal is available, like the classes of each objects are known. In these cases, Deng et al. (2011) have shown how to use class identities to construct good features by training an SVM classifier on each class and using its scaled output as a feature. They show that such features can lead to very good performance, with the added advantage that the features can be learned in parallel. The weak supervision approach that we take here aims to handle the case, which is particularly common in large scale data sets collected through web users' activity, where weaker supervision is much easier to collect.

In this paper, we used simple sampling schemes for both the similarity learning and multiple-labelling experiments. More elaborate sampling techniques such as those proposed by Weston et al. (2011), which focus on "hard negatives", may yield significant performance improvements. As these approaches typically involve rank-one gradients when implemented as online learning algorithms, they are well suited for being used in conjunction with LORETA, and this will be the subject of future work.

LORETA yields a factorized representation of the low-rank matrix. For similarity learning, these factors project to a low-dimensional space where similarity is evaluated efficiently. For classification, it can be viewed as learning two matrix components: one that projects the high dimensional data into a low dimension, and a second that learns to classify in the low dimension. In both approaches, the low-dimensional space is useful for extracting the relevant structure from the high-dimensional data, and for exploring the relations between large numbers of classes.

Acknowledgments

G.C. was supported by the Israeli science foundation Grant 1001/08, and by a Marie Curie reintegration grant PIRG06-GA-2009-256566. U.S. and D.W. were supported by the European Union under the DIRAC integrated project IST-027787. Uri Shalit is a recipient of the Google Europe Fellowship in Machine Learning, and this research is supported in part by this Google Fellowship. We would also like to thank the Gatsby Charitable Foundation for its generous support of the ICNC.

Appendix A. Proof of Lemma 2

We formally define the tangent space of a manifold at a point on the manifold, and then describe an auxiliary parametrization of the tangent space to the manifold $\mathcal{M}_k^{n,m}$ at a point $W \in \mathcal{M}_k^{n,m}$.

Definition 7 *The tangent space $T_{\mathbf{W}}\mathcal{M}$ to a manifold $\mathcal{M} \subset \mathbb{R}^n$ at a point $W \in \mathcal{M}$ is the linear space spanned by all the tangent vectors at 0 to smooth curves $\gamma : \mathbb{R} \to \mathcal{M}$ such that $\gamma(0) = W$. That is, the set of tangents in $\mathbb{R}^n$ to smooth curves within the manifold which pass through the point W.*

In order to characterize the tangent space of $\mathcal{M}_k^{n,m}$, we look into the properties of smooth curves γ, where for each $t, \gamma(t) \in \mathcal{M}_k^{n,m}$.

For any such curve, because of the rank k assumption, we may assume that for all $t \in \mathbb{R}$, there exist (non-unique) matrices $A(t) \in \mathbb{R}_*^{n\times k}$, $B(t) \in \mathbb{R}_*^{m\times k}$, such that $\gamma(t) = A(t)B(t)^T$. We now wish to find the tangent vectors to these curves. By the product rule we have:

$$\dot{\gamma}(0) = A(0)\dot{B}(0)^T + \dot{A}(0)B(0)^T.$$

Since $W=\gamma(0)=A(0)B(0)^T=AB^T$ we have for $W=AB^T$:

$$T_{\mathbf{W}}\mathcal{M}_k^{n,m}=\left\{AX^T+YB^T|X\in\mathbb{R}^{m\times k},Y\in\mathbb{R}^{n\times k}\right\}. \tag{5}$$

This is because any choice of matrices X,Y such that $X=\dot{B},Y=\dot{A}$ will give us some tangent vector, and for any tangent vector there exist such matrices. The space above is clearly a linear space. Being a tangent space to a manifold, it has the same dimension as the manifold: $(n+m)k-k^2$.

Recall the definition of the tangent space given in Lemma 1:

$$T_{\mathbf{W}}\mathcal{M}_k^{n,m}=\left\{\begin{bmatrix}A & A_\perp\end{bmatrix}\begin{bmatrix}M & N_1^T\\ N_2 & 0\end{bmatrix}\begin{bmatrix}B^T\\ B_\perp^T\end{bmatrix}:M\in\mathbb{R}^{k\times k},N_1\in\mathbb{R}^{(m-k)\times k},N_2\in\mathbb{R}^{(n-k)\times k}\right\}. \tag{6}$$

To prove Lemma 2, it is easy to verify by counting that the dimension of the space as defined in Equation (6) above is $(n+m)k-k^2$. Using the notation above, we can see that by taking $X=MB^T+N_1B_\perp^T$ and $Y=A_\perp N_2$, the space defined in Equation (6) is included in $T_{\mathbf{W}}\mathcal{M}_k^{n,m}$ as defined in Equation (5). Since it is a linear subspace of equal dimension, both spaces must be equal ■

Appendix B. Proof of Theorem 3

We state the theorem again here.

Theorem 8 *Let $W\in\mathcal{M}_k^{n,m}$, $W=AB^T$, and $W^\dagger=B^{\dagger T}A^\dagger$. Let $\xi\in T_W\mathcal{M}_k^{n,m}$, $\xi=\xi^{AB}+\xi^{AB_\perp}+\xi^{A_\perp B}$, as in 1, and let:*

$$V_1=W+\frac{1}{2}\xi^{AB}+\xi^{A_\perp B}-\frac{1}{8}\xi^{AB}W^\dagger\xi^{AB}-\frac{1}{2}\xi^{A_\perp B}W^\dagger\xi^{AB}\quad,$$
$$V_2=W+\frac{1}{2}\xi^{AB}+\xi^{AB_\perp}-\frac{1}{8}\xi^{AB}W^\dagger\xi^{AB}-\frac{1}{2}\xi^{AB}W^\dagger\xi^{AB_\perp}\quad.$$

The mapping

$$R_W(\xi)=V_1W^\dagger V_2 \tag{7}$$

is a second order retraction from a neighborhood $\Theta_W\subset T_W\mathcal{M}_k^{n,m}$ to $\mathcal{M}_k^{n,m}$.

Proof To prove that Equation (7) defines a retraction, we first show that $V_1W^\dagger V_2$ is a rank-k matrix. Note that there exist matrices $Z_1\in\mathbb{R}^{n\times k}$ and $Z_2\in\mathbb{R}^{m\times k}$ such that $V_1=Z_1B^T$ and , $V_2=AZ_2^T$. A sufficient condition for the matrices Z_1 and Z_2 to be of full rank is that the matrix M is of limited norm. Thus, for all tangent vectors lying in some neighborhood $\Theta_W\subset T_W\mathcal{M}_k^{n,m}$ of $0\in T_W\mathcal{M}_k^{n,m}$, the above relation is indeed a retraction to the manifold. In practice this is never a problem, as the set of matrices not of full rank is of zero measure, and in practice we have found these matrices to always be of full rank. Thus, $R_W(\xi)=V_1W^\dagger V_2=Z_1B^TB(B^TB)^{-1}(A^TA)^{-1}A^TAZ_2^T=Z_1Z_2^T$, which, given that Z_1 and Z_2 are of full column rank, is exactly a rank-k, $n\times m$ matrix.

Next we show that $R_W(\xi)$ is a retraction, and of second order. It is obvious that $R_W(0)=W$, since the projection of the zero vector is zero, and thus ξ^{AB}, $\xi^{AB_\perp}$ and $\xi^{A_\perp B}$ are all zero.

Expanding $V_1W^\dagger V_2$ up to second order terms in ξ, many terms cancel and we end up with:

$$\begin{aligned}R_W(\xi)&=W+\xi^{AB}+\xi^{AB_\perp}+\xi^{A_\perp B}+\xi^{A_\perp B}W^\dagger\xi^{AB_\perp}+O(\|\xi\|^3)\\&=W+\xi+\xi^{A_\perp B}W^\dagger\xi^{AB_\perp}+O(\|\xi\|^3).\end{aligned}$$

Local first order rigidity is immediately apparent. If we expand the only second order term, $\xi^{A_\perp B}W^\dagger\xi^{AB_\perp}$, we see that it equals $A_\perp N_2 N_1^T B_\perp^T$. We claim this term is orthogonal to the tangent space $T_W\mathcal{M}_k^{n,m}$. If we take, using the characterization in Lemma 2, an arbitrary tangent vector $A\tilde{M}B^T + A\tilde{N}_1^T B_\perp^T + A_\perp\tilde{N}_2 B^T$ in $T_{\mathbf{W}}\mathcal{M}_k^{n,m}$, we can calculate the inner product:

$$\begin{aligned}
&\left\langle \left(A_\perp N_2 N_1^T B_\perp^T\right), \left(A\tilde{M}B^T + A\tilde{N}_1^T B_\perp^T + A_\perp\tilde{N}_2 B^T\right)\right\rangle = \\
&tr\left(B_\perp N_1 N_2^T A_\perp^T A\tilde{M}B^T + B_\perp N_1 N_2^T A_\perp^T A\tilde{N}_1^T B_\perp^T + B_\perp N_1 N_2^T A_\perp^T A_\perp \tilde{N}_2 B^T\right) = \\
&tr\left(B_\perp N_1 N_2^T A_\perp^T A_\perp \tilde{N}_2 B^T\right) = \\
&tr\left(B^T B_\perp N_1 N_2^T A_\perp^T A_\perp \tilde{N}_2\right) = 0
\end{aligned}$$

with the equalities stemming from the fact that $A_\perp^T A = 0, B_\perp^T B = 0$, and from standard trace identities. Thus, the second order term cancels out if we project the second derivative of the curve defined by the retraction, as required by the second-order condition

$$P_W\left(\frac{\mathrm{d}R_W(\tau\xi)}{\mathrm{d}\tau^2}|_{\tau=0}\right) = 0 \quad \forall\xi\in T_W\mathcal{M}.$$

We see that the second order term is contained in the normal space. This concludes the proof that the retraction is a second order retraction. ■

Appendix C. Proof of Lemma 4

Let us see how can we calculate the needed terms explicitly. When evaluating the expression $V_1W^\dagger V_2$, we can use the algebraic relations: $WW^\dagger = P_A$ and $W^\dagger W = P_B$. From this we can conclude that: $WW^\dagger\xi^{AB} = \xi^{AB}$, $\xi^{AB}W^\dagger W = \xi^{AB}$, $\xi^{A_\perp B}W^\dagger W = \xi^{A_\perp B}$ and $WW^\dagger\xi^{AB_\perp} = \xi^{AB_\perp}$. These relations, along with many terms that cancel out, lead to the following expression:

$$\begin{aligned}
R_W(\xi) = V_1W^\dagger V_2 = &\\
&W + \xi^{AB} + \xi^{AB_\perp} + \xi^{A_\perp B} - \frac{1}{8}\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB} - \frac{3}{8}\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB_\perp} \\
&- \frac{3}{8}\xi^{A_\perp B}W^\dagger\xi^{AB}W^\dagger\xi^{AB} + \xi^{A_\perp B}W^\dagger\xi^{AB_\perp} - \xi^{A_\perp B}W^\dagger\xi^{AB}W^\dagger\xi^{AB_\perp} \\
&+ \frac{1}{16}\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB_\perp} + \frac{1}{16}\xi^{A_\perp B}W^\dagger\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB} \\
&+ \frac{1}{64}\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB} + \frac{1}{4}\xi^{A_\perp B}W^\dagger\xi^{AB}W^\dagger\xi^{AB}\xi^{AB_\perp}.
\end{aligned}$$

We now substitute the matrices M, N_1 and N_2 into the above relation. Most terms cancel out. For example, we have the identity $\xi^{AB}W^\dagger\xi^{AB} = AM^2B^T$, $\xi^{AB}W^\dagger\xi^{AB}W^\dagger\xi^{AB} = AM^3B^T$ and so forth. We obtain the following relation:

$$\begin{aligned}
R_W(\xi) = &AB^T + AMB^T + AN_1^T B_\perp^T + A_\perp N_2 B^T - \frac{1}{8}AM^3B^T \\
&- \frac{3}{8}AM^2N_1^T B_\perp^T - \frac{3}{8}A_\perp N_2 M^2 B^T + A_\perp N_2 N_1^T B_\perp^T - A_\perp N_2 M N_1^T B_\perp^T \\
&+ \frac{1}{16}AM^3N_1^T B_\perp^T + \frac{1}{16}A_\perp N_2 M^3 B^T + \frac{1}{64}AM^4B^T + \frac{1}{4}A_\perp N_2 M^2 N_1^T B_\perp^T.
\end{aligned}$$

Collecting terms by the leftmost and rightmost factors, we obtain:

$$\begin{aligned} R_W(\xi) = {} & A\left(I_k + M - \frac{1}{8}M^3 + \frac{1}{64}M^4\right)B^T \\ & + A\left(I_k - \frac{3}{8}M^2 + \frac{1}{16}M^3\right)N_1^T B_\perp^T \\ & + A_\perp N_2\left(I_k - \frac{3}{8}M^2 + \frac{1}{16}M^3\right)B^T \\ & + A_\perp N_2\left(I_k - M + \frac{1}{4}M^2\right)N_1^T B_\perp^T \quad . \end{aligned}$$

Finally, treating the first and fourth lines as a polynomial expression in M, and taking its polynomial square root, we can split the above sum into the product of an $n \times k$ matrix and a $k \times m$ matrix:

$$\begin{aligned} R_W(\xi) = {} & \left[A\left(I_k + \frac{1}{2}M - \frac{1}{8}M^2\right) + A_\perp N_2\left(I_k - \frac{1}{2}M\right)\right] \cdot \\ & \left[B\left(I_k + \frac{1}{2}M^T - \frac{1}{8}\left(M^T\right)^2\right) + B_\perp N_1\left(I_k - \frac{1}{2}M^T\right)\right]^T . \end{aligned}$$

Appendix D. Rank One Pseudoinverse Update Rule

For completeness we develop below the procedure for updating the pseudoinverse of a rank-1 perturbed matrix (Meyer, 1973), following the derivation of Petersen and Pedersen (2008). We wish to find a matrix G such that for a given matrix A along with its pseudo-inverse $A^\dagger$, and vectors of appropriate dimension c and d, we have:

$$\left(A + cd^T\right)^\dagger = A^\dagger + G.$$

We have used the fact that A has a full column rank to simplify slightly the algorithm of Petersen and Pedersen (2008).

References

P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. Technical Report UCL-INMA-2010.038, Department of Mathematical Engineering, Université catholique de Louvain, July 2010.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton Univ Press, 2008.

B. Bai, J. Weston, R. Collobert, and D. Grangier. Supervised semantic indexing. *Advances in Information Retrieval*, pages 761–765, 2009.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(1):937–965, 2006.

Algorithm 7 : Rank one pseudo-inverse update

Input: Matrices $A, A^\dagger \in \mathbb{R}_*^{n\times k}$, such that $A^\dagger$ is the pseudo-inverse of A, vectors $c \in \mathbb{R}^{n\times 1}$, $d \in \mathbb{R}^{k\times 1}$

Output: Matrix $Z^\dagger \in \mathbb{R}_*^{k\times n}$, such that $Z^\dagger$ is the pseudo-inverse of $A + cd^T$.

```
Compute:                                                                  matrix dimension
    $v = A^\dagger c$                                                              $k \times 1$
    $\beta = 1 + d^T v$                                                            $1 \times 1$
    $n = A^{\dagger T} d$                                                          $n \times 1$
    $\hat{n} = A^\dagger n$                                                        $k \times 1$
    $w = c - Av$                                                                   $n \times 1$
    if $\beta \neq 0$ AND $\|w\| \neq 0$
        $G = \frac{1}{\beta}\hat{n} w^T$                                           $k \times n$
        $s = \frac{\beta}{\|w\|^2\|n\|^2 + \beta^2}$                                $1 \times 1$
        $t = \frac{\|w\|^2}{\beta}\hat{n} + v$                                     $k \times 1$
        $\hat{G} = s \cdot t \left(\frac{\|n\|^2}{\beta} w + n\right)^T$           $k \times n$
        $G = G - \hat{G}$                                                          $k \times n$
    elseif $\beta = 0$ AND $\|w\| \neq 0$
        $G = -A^\dagger \frac{n}{\|n\|^2}$                                         $k \times 1$
        $G = Gn^T$                                                                 $k \times 1$
        $\hat{G} = v \frac{w^T}{\|w\|^2}$                                          $k \times n$
        $G = G - \hat{G}$                                                          $k \times n$
    elseif $\beta \neq 0$ AND $\|w\| = 0$
        $G = -\frac{1}{\beta} v n^T$                                               $k \times n$
    elseif $\beta = 0$ AND $\|w\| = 0$
        $\hat{v} = \frac{1}{\|v\|^2} v \left(v^T A^\dagger\right)$                 $k \times n$
        $\hat{n} = \frac{1}{\|n\|^2}\left(A^\dagger n\right) n^T$                  $k \times n$
        $G = \frac{v^T A^\dagger n}{\|v\|^2\|n\|^2} v n^T - \hat{v} - \hat{n}$     $k \times n$
    endif
    $Z^\dagger = A^\dagger + G$
```

J. Briët, F.M. de Oliveira Filho, and F. Vallentin. The Grothendieck problem with rank constraint. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems, MTNS*, 2010.

G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern*

Recognition, pages 248–255, 2009.

J. Deng, A. Berg, and L. Fei-Fei. Hierarchical Semantic Indexing for Large Scale Image Retrieval. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*, pages 785–792, 2011.

M.P. Do Carmo. *Riemannian Geometry*. Birkhauser, 1992.

L. Eldén and B. Savas. A Newton–Grassmann method for computing the best multi-linear rank-(r1, r2, r3) approximation of a tensor. *SIAM Journal on Matrix Analysis and applications*, 31(2): 248–271, 2009.

M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Electrical Engineering Department, Stanford University, 2002.

M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of the 2004 American Control Conference*, pages 3273–3278. IEEE, 2005.

A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, volume 18, page 451, 2006.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17, 2005.

D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1371–1384, 2008.

M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel. Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme. *SIAM Journal on Matrix Analysis and Applications*, 32(1):115–132, 2011.

P. Jain, B. Kulis, I.S. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems*, volume 20, pages 761–768, 2008.

P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, volume 24, pages 937–945, 2011.

M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-Rank Optimization on the Cone of Positive Semidefinite Matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010a.

M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11:517–553, 2010b.

S.M. Kakade, S. Shalev-Shwartz, and A. Tewari. Regularization techniques for learning with matrices, 2010. Arxiv preprint arXiv:0910.0610v2.

R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *The Journal of Machine Learning Research*, 99:2057–2078, 2010.

B. Kulis, M.A. Sustik, and I.S. Dhillon. Low-rank kernel learning with bregman matrix divergences. *The Journal of Machine Learning Research*, 10:341–376, 2009.

K. Lang. Learning to filter netnews. In *Proceeding of the 12th Internation Conference on Machine Learning*, pages 331–339, 1995.

C.D. Manning, P. Raghavan, H. Schutze, and Ebooks Corporation. *Introduction to Information Retrieval*, volume 1. Cambridge University Press Cambridge, UK, 2008.

R. Meka, P. Jain, C. Caramanis, and I.S. Dhillon. Rank minimization via online learning. In *Proceedings of the 25th International Conference on Machine learning*, pages 656–663, 2008.

C.D. Meyer. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

G. Meyer, S. Bonnabel, and R. Sepulchre. Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *The Journal of Machine Learning Research*, 12:593–625, 2011.

B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24 (2):227–234, 1995.

Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. In *Proceedings of the 27th International Conference on Machine Learning*, pages 823–830, 2010.

C. Oberlin and S.J. Wright. Active set identification in nonlinear programming. *SIAM Journal on Optimization*, 17(2):577–605, 2007.

K.B. Petersen and M.S. Pedersen. The matrix cookbook, Oct. 2008.

B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

S. Shalev-Shwartz, Y. Singer, and A.Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 94. ACM, 2004.

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online learning in the manifold of low-rank matrices. In *Advances in Neural Information Processing Systems 23*, pages 2128–2136. MIT Press, 2010.

B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5): 2553–2579, 2010.

B. Vandereycken, P.-A. Absil, and S. Vandewalle. Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*, pages 389–392. IEEE, 2009.

K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI–11)*, 2011.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, volume 15, pages 505–512. MIT Press, 2002.

L. Yang. An overview of distance metric learning. Technical report, School of Computer Science, Carnegie Mellon University, 2007.

Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, 1997.

Journal of Machine Learning Research 13 (2012) 459-489 Submitted 9/10; Revised 6/11; Published 2/12

Multi-Assignment Clustering for Boolean Data

Mario Frank[*] MFRANK@BERKELEY.EDU
UC Berkeley, Computer Science Division
721 Soda Hall
Berkeley, CA, 94720, USA

Andreas P. Streich[*] ANDREAS.STREICH@PHONAK.COM
Phonak AG, Advanced Concepts & Technologies
Laubisrütistrasse 28
8712 Stäfa, Switzerland

David Basin BASIN@INF.ETHZ.CH
Joachim M. Buhmann JBUHMANN@INF.ETHZ.CH
ETH Zürich, Department of Computer Science
Universitätstrasse 6
8092 Zürich, Switzerland

Editor: Tony Jebara

Abstract

We propose a probabilistic model for clustering Boolean data where an object can be simultaneously assigned to multiple clusters. By explicitly modeling the underlying generative process that combines the individual source emissions, highly structured data are expressed with substantially fewer clusters compared to single-assignment clustering. As a consequence, such a model provides robust parameter estimators even when the number of samples is low. We extend the model with different noise processes and demonstrate that maximum-likelihood estimation with multiple assignments consistently infers source parameters more accurately than single-assignment clustering. Our model is primarily motivated by the task of role mining for role-based access control, where users of a system are assigned one or more roles. In experiments with real-world access-control data, our model exhibits better generalization performance than state-of-the-art approaches.

Keywords: clustering, multi-assignments, overlapping clusters, Boolean data, role mining, latent feature models

1. Introduction

Clustering defines the unsupervised learning task of grouping a set of data items into subsets such that items in the same group are similar. While clustering data into disjoint clusters is conceptually simple, the exclusive assignment of data to clusters is often overly restrictive, especially when data is structured. In this work, we advocate a notion of clustering that is not limited to partitioning the data set. More generally, we examine the task of inferring the hidden structure responsible for generating the data. Specifically, multiple clusters can simultaneously generate a data item using

[*] These authors contributed equally. When most of this work was conducted, all authors were affiliated to ETH Zurich. This project may be found at `http://www.mariofrank.net/MACcode/index.html`.

a problem dependent link function. By adopting a generative viewpoint, such data originate from multiple sources.

Consider, for instance, individuals' movie preferences. A person might belong to the "comedy" cluster or the "classics" cluster, where each cluster membership generates a preference for the respective genre of movies. However, some people like both comedy movies and classics. In standard single-assignment clustering, a third "comedy&classics" cluster would be created for them. Under the generative viewpoint, we may assign individuals simultaneously to both of the original clusters to explain their preferences. Note that this differs from "fuzzy" clustering, where objects are partially assigned to clusters such that these fractional assignments (also called "mixed membership") add up to 1. In our approach, an object can be assigned to multiple clusters *at the same time*, that is, the assignments of an object can sum to a number larger than 1. Membership in a second cluster does not decrease the intensity of the membership in the first cluster. We call this approach *multi-assignment clustering* (MAC).

In a generative model that supports multi-assignments, one must specify how a combination of sources generates an object. In this paper, we investigate clustering for Boolean data. The combined emissions from individual sources generate an object by the Boolean OR operation. In the example of the movie preferences, this means that an individual belonging to both the comedy and the classics cluster likes a comedy film like "Ghostbusters" as much as someone from the comedy cluster, and likes the classic movie "Casablanca" as much as someone who only belongs to the classics group.

In this paper, we develop a probabilistic model for structured Boolean data. We examine various application-specific noise processes that account for the irregularities in the data and we theoretically investigate the relationships among these variants. Our experiments show that multi-assignment clustering computes more precise parameter estimates than state-of-the art clustering approaches. As a real-world application, our model defines a novel and highly competitive solution to the *role mining* problem. This task requires to infer a user-role assignment matrix and a role-permission assignment matrix from a Boolean user-permission assignment relation defining an access-control system. The generalization ability of our model in this domain outperforms other multi-assignment techniques.

The remainder of this paper is organized as follows. In the next section, we survey the literature on Boolean matrix factorization and the clustering of Boolean data. In Section 3, we derive our generative model and its variants and describe parameter inference in Section 4. In Section 5, we present experiments on synthetic and real-world data generated from multiple sources.

2. Related Work

In this section, we provide an overview of existing methods for the exploratory analysis of Boolean data. The described approaches have been developed within different research areas and have different objectives. However, they all aim to produce a structured representation of given binary data. The research areas include association-rule mining, formal concept analysis, clustering, dimension reduction, latent feature models, and database tiling. We distinguish between methods that search for an exact representation of the data and methods that approximate the representation. In the following, we review several related problem formulations and compare the approaches used to solve them.

2.1 Problem Formulations

There are different problem formulations that arise in the context of Boolean matrix factorization. In this section, we explain the most characteristic ones and relate them to each other.

2.1.1 EXACT BOOLEAN MATRIX DECOMPOSITION AND EQUIVALENT PROBLEMS

These methods aim at an exact Boolean factorization of the input matrix. The earliest formulation of such problems is presumably the set-cover problem (also called set basis problem) presented by Gimpel (1974) and Cormen et al. (2001).

Definition 1 *(**Set-Cover Problem**) Given a set of finite sets $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, find a basis $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_K\}$ with minimal cardinality K such that each $\mathbf{x}_i$ can be represented as a union of a subset of* $\mathbf{u}$.

All sets in $\mathbf{x}$ have a vector representation in a D-dimensional Boolean space, where a 1 at dimension d indicates the membership of item d in the respective set. D is the cardinality of the union of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$. The matrix $\mathbf{z} \in \{0,1\}^{N \times K}$ then indicates which subsets of $\mathbf{u}$ cover the sets in $\mathbf{x}$: $z_{ik} = 1$ indicates that $\mathbf{u}_k$ covers $\mathbf{x}_i$. Using this notation, the set-covering problem is equivalent to finding an exact Boolean decomposition of a binary matrix $\mathbf{x}$ with minimal K. An exact Boolean decomposition is $\mathbf{x} = \mathbf{z} * \mathbf{u}$, where the Boolean matrix product $*$ is defined such that

$$x_{id} = \bigvee_{k=1}^{K} (z_{ik} \wedge u_{kd}) \, . \tag{1}$$

Belohlavek and Vychodil (2010) show that the set cover problem is equivalent to Boolean factor analysis, where each factor corresponds to a row of $\mathbf{u}$. Keprt and Snásel (2004) show that the factors together with the objects assigned to them can, in turn, be regarded as formal concepts as defined in the field of Formal Concept Analysis (FCA) by Ganter and Wille (1999). Stockmeyer (1975) shows that the set-cover problem is NP-hard and the corresponding decision problem is NP-complete. Since the set-cover problem is equivalent to the other problems, this also holds for Boolean factor analysis, finding the exact Boolean decomposition of a binary matrix, and FCA. Approximation heuristics exist and are presented below.

2.1.2 APPROXIMATE BOOLEAN MATRIX DECOMPOSITION

An approximate decomposition of a matrix $\mathbf{x}$ is often more useful than an exact one. One can distinguish two problems, which we refer to as the lossy compression problem (LCP) and the structure inference problem (SIP). For LCP, two different formulations exist. In the first formulation of Miettinen et al. (2006), the size of the matrix $\mathbf{u}$ is fixed and the reconstruction error is to be minimized.

Definition 2 *(**LCP1: Minimal Deviation for given** K) For a given binary $N \times D$ matrix $\mathbf{x}$ and a given number $K < \min(N,D)$, find an $N \times K$ matrix $\mathbf{z}$ and a $K \times D$ matrix $\mathbf{u}$ such that the deviation $||\mathbf{x} - \mathbf{z} * \mathbf{u}||$ is minimal.*

Alternatively, the deviation is given, as in Vaidya et al. (2007), and the minimal $\mathbf{z}$ and $\mathbf{u}$ must be found to approximate $\mathbf{x}$.

Definition 3 *(**LCP2: Minimal** K **for Given Deviation**) For a given binary $N \times D$ matrix $\mathbf{x}$ and a given deviation δ, find the smallest number $K < min(N,D)$, a $N \times K$ matrix $\mathbf{z}$, and a $K \times D$ matrix $\mathbf{u}$ such that $||\mathbf{x} - \mathbf{z} * \mathbf{u}|| \leq \delta$.*

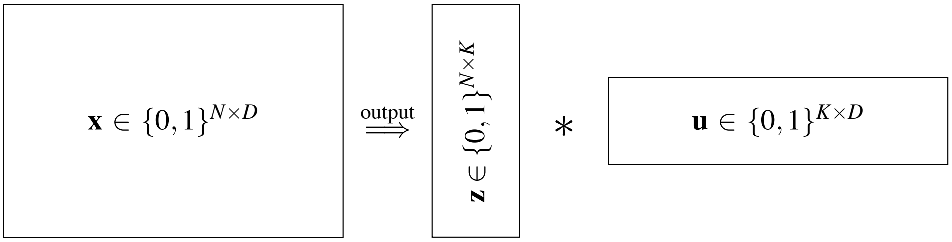

$\mathbf{x} \in \{0,1\}^{N\times D}$
output
$\Longrightarrow$
$\mathbf{z} \in \{0,1\}^{N\times K}$
$*$
$\mathbf{u} \in \{0,1\}^{K\times D}$

Figure 1: Dimensions of input data and output of the problems defined in Definitions 1–4.

The norm in both formulations of LCP is usually the Hamming distance. Both problems are NP-hard as shown by Vaidya et al. (2007).

In the structure inference problem (SIP), the matrix $\mathbf{x}$ is assumed to be generated from a structure part ($\mathbf{z}^*$,$\mathbf{u}^*$) and a random noise part. The goal is to find the decomposition ($\mathbf{z}^*$,$\mathbf{u}^*$) that recovers the structure and disregards the noise.

Definition 4 *(SIP) Let the binary $N \times D$ matrix $\mathbf{x}$ be given. Assuming that $\mathbf{x}$ was generated from a hidden structure $(\mathbf{z}^*,\mathbf{u}^*)$ and perturbed by noise Θ such that $\mathbf{x} \sim p(\mathbf{x}|\Theta,\mathbf{z}^* * \mathbf{u}^*)$, infer the underlying structure $(\mathbf{z}^*,\mathbf{u}^*)$.*

There is a substantial difference between SIP and the two lossy compression problems LCP1 and LCP2. Assuming that some of the entries are corrupted, neither the closest approximation of the original matrix nor the best compression is desirable. Instead, the goal is to infer a structure underlying the data at hand rather than to decompose the matrix with the highest possible accuracy. Since the structure of the data is repeated across the samples, whereas its noise is irregular, better structure recovery will also provide better prediction of new samples or missing observations.

2.2 Approaches

Depending on the problem formulation, there are several ways how the factorization problems are approached. In this section we provide an overview over related methods.

2.2.1 COMBINATORIAL APPROACHES

The problems LCP1 and LCP2 are NP-hard. Heuristic methods to find approximate solutions usually construct candidate sets for the rows of the matrix $\mathbf{u}$, and then greedily pick candidates such that, in each step, the reconstruction error is minimal. For the set covering problem defined in Cormen et al. (2001), the candidate set is the set of all possible formal concepts. For the approximate decomposition problem described in Miettinen et al. (2006), candidates are computed using association rule mining as presented in Agrawal et al. (1993). A predefined number of candidates is then iteratively chosen and assigned to the objects such that, in each step, the data set is optimally approximated. We will refer to this algorithm, originally presented in Miettinen et al. (2006), as the Discrete Basis Problem Solver (DBPS) and use Miettinen's implementation of DBPS in some of our experiments. In the greedy algorithm proposed in Belohlavek and Vychodil (2010), the construction of a large candidate set is avoided by iteratively constructing the next best candidate.

2.2.2 MODEL-BASED APPROACHES

Solutions to the structure inference problem as presented in Wood et al. (2006), Šingliar and Hauskrecht (2006), Kabán and Bingham (2008), and Streich et al. (2009) are often based on probabilistic models. The likelihood that is most similar to the one we propose is the noisy-OR gate introduced in Pearl (1988). Our model allows random flips in *both* directions. The noisy-OR model, which is constrained to random bit flips from zeros to ones, is thus a special case of the noise model that we present in Section 3.2.4. A detailed comparison of the relationship between the noisy-OR model and our approach follows in Section 3.

There are two models that use a noisy-OR likelihood. Noisy-OR component analysis (NOCA), as in Šingliar and Hauskrecht (2006), is based on a variational inference algorithm by Jaakkola and Jordan (1999). This algorithm computes the global probabilities $p(u_j = 1)$, but does not return a complete decomposition. A non-parametric model based on the Indian-Buffet process (Griffiths and Ghahramani, 2011) and a noisy-OR likelihood is presented in Wood et al. (2006). We call this approach infinite noisy-OR (INO). Our method differs from INO with respect to the data likelihood and with respect to optimization. While our model yields an exact solution to an approximate model, replacing the binary assignments by probabilistic assignments, the inference procedure for INO aims at solving the exact model by sampling. INO is a latent feature model, as described by Ghahramani et al. (2007), with Boolean features. Latent feature models explain data by combinations of multiple features that are indicated as active (or inactive) in a binary matrix $\mathbf{z}$. Being a member in multiple clusters (encoded in $\mathbf{z}$) is technically equivalent to having multiple features activated.

Binary independent component analysis (BICA) of Kabán and Bingham (2008) is a factor model for binary data. The combination of the binary factors is modeled with linear weights and thus deviates from the goal of finding binary decompositions as we defined it above. However, the method can be adapted to solve binary decomposition problems and performs well under certain conditions as we will demonstrate in Section 5.2.

Two other model-based approaches for clustering binary data are also related to our model, although more distantly. Kemp et al. (2006) presented a biclustering method that infers concepts in a probabilistic fashion. Each object and each feature is assigned to a single bicluster. A Dirichlet process prior (Antoniak, 1974; Ferguson, 1973) and a Beta-Bernoulli likelihood model the assignments of the objects. Heller and Ghahramani (2007) presented a Bayesian non-parametric mixture model including multiple assignments of objects to binary or real-valued centroids. When an object belongs to multiple clusters, the product over the probability distributions of all individual mixtures is considered, which corresponds to the conjunction of the mixtures. This constitutes a probabilistic model of the Boolean AND, whereas in all the above methods mentioned, as well as in our model, the data generation process uses the OR operation to combine mixture components.

In this paper, we provide a detailed derivation and an in-depth analysis of the model that we proposed in Streich et al. (2009). We thereby extend the noise part of the model to several variants and unify them in a general form. Moreover, we provide an approach for the model-order selection problem.

2.3 Applications

There are numerous applications for Boolean matrix factorization. In this paper we will focus on one specific application, the role mining problem, which was first formulated by Kuhlmann et al. (2003). This problem can be approached as a multi-assignment clustering problem since in role mining the associated data sets are clearly generated by multiple sources. Our model was motivated by this security-relevant problem. In the following, we will describe this problem and give representative examples of the main role mining methods that have been developed.

2.3.1 ROLE MINING AND RBAC

The goal of role mining is to automatically decompose a binary user-permission assignment matrix **x** into a role-based access control (RBAC) configuration consisting of a binary user-role assignment matrix **z** and a binary role-permission assignment matrix **u**. RBAC, as defined in Ferraiolo et al. (2001), is a widely used technique for administrating access-control systems where users access sensitive resources. Instead of directly assigning permissions to users, users are assigned to one or more roles (represented by the matrix **z**) and obtain the permissions contained in these roles (represented by **u**).

The major advantages of RBAC over a direct user-permission assignment (encoded in the matrix **x**) are ease of maintenance and increased security. RBAC simplifies maintenance for two reasons. First, roles can be associated with business roles, i.e. tasks in an enterprise. This business perspective on the user is more intuitive for humans than directly assigning individual low-level permissions. Second, assigning users to just a few roles is easier than assigning them to hundreds of individual permissions. RBAC increases security over access-control on a user-permission level because it simplifies the implementation and the audit of security policies. Also, it is less likely that an administrator wrongly assigns a permission to a user. RBAC is currently the access control solution of choice for many mid-size and large-scale enterprises.

2.3.2 STRUCTURE AND EXCEPTIONS IN ACCESS CONTROL MATRICES

The regularities of an access control matrix **x**, such as permissions that are assigned together to many users, constitute the structure of **x**. Exceptional user-permission assignments are not replicated over the users and thus do not contribute to the structure. There are three reasons for the existence of such exceptional assignments. First, exceptional assignments are often granted for 'special' tasks, for example if an employee temporarily substitutes for a colleague. Such exceptions may initially be well-motivated, but often the administrator forgets to remove them when the user no longer carries out the exceptional task. The second reason for exceptional assignments is simply administrative mistakes. Errors may happen when a new employee enters the company, or permissions might not be correctly updated when an employee changes position within the company. Finally, exceptional assignments can be intentionally granted to employees carrying out highly specialized tasks.

The role mining step should ideally migrate the regularities of the assignment matrix **x** to RBAC, while filtering out the remaining exceptional permission assignments. We model exceptional assignments (all three cases) with a noise model described in Section 3.2. We are not aware of any way to distinguish these three cases when only user-permission assignments are given as an input. However, being able to separate exceptional assignments from the structure in the data substantially eases the manual search for errors.

2.3.3 PRIOR ART

There is no consensus in the literature on the objective of role mining. An overview of all existing problem definitions is provided in Frank et al. (2010). We consider role mining as an inference problem, which we defined in Definition 4. Numerous algorithms for role mining exist. Molloy et al.

(2008) apply an algorithm from formal concept analysis (see Ganter and Wille, 1999) to construct candidate roles (rows in $\mathbf{u}$). The technique presented in Vaidya et al. (2007) uses an improved version of the database tiling algorithm from Han et al. (2000). In contrast to the method presented in Agrawal and Srikant (1994), their tiling algorithm avoids the construction of all concepts by using an oracle for the next best concept. A method based on a probabilistic model is proposed in Frank et al. (2008). The model is derived from the logical representation of a Boolean two-level hierarchy and is divided into several subcases. For one of the cases with only single assignments of objects, the bi-clustering method presented in Kemp et al. (2006) is used for inference.

3. Generative Model for Boolean Data from Multiple Sources

In this section we explain our model of the generation process of binary data, where data may be generated by multiple clusters. The observed data stems from an underlying structure that is perturbed by noise. We will first present our model for the structure and afterwards provide a unifying view on several noise processes presented in the literature.

We use a probabilistic model that describes the generative process. This has two advantages over discrete optimization approaches. First, considering a separate noise process for the irregularities of the data yields an interpretation for deviations between the input matrix $\mathbf{x}$ and its decomposition $(\mathbf{z},\mathbf{u})$. Second, the probabilistic representation of the sources $\mathbf{u}$ is a relaxation of the original computationally hard problem, as explained in the previous sections.

Let the observed data consist of N *objects*, each associated with D binary *dimensions*. More formally, we denote the data matrix by $\mathbf{x}$, with $\mathbf{x} \in \{0,1\}^{N\times D}$. We denote the i^{th} row of the matrix by x_{i*}, and the d^{th} column by x_{*d}. We use this notation for all matrices.

3.1 Structure Model

The systematic regularities of the observed data are captured by its structure. More specifically, the sources associated with the clusters generate the structure $\mathbf{x}^S \in \{0,1\}^{N\times D}$. The association of data items to sources is encoded in the binary assignment matrix $\mathbf{z} \in \{0,1\}^{N\times K}$, with $z_{ik}=1$ if and only if the data item i belongs to the source k, and $z_{ik}=0$ otherwise. The sum of the assignment variables for the data item i, $\sum_k z_{ik}$, can be larger than 1, which denotes that a data item i is assigned to multiple clusters. This multiplicity gives rise to the name *multi-assignment clustering* (MAC). The sources are encoded as rows of $\mathbf{u} \in \{0,1\}^{N\times K}$.

Let the set of the sources of an object be $\mathcal{L}_i := \{k \in \{1,\ldots,K\}\,|z_{ik}=1\}$. Let $\mathbb{L}$ be the set of all possible *assignment sets* and $\mathcal{L} \in \mathbb{L}$ be one such an assignment set. The value of x^S_{id} is a Boolean disjunction of the values at dimension d of all sources to which object i is assigned. The Boolean disjunction in the generation process of an x^S_{id} results in a probability for $x^S_{id}=1$, which is strictly non-decreasing in the number of associated sources $|\mathcal{L}_i|$: If any of the sources in $\mathcal{L}_i$ emits a 1 in dimension d, then $x^S_{id}=1$. Conversely, $x^S_{id}=0$ requires that all contributing sources have emitted a 0 in dimension d.

Let β_{kd} be the probability that source k emits a 0 at dimension d: $\beta_{kd} := p(u_{kd}=0)$. This parameter matrix $\beta \in [0,1]^{K\times D}$ allows us to simplify notation and to write

$$p_S\left(x^S_{id}=0\,|\,z_{i*},\beta\right) = \prod_{k=1}^{K} \beta_{kd}^{z_{ik}} \qquad \text{and } p_S\left(x^S_{id}=1\,|\,z_{i*},\beta\right) = 1 - p_S\left(x^S_{id}=0\,|\,z_{i*},\beta\right).$$

The parameter matrix β encodes these probabilities for all sources and dimensions. Employing the notion of assignment sets, one can interpret the product

$$\beta_{\mathcal{L}_i d} := \prod_{k=1}^{K} \beta_{kd}^{z_{ik}} \tag{2}$$

as the source of the assignment set $\mathcal{L}_i$. However, note that this interpretation differs from an actual single-assignment setting where $L := |\mathbb{L}|$ independent sources are assumed and must be inferred. Here, we only have $K \times D$ parameters β_{kd} whereas in single-assignment clustering, the number of source parameters would be $L \times D$, which can be up to $2^K \times D$. The expression $\beta_{\mathcal{L}_i d}$ is rather a 'proxy'-source, which we introduce just for notational convenience. The probability distribution of a x_{id} generated from this structure model given the assignments $\mathcal{L}_i$ and the sources β is then

$$p_S\left(x_{id}^S \mid \mathcal{L}_i, \beta\right) = \left(1 - \beta_{\mathcal{L}_i d}\right)^{x_{id}^S} \left(\beta_{\mathcal{L}_i d}\right)^{1-x_{id}^S}. \tag{3}$$

Note that we include the empty assignment set in the hypothesis class, i.e. a data item i need not belong to any class. The corresponding row x_{i*}^S contains only zeros and any element with the value 1 in the input matrix is explained by the noise process.

In the following sections, we describe various noise models that alter the output of the structure model. The structure part of the model together with a particular noise process is illustrated in Figure 2.

3.1.1 STRUCTURE COMPLEXITY AND SINGLE-ASSIGNMENT CLUSTERING

In the general case, which is when no restrictions on the assignment sets are given, there are $L = 2^K$ possible assignment sets. If the number of clusters to which an object can be simultaneously assigned is bounded by M, this number reduces to $L = \sum_{m=0}^{M} \binom{K}{m}$.

The particular case with $M = 1$ provides a model variant that we call *Single-Assignment Clustering* (SAC). In order to endow SAC with the same model complexity as MAC, we provide it with L clusters. Each of the assignment sets is then identified with one of the clusters. The clusters are treated (and, in particular, updated) independently of each other by computing the cluster parameters $\beta_{\mathcal{L}*}$ for each $\mathcal{L}$, discarding the dependencies in the original formulation. The underlying generative model of SAC, as well as the optimality conditions for its parameters, can be obtained by treating all assignment sets $\mathcal{L}$ independently in the subsequent equations. With all centroids computed according to Equation 2, the single-assignment clustering model yields the same probability for the data as the multi-assignment clustering model.

3.2 Noise Models and their Relationship

In this section, we first present the *mixture noise model*, which interprets the observed data as a mixture of independent emissions from the structure part and a noise source. Each bit in the matrix can thus be generated either by the structure model or by an independent global noise process. We then derive a more general formulation for this noise model. Starting there, we derive the *flip model*, where some randomly chosen bits of the signal matrix $\mathbf{x}^S$ are flipped, either from 0 to 1 or from 1 to 0. The noisy-OR model (Pearl, 1988) is a special case of the flip noise model, allowing only flips from 0 to 1.

The different noise models have different parameters. We denote the noise parameters of a model α by Θ_N^α. The full set of parameters for structure and noise is then $\Theta^\alpha := (\beta, \Theta_N^\alpha)$. As additional notation, we use the indicator function $\mathbf{I}_{\{p\}}$ for a predicate p, defined as

$$\mathbf{I}_{\{p\}} := \begin{cases} 1 & \text{if } p \text{ is true} \\ 0 & \text{otherwise .} \end{cases}$$

3.2.1 MIXTURE NOISE MODEL

In the mixture noise model, each x_{id} is generated either by the signal distribution or by a noise process. The binary indicator variable ξ_{id} indicates whether x_{id} is a noisy bit ($\xi_{id} = 1$) or a signal bit ($\xi_{id} = 0$). The observed x_{id} is then generated by

$$x_{id} = (1 - \xi_{id}) x_{id}^S + \xi_{id} x_{id}^N \, ,$$

where the generative process for the signal bit x_{id}^S is either described by the deterministic rule in Equation 1 or by the probability distribution in Equation 3. The noise bit x_{id}^N follows a Bernoulli distribution that is independent of object index i and dimension index d:

$$p_N\left(x_{id}^N \mid r\right) = r^{x_{id}^N} (1-r)^{1-x_{id}^N} \, . \tag{4}$$

Here, r is the parameter of the Bernoulli distribution indicating the probability of a 1. Combining the signal and noise distributions, the overall probability of an observed x_{id} is

$$p_M^{\text{mix}}(x_{id} \mid \mathcal{L}_i, \beta, r, \xi_{id}) = p_N(x_{id} \mid r)^{\xi_{id}} \, p_S(x_{id} \mid \mathcal{L}_i, \beta)^{1-\xi_{id}} \, . \tag{5}$$

We assume ξ_{id} to be Bernoulli distributed with a parameter $\varepsilon := p(\xi_{id} = 1)$ called the *noise fraction*. The joint probability of x_{id} and ξ_{id} given the assignment matrix $\mathbf{z}$ and all parameters is thus

$$p_M^{\text{mix}}(x_{id}, \xi \mid \mathbf{z}, \beta, r, \varepsilon) = p_M(x_{id} \mid \mathbf{z}, \beta, r, \xi) \cdot \varepsilon^{\xi_{id}} (1-\varepsilon)^{1-\xi_{id}} \, .$$

Since different x_{id} are conditionally independent given the assignments $\mathbf{z}$ and the parameters Θ^{mix}, we have

$$p_M^{\text{mix}}(\mathbf{x}, \xi \mid \mathbf{z}, \beta, r) = \prod_{id} p_M^{\text{mix}}(x_{id}, \xi \mid \mathbf{z}, \beta, r) \, .$$

The noise indicators ξ_{id} cannot be observed. We therefore marginalize out all ξ_{id} to derive the probability of $\mathbf{x}$ as

$$\begin{aligned} p_M^{\text{mix}}(\mathbf{x} \mid \mathbf{z}, \beta, r, \varepsilon) &= \sum_{\{\xi\}} p_M^{\text{mix}}(\mathbf{x}, \xi \mid \mathbf{z}, \beta, r, \varepsilon) \\ &= \prod_{id} \left(\varepsilon \cdot p_N(x_{id}) + (1-\varepsilon) \cdot p_S(x_{id})\right) \, . \end{aligned}$$

The observed data $\mathbf{x}$ is thus a mixture between the emissions of the structure part (which has weight $1-\varepsilon$) and the noise emissions (with weight ε). Introducing the auxiliary variable

$$q_{\mathcal{L}_i d}^{\text{mix}} := p_M^{\text{mix}}(x_{id} = 1 \mid \mathbf{z}, \beta, r, \varepsilon) = \varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}_i d})$$

to represent the probability that $x_{id} = 1$ under this model, we get a data-centric representation of the probability of $\mathbf{x}$ as

$$p_M^{\text{mix}}(\mathbf{x} \mid \mathbf{z}, \beta, r, \varepsilon) = \prod_{id} \left(x_{id} \, q_{\mathcal{L}_i d}^{\text{mix}} + (1 - x_{id})\left(1 - q_{\mathcal{L}_i d}^{\text{mix}}\right)\right) \, . \tag{6}$$

The parameters of the mixture noise model are $\Theta_N^{\text{mix}} := (\varepsilon, r)$. Since ε and r are independent of d and i, we will refer to ε and r as parameters of a ‘global’ noise process.

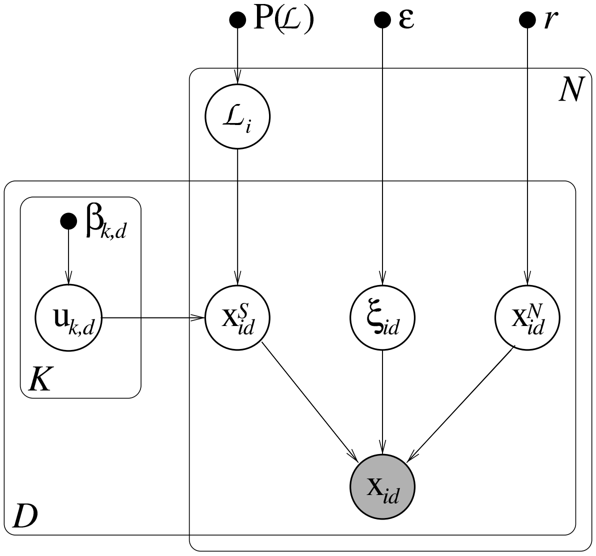


Figure 2: The generative model of Boolean MAC with mixture noise. $\mathcal{L}_i$ is the assignment set of object i, indicating which Boolean sources from $\mathbf{u}$ generated it. The bit ξ_{id} selects whether the noise-free bit x^S_{id} or the noise bit x^N_{id} is observed.

3.2.2 GENERALIZED NOISE MODEL

In this section, we generalize the mixture noise model presented above. Doing so, we achieve a generalized formulation that covers, among others, the mentioned noisy-OR model.

The overall generation process has two steps:

1. The signal part of the data is generated according to the sources, as described in Section 3.1. It is defined by the probability $p_S\left(x^S_{id} \mid \mathcal{L}_i, \beta\right)$ (Equation 3).
2. A noise process acts on the signal $\mathbf{x}^S$ and thus generates the observed data matrix $\mathbf{x}$. This noise process is described by the probability $p^\alpha(x_{id}|x^S_{id}, \Theta^\alpha_N)$, where α identifies the noise model and Θ^α_N are the parameters of the noise model α.

The overall probability of an observation x_{id} given all parameters is thus

$$p^\alpha_M\left(x_{id}|\mathcal{L}_i, \beta, \Theta^\alpha_N\right) = \sum_{x^S_{id}} p_S\left(x^S_{id} \mid \mathcal{L}_i, \beta\right) \cdot p^\alpha\left(x_{id}|x^S_{id}, \Theta^\alpha_N\right) \ .$$

3.2.3 MIXTURE NOISE MODEL

The mixture noise model assumes that each x_{id} is explained either by the structure model or by an independent global noise process. Therefore, the joint probability of $p^{\text{mix}}\left(x_{id}|x^S_{id}, \Theta^{\text{mix}}_N\right)$ can be factored as

$$p^{\text{mix}}\left(x_{id}|x^S_{id}, \Theta^{\text{mix}}_N\right) = p^{\text{mix}}_M\left(x_{id}|x^S_{id}, x^N_{id}, \xi_{id}\right) \cdot p^{\text{mix}}_N(x^N_{id}|r) \ ,$$

with

$$p^{\text{mix}}_M\left(x_{id}|x^S_{id}, x^N_{id}, \xi_{id}\right) = \left(\mathbf{I}_{\{x^S_{id}=x_{id}\}}\right)^{1-\xi_{id}} \left(\mathbf{I}_{\{x^N_{id}=x_{id}\}}\right)^{\xi_{id}} \ .$$

$p^S(x^S_{id}|\mathcal{L}_i,\beta)$ and $p^{\text{mix}}_N(x^N_{id}|r)$ are defined by Equation 3 and Equation 4 respectively. Summing out the unobserved variables x^S_{id} and x^N_{id} yields

$$\begin{aligned}
p^{\text{mix}}_M\left(x_{id}|\mathcal{L}_i,\beta,r,\xi_{id}\right) &= \sum_{x^S_{id}=0}^{1}\sum_{x^N_{id}=0}^{1} p^{\text{mix}}_M\left(x_{id},x^S_{id},x^N_{id}|\mathcal{L}_i,\beta,r,\xi_{id}\right)\\
&= p_S\left(x_{id}|\mathcal{L}_i,\beta\right)^{1-\xi_{id}}\cdot p^{\text{mix}}_N\left(x_{id}|r\right)^{\xi_{id}}\\
&= (1-\xi_{id})p_S\left(x_{id}|\mathcal{L}_i,\beta\right)+\xi_{id}p^{\text{mix}}_N\left(x_{id}|r\right)\ .
\end{aligned}$$

Integrating out the noise indicator variables ξ_{id} leads to the same representation as in Equation 5.

3.2.4 FLIP NOISE MODEL

In contrast to the previous noise model, where the likelihood is a mixture of *independent* noise and signal distributions, the flip noise model assumes that the effect of the noise depends on the signal itself. The data is generated from the same signal distribution as in the mixture noise model. Individual bits are then randomly selected and flipped. Formally, the generative process for a bit x_{id} is described by

$$x_{id} = x^S_{id}\oplus\xi_{id}\ ,$$

where $\oplus$ denotes addition modulo 2. Again, the generative process for the structure bit x^S_{id} is described by either Equation 1 or Equation 3. The value of ξ_{id} indicates whether the bit x^S_{id} is to be flipped ($\xi_{id}=1$) or not ($\xi_{id}=0$). In a probabilistic formulation, we assume that the indicator ξ_{id} for a bit-flip is distributed according to $\xi_{id}\sim p(\xi_{id}|x^S_{id},\varepsilon_0,\varepsilon_1)$. Thus, the probability of a bit-flip, given the signal and the noise parameters $(\varepsilon_0,\varepsilon_1)$, is

$$p(\xi_{id}|x^S_{id},\varepsilon_0,\varepsilon_1) = \left(\varepsilon_1^{x^S_{id}}\varepsilon_0^{1-x^S_{id}}\right)^{\xi_{id}}\left((1-\varepsilon_1)^{x^S_{id}}(1-\varepsilon_0)^{1-x^S_{id}}\right)^{1-\xi_{id}}\ ,$$

with the convention that $0^0=1$. Given the flip indicator ξ_{id} and the signal bit x^S_{id}, the final observation is deterministic:

$$p^{\text{flip}}_M(x_{id}|\xi_{id},x^S_{id}) = x_{id}^{\mathbf{I}_{\{\xi_{id}\neq x^S_{id}\}}}\left(1-x_{id}\right)^{\mathbf{I}_{\{\xi_{id}=x^S_{id}\}}}\ .$$

The joint probability distribution is then given by

$$p^{\text{flip}}\left(x_{id}|x^S_{id},\Theta^{\text{flip}}_N\right) = \sum_{\xi_{id}=0}^{1} p^{\text{flip}}_M(x_{id}|\xi_{id},x^S_{id})\cdot p(\xi_{id}|x^S_{id},\varepsilon_0,\varepsilon_1)\ .$$

3.2.5 RELATION BETWEEN THE NOISE PARAMETERS

Our unified formulation of the noise models allows us to compare the influence of the noise processes on the clean signal under different noise models. We derive the parameters of the flip noise model that is equivalent to a given mixture noise model based on the probabilities $p^{\text{mix}}(x_{id}|x^S_{id},\Theta^{\alpha}_N)$ and $p^{\text{flip}}(x_{id}|x^S_{id},\Theta^{\alpha}_N)$, for the cases $(x_{id}=1,x^S_{id}=0)$ and $(x_{id}=0,x^S_{id}=1)$:

The mixture noise model with $\Theta^{\text{mix}}_N=(\varepsilon,r)$ is equivalent to the flip noise model with $\Theta^{\text{flip}}_N=(\varepsilon\cdot r,\varepsilon\cdot(1-r))$. Conversely, we have that the flip noise model with $\Theta^{\text{flip}}_N=(\varepsilon_0,\varepsilon_1)$ is equivalent to the **mixture noise model** with $\Theta^{\text{mix}}_N=\left(\varepsilon_0+\varepsilon_1,\frac{\varepsilon_0}{\varepsilon_0+\varepsilon_1}\right)$.

Hence the two noise-processes are just different representations of the same process. We therefore use only the mixture noise model in the remainder of this paper and omit the indicator α to differentiate between the different noise models.

3.2.6 OBJECT-WISE AND DIMENSION-WISE NOISE PROCESSES

In the following, we extend the noise model presented above. Given the equivalence of mix and flip noise, we restrict ourselves to the mixture noise model.

Dimension-wise Noise. Assume a separate noise process for every dimension d, which is parameterized by r_d and has intensity ε_d. We then have

$$p(\mathbf{x} \,|\, \mathbf{z}, \beta, \varepsilon) = \prod_{i,d} \left(\varepsilon_d r_d^{x_{id}} (1 - r_d)^{1 - x_{id}} + (1 - \varepsilon_d)(1 - \beta_{\mathcal{L}_i d})^{x_{id}} \beta_{\mathcal{L}_i d}^{1 - x_{id}} \right) .$$

Object-wise Noise. Now assume a separate noise process for every object i, which is parameterized by ε_i and r_i. As before, we have

$$p(\mathbf{x} \,|\, \mathbf{z}, \beta, \varepsilon) = \prod_{i,d} \left(\varepsilon_i r_i^{x_{id}} (1 - r_i)^{1 - x_{id}} + (1 - \varepsilon_i)(1 - \beta_{\mathcal{L}_i d})^{x_{id}} \beta_{\mathcal{L}_i d}^{1 - x_{id}} \right) .$$

Note that these local noise models are very specific and could be used in the following application scenarios. In role mining, some permissions are more critical than others. Hence it appears reasonable to assume a lower error probability for the dimension representing, for example, root access to a central database server than for the dimension representing the permission to change the desktop background image. However we observed experimentally that the additional freedom in these models often leads to an over-parametrization and thus worse overall results. This problem could possibly be reduced by introducing further constraints on the parameters, such as a hierarchical order.

4. Inference

We now describe an inference algorithm for our model. While the parameters are ultimately inferred according to the maximum likelihood principle, we use the optimization method of *deterministic annealing* presented in Buhmann and Kühnel (1993) and Rose (1998). In the following, we specify the deterministic annealing scheme used in the algorithm. In Section 4.2 we then give the characteristic magnitudes and the update conditions in a general form, independent of the noise model. The particular update equations for the mixture model are then derived in detail in Section 4.3.

4.1 Annealed Parameter Optimization

The likelihood of a data matrix $\mathbf{x}$ (Equation 6) is highly non-convex in the model parameters and a direct maximization of this function will likely be trapped in local optima. Deterministic annealing is an optimization method that parameterizes a smooth transition from the convex problem of maximizing the entropy (i.e. a uniform distribution over all possible clustering solutions) to the problem of minimizing the empirical risk R. The goal of this heuristic is to reduce the risk of being trapped in a local optimum. Such methods are also known as continuation methods (see Allgower and Georg, 1980). In our case, R is the negative log likelihood. Formally, the Lagrange functional

$$F := -T \log Z = \mathbb{E}_G [R] - TH$$

is introduced, with Z being the *partition function* over all possible clustering solutions (see Equation 10), and G denotes the Gibbs distribution (see Equation 9 and Equation 8). The Lagrange parameter T (called the *computational temperature*) controls the trade-off between entropy maximization and minimization of the empirical risk. Minimizing F at a given temperature T is equivalent to constraint minimization of the empirical risk R with a lower limit on the entropy H. In other words, H is a uniform prior on the likelihood of the clustering solutions. Its weight decreases as the computational temperature T is incrementally reduced.

At every temperature T, a gradient-based expectation-maximization (EM) step computes the parameters that minimize F. The E-step computes the risks $R_{i\mathcal{L}}$ (Equation 7) of assigning data item i to the assignment set $\mathcal{L}$. The corresponding responsibilities $\gamma_{i\mathcal{L}}$ (Equation 8) are computed for all i and $\mathcal{L}$ based on the current values of the parameters. The M-step first computes the optimal values of the noise parameters. Then it uses these values to compute the optimal source parameters β. The individual steps are described in Section 4.3.

We determine the initial temperature as described in Rose (1998) and use a constant cooling rate ($T \leftarrow \vartheta \cdot T$, with $0 < \vartheta < 1$) . The cooling is continued until the responsibilities $\gamma_{i\mathcal{L}}$ for all data items i peak sharply at single assignment sets $\mathcal{L}_i$.

4.2 Characteristic Magnitudes and Update Conditions

Following our generative approach to clustering, we aim at finding the maximum likelihood solution for the parameters. Taking the logarithm of the likelihood simplifies the calculations as products become sums. Also, the likelihood function conveniently factors over the objects and features enabling us to investigate the risk of objects individually. We define the *empirical risk* of assigning an object i to the set of clusters $\mathcal{L}$ as the negative log-likelihood of the feature vector x_{i*} being generated by the sources contained in $\mathcal{L}$:

$$R_{i\mathcal{L}} := \log p(x_{i\cdot}|\mathcal{L}_i,\Theta) = -\sum_d \log\left(x_{id}\left(1-q_{\mathcal{L}d}\right)+(1-x_{id})q_{\mathcal{L}d}\right) \ . \tag{7}$$

The *responsibility* $\gamma_{i\mathcal{L}}$ of the assignment-set $\mathcal{L}$ for data item i is given by

$$\gamma_{i\mathcal{L}} := \frac{\exp\left(-R_{i\mathcal{L}}/T\right)}{\sum_{\mathcal{L}'\in\mathbb{L}}\exp\left(-R_{i\mathcal{L}'}/T\right)} \ . \tag{8}$$

The matrix γ defines a probability distribution over the space of all clustering solutions. The expected *empirical risk* $\mathbb{E}_G\left[R\right]$ of the solutions under this probability distribution G is

$$\mathbb{E}_G\left[R_{i\mathcal{L}}\right] = \sum_i\sum_{\mathcal{L}}\gamma_{i\mathcal{L}}R_{i\mathcal{L}} \ . \tag{9}$$

Finally, the *state sum* Z and the *free energy* F are defined as follows.

$$Z := \prod_i\sum_{\mathcal{L}}\exp\left(-R_{i\mathcal{L}}/T\right) \tag{10}$$

$$F := -T\log Z = -T\sum_i\log\left(\sum_{\mathcal{L}}\exp\left(-R_{i\mathcal{L}}/T\right)\right)$$

Given the above, we derive the updates of the model parameters based on the first-order condition of the free energy F. We therefore introduce the generic model parameter θ, which stands for any of the model parameters, i.e. $\theta \in \left\{\beta_{\mu\nu}, \varepsilon_0, \varepsilon_1, \varepsilon, r\right\}$. Here, μ is some particular value of source index k and ν is some particular value of dimension index d. Using this notation, the derivative of the free energy with respect to θ is given by

$$\frac{\partial F}{\partial \theta} = \sum_i \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} \frac{\partial R_{i\mathcal{L}}}{\partial \theta} = \sum_i \sum_{\mathcal{L}} \gamma_{i\mathcal{L}} \sum_d \frac{(1-2x_{id}) \frac{\partial q_{\mathcal{L}d}}{\partial \theta}}{x_{id}\,(1-q_{\mathcal{L}d}) + (1-x_{id}) q_{\mathcal{L}d}} \; .$$

4.3 Update Conditions for the Mixture Noise Model

Derivatives for the mixture noise model ($\theta \in \left\{\beta_{\mu\nu}, \varepsilon, r\right\}$) are:

$$\frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial \beta_{\mu\nu}} = (1-\varepsilon)\, \beta_{\mathcal{L}\setminus\{\mu\},d}\, \mathbf{I}_{\{\nu=d\}} \mathbf{I}_{\{\mu\in\mathcal{L}\}}, \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial \varepsilon} = 1 - r - \beta_{\mathcal{L}d}, \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L}d}}{\partial r} = -\varepsilon.$$

This results in the following first-order conditions for the mixture noise model:

$$\frac{\partial F^{\text{mix}}}{\partial \beta_{\mu\nu}} = (1-\varepsilon) \sum_{\mathcal{L}\mu\in\mathcal{L}} \beta_{\mathcal{L}\setminus\{\mu\},\nu} \left\{ \frac{\sum_{i:x_{i\nu}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}\nu})} - \frac{\sum_{i:x_{i\nu}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1 - \varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}\nu})} \right\} = 0,$$

$$\frac{\partial F^{\text{mix}}}{\partial \varepsilon} = \sum_d \left\{ \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L}d}) \sum_{i:x_{id}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}d})} - \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L}d}) \sum_{i:x_{id}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1-\varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}d})} \right\} = 0,$$

$$\frac{\partial F^{\text{mix}}}{\partial r} = \varepsilon \sum_d \left\{ \sum_{\mathcal{L}} \frac{\sum_{i:x_{id}=0} \gamma^{\text{mix}}_{i\mathcal{L}}}{1-\varepsilon r - (1-\varepsilon)(1-\beta_{\mathcal{L}d})} - \sum_{\mathcal{L}} \frac{\sum_{i:x_{id}=1} \gamma^{\text{mix}}_{i\mathcal{L}}}{\varepsilon r + (1-\varepsilon)(1-\beta_{\mathcal{L}d})} \right\} = 0.$$

There is no analytic expression for the solutions of the above equations, the parameters $\beta_{\mu\nu}$, ε, and r are thus determined numerically. In particular, we use Newton's method to determine the optimal values for the parameters. We observed that this method rapidly converges, usually needing at most 5 iterations.

The above equations contain the optimality conditions for the single-assignment clustering (SAC) model as a special case. As only assignment sets $\mathcal{L}$ with one element are allowed in this model, we can globally substitute $\mathcal{L}$ by k and get $\beta_{\mathcal{L}*} = \beta_{k*}$. Furthermore, since 1 is the neutral element for multiplication, we get $\beta_{\mathcal{L}\setminus\{\mu\},\nu} = 1$.

In the noise-free case, the value for the noise fraction is $\varepsilon = 0$. This results in a significant simplification of the update equations.

5. Experiments

In this section, we first introduce the measures that we employ to evaluate the quality of clustering solutions. Afterwards, we present results on both synthetic and real-world data.

5.1 Evaluation Criteria

For synthetic data, we evaluate the estimated sources by their Hamming distance to the true sources being used to generate the data. For real-world data, the appropriate evaluation criteria depend on the task. Independent of the task, the generalization ability of a solution indicates how well the solution fits to the unknown underlying probability distribution of the data. Moreover, as argued in Frank et al. (2010), the ability of a solution to generalize to previously unseen users is the appropriate quality criterion for the role mining problem. In the following, we introduce these two measures, parameter mismatch and generalization ability.

The following notation will prove useful. We denote by $\hat{\mathbf{z}}$ and $\hat{\mathbf{u}}$ the estimated decomposition of the matrix $\mathbf{x}$. The reconstruction of the matrix based on this decomposition is denoted by $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} := \hat{\mathbf{z}} * \hat{\mathbf{u}}$. Furthermore, in experiments with synthetic data, the signal part of the matrix is known. As indicated in Section 3, it is denoted by $\mathbf{x}^S$.

5.1.1 PARAMETER MISMATCH

Experiments with synthetic data allow us to compare the values of the true model parameters with the inferred model parameters. We report below on the accuracies of both the estimated centroids $\hat{\mathbf{u}}$ and the noise parameters.

To evaluate the accuracy of the centroid estimates, we use the average Hamming distance between the true and the estimated centroids. In order to account for the arbitrary numbering of clusters, we permute the centroid vectors u_{k*} with a permutation $\pi(k)$ such that the estimated and the true centroids agree best. Namely,

$$a(\hat{\mathbf{u}}) := \frac{1}{K \cdot D} \min_{\pi \in P_K} \sum_{k=1}^{K} \left|\left| u_{k*} - \hat{u}_{\pi(k)*} \right|\right| \ ,$$

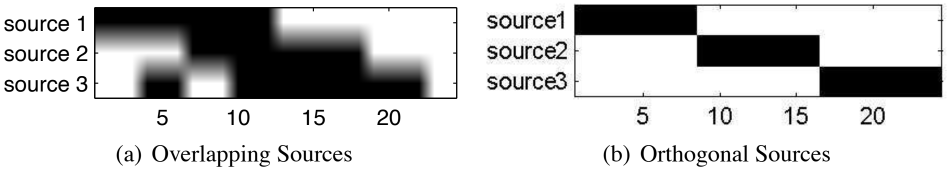
where P_K denotes the set of all permutations of K elements. Finding the $\pi \in P_K$ that minimizes the Hamming distance involves solving the assignment problem, which can be calculated in polynomial time using the Hungarian algorithm of Kuhn (2010). Whenever we know the true model parameters, we will assess methods based on parameter mismatch, always reporting this measure in percent.

5.1.2 GENERALIZATION ERROR

For real world data, the true model parameters are unknown and there might even exist a model mismatch between the learning model and the true underlying distribution that generated the input data set $\mathbf{x}^{(1)}$. Still, one can measure how well the method infers this distribution by testing if the estimated distribution generalizes to a second data set $\mathbf{x}^{(2)}$ that has been generated in the same way as $\mathbf{x}^{(1)}$. To measure this generalization ability, we first randomly split the data set along the objects into a training set $\mathbf{x}^{(1)}$ and a validation set $\mathbf{x}^{(2)}$. Then we learn the factorization $\hat{\mathbf{z}}$, $\hat{\mathbf{u}}$ based on the training set and transfer it to the validation set.

Note that the transfer of the learned solution to the validation set is not as straight-forward in such an unsupervised scenario as it is in classification. For transferring, we use the method proposed by Frank et al. (2011). For each object i in $\mathbf{x}^{(2)}$, we compute its nearest neighbor $\psi_{NN}(i)$ in $\mathbf{x}^{(1)}$ according to the Hamming distance. We then create a new matrix $\mathbf{z}'$ defined by $\mathbf{z}'_{i*} = \hat{\mathbf{z}}_{\psi_{NN}(i)*}$ for all i. As a consequence, each validation object is assigned to the same set of sources as its nearest neighbor in the training set. The possible assignment sets as well as the source parameters are thereby restricted to those that have been trained without seeing the validation data. The generalization error is then

$$G(\hat{\mathbf{z}}, \hat{\mathbf{u}}, \mathbf{x}^{(2)}, \psi_{NN}) := \frac{1}{N^{(2)} \cdot D} \left|\left| \mathbf{x}^{(2)} - \mathbf{z}' * \hat{\mathbf{u}} \right|\right| ,$$
$$\text{with } \mathbf{z}' = \left(\hat{\mathbf{z}}_{\psi_{NN}(1)*}, \hat{\mathbf{z}}_{\psi_{NN}(2)*}, \ldots, \hat{\mathbf{z}}_{\psi_{NN}(N^{(2)})*} \right)^T \ ,$$



(a) Overlapping Sources (b) Orthogonal Sources

Figure 3: Overlapping sources (left) and orthogonal sources (right) used in the experiments with synthetic data. Black indicates a 1 and white a 0 for the corresponding matrix element. In both cases, the three sources have 24 dimensions.

where $N^{(2)}$ is the number of objects in the validation data set and $*$ is the Boolean matrix product as defined in Equation 1. This measure essentially computes the fraction of wrongly predicted bits in the new data set.

As some of the matrix entries in $\mathbf{x}^{(2)}$ are interpreted as noise, it might be impossible to reach a generalization error of 0%. However, this affects all methods and all model variants. Moreover, we are ultimately interested in the total order of models with respect to this measure and not in their absolute scores. Since we assume that the noise associated with the features of different objects is independent, we deduce from a low generalization error that the algorithm can infer sources that explain—up to residual noise—the features of new objects from the same distribution. In contrast, a high generalization error implies that the inferred sources wrongly predict most of the matrix entries and thus indicates overfitting.

Note that the computation of generalization error differs from the approach taken in Streich et al. (2009). There, only $\hat{\mathbf{u}}$ is kept fixed, and $\hat{\mathbf{z}}$ is ignored when computing the generalization error. The assignment sets $\mathbf{z}'$ of the new objects are recomputed by comparing all source combinations with a fraction κ of the bits of these objects. The generalization error is the difference of the remaining $(1-\kappa)$ bits to the assigned sources. In our experiments on model-order selection, this computation of generalization error led to overfitting. As $\mathbf{z}'$ was computed independently from $\hat{\mathbf{z}}$, fitting all possible role combinations to the validation data, it supports tuning one part of the solution to this data. With the nearest neighbor-based transfer of $\hat{\mathbf{z}}$, which is computed without using the validation set, this is not possible. Overfitting is therefore detected more reliably than in Streich et al. (2009).

In order to estimate the quality of a solution, we use parameter mismatch in experiments with synthetic data and generalization error in experiments with real data.

5.2 Experiments on Synthetic Data

This section presents results from several experiments on synthetic data where we investigate the performance of different model variants and other methods. Our experiments have the following setting in common. First, we generate data by assigning objects to one or more Boolean vectors out of a set of predefined sources. Unless otherwise stated, we will use the generating sources as depicted in Figure 3. Combining the emissions of these sources via the *OR* operation generates the structure of the objects. Note that the sources can overlap, i.e. multiple sources emit a 1 at a particular dimension. In a second step, we perturb the data set by a noise process.

With synthetic data, we control all parameters, namely the number of objects and sources, the geometry of the Boolean source vectors (i.e. we vary them between overlapping sources and orthogonal sources), the fraction of bits that are affected by the noise process, and the kind of noise process. Knowing the original sources used to generate the data set enables us to measure the accuracy of the estimators, as described in Section 5.1. The goal of these experiments is to investigate the behavior of different methods under a wide range of conditions. The results will help us in interpreting the results on real-world data in the next section.

We repeat all experiments ten times, each time with different random noise. We report the median (and 65% percentiles) of the accuracy over these ten runs.

5.2.1 COMPARISON OF MAC WITH OTHER CLUSTERING TECHNIQUES

The main results of the comparison between MAC and other clustering techniques are shown in Figure 4. Each panel illustrates the results of one of the methods under five different experimental setups. We generate 50 data items from each single source as well as from each combination of two sources. Furthermore, 50 additional data items are generated without a source, i.e. they contain no structure. This experimental setting yields 350 data items in total. The overlapping sources are used as shown in Figure 3(a), and the structure is randomly perturbed by a mixture noise process. The probability of a noisy bit being 1 is kept fixed at $r = 0.5$, while the fraction of noisy bits, ε, varies between 0% and 99%. The fraction of data from multiple sources is 50% for the experiments plotted with square markers. Experiments with only 20% (80%) of the data are labeled with circles (with stars). Furthermore, we label experiments with orthogonal sources (Figure 3(b)) with 'x'. Finally, we use '+' labels for results on data with a noisy-OR noise process, i.e. $r = 1$.

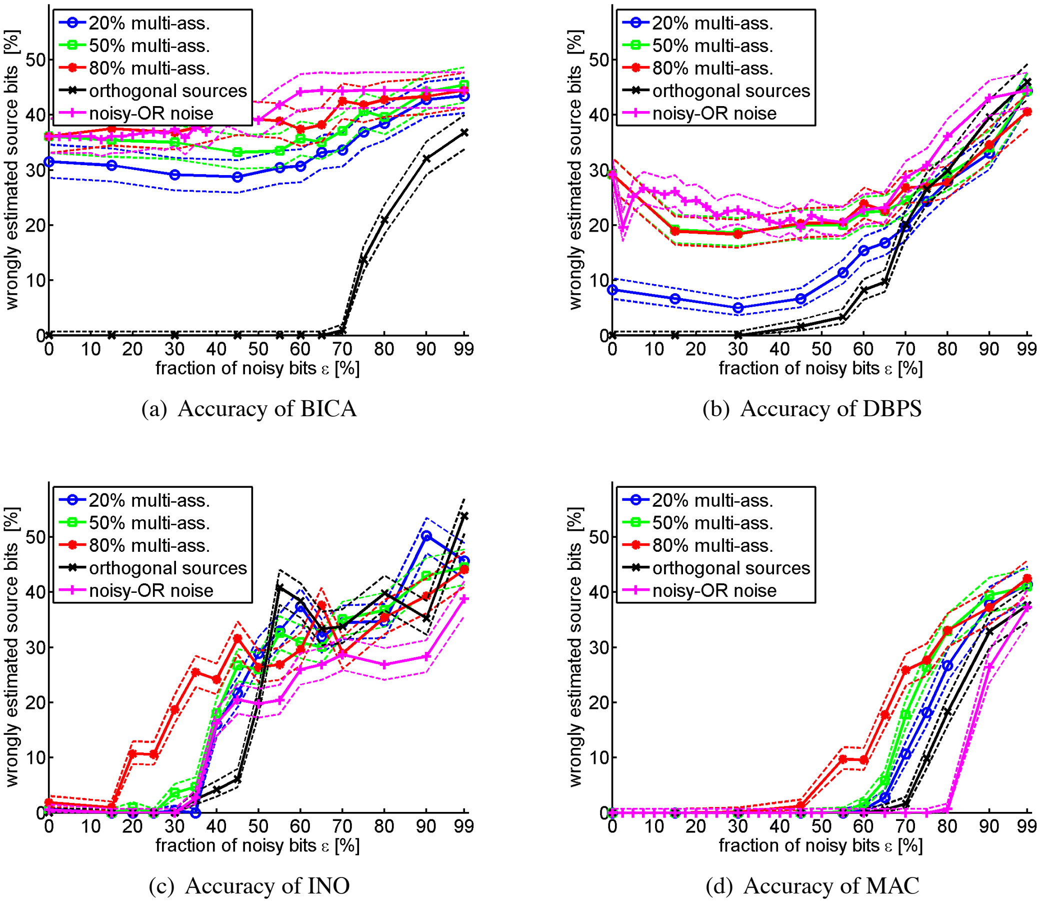
5.2.2 BINARY INDEPENDENT COMPONENT ANALYSIS (BICA)

BICA has a poor parameter accuracy in all experiments with data from overlapping clusters. This behavior is caused by the assumption of orthogonal sources, which fails to hold for such data. BICA performs better on data that was modified by the symmetric mixture noise process than on data from a noisy-OR noise process. Since BICA does not have a noise model, the data containing noise from the noisy-OR noise process leads to extra 1s in the source estimators. This effect becomes important when the noise fraction rises above 50%. We observe that, overall, the error rate does not vary much for overlapping sources.

The effect of the source geometry is particularly noticeable. On data generated by orthogonal sources, i.e. when the assumption of BICA is fulfilled, the source parameters are perfectly reconstructed for noise levels up to 65%. Only for higher noise levels, does the accuracy break down. The assumption of orthogonal source centroids is essential for BICA's performance as the poor results on data with non-orthogonal sources show. As more data items are generated by multiple, non-orthogonal sources, the influence of the mismatch between the assumption underlying BICA and the true data increases. This effect explains why the source parameter estimators for non-orthogonal centroids become less accurate when going from 20% of multi-assignments to 80%.

5.2.3 DISCRETE BASIS PROBLEM SOLVER (DBPS)

Figure 4(b) shows that this method yields accurate source parameter estimators for data generated by orthogonal sources, and, to a lesser degree, for data sets that contain a small percentage of multi-assignment data. As the fraction of multi-assignment data increases, the accuracy of DBPS decreases.



(a) Accuracy of BICA

(b) Accuracy of DBPS

(c) Accuracy of INO

(d) Accuracy of MAC

Figure 4: Accuracy of source parameter estimation for five different types of data sets in terms of mismatch to the true sources. We use (circle, square, star) symmetric Bernoulli noise and overlapping sources with three different fractions of multi-assignment data, (x) orthogonal sources and symmetric noise, and (+) overlapping sources and a noisy-or noise process. Solid lines indicate the median over 10 data sets with random noise and dashed lines show the 65% confidence intervals.

The reason for the low accuracy on multi-assignment data arises from the greedy optimization of DBPS. It selects a new source out of a candidate set such that it can explain as many objects as possible by the newly chosen source. In a setting where most of the data is created by a combination of sources, DBPS will first select a single source that equals the disjunction of the true sources because this covers the most 1s. We call this effect *combination-singlet confusion*. It is a special case of the typical problem of forward selection. Lacking a generative model for source-combinations, DBPS cannot use the observation of objects generated by source-combinations to gather evidence for the individual sources. As a consequence, the first selected source estimates fit to the source-combinations and not to the true individual sources. Often, the last selected sources are left empty, leading to a low estimation accuracy.

Note the effect of a small amount of noise on the accuracy of DBPS. The clear structure of the association matrix is perturbed, and the candidates might contain 0s in some dimensions. As a result, the roles selected in the second and subsequent steps are non-empty, making the solution more similar to the true sources. This results in the interesting effect where the accuracy increases when going from noise-free matrices to those with small amount of noise (for higher noise, it decreases again because of overfitting).

DBPS obtains accurate estimators in the setting where the data is generated by orthogonal data (labeled 'x'). Here, the candidate set does not contain sources that correspond to combinations of true sources, and the greedy optimization algorithm can only select a candidate source that corresponds to a true single source. DBPS thus performs best with respect to source parameter estimation when the generating sources are orthogonal. In contrast to BICA, which benefits from the explicit assumption of orthogonal sources, DBPS favors such sources because of the properties of its greedy optimizer.

5.2.4 INFINITE NOISY-OR (INO)

The infinite noisy-OR is a non-parametric Bayesian method. To obtain a single result, we approximate the a posteriori distribution by sampling and then choose the parameters with highest probability. This procedure estimates the maximum a posterior solution. Furthermore, in contrast to BICA, DBPS, and all MAC variants, INO determines the number of sources by itself and might obtain a value different than the number of sources used to generate the data. If the number inferred by INO is smaller than the true number, we choose the closest true sources to compute the parameter mismatch. If INO estimates a larger set of sources than than the true one, the best-matching INO sources are used. This procedure systematically overestimates the accuracy of INO, whereas INO actually solves a harder task that includes model-order selection. A deviation between the estimated number of sources and the true number mainly occurs at the mid-noise level (approximately 30% to 70% noisy bits).

In all settings, except the case where 80% of the data items are generated by multiple sources, INO yields perfect source estimators up to noise levels of 30%. For higher noise levels, its accuracy rapidly drops. While the generative model underlying INO enables this method to correctly interpret data items generated by multiple sources, a high percentage (80%) of such data poses the hardest problem for INO.

For noise fractions above approximately 50%, the source parameter estimators are only slightly better than random in all settings. On such data, the main influence comes from the noise, while the contribution of different source combinations is no longer important.

5.2.5 MULTI-ASSIGNMENT CLUSTERING (MAC)

The multi-assignment clustering method yields perfect parameter estimators for noise levels up to 40% in all experimental settings considered. The case with 80% of multi-assignment data is the most challenging one for MAC. When only 50% or 20% of the data items are generated by more than one source, the parameter estimates are accurate for noise levels up to 55% or 60% of noisy bits. When few data items originate from a single source, MAC fails to separate the contributions of the individual sources. These single-source data items function as a kind of 'anchor' and help the algorithm to converge to the true parameters of the individual sources. For very high noise levels (90% and above), the performance is again similar for all three ratios of multi-assignment data.

In comparison to the experiments with overlapping sources described in the previous paragraph, MAC profits from orthogonal centroids and yields superior parameter accuracy for noise levels above 50%. As for training data with little multi-assignment data, orthogonal centroids simplify the task of disentangling the contributions of the individual sources. When a reasonable first estimate of the source parameters can be derived from single-assignment data, a 1 in dimension d of a data item is explained either by the unique source which has a high probability of emitting a 1 in this dimension, or by noise—even if the data item is assigned to more than one source.

Interestingly, MAC's accuracy peaks when the noise is generated by a noisy-OR noise process. The reason is that observing a 1 at a particular bit creates a much higher entropy of the parameter estimate than observing a 0: a 1 can be explained by all possible combinations of sources having a 1 at this position, whereas a 0 gives strong evidence that all sources of the object are 0. As a consequence, a wrong bit being 0 is more severe than a wrong 1. The wrong 0 forces the source estimates to a particular value whereas the wrong 1 distributes its 'confusion' evenly over the sources. As the noisy-OR creates only 1s, it is less harmful. This effect could, in principle, also help other methods if they managed to appropriately disentangle combined source parameters.

5.2.6 PERFORMANCE OF MAC VARIANTS

We carry out inference with the MAC model and the corresponding Single-Assignment Clustering (SAC) model, each with and without the mixture noise model. These model variants are explained in Section 3.1.1. The results illustrated in Figure 5 are obtained using data sets with 350 objects. The objects are sampled from the overlapping sources depicted in Figure 3(a). To evaluate the solutions of the SAC variants in a fair way, we compare the estimated sources against all combinations of the true sources.

5.2.7 INFLUENCE OF SIGNAL MODEL AND NOISE MODEL

As observed in Figure 5, the source parameter estimators are much more accurate when a noise model is employed. For a low fraction of noisy bits ($< 50\%$), the estimators with a noise model are perfect, but are already wrong for 10% noise when not using a noise model. When inference is carried out using a model that lacks the ability to explain individual bits by noise, the entire data set must be explained with the source estimates. Therefore, the solutions tend to overfit the data set. With a noise model, a distinction between the structure and the irregularities in the data is possible and allows one to obtain more accurate estimates for the model parameters.

Multi-Assignment Clustering (MAC) provides more accurate estimates than SAC and the accuracy of MAC breaks down at a higher noise level than the accuracy of SAC. The reason is twofold. First, the ratio of the number of observations per model parameter differs for both model variants. MAC explains the observations with combinations of sources whereas SAC assigns each object to a single source only. SAC therefore uses only those objects for inference that are exclusively assigned to a source, while MAC also uses objects that are simultaneously assigned to other sources. Second, using the same source in different combinations with other sources implicitly provides a consistency check for the source parameter estimates. SAC lacks this effect as all source parameters are independent. The difference between MAC and SAC becomes apparent when the data set is noisy. For low fractions of noise, the accuracy is the same for both models.

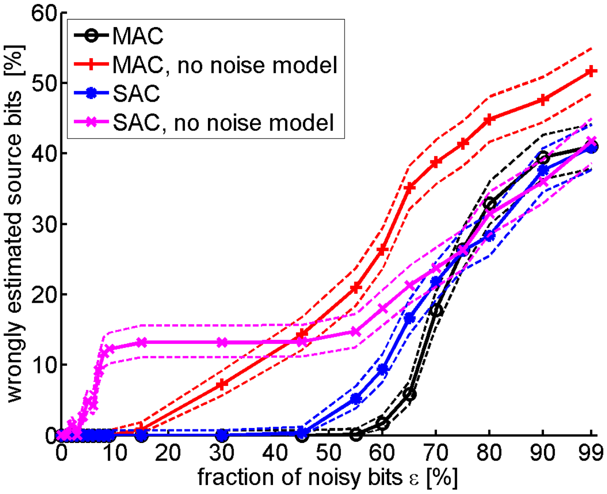


Figure 5: Average Hamming distance between true and estimated source prototypes for MAC and SAC with and without noise models respectively.

We conducted the same experiments on data sets that are ten times larger and observed the same effects as the ones described above. The sharp decrease in accuracy is shifted to higher noise levels and appears in a smaller noise window when more data is available.

5.3 Experiments on Role Mining Data

To evaluate the performance of our algorithm on real data, we apply MAC to mining RBAC roles from access control configurations. We first specify the problem setting and then report on our experimental results.

5.3.1 SETTING AND TASK DESCRIPTION

As explained in Section 2, role mining must find a suitable RBAC configuration based on a binary user-permission assignment matrix $\mathbf{x}$. An RBAC configuration is the assignment of K roles to permissions and assignments of users to these roles. A user can have multiple roles, and the bit-vectors representing the roles can overlap. The inferred RBAC configuration is encoded by the Boolean assignment matrices $(\hat{\mathbf{z}}, \hat{\mathbf{u}})$.

We emphasize the importance of the generalization ability of the RBAC configuration: The goal is not primarily to compress the existing user-permission matrix $\mathbf{x}$, but rather to infer a set of roles that generalizes well to new users. An RBAC system's security and maintainability improve when the roles do not need to be redefined whenever there is a small change in the enterprise, such as a new user being added to the system or users changing positions within the enterprise. Moreover, as previously explained, it is desirable that the role mining step identifies exceptional permission assignments. Such exceptional assignments are represented by the noise component of the mixture model. In practice, one must check whether the suspected erroneous bits are really errors or if they were (and still are!) intended. Without additional input, one can at most distinguish between reg-

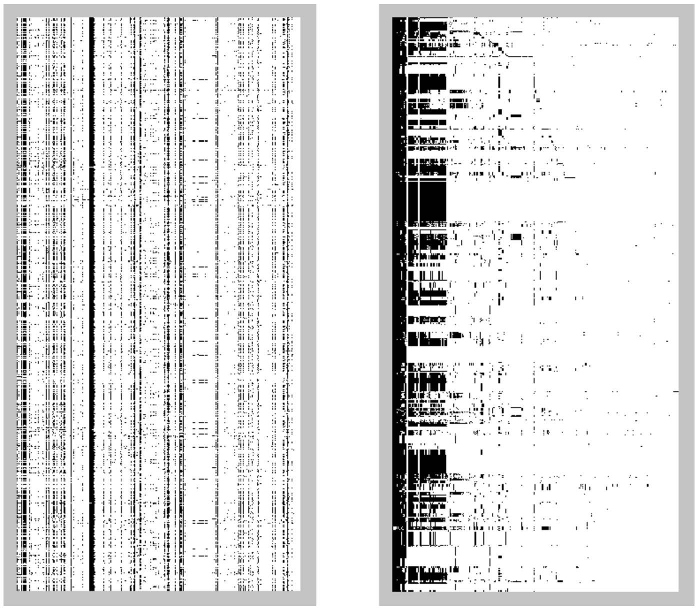
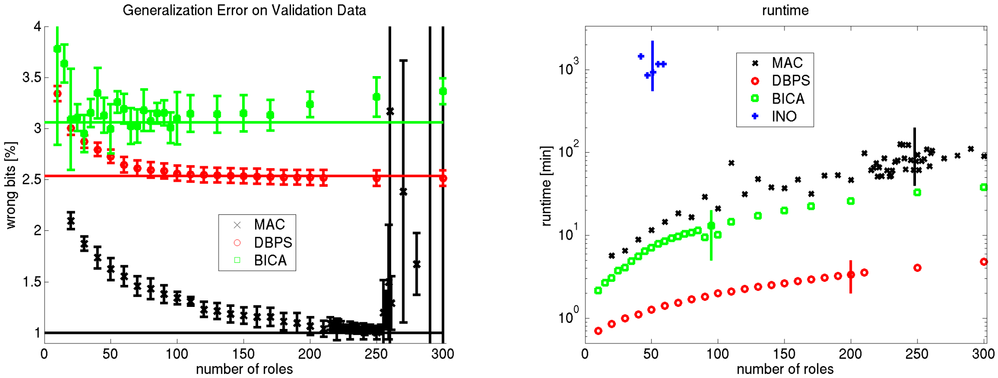


Figure 6: A 2400×500 part of the data matrix used for model-order selection. Black dots indicate a 1 at the corresponding matrix element and white dots indicate a 0. The full data matrix has size 4900×1300. Rows and columns of the right matrix are reordered such that users with the same role set and permissions of the same role are adjacent to each other, if possible. Note that there does not exist a permutation that satisfies this condition for all users and permissions simultaneously.

ularities and irregularities. This is a problem for all role mining algorithms: The interpretation of the irregularities and any subsequent corrections must be performed by a domain expert. However, minimizing the number of suspicious bits and finding a decomposition that generalizes well is already a highly significant advantage over manual role engineering. See Frank et al. (2010) for an extended discussion of this point.

In our experiments, we use a data set from our collaborator containing the user-permission assignment matrix of $N = 4900$ users and $D = 1300$ permissions. We will call this data set C_{orig} in subsequent sections. A part of this data matrix is depicted in Figure 6. Additionally, we use the publicly available access control configurations from HP labs published by Ene et al. (2008).

To evaluate the different methods on more complex data with a higher noise level, we generate another data set $\bar{\mathbf{x}}$ as follows: For the original user-permission assignment matrix of C_{orig} we combine the first 500 columns and the second 500 columns by an element-wise *OR* operation to give the structure part $\bar{\mathbf{x}}^S$. Afterwards, we replace 33% of the matrix entries by random bits to yield the modified matrix $\bar{\mathbf{x}}$. This matrix exhibits both a higher structural complexity and a substantially increased noise level than the original matrix $\mathbf{x}$. We will call this modified data set C_{mod}. We explain the individual steps of the experiments based on C_{orig} as a running example. All other experiments, those on C_{mod} and on the HP data, are carried out in the same way.



(a) Generalization Error

(b) run-time

Figure 7: Left: Generalization error on the hold-out validation set in terms of wrongly predicted bits versus the number of roles. The other external parameters for BICA and DBPS are determined by exhaustive search. Right: Run-time versus number of roles on a 2400 × 500 access-control matrix. The selected number of roles is highlighted by vertical lines.

5.3.2 MODEL-ORDER SELECTION

INO is a non-parametric model that can compute probabilities over the infinite space of all possible binary assignment matrices. It is therefore able to select the number of roles K during inference and needs no external input. For DBPS, BICA, and MAC, the number of roles must be externally selected and for DBPS and BICA, also rounding thresholds and approximation weights must be tuned. The number of roles K is the most critical parameter.

As a principle for guiding these model selection tasks, we employ the generalization error as defined in Section 5.1. Out of the total of 4900 users from C_{orig}, we use five-fold cross-validation on a subset of 3000 users. In each step, we split them into 2400 users for training the model parameters and 600 users for validating them, such that each user occurs once in the validation set and four times in the training set. The number of permissions used in this experiment is 500. We increase the number of roles until the generalization error increases. For a given number of roles, we optimize the remaining parameters (of DBPS and BICA) on the training sets and validation sets. For continuous parameters, we quantize the parameter search-space into 50 equally spaced values spanning the entire range of possible parameter values.

To restrict the cardinality of the assignment sets (for MAC), we make one trial run with a large number of roles and observe how many of the roles are involved in role combinations. A role that is involved in role combinations is at least once assigned to a user together with at least one other role. In our experiments on C_{orig}, for instance, 10% of $K = 100$ roles are used in role combinations and no roles appear in combinations with more than two roles. Therefore, for subsequent runs of the algorithm, we set $M = 2$ and limit the number of roles that can belong to a multiple assignment set to 10% of K. For large K, such a restriction drastically reduces the run-time as the solution space is much smaller than the space of all possible role combinations. See Section 5.4 for an analysis of the run-time complexity of all investigated methods.

Restricting the number of roles that can belong to a multiple assignment set risks having too few role combinations available to fit the data at hand. However, such circumstances cannot lead to underfitting when K is still to be computed in the cross-validation phase. In the worst case, an unavailable role combination would be substituted by an extra single role.

The performance of the three methods MAC, DBPS, and BICA as a function of the number of roles is depicted in Figure 7(a), left. The different models favor a substantially different number of roles on this data set (and also on other data sets, see Table 1). For MAC, there is a very clear indication of overfitting for $K > 248$. For DBPS, the generalization error monotonically decreases for $K < 150$. As K further increases, the error remains constant. In the cross-validation phase, the internal threshold parameter of DPBS is adapted to minimize the generalization error. This prevents excessive roles from being used as, with the optimal threshold, they are left empty. We select $K = 200$ for DBPS, where more roles provide no improvement. INO selects 50 roles on average. BICA favors a considerably smaller number of roles, even though the signal is not as clear. We select $K = 95$ for BICA, which is the value that minimizes the median generalization error on the validation sets.

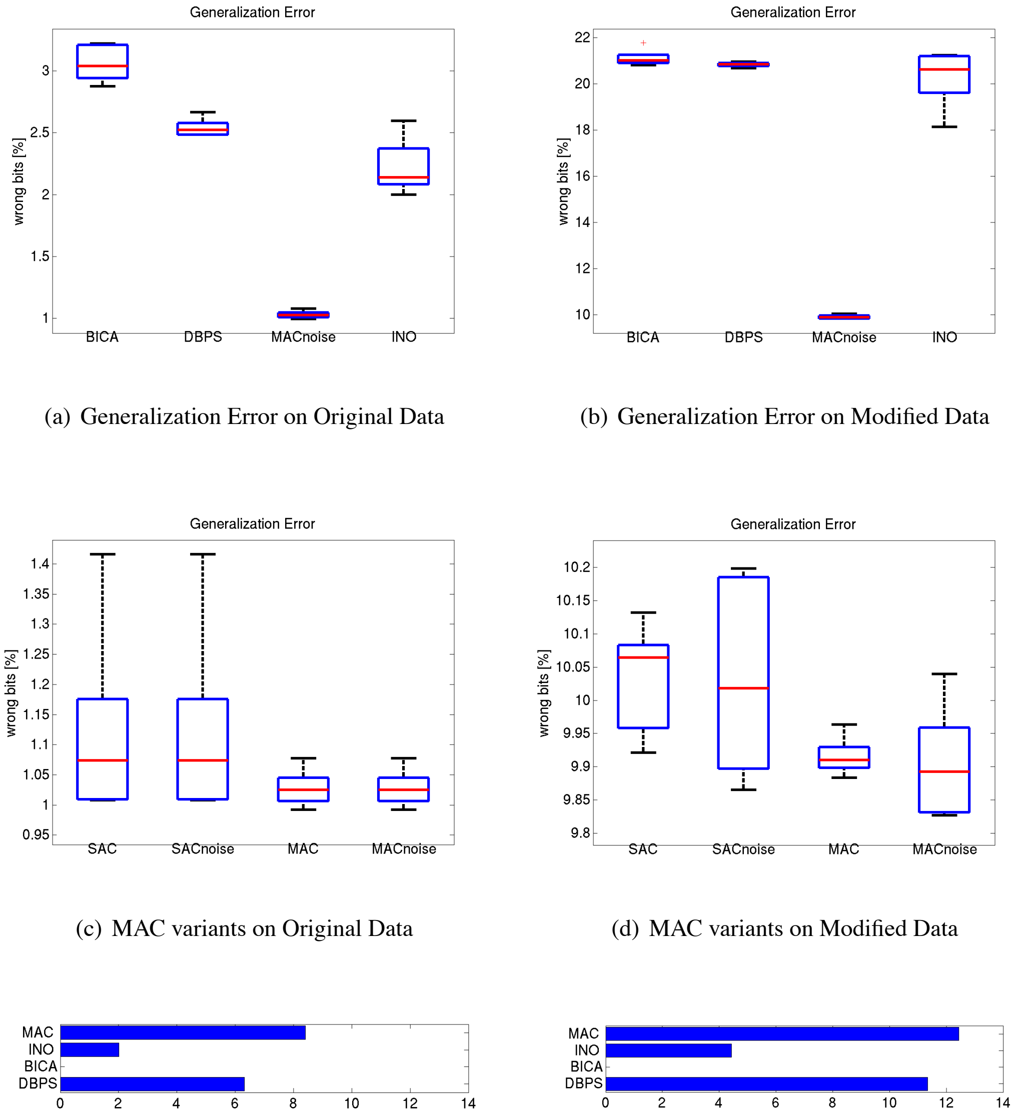
5.3.3 RESULTS OF DIFFERENT METHODS

The results of the generalization experiments for the four methods MAC, DBPS, BICA, and INO are depicted in Figure 8. Overall, all methods have a very low generalization error on the original data set. The error spans from 1% to 3% of the predicted bits. This result indicates that, on a global scale, C_{orig} has a rather clean structure. It should be stressed that most permissions in the input data set are only rarely assigned to users, whereas some are assigned to almost everyone, thereby making up most of the 1s in the matrix (see a part of the data set in Figure 6). Therefore, the most trivial role set where roles are assigned no permissions already yields a generalization error of 13.5%. Assigning everyone to a single role that contains all permissions that more than 50% percent of the users have, achieves 7.1%. One should keep this baseline in mind when interpreting the results.

INO, DBPS, and BICA span a range from 2.2% generalization error to approximately 3% with significant distance to each other. MAC achieves the lowest generalization error with slightly more than 1%. It appears that INO is misled by its noisy-OR noise model, which seems to be inappropriate in this case. MAC estimates the fraction of noisy bits by $\hat{\varepsilon} \approx 2.8\%$ and the probability for a noisy bit to be 1 by $\hat{r} \approx 20\%$. This estimate clearly differs from a noisy-OR noise process (which would have $r = 1$). With more than 3% generalization error, BICA performs worst. As all other methods estimate a considerable centroid overlap, the assumption of orthogonal (non-overlapping) centroids made by BICA seems to be inappropriate here and might be responsible for the higher error.

In our experiments on the modified data set with more structure and a higher noise level, Figure 8(b), all methods have significantly higher generalization errors, varying between approximately 10% to 21%. The trivial solution of providing each user all those permissions assigned to more than 50% of the users, leads to an error of 23.3%. Again, MAC with 10% generalization error yields significantly lower generalization error than all the other methods. INO, DBPS, and BICA perform almost equally well each with a median error of 20% to 21%. A generalization error of 10% is still very good as this data set contains at least 33% random bits, even though a random bit can take the correct value by chance.

The lower row of Figure 8 shows the average role overlap between the roles obtained by the different methods. This overlap measures the average number of permissions that the inferred roles



(a) Generalization Error on Original Data

(b) Generalization Error on Modified Data

(c) MAC variants on Original Data

(d) MAC variants on Modified Data

(e) Average Role Overlap (%)

(f) Average Role Overlap (%)

Figure 8: Generalization experiment on real data. Graphs (a)-(d) show the generalization error obtained with the inferred roles, and graphs (e)-(f) display the average overlap between roles.

have in common. For BICA, the roles never overlap, by the definition of the method. For all other methods, the increased overlap of the data's structure is reflected in the estimated roles. The decrease in the difference in performance between BICA and the other models after processing the modified data set indicates that the main difficulty for models that can represent overlapping roles is the increased noise level rather than the overlapping structure. We will return to the influence of the data set in our discussion of the results of the MAC model variants in the next section.

5.3.4 RESULTS OF MAC MODEL VARIANTS

To investigate the influence of the various model variants of MAC, we compare the performance reported above for MAC with i) the results obtained by the single-assignment clustering variant (SAC) of the model and ii) with the model variants without a noise part. The middle row of Figure 8 shows the generalization error of SAC and MAC, both with and without a noise model. On the original data set, Figure 8(c), all model variants perform almost equally well. The noise model seems to have little or no impact, whereas the multi-assignments slightly influence the generalization error. Taking MAC's estimated fraction of noisy bits $\hat{\varepsilon} \approx 2.8\%$ into account, we interpret this result by referring to the experiments with synthetic data. There the particular model variant has no influence on the parameter accuracy when the noise level is below 5% (see Figure 5.2.7). As we seem to operate with such low noise levels here, it is not surprising that the model variants do not exhibit a large difference on that data set. On the modified data with more complex structure and with a higher noise level than the original data (Figure 8(d)), the difference between multi-assignments and single-assignments becomes more apparent. Both MAC and SAC benefit from a noise part in the model, but the multi-assignments have a higher influence.

5.3.5 RESULTS ON HP DATA

With all methods described above, we learn RBAC configurations on the publicly available data sets from HP labs (first presented by Ene et al., 2008). The data set 'customer' is the access control matrix of an HP customer. 'americas small' is the configuration of Cisco firewalls that provide users limited access to HP network resources. The data set 'emea' is created in a similar way and 'firewall 1' and 'firewall 2' are created by Ene et al. (2008) by analyzing Checkpoint firewalls. Finally, 'domino' is the access profiles of a Lotus Domino server.

We run the same analysis as on C_{orig}. For the data sets 'customer', 'americas small', and 'firewall 1', we first make a trial run with many roles to identify the maximum cardinality of assignment sets M that MAC uses. We then restrict the hypothesis space of the model accordingly. For 'customer' and 'firewall 1', we use $M = 3$, for 'americas small' we use $M = 2$. For the smaller data sets, we simply offered MAC all possible role configurations, although the model does not populate all of them.

In the cross-validation phase we select the number of roles for each of the methods (except for INO), and the thresholds for BICA and DBPS in the previously described way. Afterwards we compute the generalization error on hold-out test data.

Our experimental findings are summarized in Table 1. We report the favored number of roles, the median generalization error and its average difference to the 25% and 75%-percentiles, and the run-time of one run, respectively. Overall, the MAC variants achieve the lowest generalization error within the variance of this measure. For 'americas small' and 'emea' all methods generalize equally well (note the high variance for 'emea', which is an effect of the small sample size and the high dimensionality of that data set). Here differences between the methods are dominated by run-time and the number of roles that have been found. For 'dominos', INO and BICA are almost as good as MAC, although with a significantly higher variance. Visual inspection of the 'dominos' matrix indicates that this data set has a sparse and simple structure. Differences between the methods are most pronounced on the two 'firewall' data sets. Remarkably, INO finds 80 roles for 'emea', although this data set has only 35 users.

Given the overall good generalization performance of MAC, we conclude that this model is a good ‘allrounder’. This also confirms our findings in the experiments with synthetic data. Each of the other methods shows a good performance on individual data sets but not as reliably as MAC. Comparison with the results on synthetic data suggests that their differing performance on different data sets is either due to different fractions of random noise or to true underlying sources with different overlap.

	customer			americas small		
	10,021 users × 277 perms.			3,477 users × 1,587 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	187	2.40 ± 0.03	49	139	1.03 ± 0.01	80
DBPS	178	2.54 ± 0.05	43	105	1.00 ± 0.03	187
INO	20	7.8 ± 1.6	996	65.6	1.05 ± 0.01	3691
BICA	82	2.66 ± 0.02	200	63	1.00 ± 0.01	64

	firewall1			firewall2		
	365 users × 709 perms.			325 users × 590 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	49	4.57 ± 0.01	10	10	3.40 ± 0.00	1.8
DBPS	21	13.6 ± 3.1	5	4	19.5 ± 4.4	2
INO	38.2	8.04 ± 0.00	96	6.2	11.15 ± 0.00	14
BICA	18	12.8 ± 3.0	2.1	4	19.9 ± 4.5	0.9

	dominos			emea		
	79 users × 231 perms.			35 users × 3,046 perms.		
	k	gen. error [%]	run-time [min]	k	gen. error [%]	run-time [min]
MAC	7	1.73 ± 0.00	1.1	3	8.7 ± 1.2	0.7
DBPS	9	2.3 ± 0.5	0.2	8	7.3 ± 2.6	1.1
INO	26	1.7 ± 0.1	9.0	80.4	10.1 ± 2.4	204
BICA	3	1.9 ± 0.3	0.1	5	8.6 ± 2.8	1.0

Table 1: Results on HP labs data for different methods. We report the number of roles, the median run-time of one run, as well as the median generalization error and the half inter-percentile distance between 25% and 75%.

5.4 Complexity and Runtime

The complexity of the optimization problem is determined by the number of objects and features and by the number of possible assignment sets $L := |\mathbb{L}|$. As L can be large for even a small number of clusters, the complexity is dominated by that number. Let the number of clusters that a data item can simultaneously belong to be limited by the *degree* M, i.e. $\max_{\mathcal{L} \in \mathbb{L}} |\mathcal{L}| = M$. Then the size of the assignment set is limited by $L = \sum_{m=0}^{M} \binom{K}{m} \leq 2^K$. Even for moderately sized K and M, this dependence results in computationally demanding optimization problems both for the inference step as well as for assigning new data items to previously obtained clusters. However, if the data at hand truly exhibits such a high complexity (high K and M) then also a single assignment model needs such a high complexity (to prevent the model from underfitting). In this case, a SAC model must learn L sources, while the MAC variant learns the L possible combinations out of K sources. The number of responsibilities $\gamma_{i\mathcal{L}}$ (Equation 8) to be computed in the E-step is the same for both models. However, in the M-step, MAC shares the source parameters while SAC must estimate them separately. We will shortly elaborate on the relationship between MAC and SAC from the inference perspective. Coming back to the complexity, the high number of responsibilities $\gamma_{i\mathcal{L}}$ to be computed for MAC appears to be a model-order selection issue. One can drastically reduce its complexity by limiting the number of assignment sets as described in Section 5.3.2.

In our experiments on real-world data in Section 5.3, we monitored the run-time, which is depicted in Figure 7(b). Each point represents the runtime for a single run of the different algorithms on an access-control matrix with $N = 2400$ users and $D = 500$ permissions. The number of roles chosen by the respective method is indicated by a vertical line. For INO we report the median number of roles selected. Note that in one run of INO, the model-order selection task is solved 'on-the-fly' while the other methods require multiple runs and an external validation. This overhead is reflected in the runtime. Considerable care is required in interpreting these results since the different methods were implemented by different authors in different languages (Matlab for INO, BICA and MAC, and C++ for DBPS). The DBPS implementation in C++ is impressively fast while the trend of the generalization error over the number of roles is roughly comparable to MAC and BICA. Thus, for large and demanding data sets, one could employ DBPS as a fast 'scout' to obtain an educated guess of the model-order. In conclusion, for all the investigated algorithms the runtime is not a limiting factor in role mining. This computation is only performed once when migrating an access-control system to another one. It is therefore not a problem if the computation takes hours.

5.5 Relationship Between SAC and MAC

In the following, we show that MAC can be interpreted as a SAC model with a parameter sharing rule. In the limit of many observations, MAC is equivalent to SAC with proxy-sources substituting MAC's source combinations. In order to understand the parameter sharing underlying MAC, we write the set of admissible assignment sets $\mathbb{L}$ as a Boolean matrix $\mathbf{z}^{\mathbb{L}} \in \{0,1\}^{L\times K}$. Assuming an arbitrary but fixed numbering of assignment sets in $\mathbb{L}$, $z^{\mathbb{L}}_{lk} = 1$ means that the l^{th} assignment set contains source k, and $z^{\mathbb{L}}_{lk} = 0$ otherwise. Hence, the assignment matrix $\mathbf{z}$ decomposes into $\mathbf{z} = \mathbf{z}^{\mathcal{L}} * \mathbf{z}^{\mathbb{L}}$, where $\mathbf{z}^{\mathcal{L}} \in \{0,1\}^{N\times L}$ denotes the exclusive assignment of objects to assignment sets ($z^{\mathcal{L}}_{il}$ iff object i has assignment set l, and $\sum_l z^{\mathcal{L}}_{il} = 1$ for all i). Using this notation, the decomposition $\mathbf{x} \approx \mathbf{z} * \mathbf{u}$ can be extended to

$$\mathbf{x} \approx \left(\mathbf{z}^{\mathcal{L}} * \mathbf{z}^{\mathbb{L}}\right) * \mathbf{u} = \mathbf{z}^{\mathcal{L}} * \left(\mathbf{z}^{\mathbb{L}} * \mathbf{u}\right) = \mathbf{z}^{\mathcal{L}} * \mathbf{u}^{\text{SAC}} \ ,$$

where we have defined $\mathbf{u}^{\text{SAC}} := \mathbf{z}^{\mathbb{L}} * \mathbf{u}$ as the proxy-source parameters of the single-assignment clustering model. The same notion of proxy-sources, substituting the disjunction of individual sources, is used in Equation 2 for the probabilistic source parameters. Asymptotically, the two models are equivalent. However, SAC must estimate $L \cdot D$ parameters, while the MAC model only uses $K \cdot D$ parameters. By sharing the parameters of the assignment sets, MAC reduces the number of parameters to be estimated and thereby increases the number of data items available per parameter. Moreover, the sharing rule provides a mutual inconsistency check for the involved parameter estimates. This check is not available if parameters are estimated independently. These two points explain the higher accuracy in the parameter estimators, which we observe in the experiments reported in Section 5.2.

6. Conclusion and Outlook

We have presented a probabilistic method to cluster vectors of Boolean data. In contrast to the conventional approach of mutually exclusive cluster assignments, our method enables a data item to belong to multiple clusters. In our generative model, the clusters are the sources that generate the structure in the data and irregularities are explained by an independent noise process. In a detailed analysis of our model variants, we demonstrate that the proposed method outperforms state-of-the-art techniques with respect to parameter estimation accuracy and generalization ability. In experiments on a real world data set from the domain of role-based access control, our model achieves significantly lower generalization error than state-of-the-art techniques.

Throughout this paper, the Boolean *OR* combines the emissions of multiple sources. However, the proposed concept is neither limited to the Boolean *OR* nor to Boolean data. Further work will address the combination of other kinds of data and other combination rules such as additive combinations of real numbers.

Acknowledgments

This work was partially supported by the Zurich Information Security Center and by the CTI grant Nr. 8539.2;2 EPSS-ES. We thank the reviewers for their valuable comments.

References

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *Int Conf on Management of Data*, 22(2):207–216, 1993.

Eugene L. Allgower and Kurt Georg. Simplicial and continuation methods for approximations, fixed points and solutions to systems of equations. *SIAM Review*, 22:28–85, 1980.

Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, November 1974.

Radim Belohlavek and Vilem Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Comput. Syst. Sci.*, 76(1):3–20, 2010.

Joachim M. Buhmann and Hans Kühnel. Vector quantization with complexity costs. In *IEEE Trans on Information Theory*, volume 39, pages 1133–1145. IEEE, 1993.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 2nd ed.* MIT Press, 2001.

Alina Ene, William Horne, Nikola Milosavljevic, Prasad Rao, Robert Schreiber, and Robert E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *SACMAT '08: Proceeding of the 13th ACM Symposium on Access Control Models and Technologies*, pages 1–10, 2008.

Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1 (2):209–230, 1973.

David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4 (3):224–274, 2001.

Mario Frank, David Basin, and Joachim M. Buhmann. A class of probabilistic models for role engineering. In *CCS '08: Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 299–310, New York, NY, USA, 2008. ACM.

Mario Frank, Joachim M. Buhmann, and David Basin. On the definition of role mining. In *SACMAT '10: Proceeding of the 15th ACM Symposium on Access Control Models and Technologies*, pages 35–44, New York, NY, USA, 2010. ACM.

Mario Frank, Morteza Chehreghani, and Joachim M. Buhmann. The minimum transfer cost principle for model-order selection. In *ECML PKDD '11: Machine Learning and Knowledge Discovery in Databases*, pages 423–438. Springer Berlin / Heidelberg, 2011.

Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis - Mathematical Foundations*. Springer, 1999.

Zoubin Ghahramani, Thomas L. Griffiths, and Peter Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics 8. Oxford University Press*, pages 201–225, 2007.

James F. Gimpel. The minimization of spatially-multiplexed character sets. *Communications of the ACM*, 17(6):315–318, 1974.

Thomas L. Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, New York, NY, USA, 2000. ACM.

Katherine A. Heller and Zoubin Ghahramani. A nonparametric bayesian approach to modeling overlapping clusters. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-2007)*, pages 297–304, 2007.

Tommi S. Jaakkola and Michael I. Jordan. Variational probabilistic inference and the qmr-dt network. *Journal of Artificial Intelligence Research*, 10(1):291–322, 1999.

Ata Kabán and Ella Bingham. Factorisation and denoising of 0-1 data: A variational approach. *Neurocomputing*, 71(10-12):2291–2308, 2008.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Nat Conf on Artificial Intelligence*, pages 763–770, 2006.

Ales Keprt and Václav Snásel. Binary factor analysis with help of formal concepts. In *Proc. of CLA 2004*, pages 90–101, 2004.

Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. Role mining — revealing business roles for security administration using data mining technology. In *SACMAT'03: Proceeding of the 8th ACM Symp on Access Control Models and Technologies*, pages 179–186, New York, NY, USA, 2003. ACM.

Harold W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008*, pages 29–47. Springer Berlin Heidelberg, 2010.

Pauli Miettinen, Taneli Mielikäinen, Aris Gionis, Gautam Das, and Heikki Mannila. The Discrete Basis Problem. In *Proc. of Principles and Practice of Knowledge Discovery in Databases*, pages 335–346. Springer, 2006.

Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. Mining roles with semantic meanings. In *SACMAT '08: Proceeding of the 13th ACM Symposium on Access Control Models and Technologies*, pages 21–30, 2008.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proc. of the IEEE*, pages 2210–2239, 1998.

Larry J. Stockmeyer. The set basis problem is NP-complete. *Report RC5431, IBM Watson Research*, 1975.

Andreas P. Streich, Mario Frank, David Basin, and Joachim M. Buhmann. Multi-assignment clustering for Boolean data. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 969–976, New York, NY, USA, 2009. ACM.

Jaideep Vaidya, Vijay Atluri, and Qi Guo. The Role Mining Problem: Finding a minimal descriptive set of roles. In *SACMAT '07: Proceeding of the 12th ACM Symposium on Access Control Models and Technologies*, pages 175–184. ACM Press, 2007.

Tomáš Šingliar and Miloš Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.

Frank Wood, Thomas L. Griffiths, and Zoubin Ghahramani. A non-parametric Bayesian method for inferring hidden causes. In *Conference on Uncertainty in Artificial Intelligence*, pages 536–543. AUAI Press, 2006.

Journal of Machine Learning Research 13 (2012) 491-518 Submitted 4/11; Revised 12/11; Published 2/12

Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks

Vikas C. Raykar VIKAS.RAYKAR@SIEMENS.COM
Shipeng Yu SHIPENG.YU@SIEMENS.COM
Siemens Healthcare
51 Valley Stream Parkway, E51
Malvern, PA 19355, USA

Editor: Ben Taskar

Abstract

With the advent of crowdsourcing services it has become quite cheap and reasonably effective to get a data set labeled by multiple annotators in a short amount of time. Various methods have been proposed to estimate the consensus labels by correcting for the bias of annotators with different kinds of expertise. Since we do not have control over the quality of the annotators, very often the annotations can be dominated by spammers, defined as annotators who assign labels randomly without actually looking at the instance. Spammers can make the cost of acquiring labels very expensive and can potentially degrade the quality of the final consensus labels. In this paper we propose an empirical Bayesian algorithm called SpEM that iteratively eliminates the spammers and estimates the consensus labels based only on the good annotators. The algorithm is motivated by defining a spammer score that can be used to rank the annotators. Experiments on simulated and real data show that the proposed approach is better than (or as good as) the earlier approaches in terms of the accuracy and uses a significantly smaller number of annotators.

Keywords: crowdsourcing, multiple annotators, ranking annotators, spammers

1. Introduction

Annotating a data set is one of the major bottlenecks in using supervised learning to build good predictive models. Getting a data set labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services (Amazon's Mechanical Turk[1] being a prime example) it has become quite easy and inexpensive to acquire labels from a large number of annotators in a short amount of time (see Sheng et al. 2008, Snow et al. 2008, and Sorokin and Forsyth 2008 for some natural language processing and computer vision case studies). For example in AMT the *requesters* are able to pose tasks known as HITs (Human Intelligence Tasks). Workers (called *providers*) can then browse among existing tasks and complete them for a small monetary payment set by the requester.

A major drawback of most crowdsourcing services is that we do not have control over the quality of the annotators. The annotators usually come from a diverse pool including genuine experts, novices, biased annotators, malicious annotators, and spammers. Hence in order to get good quality labels requestors typically get each instance labeled by multiple annotators and these multiple annotations are then consolidated either using a simple majority voting or more sophisticated methods

[1] Amazon's Mechanical Turk can be found at `https://www.mturk.com`.

that model and correct for the annotator biases (Dawid and Skene, 1979; Smyth et al., 1995; Raykar et al., 2009, 2010; Yan et al., 2010) and/or task complexity (Carpenter, 2008; Whitehill et al., 2009; Welinder et al., 2010).

In this paper we are interested in the situation where the annotations are dominated by *spammers*. In our context a spammer is a low quality annotator who assigns random labels (maybe because the annotator does not understand the labeling criteria, does not look at the instances when labeling, or maybe a bot pretending to be a human annotator). Spammers can significantly *increase the cost* of acquiring annotations (since they need to be paid) and at the same time *decrease the accuracy* of the final consensus labels. A mechanism to detect and eliminate spammers is a desirable feature for any crowdsourcing market place. For example one can give monetary bonuses to good annotators and deny payments to spammers. This paper makes two novel contributions:[2]

1. ***Spammer score to rank annotators*** The first contribution of this paper is to formalize the notion of a spammer for binary and categorical labels. More specifically we define a *scalar metric* which can be used to *rank the annotators*, with the spammers having a score close to zero and the good annotators having a score close to one. We summarize the multiple parameters corresponding to each annotator into a single score indicative of how spammer like the annotator is. While this metric was implicit for binary labels in earlier works (Dawid and Skene, 1979; Smyth et al., 1995; Carpenter, 2008; Raykar et al., 2009; Donmez et al., 2009) the extension to categorical labels is novel and is quite different for the error rate computed from the confusion rate matrix. An attempt to quantify the quality of the workers based on the confusion matrix was recently made by Ipeirotis et al. (2010) where they transformed the observed labels into posterior soft labels based on the estimated confusion matrix. While we obtain somewhat similar annotator rankings, we differ from this work in that our score is directly defined in terms of the annotator parameters. Having the score defined only in terms of the annotator parameters makes it easy to specify a prior for Bayesian approaches to eliminate spammers and consolidate annotations.

2. ***Algorithm to eliminate spammers*** The second contribution is that we propose an algorithm to consolidate annotations that eliminates spammers automatically. One of the commonly used strategy is to inject some items into the annotations *with known labels* (gold standard) and use them to evaluate the annotators and thus eliminate the spammers.[3] Typically we would like to detect the spammers with as few instances as possible and eliminate them from further annotations. In this work we propose an algorithm called SpEM that eliminates the spammers *without using any gold standard* and estimates the consensus ground truth based only on the good annotators. The same algorithm can also be used if some labels are also known.

   We build on the earlier works of Dawid and Skene (1979), Smyth et al. (1995), and Raykar et al. (2009, 2010) who proposed algorithms that correct for the annotator biases by estimating the annotator accuracy and the actual true label jointly. A simple strategy would be to use these algorithms to estimate the annotator parameters, detect and eliminate the spammers (as defined by our proposed spammer score) and refit the model with only the good annotators. However this approach is not a principled approach and might be hard to control (for example, how to define spammers and how many to remove, etc). The algorithm we propose is essentially a formalization of this strategy. Our final algorithm essentially repeats

[2] A preliminary version of this paper (Raykar and Yu, 2011) mainly discussed the score to rank annotators.

[3] This is the strategy used by CrowdFlower (`http://crowdflower.com/docs/gold`).

this, it *iteratively* eliminates the spammers and re-estimates the labels based only on the good annotators. A crucial element of our proposed algorithm is that we eliminate spammers by thresholding on a hyperparameter of the prior (automatically estimated from the data) rather than directly thresholding on the estimated spammer score.

The rest of the paper is organized as follows. In Section 2 we model the annotators in terms of the sensitivity and specificity for binary labels and extend it to categorical labels. Based on this model the notion of a spammer is formalized in Section 3. In Section 4 we propose a Bayesian point estimate by using a prior (Section 4.2) derived from the proposed spammer score designed to favor spammer detection. This is essentially a modification of the Expectation Maximization (EM) algorithm proposed by Dawid and Skene (1979), Smyth et al. (1995), and Raykar et al. (2009, 2010). The hyperparameters of this prior are estimated via an empirical Bayesian method in Section 5 leading to the proposed SpEM algorithm (Algorithm 1) that iteratively eliminates the spammers and re-estimates the ground truth based only on the good annotators. In Section 6 we discuss this algorithm in context of other methods and also propose a few extensions. in Section 7 we extend the same ideas to categorical labels. In Section 8 we extensively validate our approach using both simulated data and real data collected using AMT and other sources from different domains.

2. Annotator Model

An annotator provides a noisy version of the true label. Let $y_i^j \in \{0,1\}$ be the label assigned to the i^{th} instance by the j^{th} annotator, and let y_i be the actual (unobserved) label. Following the approach of Raykar et al. (2009, 2010) we model the accuracy of the annotator separately on the positive and the negative examples. If the true label is one, the *sensitivity* (true positive rate) for the j^{th} annotator is defined as the probability that the annotator labels it as one.

$$\alpha^j := \Pr[y_i^j = 1 | y_i = 1].$$

On the other hand, if the true label is zero, the *specificity* (1−false positive rate) is defined as the probability that the annotator labels it as zero.

$$\beta^j := \Pr[y_i^j = 0 | y_i = 0].$$

With this model we have implicitly assumed that α^j and β^j do not depend on the instance. Extensions of this basic model have been proposed to include item level difficulty (Carpenter, 2008; Whitehill et al., 2009) and also to explicitly model the annotator performance based on the instance feature vector (Yan et al., 2010). In principle the proposed algorithm can be extended to these kind of complicated models (with more parameters), however for simplicity we use the basic model proposed in Raykar et al. (2009, 2010) in our formulation.

The same model can be extended to categorical labels. Suppose there are $C \geq 2$ categories. We introduce a multinomial parameter $\boldsymbol{\alpha}_c^j = (\alpha_{c1}^j, \ldots, \alpha_{cC}^j)$ for each annotator, where

$$\alpha_{ck}^j := \Pr[y_i^j = k | y_i = c], \qquad \sum_{k=1}^{C} \alpha_{ck}^j = 1.$$

The term α_{ck}^j denotes the probability that annotator j assigns class k to an instance given the true class is c. When $C = 2$, α_{11}^j and α_{00}^j are sensitivity and specificity, respectively.

3. Who is a Spammer? Score to Rank Annotators

Intuitively, *a spammer assigns labels randomly*, maybe because the annotator does not understand the labeling criteria, does not look at the instances when labeling, or maybe a bot pretending to be a human annotator. More precisely an annotator is a spammer if the probability of observed label y_i^j being one given the true label y_i is independent of the true label, that is,

$$\Pr[y_i^j = 1|y_i] = \Pr[y_i^j = 1]. \tag{1}$$

This means that the annotator is assigning labels randomly by flipping a coin with bias $\Pr[y_i^j = 1]$ without actually looking at the data. Equivalently (1) can be written as

$$\begin{aligned} \Pr[y_i^j = 1|y_i = 1] &= \Pr[y_i^j = 1|y_i = 0], \\ \alpha^j &= 1 - \beta^j. \end{aligned} \tag{2}$$

Hence in the context of the annotator model defined in Section 2, a spammer is an annotator for whom

$$\alpha^j + \beta^j - 1 = 0.$$

This corresponds to the diagonal line on the Receiver Operating Characteristic (ROC) plot (see Figure 1).[4] If $\alpha^j + \beta^j - 1 < 0$ then the annotator lies below the diagonal line and is a malicious annotator who flips the labels. Note that a malicious annotator has discriminatory power if we can detect them and flip their labels. In fact the methods proposed in Dawid and Skene (1979) and Raykar et al. (2010) can automatically flip the labels for the malicious annotators. Hence we define the spammer score for an annotator as

$$\mathcal{S}^j = (\alpha^j + \beta^j - 1)^2. \tag{3}$$

An annotator is a spammer if $\mathcal{S}^j$ is close to zero. Good annotators have $\mathcal{S}^j > 0$ while a perfect annotator has $\mathcal{S}^j = 1$.

Another interpretation of a spammer can be seen from the log odds. Using Bayes' rule the posterior log-odds can be written as

$$\log \frac{\Pr[y_i = 1|y_i^j]}{\Pr[y_i = 0|y_i^j]} = \log \frac{\Pr[y_i^j|y_i = 1]}{\Pr[y_i^j|y_i = 0]} + \log \frac{p}{1-p},$$

where $p := \Pr[y_i = 1]$ is the prevalence of the positive class. If an annotator is a spammer (that is (2) holds) then

$$\log \frac{\Pr[y_i = 1|y_i^j]}{\Pr[y_i = 0|y_i^j]} = \log \frac{p}{1-p}.$$

Essentially the annotator provides no information in updating the posterior log-odds and hence does not contribute to the estimation of the actual true label.

[4] Note that $(\alpha^j + \beta^j)/2$ is equal to the area shown in the plot and can be considered as a non-parametric approximation to the area under the ROC curve (AUC) based on one observed point $(1 - \beta^j, \alpha^j)$. It is also equal to the Balanced Classification Rate (BCR). So a spammer can also be defined as having BCR or AUC equal to 0.5. Another way to think about this is that instead of using sensitivity and specificity we can re-parameterize an annotator in terms of an accuracy parameter ($(\alpha^j + \beta^j)/2$) and a bias parameter ($(\alpha^j - \beta^j)/2$). A spammer is an annotator with accuracy equal to 0.5. The biased ($\alpha^j - \beta^j$ is large) or malicious annotators ($\alpha^j + \beta^j < 1$) (see Figure 1) are also sometimes called the spammers since they can potentially degrade the consensus labels, but in this paper we do not focus on them, since their annotations can be calibrated or reversed by the EM algorithm.

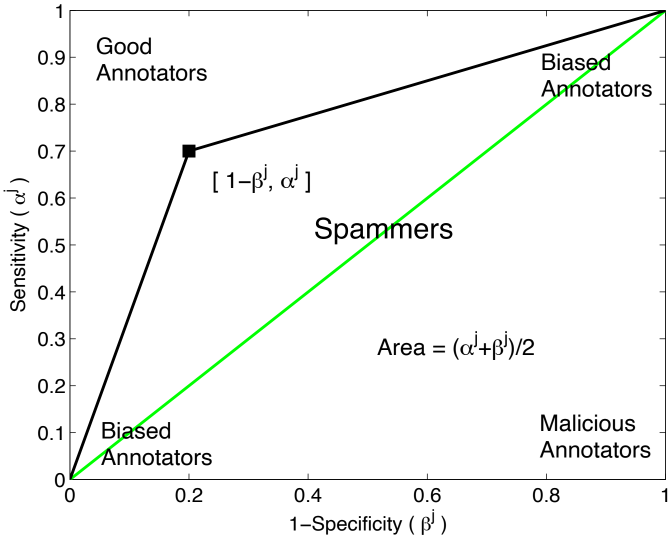


Figure 1: For binary labels each annotator is modeled by his/her sensitivity and specificity. A spammer lies on the diagonal line (that is, $\alpha^j = 1 - \beta^j$) on this ROC plot.

3.1 Accuracy

This notion of a spammer is quite different from that of the *accuracy* of an annotator. An annotator with high accuracy is a good annotator but one with low accuracy is not necessarily a spammer. The accuracy of the j^{th} annotator is computed as

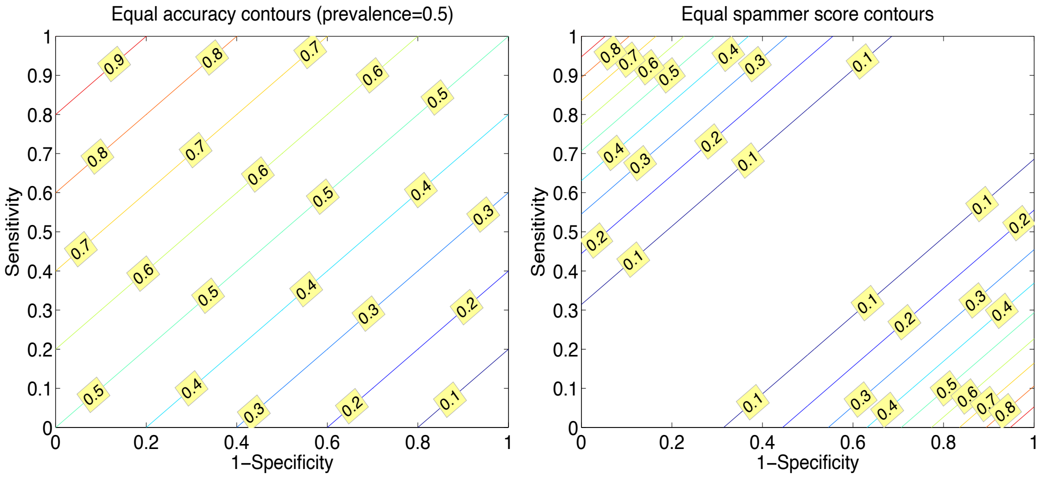
$$\text{Accuracy}^j = \Pr[y_i^j = y_i] = \sum_{k=0}^{1} \Pr[y_i^j = 1|y_i = k]\Pr[y_i = k] = \alpha^j p + \beta^j (1-p), \qquad (4)$$

where $p := \Pr[y_i = 1]$ is the prevalence of the positive class. Note that accuracy depends on prevalence. Our proposed spammer score does not depend on prevalence and essentially quantifies the annotator's inherent discriminatory power. Figure 2(a) shows the contours of equal accuracy on the ROC plot. Note that annotators below the diagonal line (malicious annotators) have low accuracy. The malicious annotators flip their labels and as such are not spammers if we can detect them and then correct for the flipping. In fact the EM algorithms (Dawid and Skene, 1979; Raykar et al., 2010) can correctly flip the labels for the malicious annotators and hence they should not be treated as spammers. Figure 2(b) also shows the contours of equal score for our proposed score and it can be seen that the malicious annotators have a high score and only annotators along the diagonal have a low score (spammers).

3.2 Categorical Labels

We now extend the notion of spammers to categorial labels. As earlier a spammer assigns labels randomly, that is,

$$\Pr[y_i^j = k|y_i] = \Pr[y_i^j = k], \forall k.$$



(a) Accuracy

(b) Spammer score

Figure 2: (a) Contours of equal accuracy (4) and (b) equal spammer score (3).

This is equivalent to $\Pr[y_i^j = k | y_i = c] = \Pr[y_i^j = k | y_i = c'], \forall c, c', k = 1, \ldots, C$, which means knowing the true class label being c or c' does not change the probability of the annotator's assigned label. This indicates that the annotator j is a spammer if

$$\alpha_{ck}^j = \alpha_{c'k}^j, \forall c, c', k = 1, \ldots, C. \tag{5}$$

Let $\mathbf{A}^j$ be the $C \times C$ confusion rate matrix with entries $[\mathbf{A}^j]_{ck} = \alpha_{ck}$, a spammer would have all the rows of $\mathbf{A}^j$ equal to one another, for example, an annotator with a confusion matrix $\mathbf{A}^j = \begin{bmatrix} 0.50 & 0.25 & 0.25 \\ 0.50 & 0.25 & 0.25 \\ 0.50 & 0.25 & 0.25 \end{bmatrix}$, is a spammer for a three class categorical annotation problem. Essentially $\mathbf{A}^j$ is a rank one matrix of the form $\mathbf{A}^j = \mathbf{e}\mathbf{v}_j^\top$, for some column vector $\mathbf{v}_j \in \mathbb{R}^C$ that satisfies $\mathbf{v}_j^\top \mathbf{e} = 1$, where $\mathbf{e}$ is column vector of ones. In the binary case we had this natural notion of spammer as an annotator for whom $\alpha^j + \beta^j - 1$ was close to zero. One natural way to summarize (5) would be in terms of the distance (Frobenius norm) of the confusion matrix to the closest rank one approximation, that is,

$$\mathcal{S}^j := \|\mathbf{A}^j - \mathbf{e}\hat{\mathbf{v}}_j^\top\|_F^2, \tag{6}$$

where $\hat{\mathbf{v}}_j$ solves

$$\hat{\mathbf{v}}_j = \arg\min_{\mathbf{v}_j} \|\mathbf{A}^j - \mathbf{e}\mathbf{v}_j^\top\|_F^2 \qquad \text{subject to} \quad \mathbf{v}_j^\top \mathbf{e} = 1. \tag{7}$$

Solving (7) yields $\hat{\mathbf{v}}_j = (1/C)\mathbf{A}^{j\top}\mathbf{e}$, which is the mean of the rows of $\mathbf{A}^j$. Then from (6) we have

$$\mathcal{S}^j = \left\| \left(\mathbf{I} - \frac{1}{C}\mathbf{e}\mathbf{e}^\top \right) \mathbf{A}^j \right\|_F^2 = \frac{1}{C} \sum_{c<c'} \sum_k (\alpha_{ck}^j - \alpha_{c'k}^j)^2.$$

This is equivalent to subtracting the mean row from each row of the confusion matrix and then summing up the squares of all the entries. So a spammer is an annotator for whom $\mathcal{S}^j$ is close to zero. A perfect annotator has $\mathcal{S}^j = C-1$. We normalize this score to lie between 0 and 1.

$$\mathcal{S}^j = \frac{1}{C(C-1)} \sum_{c<c'} \sum_k (\alpha^j_{ck} - \alpha^j_{c'k})^2$$

When $C=2$ this is equivalent to the score proposed earlier for binary labels.

4. Algorithm to Consolidate Multiple Annotations

Using the spammer score proposed in the earlier section to define a prior we describe an empirical Bayesian algorithm to consolidate the multiple annotations and eliminate the spammers simultaneously. For ease of exposition we first start with binary labels and later extend it to categorical labels in Section 7.

4.1 Likelihood

Let N be the number of instances and M be the number annotators. Let $\mathcal{D} = \{y_i^1, \ldots, y_i^M\}_{i=1}^N$ be the observed annotations from the M annotators, and let $p = \Pr[y_i = 1]$ be the prevalence of the positive class. Assuming the instances are independent, the likelihood of the parameters $\boldsymbol{\theta} = [\alpha^1, \beta^1, \ldots, \alpha^M, \beta^M, p]$ given the observations $\mathcal{D}$ can be factored as $\Pr[\mathcal{D}|\boldsymbol{\theta}] = \prod_{i=1}^N \Pr[y_i^1, \ldots, y_i^M|\boldsymbol{\theta}]$. Under the assumption that the annotation labels $y_i^1, \ldots, y_i^M$ are independent given the true label y_i, the log likelihood can be written as

$$\log \Pr[\mathcal{D}|\boldsymbol{\theta}] = \sum_{i=1}^N \log \sum_{y_i=0}^{1} \prod_{j=1}^{M} \Pr[y_i^j|y_i, \boldsymbol{\theta}] \cdot \Pr[y_i|\boldsymbol{\theta}] = \sum_{i=1}^N \log \left[a_i p + b_i (1-p)\right], \tag{8}$$

where we denote

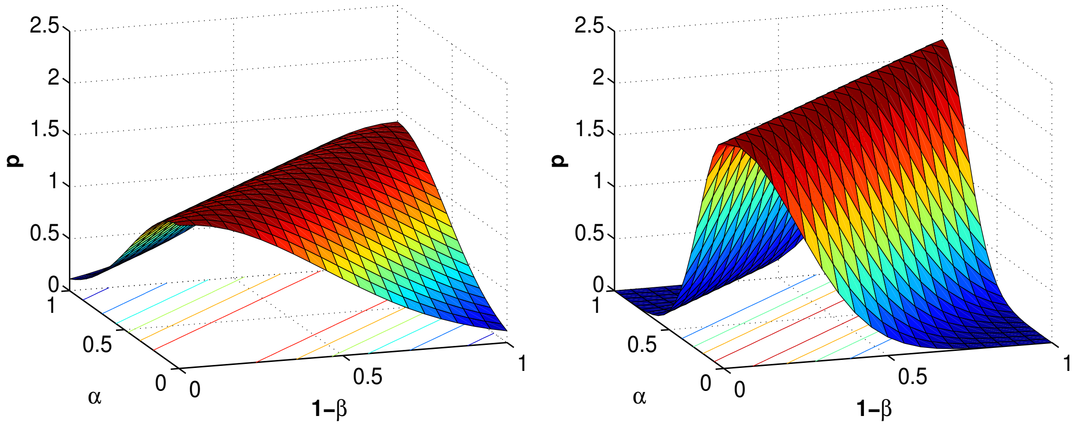
$$a_i = \prod_{j=1}^M \Pr[y_i^j | y_i = 1, \alpha^j] = \prod_{j=1}^M [\alpha^j]^{y_i^j} [1-\alpha^j]^{1-y_i^j},$$
$$b_i = \prod_{j=1}^M \Pr[y_i^j | y_i = 0, \beta^j] = \prod_{j=1}^M [\beta^j]^{1-y_i^j} [1-\beta^j]^{y_i^j}.$$

This log likelihood can be efficiently maximized by the Expectation Maximization (EM) algorithm (Dempster et al., 1977) leading to the iterative algorithm proposed in the earlier works (Dawid and Skene, 1979; Smyth et al., 1995; Raykar et al., 2010).

4.2 Automatic Spammer Detection Prior

Several authors have proposed a Bayesian approach by imposing a prior on the parameters (Raykar et al., 2009; Carpenter, 2008). For example, Raykar et al. (2009) assigned a beta prior for each α^j and β^j independently. Since we are interested in the situation when the annotations are mostly dominated by spammers, based on the score $\mathcal{S}^j$ (3) derived earlier we propose a prior called *Automatic Spammer Detection* (ASD) prior which favors the spammers. Specifically we assign the following prior to the pair $\{\alpha^j, \beta^j\}$ with a separate precision parameter $\lambda^j > 0$ (hyperparameter) for each annotator:

$$\Pr[\alpha^j, \beta^j | \lambda^j] = \frac{1}{N(\lambda^j)} \exp\left(-\frac{\lambda^j (\alpha^j + \beta^j - 1)^2}{2}\right). \tag{9}$$



(a) $\lambda = 5$

(b) $\lambda = 20$

Figure 3: The proposed Automatic Spammer Detection prior (9) for different values of λ^j.

where the normalization term N is given by (see Appendix A)

$$N(\lambda^j) \;=\; \int_0^1 \int_0^1 \exp\left(-\frac{\lambda^j(\alpha^j+\beta^j-1)^2}{2}\right) d\alpha^j d\beta^j = \sqrt{\frac{2\pi}{\lambda^j}}\left(\frac{2}{\sqrt{\lambda^j}}\int_0^{\sqrt{\lambda^j}} \Phi(t)dt - 1\right),$$

where Φ is the Gaussian cumulative distribution function. This prior is effectively a truncated Gaussian on $\alpha^j + \beta^j - 1$ with mean zero and variance $1/\lambda^j$. Figure 3 illustrates the prior for two different values of the precision parameter. When λ^j is large the prior is sharply peaked along the diagonal corresponding to the spammers on the ROC plot.

We also assume that the ASD priors for each annotator are independent. For sake of completeness we further assume a beta prior for the prevalence, that is, $\text{Beta}(p|p_1,p_2)$. Denote $\boldsymbol{\lambda} = [\lambda^1,\ldots,\lambda^M,p_1,p_2]$, we have

$$\Pr[\boldsymbol{\theta}|\boldsymbol{\lambda}] = \text{Beta}(p|p_1,p_2)\prod_{j=1}^{M}\Pr[\alpha^j,\beta^j|\lambda^j]. \tag{10}$$

4.3 Maximum-a-posteriori Estimate Via EM Algorithm

Given the log likelihood (8) and the prior (10), the task is to estimate the parameters $\boldsymbol{\theta} = [\alpha^1,\beta^1,\ldots,\alpha^M,\beta^M,p]$. The maximum-a-posteriori (MAP) estimator is found by maximizing the log-posterior, that is,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}}\{\ln\Pr[\mathcal{D}|\boldsymbol{\theta}] + \ln\Pr[\boldsymbol{\theta}]\}.$$

An EM algorithm can be derived for MAP estimation by relying on the interpretation of Neal and Hinton (1998) which is an efficient iterative procedure to compute the solution in presence of missing/hidden data. We will use the unknown hidden true label $\boldsymbol{y} = [y_1,\ldots,y_N]$ as the missing data. The complete data log-likelihood can be written as

$$\log\Pr[\mathcal{D},\boldsymbol{y}|\boldsymbol{\theta}] = \sum_{i=1}^{N}\left[y_i\log pa_i + (1-y_i)\log(1-p)b_i\right].$$

Each iteration of the EM algorithm consists of two steps: an Expectation(E)-step and a Maximization(M)-step. The M-step involves maximization of a lower bound on the log-posterior that is refined in each iteration by the E-step.

E-step: Given the observation $\mathcal{D}$ and the current estimate of the model parameters $\boldsymbol{\theta}$, the conditional expectation (which is a lower bound on the true likelihood) is computed as

$$\mathbb{E}\left\{\log \Pr[\mathcal{D}, \boldsymbol{y}|\boldsymbol{\theta}]\right\} = \sum_{i=1}^{N} \Big[\mu_i \log p a_i + (1-\mu_i)\log(1-p)b_i\Big],$$

where the expectation is with respect to $\Pr[\boldsymbol{y}|\mathcal{D},\boldsymbol{\theta}]$, and $\mu_i = \Pr[y_i = 1|y_i^1,\ldots,y_i^M,\boldsymbol{\theta}]$ is the expected label for y_i conditioned on the observed annotations and the model parameters. Using Bayes theorem we can compute

$$\mu_i \propto \Pr[y_i^1,\ldots,y_i^M|y_i = 1,\boldsymbol{\theta}]\cdot\Pr[y_i = 1|\boldsymbol{\theta}] = \frac{a_i p}{a_i p + b_i(1-p)}. \tag{11}$$

M-step: Based on the current estimate μ_i and the observations $\mathcal{D}$, we can estimate p by maximizing the lower bound on the log posterior, $\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}$, where

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta}} &= \mathbb{E}\left\{\log \Pr[\mathcal{D}, \boldsymbol{y}|\theta]\right\} + \log \Pr[\boldsymbol{\theta}|\boldsymbol{\lambda}] \\ &= \sum_{i=1}^{N} \Big[\mu_i \log p a_i + (1-\mu_i)\log(1-p)b_i\Big] + \log \text{Beta}(p|p_1,p_2) \\ &\quad - \sum_{j=1}^{M} \frac{\lambda^j}{2}(\alpha^j + \beta^j - 1)^2 - \sum_{j=1}^{M} \log N(\lambda^j).\end{aligned} \tag{12}$$

Equating the derivative of $\mathcal{L}_{\boldsymbol{\theta}}$ with respect to p to zero, we estimate p as

$$p = \frac{p_1 - 1 + \sum_{i=1}^{N}\mu_i}{p_1 + p_2 - 2 + N}. \tag{13}$$

The derivative with respect to α^j and β^j can be computed as follows:

$$\frac{\partial \mathcal{L}_{\boldsymbol{\theta}}}{\partial \alpha^j} = \frac{\sum_{i=1}^{N}\mu_i y_i^j - \alpha^j \sum_{i=1}^{N}\mu_i}{\alpha^j(1-\alpha^j)} - \lambda^j(\alpha^j + \beta^j - 1), \tag{14}$$

$$\frac{\partial \mathcal{L}_{\boldsymbol{\theta}}}{\partial \beta^j} = \frac{\sum_{i=1}^{N}(1-\mu_i)(1-y_i^j) - \beta^j \sum_{i=1}^{N}(1-\mu_i)}{\beta^j(1-\beta^j)} - \lambda^j(\alpha^j + \beta^j - 1). \tag{15}$$

Equating these derivatives to zero we obtain two cubic equations[5] involving α^j and β^j, respectively. We can iteratively solve one cubic equation (for example, for α^j) by fixing the counterpart (for

[5] The pair of cubic equations are given by

$$\begin{aligned}\lambda^j(\alpha^j)^3 + (\beta^j\lambda^j - 2\lambda^j)(\alpha^j)^2 - (\lambda^j - \beta^j\lambda^j - \sum_{i=1}^{N}\mu_i)\alpha^j + (\sum_{i=1}^{N}\mu_i y_i^j) &= 0 \\ \lambda^j(\beta^j)^3 + (\alpha^j\lambda^j - 2\lambda^j)(\beta^j)^2 - (\lambda^j - \alpha^j\lambda^j - \sum_{i=1}^{N}\mu_i)\beta^j + (\sum_{i=1}^{N}\mu_i y_i^j) &= 0\end{aligned}$$

For each equation we retain only the root that lies in the range $[0,1]$.

example, β^j) till convergence. Also note that when $\lambda^j = 0$ we get the standard EM algorithm proposed by Dawid and Skene (1979). These two steps (the E- and M-step) can be iterated till convergence. We use majority voting $\mu_i = 1/M \sum_{j=1}^{M} y_i^j$ as the initialization for μ_i to start the EM-algorithm.

5. Algorithm to Eliminate Spammers

For each annotator we imposed the Automatic Spammer Detection prior of the form $\Pr[\alpha^j, \beta^j | \lambda^j] \propto \exp\left(-\lambda^j(\alpha^j + \beta^j - 1)^2/2\right)$, parameterized by precision hyperparameter λ^j. If we know the hyperparameters $\boldsymbol{\lambda} = [\lambda^1, \ldots, \lambda^M]$ we can compute the MAP estimate efficiently via the EM algorithm as described in the previous section. However it is crucial that we use the right λ^j for each annotator for two reasons: (1) For the good annotators we want the precision term to be small so that we do not over penalize the good annotators. (2) We can use the estimated λ^j to detect spammers. Clearly, as the precision λ^j increases, that is, the variance tends to zero, thus concentrating the prior sharply around the random diagonal line in the ROC plot. Hence, regardless of the evidence of the training data, the posterior will also be sharply concentrated around $\alpha^j + \beta^j = 1$, thus that annotator will not affect the ground truth and hence, it can be effectively removed. Therefore, the discrete optimization problem corresponding to spammer detection (should each annotator be included or not?), can be more easily solved via an easier continuous optimization over hyperparameters. In this section we adopt an empirical Bayesian strategy (specifically the *type-II maximum likelihood*) to automatically learn the hyperparameters from the data itself. This is in the spirit of the commonly used automatic relevance determination (ARD) prior used for feature selection by relevance vector machine (Tipping, 2001) and Gaussian process classification (Rasmussen and Williams, 2006).

5.1 Evidence Maximization

In *type-II maximum likelihood* approach, the hyperparameters $\boldsymbol{\lambda}$ are chosen to maximize the marginal likelihood (or equivalently the log marginal likelihood), that is,

$$\widehat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \Pr[\mathcal{D}|\boldsymbol{\lambda}] = \arg\max_{\boldsymbol{\lambda}} \log \Pr[\mathcal{D}|\boldsymbol{\lambda}],$$

where the marginal likelihood $\Pr[\mathcal{D}|\boldsymbol{\lambda}]$ is essentially the *evidence* for $\boldsymbol{\lambda}$ with the parameters $\boldsymbol{\theta}$ marginalized or integrated out.

$$\Pr[\mathcal{D}|\boldsymbol{\lambda}] = \int_{\boldsymbol{\theta}} \Pr[\mathcal{D}|\boldsymbol{\theta}] \Pr[\boldsymbol{\theta}|\boldsymbol{\lambda}] d\boldsymbol{\theta}.$$

Since this integral is analytically intractable we use the Laplace method which involves a second order Taylor series approximation around the MAP estimate.

5.2 Laplace Approximation

The marginal likelihood can be rewritten as follows, $\Pr[\mathcal{D}|\boldsymbol{\lambda}] = \int_{\boldsymbol{\theta}} \exp[\Psi(\boldsymbol{\theta})] d\boldsymbol{\theta}$ where

$$\Psi(\boldsymbol{\theta}) = \log \Pr[\mathcal{D}|\boldsymbol{\theta}] + \log \Pr[\boldsymbol{\theta}|\boldsymbol{\lambda}].$$

We approximate Ψ using a second order Taylor series around the MAP estimate $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$,

$$\Psi(\boldsymbol{\theta}) \approx \Psi(\widehat{\boldsymbol{\theta}}_{\text{MAP}}) + \frac{1}{2}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{MAP}}) \mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda}) (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{MAP}})^\top,$$

where $\mathbf{H}$ is the Hessian matrix. We have made use of the fact that the gradient of Ψ evaluated at the MAP estimate $\widehat{\boldsymbol{\theta}}_{\text{MAP}}$ is zero. Hence we have the following approximation to the log-marginal likelihood.

$$\log \Pr[\mathcal{D}|\boldsymbol{\lambda}] \approx \log \Pr[\mathcal{D}|\widehat{\boldsymbol{\theta}}_{\text{MAP}}] + \log \Pr[\widehat{\boldsymbol{\theta}}_{\text{MAP}}|\boldsymbol{\lambda}] - \frac{1}{2}\log\det[-\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})] + \frac{d}{2}\log 2\pi.$$

The hyperparameters $\boldsymbol{\lambda}$ are found by maximizing this approximation to the log marginal likelihood. We use a simple iterative re-estimation method by setting the first derivative to zero. The derivative can be written as (see Appendix B for more details)

$$\frac{\partial}{\partial \lambda^j}\log \Pr[\mathcal{D}|\boldsymbol{\lambda}] \approx -\frac{1}{2}(\widehat{\alpha}^j + \widehat{\beta}^j - 1)^2 + \frac{1}{2\lambda^j}\delta(\lambda^j) - \frac{1}{2}\sigma(\lambda^j),$$

where we have defined

$$\delta(\lambda^j) = 2 - \frac{\sqrt{2\pi\lambda^j}\,\text{erf}(\sqrt{\lambda^j/2})}{\sqrt{2\pi\lambda^j}\,\text{erf}(\sqrt{\lambda^j/2}) + 2\exp(-\lambda^j/2) - 2}, \tag{16}$$

in which $\text{erf}(x) = (2/\sqrt{\pi})\int_0^x \exp(-t^2)dt$ is the error function, and

$$\sigma(\lambda^j) = \text{Tr}\left[\mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})\frac{\partial}{\partial \lambda^j}\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})\right].$$

See Appendix B for more details on computation of $\sigma(\lambda^j)$. Assuming $\delta^j = \delta(\lambda^j)$ and $\sigma^j = \sigma(\lambda^j)$ does not depend on λ^j, a simple update rule for the hyperparameters can be written by equating the first derivative to zero.[6]

$$\lambda^j = \frac{\delta^j}{(\widehat{\alpha}^j + \widehat{\beta}^j - 1)^2 + \sigma^j}. \tag{17}$$

One way to think of this is that the penalization is inversely proportional to $(\widehat{\alpha}^j + \widehat{\beta}^j - 1)^2$, that is, good annotators get penalized less while the spammers suffer a large penalization. Figure 4(b) plots the estimated hyperparameter $\widehat{\lambda}^j$ for each annotator as a function of the iteration number for a simulation setup shown in Figure 4(a). The simulation has 5 good annotators and 20 spammers. It can be seen that as expected for the good annotators $\widehat{\lambda}^j$ starts decreasing[7] while for the spammers $\widehat{\lambda}^j$ starts increasing with iterations.[8] By using a suitable pruning threshold we can detect and eliminate the spammers.

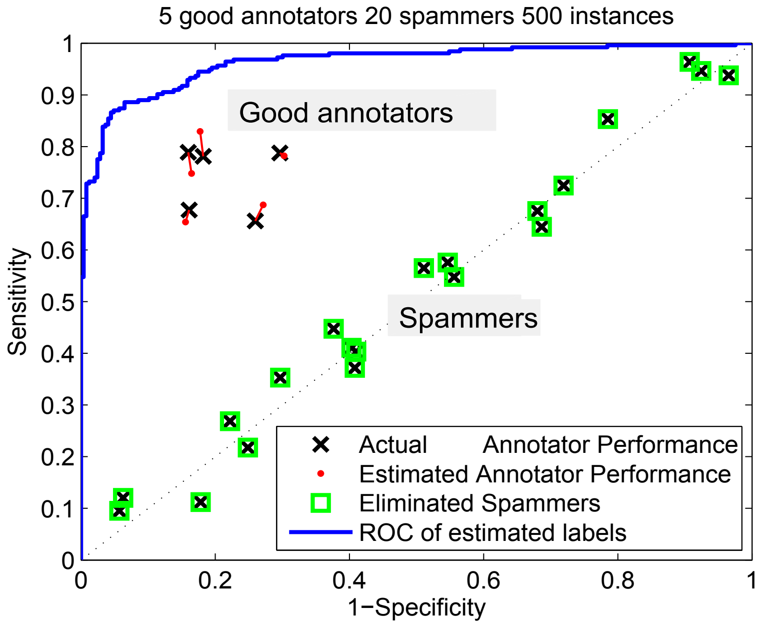
The final algorithm has two levels of iterations (see Algorithm 1): in an outer loop we update the hyper-parameters $\widehat{\lambda}^j$ and in an inner loop we find the MAP estimator for sensitivity and specificity given the hyper-parameters. At each iteration we eliminate all the annotators for whom the estimated $\widehat{\lambda}^j$ is greater than a certain pruning threshold T.[9]

[6] In practice, one can iterate (17) and (16) several times to get better estimate for λ^j.

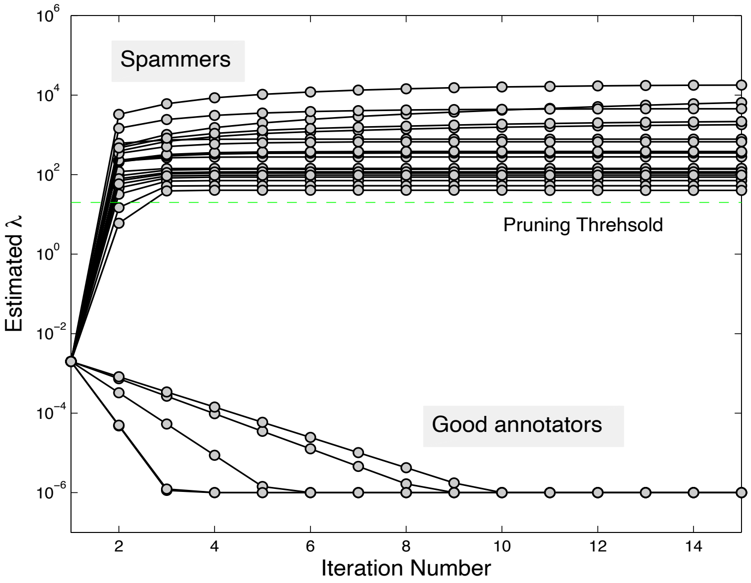
[7] For numerical stability we do not let the hyper parameter go below 10^{-6}.

[8] We have different rates of convergence for the good annotators and the spammers. This is because of our assumption that δ (16) does not depend on λ. This is almost true for large λ and is not a good approximation for small λ.

[9] For all our experiments for each annotator we set the pruning threshold to 0.1 times the number of instances labeled by him.



(a) Setup



(b) Estimated Hyperparameters

Figure 4: *Illustration of spammer elimination via evidence maximization* (a) The black cross plots the actual sensitivity and specificity of each annotator. The simulation has 5 good annotators and 20 spammers and 500 instances. The red dot plots the sensitivity and specificity as estimated by the SpEM algorithm. The green squares show the annotators eliminated as spammers. (b) The estimated hyperparameter λ^j for each annotator as a function of the iteration number. The pruning threshold is also shown on the plot.

Algorithm 1 SpEM

```
Require: Annotations $y_i^j \in \{0,1\}$, $j = 1,\dots,M$, $i = 1,\dots,N$ from $M$ annotators on $N$ instances.
Initialize $\lambda^j = 1/N$, for $j = 1,\dots,M$.
Initialize $\mathcal{A} = \{1,\dots,M\}$ the set of good annotators.
Initialize $\mu_i = 1/M \sum_{j=1}^{M} y_i^j$ using soft majority voting.
repeat {Outer loop with evidence maximization}
  repeat {EM loop}
    {M-step}
    Update $p$ based on (13).
    Update $\alpha^j$, $\beta^j$ based on (14)-(15), $\forall j \in \mathcal{A}$.
    {E-step}
    Estimate $\mu_i$ using (11), $\forall i = 1,\dots,N$.
  until Change of expected posterior (12) $< \varepsilon_1$.
  {Evidence Maximization}
  for all $j \in \mathcal{A}$ do
    Update $\lambda^j$ based on (17).
    if $\lambda^j > T$ (the pruning threshold) then
      $\mathcal{A} \leftarrow \mathcal{A} \backslash \{j\}$
    end if
  end for
until Change of expected posterior (12) $< \varepsilon_2$.
Ensure: Detected spammers in set $\{1,\dots,M\} \backslash \mathcal{A}$.
Ensure: Non-spammers in $\mathcal{A}$ with sensitivity $\alpha^j$ and specificity $\beta^j$, for $j \in \mathcal{A}$.
Ensure: Prevalence factor $p$ and expected hidden label $\mu_i$, $\forall i = 1,\dots,N$.
```

In all our experiments we set the convergence tolerance $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. The pruning threshold was set to $T = 0.1N$.

6. Discussions

1. **Can we use the EM algorithm directly to eliminate spammers?** Majority Voting and EM algorithm do not have a mechanism to explicitly detect spammers. However we could define an annotator as a spammer if the estimated $|\widehat{\alpha}^j + \widehat{\beta}^j - 1| \leq \varepsilon$. However it is not clear what is the right ε to use. Also the spammers influence the estimation of $\widehat{\alpha}^j$ and $\widehat{\beta}^j$ for the good annotators. A fix to this would be to eliminate the spammers and get an improved estimate of the ground truth. In principle this process could be repeated till convergence, which essentially boils down to a discrete version of our proposed SpEM algorithm.

2. **What is the advantage of different shrinkage for each annotator ?** We could have imposed a common shrinkage prior (that is, same $\lambda^j \equiv \lambda$ for all annotators) and then estimated one λ as shown earlier. While this is a valid approach, the advantage of our ASD prior is that the amount of shrinkage for each annotator is different and depends on how good the annotator is, that is, good annotators suffer less shrinkage while spammers suffer severe shrinkage.

3. **Missing annotations** The proposed SpEM algorithm can be easily extended to handle missing annotations (which is more realistic scenario in crowdsourcing marketplaces). Let M_i be the number of annotators labeling the i^{th} instance, and let N_j be the number of instances labeled

by the j^{th} annotator. Then in the EM loop, we just need to replace N by N_j for estimating α^j and β^j in (14) and (15), and replace M by M_i for updating μ_i in (11).

4. **Training a classifier directly** The proposed algorithm can be readily extended to learn a classifier along with the ground truth (Raykar et al., 2009). Let instance i have features $\boldsymbol{x}_i \in \mathbb{R}^d$, and define the classification problem as learning $\boldsymbol{w} \in \mathbb{R}^d$ such that $\Pr[y_i = 1|\boldsymbol{x}_i, \boldsymbol{w}] = p_i = f(\boldsymbol{w}^\top \boldsymbol{x}_i)$, with f a mapping function (for example, logistic function). To learn $\boldsymbol{w}$ in SpEM we just need to replace (13) with a Newton-Raphson step to update $\boldsymbol{w}$, and replace p with p_i in (11).

5. **Partially known gold standard** If the actual ground truth is available for some instances, SpEM can readily incorporate them into the learning loop. The only change we need to make is to estimate μ_i in (11) only for the instances for which the ground truth is not available, and fix $\mu_i = y_i$ if the ground truth y_i is available. Therefore, the gold standard instances and unlabeled instances will be used together to estimate the sensitivity and specificity of each annotator (and also to estimate the labels).

7. Extension to Categorical Annotations

We now extend the proposed algorithm to handle categorical annotations. A simple solution for categorical outcomes is to use a one-against-all strategy and run the binary SpEM C times, each time obtaining a spammer indicator λ^j for each annotator. One might then identify an annotator j as a spammer if all of the λ^j in the C runs indicate that this is a spammer. However in this section we provide a more principled solution in line with the framework proposed for binary labels. Following the same motivation as before, we define the ASD prior as follows

$$\Pr[\mathbf{A}^j|\lambda^j] = \frac{1}{N(\lambda^j)} \exp\left(-\frac{\lambda^j}{2C} \sum_{c<c'} \sum_{k=1}^{C} \left(\alpha^j_{ck} - \alpha^j_{c'k}\right)^2\right),$$

which gives more probability mass to a spammer. A similar EM algorithm can be developed under this prior, and evidence maximization follows naturally with Laplace approximation. Under the same assumptions as earlier, the log-likelihood of the parameters $\boldsymbol{\theta} = [\mathbf{A}^1, \ldots, \mathbf{A}^M, p_1, \ldots, p_C]$ is

$$\log \Pr[\mathcal{D}|\boldsymbol{\theta}] = \sum_{i=1}^{N} \log \left[\sum_{c=1}^{C} \Pr(y_i = c) \prod_{j=1}^{M} \Pr(y_i^j | y_i = c)\right] = \sum_{i=1}^{N} \log \left[\sum_{c=1}^{C} p_c \prod_{j=1}^{M} \prod_{k=1}^{C} (\alpha^j_{ck})^{\delta(y_i^j, k)}\right],$$

where $p_c = \Pr(y_i = c)$ and $\delta(u, v) = 1$ if $u = v$ and 0 otherwise. If we know the missing labels y the complete log likelihood can be written as

$$\log \Pr[\mathcal{D}, y|\boldsymbol{\theta}] = \sum_{i=1}^{N} \sum_{c=1}^{C} \delta(y_i, c) \log \left[p_c \prod_{j=1}^{M} \prod_{k=1}^{C} (\alpha^j_{ck})^{\delta(y_i^j, k)}\right].$$

In the E-step we compute the conditional expectation as

$$\mathbb{E}\left\{\log \Pr[\mathcal{D}, \boldsymbol{y}|\boldsymbol{\theta}]\right\} = \sum_{i=1}^{N} \sum_{c=1}^{C} \mu_{ic} \log \left[p_c \prod_{j=1}^{M} \prod_{k=1}^{C} (\alpha^j_{ck})^{\delta(y_i^j, k)}\right]$$

where $\mu_{ic} = \Pr[y_i = c | y_i^1, \ldots, y_i^M, \boldsymbol{\theta}]$ and is computed as $\mu_{ic} \propto p_c \prod_{j=1}^M \prod_{k=1}^C (\alpha_{ck}^j)^{\delta(y_i^j,k)}$. Based on the current estimate μ_{ic} in the M-step we can estimate the parameters by maximizing the lower bound on the log posterior (along with the Lagrange multipliers γ), $\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}$, where

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^N \sum_{c=1}^C \mu_{ic} \left[\log p_c + \sum_{j=1}^M \sum_{k=1}^C \delta(y_i^j,k) \log \alpha_{ck}^j \right] \\ &- \sum_{j=1}^M \frac{\lambda^j}{2C} \sum_{c<c'} \sum_{k=1}^C \left(\alpha_{ck}^j - \alpha_{c'k}^j\right)^2 - \sum_{j=1}^M \log N(\lambda^j) + \sum_{j=1}^M \sum_{c=1}^C \gamma_c^j \left(1 - \sum_{k=1}^C \alpha_{ck}^j\right).\end{aligned}$$

We update the prevalence as $p_c = (1/N) \sum_{i=1}^N \mu_{ic}$ and for the α_{ck}^j we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \alpha_{ck}^j} &= \frac{\sum_{i=1}^N \mu_{ic} \delta(y_i^j,k)}{\alpha_{ck}^j} - \frac{\lambda^j}{C} \sum_{c' \neq c} \left(\alpha_{ck}^j - \alpha_{c'k}^j\right) - \gamma_c^j = 0, \qquad (18) \\ \frac{\partial \mathcal{L}}{\partial \gamma_c^j} &= \sum_{k=1}^C \alpha_{ck}^j - 1 = 0.\end{aligned}$$

The practical solution to solve[10] this for every α_{ck}^j is to fix the $\alpha_{c'k}^j$ for $c' \neq c$, solve the equation array with a fixed γ_c^j, and then update γ_c^j as

$$\gamma_c^j = \frac{1}{C} \sum_{k=1}^C \frac{\sum_{i=1}^N \mu_{ic} \delta(y_i^j,k)}{\alpha_{ck}^j},$$

which follows by summing (18) for all k. As earlier in order to determine the hyperparameters we obtain a simple iterative update by setting the derivative of the approximate log-marginal likelihood to zero.

$$\frac{\partial}{\partial \lambda^j} \log \Pr[\mathcal{D} | \boldsymbol{\lambda}] \approx -\frac{1}{2C} \sum_{c<c'} \sum_{k=1}^C \left(\alpha_{ck}^j - \alpha_{c'k}^j\right)^2 - \frac{1}{N(\lambda^j)} \frac{\partial}{\partial \lambda^j} N(\lambda^j) - \frac{1}{2} \sigma(\lambda^j),$$

where we have defined

$$\sigma(\lambda) = \text{Tr}\left[\mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda}) \frac{\partial}{\partial \lambda} \mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})\right].$$

and

$$-\frac{1}{N(\lambda^j)} \frac{\partial}{\partial \lambda^j} N(\lambda^j) = \frac{1}{2\lambda^j} \delta(\lambda^j).$$

See Appendix C for more details on computation of σ and δ. Then the update is given by

$$\widehat{\lambda}^j = \frac{\delta(\lambda^j)}{(1/C) \sum_{c<c'} \sum_{k=1}^C \left(\alpha_{ck}^j - \alpha_{c'k}^j\right)^2 + \sigma(\lambda^j)}.$$

[10] Keeping all terms except α_{ck}^j fixed this is a quadratic equation $A(\alpha_{ck}^j)^2 + B(\alpha_{ck}^j) + C = 0$ where $A = \frac{\lambda^j(C-1)}{C}$, $B = \gamma_c^j - \frac{\lambda^j}{C} \sum_{c' \neq c} \alpha_{c'k}^j$, and $C = -\sum_{i=1}^N \mu_{ic} \delta(y_i^j,k)$. We keep the root which lies between 0 and 1.

8. Experimental Validation

We first experimentally validate the proposed algorithm on simulated data. Figure 5(a) shows the simulation setup consisting of 5 good annotators (shown as red squares) and 100 spammers (shown as black crosses). The good annotators have sensitivity and specificity between 0.65 and 0.85. All the spammers lie around the diagonal. We compare our proposed SpEM algorithm against the commonly used Majority Voting and the EM algorithm (Dawid and Skene, 1979; Smyth et al., 1995; Raykar et al., 2009, 2010). All these methods estimate a probabilistic version ([0 1]) of the binary ground truth ($\{0,1\}$). Since we simulate the instances we know the actual binary ground truth and hence can compute the area under the ROC curve (AUC) of the estimated probabilistic ground truth.

8.1 Effect of Increasing Spammers

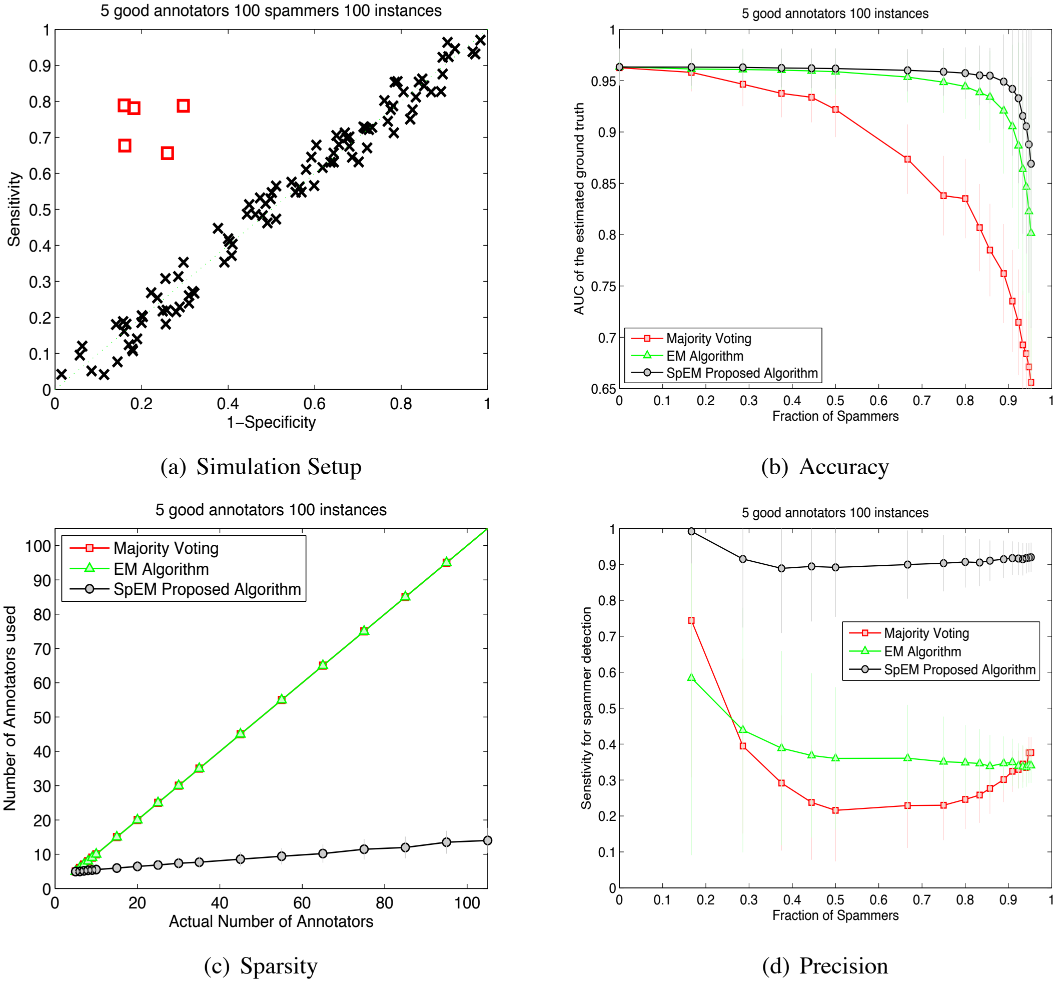
For the first experiment we deliberately choose 100 instances (with prevalence $p = 0.5$), since it is beneficial if we can detect the spammers with fewer instances. Figure 5(b) plots AUC of the estimated probabilistic ground truth as a function of the fraction of spammers (number of spammers/total number of annotators), for each point we keep all the five good annotators and keep adding more annotators from the pool of 100 spammers. All plots show the mean and one standard deviation error bars (over 100 repetitions). The pruning threshold for the SpEM algorithm was set to 20. Figure 5(d) plots the sensitivity for spammer detection which is essentially the fraction of spammers correctly detected. The following observations can be made:

1. As the fraction of spammers increases the performance of the Majority Voting degrades drastically as compared to the EM and the SpEM algorithm (refer Figure 5(b)). The proposed SpEM algorithm has a better AUC than the EM algorithm especially when the spammers dominate (when the fraction of spammers is greater than 0.7 in Figure 5(b)). The variability (one standard deviation error bars) for all the methods increases as the number of spammers increases.

2. The clear advantage of the proposed SpEM algorithm can be seen in Figure 5(d) where it can identify almost 90% of the spammers correctly as compared the EM algorithm with can identify about 40% correctly. Majority Voting and EM algorithm do not have a mechanism to explicitly detect spammers, we define an annotator as a spammer if the estimated $|\widehat{\alpha}^j + \widehat{\beta}^j - 1| \leq \varepsilon$ (We have used $\varepsilon = 0.05$ in our experiments.[11].)

3. The SpEM algorithm iteratively eliminates the spammers and then re-estimates the ground truth without the spammers. Figure 5(c) plots the actual number of annotators that were used in the final model. Note that the EM and the Majority Voting use all the annotators to estimate the model parameters while the SpEM algorithm uses only a small fraction of the annotators.

To summarize, the proposed SpEM algorithm is slightly more accurate than the EM algorithm and at the same time uses a small fraction of the annotators thus effectively eliminating most of the spammers.



(a) Simulation Setup

(b) Accuracy

(c) Sparsity

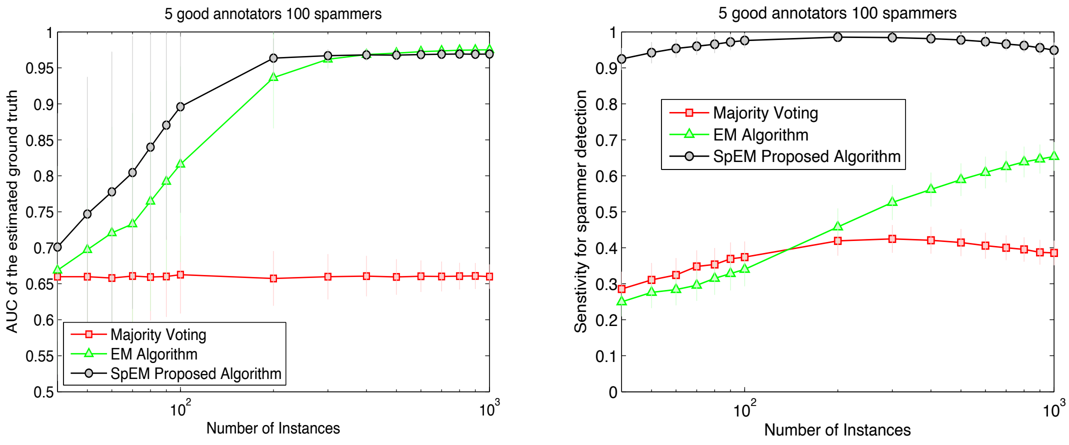
(d) Precision

Figure 5: *Effect of increasing the number of spammers* (Section 8.1) (a) The simulation setup has 5 good annotators (red squares) and 100 spammers (black crosses) and 100 instances. (b) The AUC of the estimated ground truth as a function of the fraction of spammers. (c) The actual number of annotators that were used. (d) The fraction of spammers correctly detected. All plots show mean and one standard deviation error bars (over 100 repetitions).

8.2 Effect of Increasing Instances

For the proposed algorithm to be practically useful we would like to detect the spammers with as few examples as possible so that they can be eliminated early on. Figure 6 plots the performance for the same setup as earlier as a function of the number of instances. From Figure 6(a) we see that the AUC for the proposed method is much better than the EM algorithm especially for smaller number of instances. As the number of instances increases the accuracy of the EM algorithm is as good as the proposed SpEM algorithm. The EM algorithm (and also the proposed SpEM) automatically gives

[11] The 0.05 value is just a heuristic based on a band around the diagonal of the ROC plot.



(a) Accuracy

(b) Sparsity

Figure 6: *Effect of increasing the number of instances* (Section 8.2) (a) The AUC of the estimated ground truth as a function of the number of instances. (b) The fraction of actual spammers that were eliminated. All plots show the mean and one standard deviation error bars (over 100 repetitions). The simulation setup has 5 good annotators and 100 spammers. The pruning threshold was set to $0.1N$ where N is the total number of instances.

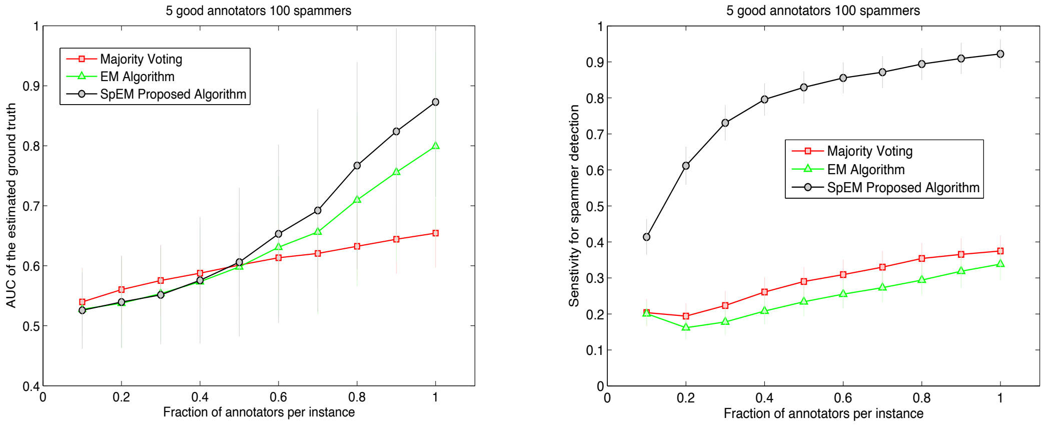
less emphasis for annotators with small $|\widehat{\alpha}^j + \widehat{\beta}^j - 1|$. The reason SpEM achieves better accuracy is that the parameters $\widehat{\alpha}^j$ and $\widehat{\beta}^j$ are better estimated because of the ASD prior we imposed. This also explains the fact that when we have a large number of instances both the EM and SpEM algorithm estimate the parameters equally well. The main benefit can be seen in Figure 6(b) where the SpEM algorithm can eliminate most of the spammers. For example with just 50 examples the SpEM algorithm can detect $> 90\%$ of the spammers and at the same time achieve a higher accuracy.

8.3 Effect of Missing Labels

In a realistic scenario an annotator does not label all the instances. Figure 7 plots the behavior of the different algorithms as a function of the fraction of annotators labeling each instance. When each annotator labels only a few instances all three algorithms achieve very similar performance in terms of the AUC. However the proposed SpEM algorithm can still eliminate more spammers then the EM algorithm.

8.4 Effect of Prevalence

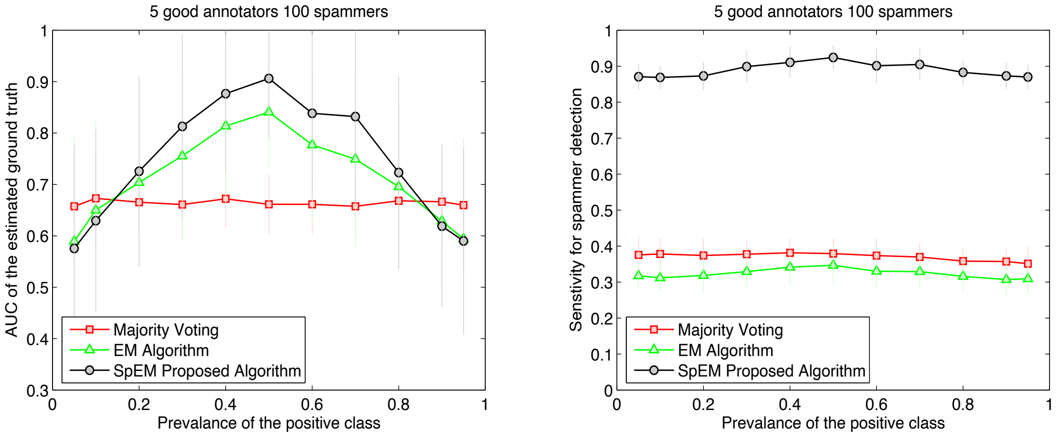
Figure 8 plots the behavior of the different algorithms as a function of the prevalence of the positive class. Note that when the prevalence is low the majority voting seems superior to other algorithms in terms of AUC. When the prevalence is small (or large) there are very few examples to reliably estimate the sensitivity (or specificity).



(a) Accuracy

(b) Precision

Figure 7: *Effect of missing labels* (Section 8.3) (a) The AUC of the estimated labels as a function of the fraction of annotators labeling each instance. (b) The fraction of actual spammers that were eliminated. All plots show the mean and one standard deviation error bars (over 100 repetitions). The simulation setup has 5 good annotators and 50 spammers.



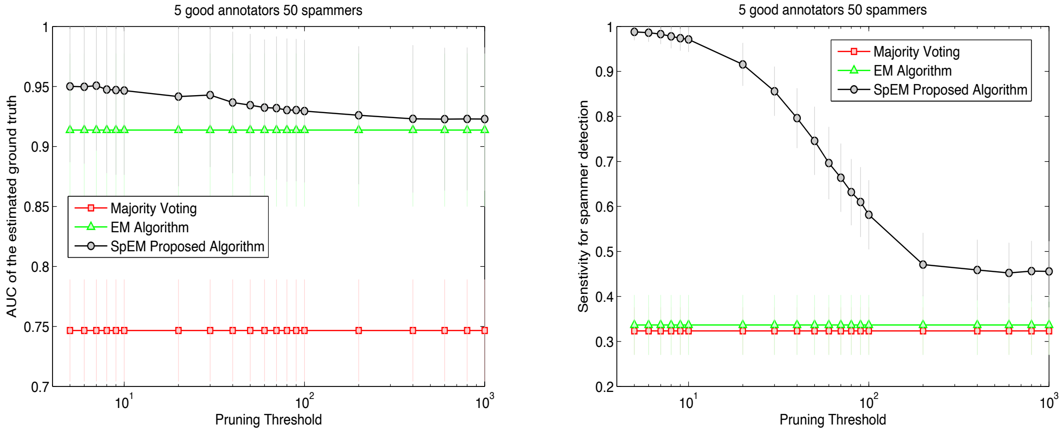
(a) Accuracy

(b) Precision

Figure 8: *Effect of prevalence* (Section 8.4) (a) The AUC of the estimated labels as a function of the prevalence of the positive class. (b) The fraction of actual spammers that were eliminated. All plots show the mean and one standard deviation error bars (over 100 repetitions). The simulation setup has 5 good annotators and 50 spammers.

8.5 Effect of Pruning Threshold

The only tunable parameter of the SpEM algorithm is the pruning threshold. For all our experiments for each annotator we set the pruning threshold to 0.1 times the number of instances labeled by the



(a) Accuracy

(b) Precision

Figure 9: *Effect of pruning threshold* (Section 8.5) (a) The AUC of the estimated labels as a function of the pruning threshold. (b) The fraction of actual spammers that were eliminated. All plots show the mean and one standard deviation error bars (over 100 repetitions). The simulation setup has 5 good annotators and 50 spammers.

annotator. However we can use this parameter to control the number of annotators we want to use. Figure 9 plots the performance for the same setup as earlier for different pruning thresholds. From Figure 9(b) we see that as the pruning threshold decreases the sensitivity for spammer elimination increases thereby using less annotators. Interestingly the accuracy also increases. If we had imposed a common shrinkage prior (that is, same λ^j for all annotators) then we would expect a drop in accuracy as the model becomes more sparse. The advantage of our ASD prior is that the amount of shrinkage for each annotator is different and depends on how accurate the annotator is, more accurate annotators suffer less shrinkage while spammers suffer severe shrinkage.

8.6 Experiments On Crowdsourced Data

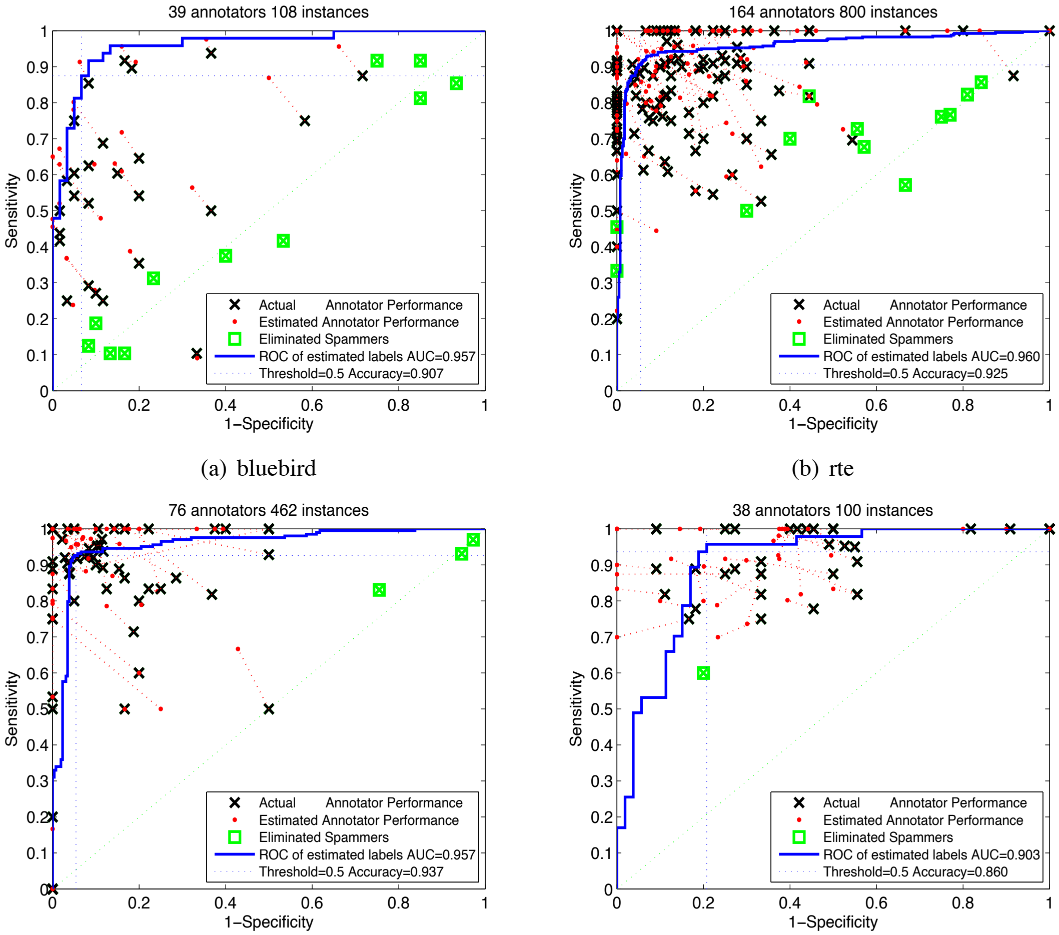
We report results on some publicly available linguistic and image annotation data collected using the Amazon Mechanical Turk and other sources. Table 1 summarizes the data sets along with a brief description of the tasks. Table 2 summarizes the results for the binary data sets with known ground truth. We compare the proposed SpEM, EM (Dawid and Skene, 1979; Raykar et al., 2010), and the Majority Voting (MV) algorithm in terms of AUC and accuracy. To compute the accuracy we use a threshold of 0.5 on the estimated probabilities. In terms of the AUC all three algorithms have similar performance. In terms of accuracy the SpEM and EM were better than the MV algorithm. The table also shows the number of annotators eliminated as spammers by the proposed algorithm. Figure 10 plots the actual and the estimated annotator performance for the SpEM algorithm for binary data sets with known ground truth.

Data Set	Type	N	M	M^*	N^*	Brief Description
bluebird	binary	108	39	39/39	108/108	bird identification (Welinder et al., 2010) The annotator had to identify whether there was a Indigo Bunting or Blue Grosbeak in the image.
rte	binary	800	164	10/10	49/20	recognizing textual entailment (Snow et al., 2008) The annotator is presented with two sentences and given a binary choice of whether the second hypothesis sentence can be inferred from the first.
temp	binary	462	76	10/10	61/16	event annotation (Snow et al., 2008) Annotators are presented with a dialogue and a pair of verbs from the dialogue, and need to label whether the event described by the first verb occurs before or after the second.
localview×	binary	832	97	5/5	43/14	word sense disambiguation (Parent and Eskenazi, 2010) Workers were asked to indicate if two definitions of a word were related to the same meaning or different meanings.
valence	ordinal	100	38	10/10	26/20	affect recognition (Snow et al., 2008) Each annotator is presented with a short headline and asked to rate the overall positive or negative valence of the emotional content of the headline.
sentiment×	categorical/3	1660	33	6/6	291/175	Irish economic sentiment analysis (Brew et al., 2010) Articles from three Irish online news sources were annotated by a group of 33 volunteer users, who were encouraged to label the articles as positive, negative, or irrelevant.

Table 1: *Data Sets*. N is the number of instances and M is the number of annotators. M^* is the mean/median number of annotators per instance. N^* is the mean/median number of instances labeled by each annotator. All the data sets except those marked × have a known gold standard. Except sentiment data set all others were collected using Amazons's Mechanical Turk. The valence data set was converted to a binary scale in our experiments.

Data	Spammers	AUC			Accuracy		
	S	SpEM	EM	MV	SpEM	EM	MV
bluebird	11/39	.96	.95	.88	.91	.90	.76
rte	12/164	.96	.96	.96	.93	.93	.92
temp	3/76	.96	.96	.97	.94	.94	.94
valence	1/38	.90	.91	.94	.86	.86	.80
localview×	12/97	-	-	-	-	-	-
sentiment×	1/33	-	-	-	-	-	-

Table 2: *Comparison of the various methods for the data sets in Table 1*. SpEM is the proposed algorithm, EM is the algorithm proposed in Dawid and Skene (1979) and Raykar et al. (2010), and MV is the soft majority voting algorithm. S is the number of annotators eliminated as spammers by the proposed algorithm. The accuracy and AUC are shown only for data sets with known gold standard.



(a) bluebird

(b) rte

(c) temp

(d) valence

Figure 10: *SpEM results for binary datsets shown in Table 2* The black cross plots the actual sensitivity and specificity of each annotator. The red dot plots the sensitivity and specificity estimated by the SpEM algorithm. The green squares show the annotators eliminated as spammers. We plot the ROC for the estimated ground truth and the operating point corresponding to a threshold of 0.5.

8.7 Ranking Annotators

The proposed spammer score can be used to rank the annotators. Figure 11 plots the spammer scores and rankings obtained for four data sets. The mean and the 95% CI obtained via bootstrapping are also shown. The number at the top of the CI bar shows the number of instances annotated by that annotator. The rankings are based on the lower limit of the 95% CI which factors the number of instances labeled by the annotator into the ranking. An annotator who labels only a few instances will have very wide CI. Some annotators who label only a few instances may have a high mean spammer score but the CI will be wide and hence ranked lower. Ideally we would like to have annotators with a high score and at the same time label a lot of instances so that we can reliably identify them. The authors (Brew et al., 2010) for the sentiment data set shared with us some of the qualitative observations regarding the annotators and they somewhat agree with our rankings. For example the authors made the following comments about Annotator 7 *"Quirky annotator - had a lot of debate about what was the meaning of the annotation question. I'd say he changed his labeling strategy at least once during the process"*. Our proposed score gave a low rank to this annotator.

9. Conclusion

In this paper we formalized the notion of a spammer for binary and categorical annotations. Using the score to define a prior we proposed an empirical Bayesian algorithm called SpEM that simultaneously estimates the consensus ground truth and also eliminates the spammers. Experiments on simulated and real data show that the proposed approach is better than (or as good as) the earlier approaches in terms of the accuracy and uses a significantly smaller number of annotators.

Appendix A. ASD Prior Normalization

In this appendix we analytically derive the normalization term for the proposed ASD prior. The normalization term $N(\lambda^j)$ can be computed as

$$\begin{aligned}
N(\lambda^j) &= \int_0^1 \int_0^1 \exp\left(-\frac{\lambda^j(\alpha^j+\beta^j-1)^2}{2}\right) d\alpha^j d\beta^j \\
&= \int_0^1 \left[\int_0^1 \sqrt{\frac{2\pi}{\lambda^j}} \mathcal{N}\left(\beta^j; 1-\alpha^j, \frac{1}{\lambda^j}\right) d\beta^j\right] d\alpha^j \\
&= \sqrt{\frac{2\pi}{\lambda^j}} \left[\int_0^1 \Phi(\sqrt{\lambda^j}\alpha^j) d\alpha^j - \int_0^1 \Phi[\sqrt{\lambda^j}(\alpha^j-1)] d\alpha^j\right],
\end{aligned}$$

where $\Phi(x) = (1/\sqrt{2\pi})\int_{-\infty}^{x} \exp(-t^2/2)dt$ is the Gaussian cumulative distribution function and $\mathcal{N}(x;u,v)$ the Gaussian distribution of x with mean u and variance v. Using the fact that $\int \Phi(x)dx = x\Phi(x) + \phi(x)$, where ϕ is the standard normal and $\Phi(x) = (1/2)[1+\text{erf}(t/\sqrt{2})]$ the normalization

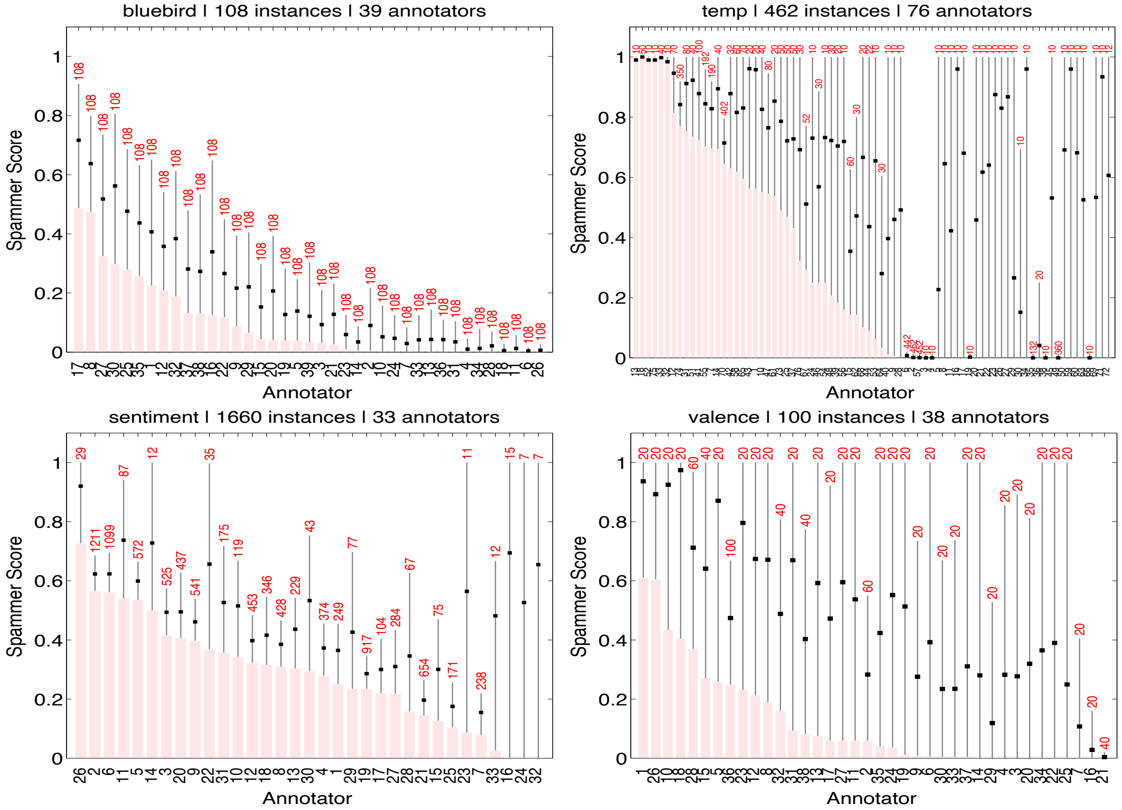


Figure 11: *Annotator Rankings* The rankings obtained for the data sets in Table 1. The spammer score ranges from 0 to 1, the lower the score, the more spammy the annotator. The mean spammer score and the 95% confidence intervals (CI) are shown, obtained from 100 bootstrap replications. The annotators are ranked based on the lower limit of the 95% CI. The number at the top of the CI bar shows the number of instances annotated by that annotator. Note that the CIs are wider when the annotator labels only a few instances.

term can be further simplified as follows,

$$
\begin{aligned}
N(\lambda^j) &= \frac{\sqrt{2\pi}}{\lambda^j}\left(\sqrt{\lambda^j}(2\Phi(\sqrt{\lambda^j})-1)+2\phi(\sqrt{\lambda^j})-2\phi(0)\right)\\
&= \frac{\sqrt{2\pi}}{\lambda^j}\left(\sqrt{\lambda^j}\mathrm{erf}(\sqrt{\lambda^j/2})+2\phi(\sqrt{\lambda^j})-2\phi(0)\right)\\
&= \frac{1}{\lambda^j}\left(\sqrt{2\pi\lambda^j}\mathrm{erf}(\sqrt{\lambda^j/2})+2\exp(-\lambda^j/2)-2\right)\\
&= \sqrt{\frac{2\pi}{\lambda^j}}\left(\frac{2}{\sqrt{\lambda^j}}\int_0^{\sqrt{\lambda^j}}\Phi(t)dt-1\right).
\end{aligned}
$$

Appendix B. Derivatives of the Log-marginal—Binary Case

The derivative of the approximation to the log-marginal likelihood can be written as

$$\begin{aligned}\frac{\partial}{\partial \lambda^j} \log \Pr[\mathcal{D}|\boldsymbol{\lambda}] &\approx \frac{\partial}{\partial \lambda^j} \log \Pr[\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}|\boldsymbol{\lambda}] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}},\boldsymbol{\lambda})\frac{\partial}{\partial \lambda^j}\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}},\boldsymbol{\lambda})\right] \\ &= \frac{\partial}{\partial \lambda^j} \log \Pr[\widehat{\alpha}^j,\widehat{\beta}^j|\lambda^j] - \frac{1}{2}\sigma(\lambda^j)\end{aligned}$$

where we have defined $\sigma(\lambda) = \mathrm{Tr}\left[\mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}},\boldsymbol{\lambda})\frac{\partial}{\partial\lambda}\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}},\boldsymbol{\lambda})\right]$. From the ASD prior we can show that

$$\frac{\partial}{\partial \lambda^j} \log \Pr[\widehat{\alpha}^j,\widehat{\beta}^j|\lambda^j] = -\frac{1}{2}(\widehat{\alpha}^j+\widehat{\beta}^j-1)^2 + \frac{1}{2\lambda^j}\delta(\lambda^j),$$

where we have defined

$$\delta(\lambda) = \left[2 - \frac{\sqrt{2\pi\lambda}\,\mathrm{erf}(\sqrt{\lambda/2})}{\sqrt{2\pi\lambda}\,\mathrm{erf}(\sqrt{\lambda/2}) + 2\exp(-\lambda/2) - 2}\right].$$

To compute $\sigma(\lambda^j)$, let us compute the Hessian matrix first. Since $\log \Pr[\mathcal{D}|\boldsymbol{\theta}]$ is again not tractable, we use the following lower bound (as used by the EM algorithm earlier) to compute the likelihood term:

$$\log \Pr[\mathcal{D}|\boldsymbol{\theta}] \geq \sum_{i=1}^{N}\left[\mu_i \log p a_i + (1-\mu_i)\log(1-p)b_i\right],$$

where $\mu_i = \Pr[\hat{y}_i|\boldsymbol{\theta}]$ is the expected class label for item i (calculated in the E-step). Then we have

$$\begin{aligned}\Psi(\boldsymbol{\theta}) &= \log \Pr[\mathcal{D}|\boldsymbol{\theta}] + \log \Pr[\boldsymbol{\theta}|\boldsymbol{\lambda}] \\ &\approx \sum_{i=1}^{N}\left[\mu_i \log p a_i + (1-\mu_i)\log(1-p)b_i\right] - \sum_{j=1}^{M}\frac{\lambda^j}{2}(\alpha^j+\beta^j-1)^2 - \sum_{j=1}^{M}\log C(\lambda^j).\end{aligned}$$

Note that this is equal to $\mathcal{L}_{\boldsymbol{\theta}}$ as defined in (12). The first-order derivatives with respect to α^j and β^j are:

$$\begin{aligned}\frac{\partial \Psi(\boldsymbol{\theta})}{\partial \alpha^j} &= \frac{\sum_{i=1}^{N}\mu_i y_i^j - \alpha^j\sum_{i=1}^{N}\mu_i}{\alpha^j(1-\alpha^j)} - \lambda^j(\alpha^j+\beta^j-1), \\ \frac{\partial \Psi(\boldsymbol{\theta})}{\partial \beta^j} &= \frac{\sum_{i=1}^{N}(1-\mu_i)(1-y_i^j) - \beta^j\sum_{i=1}^{N}(1-\mu_i)}{\beta^j(1-\beta^j)} - \lambda^j(\alpha^j+\beta^j-1).\end{aligned}$$

The second-order derivatives are:

$$\frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \alpha^j \partial \alpha^j} = \frac{\sum_i \mu_i y_i^j \cdot (2\alpha^j-1) - (\alpha^j)^2\sum_i \mu_i}{[\alpha^j(1-\alpha^j)]^2} - \lambda^j \tag{19}$$

$$\frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \alpha^j \partial \beta^j} = \frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \beta^j \partial \alpha^j} = -\lambda^j$$

$$\frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \beta^j \partial \beta^j} = \frac{\sum_i (1-\mu_i)(1-y_i^j)\cdot(2\beta^j-1) - (\beta^j)^2\sum_i(1-\mu_i)}{[\beta^j(1-\beta^j)]^2} - \lambda^j \tag{20}$$

$$\frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \alpha^j \partial \alpha^k} = \frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \alpha^j \partial \beta^k} = \frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \beta^j \partial \alpha^k} = \frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \beta^j \partial \beta^k} = 0,\ \forall j \neq k.$$

If we arrange all the parameters column-wise as a vector $\{\alpha^1, \beta^1, \ldots, \alpha^M, \beta^M\}$, then the Hessian matrix can be written in a block diagonal form $\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda}) = \mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \mathbf{B}(\boldsymbol{\lambda})$, where matrix $\mathbf{A}$ (a diagonal matrix with entries equal to the first terms in (19) and (20)) depends on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ only, and matrix $\mathbf{B}$ is a block diagonal matrix of the form

$$\mathbf{B}(\boldsymbol{\lambda}) = \begin{pmatrix} \lambda^1 & \lambda^1 & 0 & 0 & \cdots & 0 & 0 \\ \lambda^1 & \lambda^1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \lambda^2 & \lambda^2 & \cdots & 0 & 0 \\ 0 & 0 & \lambda^2 & \lambda^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda^M & \lambda^M \\ 0 & 0 & 0 & 0 & \cdots & \lambda^M & \lambda^M \end{pmatrix}.$$

It is now easy to take the derivative of $\mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})$ with respect to λ^j and compute $\sigma(\lambda^j)$. Let $\boldsymbol{\Sigma} = \mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})$, a block diagonal matrix, then we have we have

$$\sigma(\lambda^j) = \text{Tr}\left[\mathbf{H}^{-1}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda}) \frac{\partial}{\partial \lambda^j} \mathbf{H}(\widehat{\boldsymbol{\theta}}_{\text{MAP}}, \boldsymbol{\lambda})\right] = -\boldsymbol{\Sigma}_{2j-1,2j-1} - \boldsymbol{\Sigma}_{2j-1,2j} - \boldsymbol{\Sigma}_{2j,2j-1} - \boldsymbol{\Sigma}_{2j,2j}.$$

That is, $\sigma(\lambda^j)$ is computed by taking the negative of the element-wise sum of the sub-matrix $\boldsymbol{\Sigma}(2j-1:2j, 2j-1:2j)$.

Appendix C. Derivatives of the Log Marginal—Categorical Labels

Similarly the second-order derivatives for the categorical case can be written as

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial \alpha^j_{ck} \partial \alpha^j_{ck}} &= -\frac{\sum_i \mu_{ic} \delta(y^j_i, k)}{[\alpha^j_{ck}]^2} - \frac{(C-1)\lambda^j}{C}, \\ \frac{\partial^2 \mathcal{L}}{\partial \alpha^j_{ck} \partial \alpha^j_{c'k}} &= \frac{\lambda^j}{C}, \\ \frac{\partial^2 \mathcal{L}}{\partial \alpha^j_{ck} \partial \alpha^j_{ck'}} &= \frac{\partial^2 \mathcal{L}}{\partial \alpha^j_{ck} \partial \alpha^j_{c'k'}} = 0. \end{aligned} \tag{21}$$

If we rearrange all the parameters in the multinomial term $\boldsymbol{\alpha}^j$ column-wise as a vector of the form $\{\alpha^j_{11}, \alpha^j_{21}, \ldots, \alpha^j_{C1}, \alpha^j_{12}, \ldots, \alpha^j_{C2}, \ldots, \alpha^j_{CC}\}$, then Hessian matrix for the parameters $\theta = \{\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^M\}$ can be written in a block diagonal form as $\mathbf{H} = \text{diag}(\mathbf{H}^1, \ldots, \mathbf{H}^M)$, with $\mathbf{H}^j = \text{diag}(\mathbf{D}^j_1, \ldots, \mathbf{D}^j_C)$, where each matrix $\mathbf{D}^j_c$ is a $C \times C$ matrix of the form $\mathbf{D}^j_c = \mathbf{A}_c(\boldsymbol{\alpha}^j) + \mathbf{B}(\lambda^j)$, where $\mathbf{A}_c(\boldsymbol{\alpha}^j)$ is a diagonal matrix with entries equal to the first term in (21) and $\mathbf{B}(\lambda^j) = \lambda^j\big((1/C)\mathbf{e}\mathbf{e}^\top - \mathbf{I}_C\big)$.

$$\mathbf{B}^j_c(\lambda^j) = \frac{1}{C}\begin{pmatrix} -(C-1)\lambda^j & \lambda^j & \lambda^j & \cdots & \lambda^j \\ \lambda^j & -(C-1)\lambda^j & \lambda^j & \cdots & \lambda^j \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda^j & \cdots & \lambda^j & -(C-1)\lambda^j & \lambda^j \\ \lambda^j & \cdots & \lambda^j & \lambda^j & -(C-1)\lambda^j \end{pmatrix}$$

Therefore, $\frac{\partial}{\partial\lambda^j}\mathbf{H}^j$ is a $C^2 \times C^2$ block diagonal matrix with $(1/C)\mathbf{e}\mathbf{e}^\top - \mathbf{I}_C$ on every diagonal. This greatly simplifies the computation of $\sigma(\lambda^j)$.

Since computing the normalization constant $N(\lambda^j)$ is analytically hard we numerically calculate $\delta(\lambda^j)$ by observing that

$$\begin{aligned}\delta(\lambda^j) &= -\frac{2\lambda^j}{N(\lambda^j)}\frac{\partial}{\partial\lambda^j}N(\lambda^j) \\ &= \lambda^j \cdot \frac{\int_S \exp\left(-\frac{\lambda^j}{2}\left\|\left(\mathbf{I}-\frac{1}{C}\mathbf{e}\mathbf{e}^\top\right)\mathbf{A}^j\right\|_F^2\right)\cdot\left\|\left(\mathbf{I}-\frac{1}{C}\mathbf{e}\mathbf{e}^\top\right)\mathbf{A}^j\right\|_F^2 d\mathbf{A}^j}{\int_S \exp\left(-\frac{\lambda^j}{2}\left\|\left(\mathbf{I}-\frac{1}{C}\mathbf{e}\mathbf{e}^\top\right)\mathbf{A}^j\right\|_F^2\right) d\mathbf{A}^j},\end{aligned}$$

where $S = \{\mathbf{A}^j = [\alpha^j_{ck}] \in \mathbb{R}^{C\times C} | \alpha_{ck} \in [0,1], \mathbf{A}^j\mathbf{e} = \mathbf{e}\}$. We compute the integration numerically via sampling.

References

A. Brew, D. Greene, and P. Cunningham. Using crowdsourcing and active learning to track sentiment in online media. In *Proceedings of the 6th Conference on Prestigious Applications of Intelligent Systems (PAIS'10)*, 2010.

B. Carpenter. Multilevel Bayesian models of categorical data annotation. Technical Report available at http://lingpipe-blog.com/lingpipe-white-papers/, 2008.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979.

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*, 39(1):1–38, 1977.

P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *KDD 2009: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2009.

P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP'10)*, pages 64–67, 2010.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

G. Parent and M. Eskenazi. Clustering dictionary definitions using amazon mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 21–29. Association for Computational Linguistics, 2010.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

V. C. Raykar and S Yu. Ranking annotators for crowdsourced labeling tasks. In *Advances in Neural Information Processing Systems 24*, pages 1809–1817. 2011.

V. C. Raykar, S. Yu, L .H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*, pages 889–896, 2009.

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, April 2010.

V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622, 2008.

P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems 7*, pages 1085–1092. 1995.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 254–263, 2008.

A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. In *Proceedings of the First IEEE Workshop on Internet Vision at CVPR 08*, pages 1–8, 2008.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, pages 2424–2432. 2010.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, pages 2035–2043. 2009.

Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, and J. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, pages 932–939, 2010.

Journal of Machine Learning Research 13 (2012) 519-547 Submitted 2/11; Revised 7/11; Published 3/12

Metric and Kernel Learning Using a Linear Transformation

Prateek Jain PRAJAIN@MICROSOFT.EDU
Microsoft Research India
#9 Lavelle Road
Bangalore 560 003, India

Brian Kulis KULIS@CSE.OHIO-STATE.EDU
599 Dreese Labs
Ohio State University
Columbus, OH 43210, USA

Jason V. Davis JVDAVIS@GMAIL.COM
Etsy Inc.
55 Washington Street, Ste. 512
Brooklyn, NY 11201

Inderjit S. Dhillon INDERJIT@CS.UTEXAS.EDU
The University of Texas at Austin
1 University Station C0500
Austin, TX 78712, USA

Editors: Sören Sonnenburg, Francis Bach and Cheng Soon Ong

Abstract

Metric and kernel learning arise in several machine learning applications. However, most existing metric learning algorithms are limited to learning metrics over low-dimensional data, while existing kernel learning algorithms are often limited to the transductive setting and do not generalize to new data points. In this paper, we study the connections between metric learning and kernel learning that arise when studying metric learning as a linear transformation learning problem. In particular, we propose a general optimization framework for learning metrics via linear transformations, and analyze in detail a special case of our framework—that of minimizing the LogDet divergence subject to linear constraints. We then propose a general regularized framework for learning a kernel matrix, and show it to be *equivalent* to our metric learning framework. Our theoretical connections between metric and kernel learning have two main consequences: 1) the learned kernel matrix parameterizes a linear transformation kernel *function* and can be applied inductively to new data points, 2) our result yields a constructive method for kernelizing most existing Mahalanobis metric learning formulations. We demonstrate our learning approach by applying it to large-scale real world problems in computer vision, text mining and semi-supervised kernel dimensionality reduction.

Keywords: metric learning, kernel learning, linear transformation, matrix divergences, logdet divergence

1. Introduction

One of the basic requirements of many machine learning algorithms (e.g., semi-supervised clustering algorithms, nearest neighbor classification algorithms) is the ability to compare two objects to compute a similarity or distance between them. In many cases, off-the-shelf distance or similarity functions such as the Euclidean distance or cosine similarity are used; for example, in text retrieval applications, the cosine similarity is a standard function to compare two text documents. However, such standard distance or similarity functions are not appropriate for all problems.

Recently, there has been significant effort focused on task-specific learning for comparing data objects. One prominent approach has been to learn a distance metric between objects given additional side information such as pairwise similarity and dissimilarity constraints over the data. A class of distance metrics that has shown excellent generalization properties is the learned *Mahalanobis distance* function (Davis et al., 2007; Xing et al., 2002; Weinberger et al., 2005; Goldberger et al., 2004; Shalev-Shwartz et al., 2004). The Mahalanobis distance can be viewed as a method in which data is subject to a *linear transformation*, and the goal of such metric learning methods is to learn the linear transformation for a given task. Despite their simplicity and generalization ability, Mahalanobis distances suffer from two major drawbacks: 1) the number of parameters to learn grows quadratically with the dimensionality of the data, making it difficult to learn distance functions over high-dimensional data, 2) learning a linear transformation is inadequate for data sets with non-linear decision boundaries.

To address the latter shortcoming, *kernel learning* algorithms typically attempt to learn a kernel matrix over the data. Limitations of linear methods can be overcome by employing a non-linear input kernel, which implicitly maps the data non-linearly to a high-dimensional feature space. However, many existing kernel learning methods are still limited in that the learned kernels do not generalize to new points (Kwok and Tsang, 2003; Kulis et al., 2006; Tsuda et al., 2005). These methods are therefore restricted to learning in the transductive setting where all the data (labeled and unlabeled) is assumed to be given upfront. There has been some work on learning kernels that generalize to new points, most notably work on hyperkernels (Ong et al., 2005), but the resulting optimization problems are expensive and cannot be scaled to large or even medium-sized data sets. Another approach is multiple kernel learning (Lanckriet et al., 2004), which learns a mixture of base kernels; this approach is inductive but the class of learnable kernels can be restrictive.

In this paper, we explore metric learning with linear transformations over arbitrarily high-dimensional spaces; as we will see, this is equivalent to learning a *linear transformation kernel function* $\phi(\boldsymbol{x})^T W \phi(\boldsymbol{y})$ given an input kernel function $\phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$. In the first part of the paper, we formulate a metric learning problem that uses a particular loss function called the LogDet divergence, for learning the positive definite matrix W. This loss function is advantageous for several reasons: it is defined only over positive definite matrices, which makes the optimization simpler, as we will be able to effectively ignore the positive definiteness constraint on W. Furthermore, the loss function has precedence in optimization (Fletcher, 1991) and statistics (James and Stein, 1961). An important advantage of our method is that the proposed optimization algorithm is scalable to very large data sets of the order of millions of data objects. But perhaps most importantly, the loss function permits efficient kernelization, allowing efficient learning of a linear transformation in kernel space. As a result, unlike transductive kernel learning methods, our method easily handles out-of-sample extensions, that is, it can be applied to unseen data.

We build upon our results of kernelization for the LogDet formulation to develop a general framework for learning linear transformation kernel functions and show that such kernels can be efficiently learned over a wide class of convex constraints and loss functions. Our result can be viewed as a representer theorem, where the optimal parameters can be expressed purely in terms of the training data. In our case, even though the matrix W may be infinite-dimensional, it can be fully represented in terms of the constrained data points, making it possible to compute the learned kernel function over arbitrary points.

We demonstrate the benefits of a generalized framework for inductive kernel learning by applying our techniques to the problem of inductive kernelized semi-supervised dimensionality reduction. By choosing the trace-norm as a loss function, we obtain a novel kernel learning method that learns *low-rank* linear transformations; unlike previous kernel dimensionality methods, which are either unsupervised or cannot easily be applied inductively to new data, our method intrinsically possesses both desirable properties.

Finally, we apply our metric and kernel learning algorithms to a number of challenging learning problems, including ones from the domains of computer vision and text mining. Unlike existing techniques, we can learn linear transformation-based distance or kernel functions over these domains, and we show that the resulting functions lead to improvements over state-of-the-art techniques for a variety of problems.

2. Related Work

Most of the existing work in metric learning has been done in the Mahalanobis distance (or metric) learning paradigm, which has been found to be a sufficiently powerful class of metrics for a variety of data. In one of the earliest papers on metric learning, Xing et al. (2002) propose a semidefinite programming formulation under similarity and dissimilarity constraints for learning a Mahalanobis distance, but the resulting formulation is slow to optimize and has been outperformed by more recent methods. Weinberger et al. (2005) formulate the metric learning problem in a large margin setting, with a focus on k-NN classification. They also formulate the problem as a semidefinite programming problem and consequently solve it using a method that combines sub-gradient descent and alternating projections. Goldberger et al. (2004) proceed to learn a linear transformation in the fully supervised setting. Their formulation seeks to 'collapse classes' by constraining within-class distances to be zero while maximizing between-class distances. While each of these algorithms was shown to yield improved classification performance over the baseline metrics, their constraints do not generalize outside of their particular problem domains; in contrast, our approach allows arbitrary linear constraints on the Mahalanobis matrix. Furthermore, these algorithms all require eigenvalue decompositions or semi-definite programming, which is at least cubic in the dimensionality of the data.

Other notable works for learning Mahalanobis metrics include Pseudo-metric Online Learning Algorithm (POLA) (Shalev-Shwartz et al., 2004), Relevant Components Analysis (RCA) (Schultz and Joachims, 2003), Neighborhood Components Analysis (NCA) (Goldberger et al., 2004), and locally-adaptive discriminative methods (Hastie and Tibshirani, 1996). In particular, Shalev-Shwartz et al. (2004) provided the first demonstration of Mahalanobis distance learning in kernel space. Their construction, however, is expensive to compute, requiring cubic time per iteration to update the parameters. As we will see, our LogDet-based algorithm can be implemented more efficiently.

Non-linear transformation based metric learning methods have also been proposed, though these methods usually suffer from suboptimal performance, non-convexity, or computational complexity. Examples include the convolutional neural net based method of Chopra et al. (2005); and a general Riemannian metric learning method (Lebanon, 2006).

Most of the existing work on kernel learning can be classified into two broad categories. The first category includes parametric approaches, where the learned kernel function is restricted to be of a specific form and then the relevant parameters are learned according to the provided data. Prominent methods include multiple kernel learning (Lanckriet et al., 2004), hyperkernels (Ong et al., 2005), and hyper-parameter cross-validation (Seeger, 2006). Most of these methods either lack modeling flexibility, require non-convex optimization, or are restricted to a supervised learning scenario. The second category includes non-parametric methods, which explicitly model geometric structure in the data. Examples include spectral kernel learning (Zhu et al., 2005), manifold-based kernel learning (Bengio et al., 2004), and kernel target alignment (Cristianini et al., 2001). However, most of these approaches are limited to the transductive setting and cannot be used to naturally generalize to new points. In comparison, our kernel learning method combines both of the above approaches. We propose a general non-parametric kernel *matrix* learning framework, similar to methods of the second category. However, based on our choice of regularization and constraints, we show that our learned kernel matrix corresponds to a linear transformation kernel function parameterized by a PSD matrix. As a result, our method can be applied to inductive settings without sacrificing significant modeling power. In addition, our kernel learning method naturally provides kernelization for many existing metric learning methods. Recently, Chatpatanasiri et al. (2010) showed kernelization for a class of metric learning algorithms including LMNN and NCA; as we will see, our result is more general and we can prove kernelization over a larger class of problems and can also reduce the number of parameters to be learned. Furthermore, our methods can be applied to a variety of domains and with a variety of forms of side-information. Independent of our work, Argyriou et al. (2010) recently proved a representer type of theorem for spectral regularization functions. However, the framework they consider is different than ours in that they are interested in sensing an underlying high-dimensional matrix using given measurements.

The research in this paper combines and extends work done in Davis et al. (2007), Kulis et al. (2006), Davis and Dhillon (2008), and Jain et al. (2010). The focus in Davis et al. (2007) and Davis and Dhillon (2008) was solely on the LogDet divergence, while the main goal in Kulis et al. (2006) was to demonstrate the computational benefits of using the LogDet and von Neumann divergences for learning low-rank kernel matrices. In Jain et al. (2010), we showed the equivalence between a general class of kernel learning problems and metric learning problems. In this paper, we unify and summarize the work in the existing conference papers and also provide detailed proofs of the theorems in Jain et al. (2010).

3. LogDet Divergence Based Metric Learning

We begin by studying a particular method, based on the LogDet divergence, for learning metrics via learning linear transformations given pairwise distance constraints. We discuss kernelization of this formulation and present efficient optimization algorithms. Finally, we address limitations of the method when the amount of training data is large, and propose a modified algorithm to efficiently learn a kernel under such circumstances. In subsequent sections, we will take the ingredients developed in this section and show how to generalize them to adapt to a much larger class of loss functions and constraints, which will encompass most of the previously-studied approaches for Mahalanobis metric learning.

3.1 Mahalanobis Distances and Parameterized Kernels

First we introduce the framework for metric and kernel learning that is employed in this paper. Given a data set of objects $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n], \boldsymbol{x}_i \in \mathbb{R}^{d_0}$ (when working in kernel space, the data matrix will be represented as $\Phi = [\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_n)]$, where ϕ is the mapping to feature space, that is, $\phi : \mathbb{R}^{d_0} \to \mathbb{R}^d$), we are interested in finding an appropriate distance function to compare two objects. We consider the Mahalanobis distance, parameterized by a positive definite matrix W; the squared distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by

$$d_W(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T W(\boldsymbol{x}_i - \boldsymbol{x}_j). \tag{1}$$

This distance function can be viewed as learning a linear transformation of the data and measuring the squared Euclidean distance in the transformed space. This is seen by factorizing the matrix $W = G^T G$ and observing that $d_W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|G\boldsymbol{x}_i - G\boldsymbol{x}_j\|_2^2$. However, if the data is not linearly separable in the input space, then the resulting distance function may not be powerful enough for the desired application. As a result, we are interested in working in kernel space; that is, we will express the Mahalanobis distance in kernel space using an appropriate mapping ϕ from input to feature space:

$$d_W(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)) = (\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j))^T W(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)).$$

Note that when we choose ϕ to be the identity, we obtain (1); we will use the more general form throughout this paper. As is standard with kernel-based algorithms, we require that this distance be computable given the ability to compute the kernel function $\kappa_0(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$. We can therefore equivalently pose the problem as learning a parameterized kernel function $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T W \phi(\boldsymbol{y})$ given some input kernel function $\kappa_0(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$.

To learn the resulting metric/kernel, we assume that we are given constraints on the desired distance function. In this paper, we assume that pairwise similarity and dissimilarity constraints are given over the data—that is, pairs of points that should be similar under the learned metric/kernel, and pairs of points that should be dissimilar under the learned metric/kernel. Such constraints are natural in many settings; for example, given class labels over the data, points in the same class should be similar to one another and dissimilar to points in different classes. However, our approach is general and can accommodate other potential constraints over the distance function, such as relative distance constraints.

The main challenge is in finding an appropriate loss function for learning the matrix W so that 1) the resulting algorithm is scalable and efficiently computable in kernel space, 2) the resulting metric/kernel yields improved performance on the underlying learning problem, such as classification, semi-supervised clustering etc. We now move on to the details.

3.2 LogDet Metric Learning

The LogDet divergence between two positive definite matrices[1] $W, W_0 \in \mathbb{R}^{d \times d}$ is defined to be

$$D_{\ell d}(W, W_0) = \mathrm{tr}(W W_0^{-1}) - \log\det(W W_0^{-1}) - d.$$

We are interested in finding W that is closest to W_0 as measured by the LogDet divergence but that satisfies our desired constraints. When $W_0 = I$, we can interpret the learning problem as a maximum entropy problem. Given a set of similarity constraints $\mathcal{S}$ and dissimilarity constraints $\mathcal{D}$, we propose

[1] The definition of LogDet divergence can be extended to the case when W_0 and W are rank deficient by appropriate use of the pseudo-inverse. The interested reader may refer to Kulis et al. (2008).

the following problem:

$$\min_{W\succeq 0} D_{\ell\mathrm{d}}(W,I), \qquad \text{s.t. } d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) \leq u,\ (i,j)\in\mathcal{S},$$
$$d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) \geq \ell,\ (i,j)\in\mathcal{D}. \tag{2}$$

We make a few remarks about this formulation. The above problem was proposed and studied in Davis et al. (2007). LogDet has many important properties that make it useful for machine learning and optimization, including scale-invariance and preservation of the range space; see Kulis et al. (2008) for a detailed discussion on the properties of LogDet. Beyond this, we prefer LogDet over other loss functions (including the squared Frobenius loss as used in Shalev-Shwartz et al., 2004 or a linear objective as in Weinberger et al., 2005) due to the fact that the resulting algorithm turns out to be simple and efficiently kernelizable, as we will see. We note that formulation (2) minimizes the LogDet divergence to the identity matrix I. This can easily be generalized to arbitrary positive definite matrices W_0. Further, (2) considers simple similarity and dissimilarity constraints over the learned Mahalanobis distance, but other linear constraints are possible. Finally, the above formulation assumes that there exists a feasible solution to the proposed optimization problem; extensions to the infeasible case involving slack variables are discussed later (see Section 3.5).

3.3 Kernelizing the LogDet Metric Learning Problem

We now consider the problem of kernelizing the metric learning problem. Subsequently, we will present an efficient algorithm and discuss generalization to new points.

Given a set of n constrained data points, let K_0 denote the input kernel matrix for the data, that is, $K_0(i,j)=\kappa_0(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$. Note that the squared Euclidean distance in kernel space may be written as $K(i,i)+K(j,j)-2K(i,j)$, where K is the learned kernel matrix; equivalently, we may write the distance as $\mathrm{tr}(K(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T)$, where $\boldsymbol{e}_i$ is the i-th canonical basis vector. Consider the following problem to find K:

$$\min_{K\succeq 0} D_{\ell\mathrm{d}}(K,K_0), \qquad \text{s.t. } \mathrm{tr}(K(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T) \leq u, \qquad (i,j)\in\mathcal{S},$$
$$\mathrm{tr}(K(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T) \geq \ell, \qquad (i,j)\in\mathcal{D}. \tag{3}$$

This kernel learning problem was first proposed in the transductive setting in Kulis et al. (2008), though no extensions to the inductive case were considered. Note that problem (2) optimizes over a $d\times d$ matrix W, while the kernel learning problem (3) optimizes over an $n\times n$ matrix K. We now present our key theorem connecting (2) and (3).

Theorem 1 *Let $K_0\succ 0$. Let W^* be the optimal solution to problem* (2) *and let K^* be the optimal solution to problem* (3)*. Then the optimal solutions are related by the following:*

$$K^*=\Phi^T W^*\Phi,\ \ W^*=I+\Phi S\Phi^T,$$
$$\textit{where } S=K_0^{-1}(K^*-K_0)K_0^{-1}, \quad K_0=\Phi^T\Phi, \quad \Phi=[\phi(\boldsymbol{x}_1),\phi(\boldsymbol{x}_2),\ldots,\phi(\boldsymbol{x}_n)].$$

The above theorem shows that the LogDet metric learning problem (2) can be solved implicitly by solving an equivalent kernel learning problem (3). In fact, in Section 4 we show an equivalence between metric and kernel learning for a general class of regularization functions. The above theorem follows as a corollary to our general theorem (see Theorem 4), which will be proven later.

Next, we generalize the above theorem to regularize against arbitrary positive definite matrices W_0.

Corollary 2 *Consider the following problem:*

$$\min_{W\succeq 0} D_{\ell d}(W,W_0),\ s.t.\ d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j))\leq u,\ (i,j)\in\mathcal{S},$$
$$d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j))\geq \ell,\ (i,j)\in\mathcal{D}. \tag{4}$$

Let W^* be the optimal solution to problem (4) *and let K^* be the optimal solution to problem* (3). *Then the optimal solutions are related by the following:*

$$K^*=\Phi^T W^*\Phi, \qquad W^*=W_0+W_0\Phi S\Phi^T W_0,$$
$$\textit{where } S=K_0^{-1}(K^*-K_0)K_0^{-1}, \quad K_0=\Phi^T W_0\Phi, \quad \Phi=[\phi(\boldsymbol{x}_1),\phi(\boldsymbol{x}_2),\ldots,\phi(\boldsymbol{x}_n)].$$

Proof Note that $D_{\ell\mathrm{d}}(W,W_0)=D_{\ell\mathrm{d}}(W_0^{-1/2}WW_0^{-1/2},I)$. Let $\widetilde{W}=W_0^{-1/2}WW_0^{-1/2}$. Problem (4) is now equivalent to:

$$\min_{\widetilde{W}\succeq 0} \quad D_{\ell\mathrm{d}}(\widetilde{W},I), \qquad \text{s.t.} \quad d_{\widetilde{W}}(\tilde{\phi}(\boldsymbol{x}_i),\tilde{\phi}(\boldsymbol{x}_j))\leq u \qquad (i,j)\in\mathcal{S},$$
$$d_{\widetilde{W}}(\tilde{\phi}(\boldsymbol{x}_i),\tilde{\phi}(\boldsymbol{x}_j))\geq \ell \qquad (i,j)\in\mathcal{D}, \tag{5}$$

where $\widetilde{W}=W_0^{-1/2}WW_0^{-1/2}$, $\widetilde{\Phi}=W_0^{1/2}\Phi$ and $\widetilde{\Phi}=[\tilde{\phi}(\boldsymbol{x}_1),\tilde{\phi}(\boldsymbol{x}_2),\ldots,\tilde{\phi}(\boldsymbol{x}_n)]$. Now using Theorem 1, the optimal solution $\widetilde{W}^*$ of problem (5) is related to the optimal K^* of problem (3) by $K^*=\widetilde{\Phi}^T\widetilde{W}^*\widetilde{\Phi}=\Phi^T W_0^{1/2}W_0^{-1/2}W^*W_0^{-1/2}W_0^{1/2}\Phi=\Phi^T W^*\Phi$. Similarly, $W^*=W_0^{1/2}\widetilde{W}^*W_0^{1/2}=W_0+W_0\Phi S\Phi^T W_0$ where $S=K_0^{-1}(K^*-K_0)K_0^{-1}$. ■

Since the kernelized version of LogDet metric learning is also a linearly constrained optimization problem with a LogDet objective, similar algorithms can be used to solve either problem. This equivalence implies that we can *implicitly* solve the metric learning problem by instead solving for the optimal kernel matrix K^*. Note that using LogDet divergence as objective function has two significant benefits over many other popular loss functions: 1) the metric and kernel learning problems (2), (3) are both equivalent and therefore solving the kernel learning formulation directly provides an out of sample extension (see Section 3.4 for details), 2) projection with respect to the LogDet divergence onto a single distance constraint has a closed-form solution, thus making it amenable to an efficient cyclic projection algorithm (refer to Section 3.5).

3.4 Generalizing to New Points

In this section, we see how to generalize to new points using the learned kernel matrix K^*.

Suppose that we have solved the kernel learning problem for K^* (from now on, we will drop the * superscript and assume that K and W are at optimality). The distance between two points $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ that are in the training set can be computed directly from the learned kernel matrix as $K(i,i)+K(j,j)-2K(i,j)$. We now consider the problem of computing the learned distance between two points $\phi(\boldsymbol{z}_1)$ and $\phi(\boldsymbol{z}_2)$ that may not be in the training set.

In Theorem 1, we showed that the optimal solution to the metric learning problem can be expressed as $W = I + \Phi S \Phi^T$. To compute the Mahalanobis distance in kernel space, we see that the inner product $\phi(\boldsymbol{z}_1)^T W \phi(\boldsymbol{z}_2)$ can be computed entirely via inner products between points:

$$\begin{aligned} \phi(\boldsymbol{z}_1)^T W \phi(\boldsymbol{z}_2) &= \phi(\boldsymbol{z}_1)^T (I + \Phi S \Phi^T) \phi(\boldsymbol{z}_2) = \phi(\boldsymbol{z}_1)^T \phi(\boldsymbol{z}_2) + \phi(\boldsymbol{z}_1)^T \Phi S \Phi^T \phi(\boldsymbol{z}_2), \\ &= \kappa_0(\boldsymbol{z}_1, \boldsymbol{z}_2) + \boldsymbol{k}_1^T S \boldsymbol{k}_2, \text{where } \boldsymbol{k}_i = [\kappa_0(\boldsymbol{z}_i, \boldsymbol{x}_1), \ldots, \kappa_0(\boldsymbol{z}_i, \boldsymbol{x}_n)]^T. \end{aligned} \tag{6}$$

Thus, the expression above can be used to evaluate kernelized distances with respect to the learned kernel function between arbitrary data objects.

In summary, the connection between kernel learning and metric learning allows us to generalize our metrics to new points in kernel space. This is performed by first solving the kernel learning problem for K, then using the learned kernel matrix and the input kernel function to compute learned distances using (6).

3.5 Kernel Learning Algorithm

Given the connection between the Mahalanobis metric learning problem for the $d \times d$ matrix W and the kernel learning problem for the $n \times n$ kernel matrix K, we develop an algorithm for efficiently performing metric learning in kernel space. Specifically, we provide an algorithm (see Algorithm 1) for solving the kernelized LogDet metric learning problem (3).

First, to avoid problems with infeasibility, we incorporate *slack variables* into our formulation. These provide a tradeoff between minimizing the divergence between K and K_0 and satisfying the constraints. Note that our earlier results (see Theorem 1) easily generalize to the slack case:

$$\begin{aligned} \min_{K, \boldsymbol{\xi}} \quad & D_{\ell d}(K, K_0) + \gamma \cdot D_{\ell d}(\text{diag}(\boldsymbol{\xi}), \text{diag}(\boldsymbol{\xi}_0)) \\ \text{s.t.} \quad & \text{tr}(K(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T) \leq \xi_{ij} \quad (i,j) \in \mathcal{S}, \\ & \text{tr}(K(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T) \geq \xi_{ij} \quad (i,j) \in \mathcal{D}. \end{aligned} \tag{7}$$

The parameter γ above controls the tradeoff between satisfying the constraints and minimizing $D_{\ell d}(K, K_0)$, and the entries of $\boldsymbol{\xi}_0$ are set to be u for corresponding similarity constraints and ℓ for dissimilarity constraints.

To solve problem (7), we employ the technique of *Bregman projections*, as discussed in the transductive setting (Kulis et al., 2008). At each iteration, we choose a constraint (i, j) from $\mathcal{S}$ or $\mathcal{D}$. We then apply a Bregman projection such that K satisfies the constraint after projection; note that the projection is not an orthogonal projection but is rather tailored to the particular function that we are optimizing. Algorithm 1 details the steps for Bregman's method on this optimization problem. Each update is a rank-one update

$$K \leftarrow K + \beta K(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T K,$$

where β is a projection parameter that can be computed in closed form (see Algorithm 1).

Algorithm 1 has a number of key properties which make it useful for various kernel learning tasks. First, the Bregman projections can be computed in closed form, assuring that the projection updates are efficient ($O(n^2)$). Note that, if the feature space dimensionality d is less than n then a similar algorithm can be used directly in the feature space (see Davis et al., 2007). Instead of LogDet, if we use the von Neumann divergence, another potential loss function for this problem,

Algorithm 1 Metric/Kernel Learning with the LogDet Divergence

Input: K_0: input $n \times n$ kernel matrix, $\mathcal{S}$: set of similar pairs, $\mathcal{D}$: set of dissimilar pairs, u, ℓ: distance thresholds, γ: slack parameter
Output: K: output kernel matrix

```
1. $K \leftarrow K_0, \lambda_{ij} \leftarrow 0 \ \forall \ ij$
2. $\xi_{ij} \leftarrow u$ for $(i,j) \in \mathcal{S}$; otherwise $\xi_{ij} \leftarrow \ell$
3. repeat
    3.1. Pick a constraint $(i,j) \in \mathcal{S}$ or $\mathcal{D}$
    3.2. $p \leftarrow (\boldsymbol{e}_i - \boldsymbol{e}_j)^T K (\boldsymbol{e}_i - \boldsymbol{e}_j)$
    3.3. $\delta \leftarrow 1$ if $(i,j) \in \mathcal{S}$, $-1$ otherwise
    3.4. $\alpha \leftarrow \min\left(\lambda_{ij}, \frac{\delta\gamma}{\gamma+1}\left(\frac{1}{p} - \frac{1}{\xi_{ij}}\right)\right)$
    3.5. $\beta \leftarrow \delta\alpha/(1-\delta\alpha p)$
    3.6. $\xi_{ij} \leftarrow \gamma\xi_{ij}/(\gamma + \delta\alpha\xi_{ij})$
    3.7. $\lambda_{ij} \leftarrow \lambda_{ij} - \alpha$
    3.8. $K \leftarrow K + \beta K(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T K$
4. until convergence
return $K$
```

$O(n^2)$ updates are possible, but are much more complicated and require use of the fast multipole method, which cannot be employed easily in practice. Secondly, the projections maintain positive definiteness, which avoids any eigenvector computation or semidefinite programming. This is in stark contrast with the Frobenius loss, which requires additional computation to maintain positive definiteness, leading to $O(n^3)$ updates.

3.6 Metric/Kernel Learning with Large Data Sets

In Sections 3.1 and 3.3, we proposed a LogDet divergence-based Mahalanobis metric learning problem (2) and an equivalent kernel learning problem (3). The number of parameters involved in these problems is $O(\min(n,d)^2)$, where n is the number of training points and d is the dimensionality of the data. The quadratic dependence affects not only the running time for training and testing, but also requires estimating a large number of parameters. For example, a data set with 10,000 dimensions leads to a Mahalanobis matrix with 100 million entries. This represents a fundamental limitation of existing approaches, as many modern data mining problems possess relatively high dimensionality.

In this section, we present a heuristic for learning structured Mahalanobis distance (kernel) functions that scale linearly with the dimensionality (or training set size). Instead of representing the Mahalanobis distance/kernel matrix as a full $d \times d$ (or $n \times n$) matrix with $O(\min(n,d)^2)$ parameters, our methods use compressed representations, admitting matrices parameterized by $O(\min(n,d))$ values. This enables the Mahalanobis distance/kernel function to be learned, stored, and evaluated efficiently in the context of high dimensionality and large training set size. In particular, we propose a method to efficiently learn an identity plus low-rank Mahalanobis distance matrix and its equivalent kernel function.

Now, we formulate this approach, which we call the high-dimensional identity plus low-rank (IPLR) metric learning problem. Consider a low-dimensional subspace in $\mathbb{R}^d$ and let the columns of U form an orthogonal basis of this subspace. We will constrain the learned Mahalanobis distance matrix to be of the form:

$$W = I^d + W_l = I^d + ULU^T,$$

where I^d is the $d \times d$ identity matrix, W_l denotes the low-rank part of W and $L \in \mathbb{S}_+^{k\times k}$ with $k \ll \min(n,d)$. Analogous to (2), we propose the following problem to learn an identity plus low-rank Mahalanobis distance function:

$$\min_{W,L\succeq 0} \quad D_{\ell\mathrm{d}}(W,I^d) \qquad \text{s.t.} \quad d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) \le u \quad (i,j)\in\mathcal{S},$$
$$d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) \ge \ell \quad (i,j)\in\mathcal{D}, \qquad W = I^d + ULU^T. \tag{8}$$

Note that the above problem is identical to (2) except for the added constraint $W = I^d + ULU^T$. Let $F = I^k + L$. Now we have

$$\begin{aligned} D_{\ell\mathrm{d}}(W,I^d) &= \mathrm{tr}(I^d + ULU^T) - \log\det(I^d + ULU^T) - d, \\ &= \mathrm{tr}(I^k + L) + d - k - \log\det(I^k + L) - d = D_{\ell\mathrm{d}}(F,I^k), \end{aligned} \tag{9}$$

where the second equality follows from the fact that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ and Sylvester's determinant lemma. Also note that for all $C \in \mathbb{R}^{n\times n}$,

$$\begin{aligned} \mathrm{tr}(W\Phi C\Phi^T) &= \mathrm{tr}((I^d + ULU^T)\Phi C\Phi^T) = \mathrm{tr}(\Phi C\Phi^T) + \mathrm{tr}(LU^T\Phi C\Phi^T U), \\ &= \mathrm{tr}(\Phi C\Phi^T) - \mathrm{tr}(\Phi' C\Phi'^T) + \mathrm{tr}(F\Phi' C\Phi'^T), \end{aligned}$$

where $\Phi' = U^T\Phi$ is the reduced-dimensional representation of Φ. Therefore,

$$\begin{aligned} d_W(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) &= \mathrm{tr}(W\Phi(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T\Phi^T) \\ &= d_I(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) - d_I(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)) + d_F(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)). \end{aligned} \tag{10}$$

Using (9) and (10), problem (8) is equivalent to the following:

$$\begin{aligned} \min_{F\succeq 0} \quad & D_{\ell\mathrm{d}}(F,I^k), \\ \text{s.t.} \quad & d_F(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)) \le u - d_I(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) + d_I(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)), \ (i,j)\in\mathcal{S}, \\ & d_F(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)) \ge \ell - d_I(\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)) + d_I(\phi'(\boldsymbol{x}_i),\phi'(\boldsymbol{x}_j)), \ (i,j)\in\mathcal{D}. \end{aligned} \tag{11}$$

Note that the above formulation is an instance of problem (2) and can be solved using an algorithm similar to Algorithm 1. Furthermore, the above problem solves for a $k\times k$ matrix rather than a $d\times d$ matrix seemingly required by (8). The optimal W^* is obtained as $W^* = I^d + U(F^* - I^k)U^T$.

Next, we show that problem (11) and equivalently (8) can be solved efficiently in feature space by selecting an appropriate basis R ($U = R(R^TR)^{-1/2}$). Let $R = \Phi J$, where $J \in \mathbb{R}^{n\times k}$. Note that $U = \Phi J(J^TK_0J)^{-1/2}$ and $\Phi' = U^T\Phi = (J^TK_0J)^{-1/2}J^TK_0$, that is, $\Phi' \in \mathbb{R}^{k\times n}$ can be computed efficiently in the feature space (requiring inversion of only a $k\times k$ matrix). Hence, problem (11) can be solved efficiently in feature space using Algorithm 1, and the optimal kernel K^* is given by

$$K^* = \Phi^T W^*\Phi = K_0 + K_0J(J^TK_0J)^{-1/2}(F^* - I^k)(J^TK_0J)^{-1/2}J^TK_0.$$

Note that (11) can be solved via Algorithm 1 using $O(k^2)$ computational steps per iteration. Additionally, $O(\min(n,d)k)$ steps are required to prepare the data. Also, the optimal solution W^* (or

K^*) can be stored implicitly using $O(\min(n,d)k)$ memory and similarly, the Mahalanobis distance between any two points can be computed in $O(\min(n,d)k)$ time.

The metric learning problem presented here depends critically on the basis selected. For the case when d is not significantly larger than n and feature space vectors Φ are available explicitly, the basis R can be selected by using one of the following heuristics (see Section 5, Davis and Dhillon, 2008 for more details):

- Using the top k singular vectors of Φ.
- Clustering the columns of Φ and using the mean vectors as the basis R.
- For the fully-supervised case, if the number of classes (c) is greater than the required dimensionality (k) then cluster the class-mean vectors into k clusters and use the obtained cluster centers for forming the basis R. If $c < k$ then cluster each class into k/c clusters and use the cluster centers to form R.

For learning the kernel function, the basis $R = \Phi J$ can be selected by: 1) using a randomly sampled coefficient matrix J, 2) clustering Φ using kernel k-means or a spectral clustering method, 3) choosing a random subset of Φ, that is, the columns of J are random indicator vectors. A more careful selection of the basis R should further improve accuracy of our method and is left as a topic for future research.

4. Kernel Learning with Other Convex Loss Functions

One of the key benefits of our kernel learning formulation using the LogDet divergence (3) is in the ability to efficiently learn a linear transformation (LT) kernel *function* (a kernel of the form $\phi(\boldsymbol{x})^T W \phi(\boldsymbol{y})$ for some matrix $W \succeq 0$) which allows the learned kernel function to be computed over new data points. A natural question is whether one can learn similar kernel functions with other loss functions, such as those considered previously in the literature for Mahalanobis metric learning.

In this section, we propose and analyze a general kernel matrix learning problem similar to (3) but using a more general class of loss functions. As in the LogDet loss function case, we show that our kernel matrix learning problem is equivalent to learning a linear transformation (LT) kernel *function* with a specific loss function. This implies that the learned LT kernel function can be naturally applied to new data. Additionally, since a large class of metric learning methods can be seen as learning a LT kernel function, our result provides a constructive method for kernelizing these methods. Our analysis recovers some recent kernelization results for metric learning, but also implies several new results.

4.1 A General Kernel Learning Framework

Recall that $\kappa_0 : \mathbb{R}^{d_0} \times \mathbb{R}^{d_0} \to \mathbb{R}$ is the input kernel function. We assume that the data vectors in X have been mapped via ϕ, resulting in $\Phi = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n)]$. As before, denote the input kernel matrix as $K_0 = \Phi^T \Phi$. The goal is to learn a kernel function κ that is regularized against κ_0 but incorporates the provided side-information. As in the LogDet formulation, we will first consider a transductive scenario, where we learn a kernel matrix K that is regularized against K_0 while satisfying the available side-information.

Recall that the LogDet divergence based loss function in the kernel matrix learning problem (3) is given by:

$$\begin{aligned} D_{\ell\mathrm{d}}(K,K_0) &= \mathrm{tr}(KK_0^{-1}) - \log\det(KK_0^{-1}) - n, \\ &= \mathrm{tr}(K_0^{-1/2}KK_0^{-1/2}) - \log\det(K_0^{-1/2}KK_0^{-1/2}) - n. \end{aligned}$$

The kernel matrix learning problem (3) can be rewritten as:

$$\begin{aligned} \min_{K\succeq 0} \quad f(K_0^{-1/2}KK_0^{-1/2}), \qquad \text{s.t.} \quad & \mathrm{tr}(K(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T) \le u, \quad (i,j)\in S, \\ & \mathrm{tr}(K(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T) \ge \ell, \quad (i,j)\in \mathcal{D}, \end{aligned}$$

where $f(A) = \mathrm{tr}(A) - \log\det(A)$.

In this section, we will generalize our optimization problem to include more general loss functions beyond the LogDet-based loss function f specified above. We also generalize our constraints to include arbitrary constraints over the kernel matrix K rather than just the pairwise distance constraints in the above problem. Using the above specified generalizations, the optimization problem that we obtain is given by:

$$\min_{K\succeq 0} f(K_0^{-1/2}KK_0^{-1/2}), \quad \text{s.t. } g_i(K) \le b_i,\ 1\le i\le m, \tag{12}$$

where f and g_i are functions from $\mathbb{R}^{n\times n}\to\mathbb{R}$. We call f the *loss function* (or regularizer) and g_i the *constraints*. Note that if f and constraints g_i's are all convex, then the above problem can be solved optimally (under mild conditions) using standard convex optimization algorithms (Groschel et al., 1988). Note that our results also hold for unconstrained variants of the above problem, as well as variants with slack variables.

In general, such formulations are limited in that the learned kernel cannot readily be applied to new data points. However, we will show that the above proposed problem is equivalent to learning linear transformation (LT) kernel functions. Formally, an LT kernel function κ_W is a kernel function of the form $\kappa(\boldsymbol{x},\boldsymbol{y}) = \phi(\boldsymbol{x})^T W\phi(\boldsymbol{y})$, where W is a positive semi-definite (PSD) matrix. A natural way to learn an LT kernel function would be to learn the parameterization matrix W using the provided side-information. To this end, we consider the following generalization of our LogDet based learning problem (2):

$$\min_{W\succeq 0} f(W), \quad \text{s.t. } g_i(\Phi^T W\Phi) \le b_i,\ 1\le i\le m, \tag{13}$$

where, as before, the function f is the loss function and the functions g_i are the constraints that encode the side information. The constraints g_i are assumed to be a function of the matrix $\Phi^T W\Phi$ of learned kernel values over the training data. Note that most Mahalanobis metric learning methods may be viewed as a special case of the above framework (see Section 5). Also, for data mapped to high-dimensional spaces via kernel functions, this problem is seemingly impossible to optimize since the size of W grows quadratically with the dimensionality.

4.2 Analysis

We now analyze the connection between the problems (12) and (13). We will show that the solutions to the two problems are equivalent, that is, by optimally solving one of the problems, the solution to the other can be computed in closed form. Further, this result will yield insight into the type of kernel that is learned by the kernel learning problem.

We begin by defining the class of loss functions considered in our analysis.

Definition 3 *We say that $f:\mathbb{R}^{n\times n}\rightarrow\mathbb{R}$ is a **spectral function** if $f(A)=\sum_i f_s(\lambda_i)$, where $\lambda_1,...,\lambda_n$ are the eigenvalues of A and $f_s:\mathbb{R}\rightarrow\mathbb{R}$ is a real-valued scalar function. Note that if f_s is a convex scalar function, then f is also convex.*

Note that the LogDet based loss function in (3) is a spectral function. Similarly, most of the existing metric learning formulations have a spectral function as their objective function.

Now we state our main result that for a spectral function f, problems (12) and (13) are equivalent.

Theorem 4 *Let $K_0=\Phi^T\Phi\succ 0$, f be a spectral function as in Definition 3 and assume that the global minimum of the corresponding strictly convex scalar function f_s is $\alpha>0$. Let W^* be an optimal solution to* (13) *and K^* be an optimal solution to* (12)*. Then,*

$$W^*=\alpha I^d+\Phi S\Phi^T,$$

where $S=K_0^{-1}(K^*-\alpha K_0)K_0^{-1}$. Furthermore, $K^*=\Phi^T W^*\Phi$.

The first part of the theorem demonstrates that, given an optimal solution K^* to (12), one can construct the corresponding solution W^* to (13), while the second part shows the reverse. Note the similarities between this theorem and the earlier Theorem 1. We provide the proof of this theorem below. The main idea behind the proof is to first show that the optimal solution to (13) is always of the form $W=\alpha I^d+\Phi S\Phi^T$, and then we obtain the closed form expression for S using simple algebraic manipulations.

First we introduce and analyze an auxiliary optimization problem that will help in proving the above theorem. Consider the following problem:

$$\min_{W\succeq 0,L} f(W),\quad \text{s.t. } g_i(\Phi^T W\Phi)\le b_i,\ 1\le i\le m,\quad W=\alpha I^d+ULU^T, \tag{14}$$

where $L\in\mathbb{R}^{k\times k}$, $U\in\mathbb{R}^{d\times k}$ is a column orthogonal matrix, and I^d is the $d\times d$ identity matrix. In general, k can be significantly smaller than $\min(n,d)$. Note that the above problem is identical to (13) except for an added constraint $W=\alpha I^d+ULU^T$. We now show that (14) is equivalent to a problem over $k\times k$ matrices. In particular, (14) is equivalent to (15) defined below.

Lemma 5 *Let f be a spectral function as in Definition 3 and let $\alpha>0$ be any scalar. Then,* (14) *is equivalent to:*

$$\min_{L\succeq-\alpha I^k} f(\alpha I^k+L),\quad s.t.\ \ g_i(\alpha\Phi^T\Phi+\Phi^T ULU^T\Phi)\le b_i,\ 1\le i\le m. \tag{15}$$

Proof The last constraint in (14) asserts that $W=\alpha I^d+ULU^T$, which implies that there is a one-to-one mapping between W and L: given W, L can be computed and vice-versa. As a result, we can eliminate the variable W from (14) by substituting αI^d+ULU^T for W (via the last constraint in (14)). The resulting optimization problem is:

$$\min_{L\succeq-\alpha I^k} f(\alpha I^d+ULU^T),\quad \text{s.t. } g_i(\alpha\Phi^T\Phi+\Phi^T ULU^T\Phi)\le b_i,\ 1\le i\le m. \tag{16}$$

Note that (15) and (16) are the same except for their objective functions. Below, we show that both the objective functions are equal up to a constant, so they are interchangeable in the optimization problem. Let $U' \in \mathbb{R}^{d\times d}$ be an orthonormal matrix obtained by completing the basis represented by U, that is, $U' = [U\ U_\perp]$ for some $U_\perp \in \mathbb{R}^{d\times(d-k)}$ s.t. $U^T U_\perp = 0$ and $U_\perp^T U_\perp = I^{d-k}$. Now, $W = \alpha I^d + ULU^T = U'\left(\alpha I^d + \begin{bmatrix} L & 0\\ 0 & 0\end{bmatrix}\right)U'^T$. It is straightforward to see that for a spectral function f, $f(VWV^T) = f(W)$, where V is an orthogonal matrix. Also, $\forall A, B \in \mathbb{R}^{d\times d}$, $f\left(\begin{bmatrix} A & 0\\ 0 & B\end{bmatrix}\right) = f(A) + f(B)$. Using the above observations, we get:

$$f(W) = f(\alpha I^d + ULU^T) = f\left(\begin{bmatrix} \alpha I^k + L & 0\\ 0 & \alpha I^{d-k}\end{bmatrix}\right) = f(\alpha I^k + L) + (d-k)f(\alpha). \quad (17)$$

Therefore, the objective functions of (15) and (16) differ by only a constant, that is, they are equivalent with respect to the optimization problem. The lemma follows. ■

We now show that for strictly convex spectral functions (see Definition 3) the optimal solution W^* to (13) is of the form $W^* = \alpha I^d + \Phi S \Phi^T$, for some S.

Lemma 6 *Suppose f, K_0 and α satisfy the conditions given in Theorem 4. Then, the optimal solution to* (13) *is of the form $W^* = \alpha I^d + \Phi S \Phi^T$, where S is a $n \times n$ matrix.*

Proof Note that $K_0 \succ 0$ implies that $d \geq n$. Our results can be extended when $d < n$, that is, $K_0 \succeq 0$, by using the pseudo-inverse of K_0 instead of the inverse. However, for simplicity we only present the full-rank case.

Now, let $W = U\Lambda U^T = \sum_j \lambda_j \boldsymbol{u}_j \boldsymbol{u}_j^T$ be the eigenvalue decomposition of W. Consider a constraint $g_i(\Phi^T W \Phi) \leq b_i$ as specified in (13). Note that if the j-th eigenvector $\boldsymbol{u}_j$ of W is orthogonal to the range space of Φ, that is, $\Phi^T \boldsymbol{u}_j = 0$, then the corresponding eigenvalue λ_j is not constrained (except for the non-negativity constraint imposed by the positive semi-definiteness constraint). Since the range space of Φ is at most n-dimensional, we can assume that $\lambda_j \geq 0, \forall j > n$ are not constrained by the linear inequality constraints in (13).

Since f satisfies the conditions of Theorem 4, $f(W) = \sum_j f_s(\lambda_j)$. Also, $f_s(\alpha) = \min_x f_s(x)$. Hence, to minimize $f(W)$, we can select $\lambda_j^* = \alpha \geq 0, \forall j > n$ (note that the non-negativity constraint is also satisfied here). Furthermore, the eigenvectors $\boldsymbol{u}_j$, $\forall j \leq n$, lie in the range space of Φ, that is, $\forall j \leq n$, $\boldsymbol{u}_j = \Phi \boldsymbol{z}_j$ for some $\boldsymbol{z}_j \in \mathbb{R}^n$. Therefore,

$$W^* = \sum_{j=1}^{n} \lambda_j^* \boldsymbol{u}_j^* \boldsymbol{u}_j^{*T} + \alpha \sum_{j=n+1}^{d} \boldsymbol{u}_j^* \boldsymbol{u}_j^{*T} = \sum_{j=1}^{n} (\lambda_j^* - \alpha) \boldsymbol{u}_j^* \boldsymbol{u}_j^{*T} + \alpha \sum_{j=1}^{d} \boldsymbol{u}_j^* \boldsymbol{u}_j^{*T} = \Phi S \Phi^T + \alpha I^d,$$

where $S = \sum_{j=1}^n (\lambda_j^* - \alpha) \boldsymbol{z}_j^* \boldsymbol{z}_j^{*T}$. ■

Now we use Lemmas 5 and 6 to prove Theorem 4.

Proof [Proof of Theorem 4] Let $\Phi = U_\Phi \Sigma V_\Phi^T$ be the singular value decomposition (SVD) of Φ. Note that $K_0 = \Phi^T \Phi = V_\Phi \Sigma^2 V_\Phi^T$, so $\Sigma V_\Phi^T = V_\Phi^T K_0^{1/2}$. Also, assuming $\Phi \in \mathbb{R}^{d\times n}$ to be full-rank and $d > n$, $V_\Phi V_\Phi^T = I$.

Using Lemma 6, the optimal solution to (13) is restricted to be of the form $W = \alpha I^d + \Phi S \Phi^T = \alpha I^d + U_\Phi \Sigma V_\Phi^T S V_\Phi \Sigma U_\Phi^T = \alpha I^d + U_\Phi V_\Phi^T K_0^{1/2} S K_0^{1/2} V_\Phi U_\Phi^T = \alpha I^d + U_\Phi V_\Phi^T L V_\Phi U_\Phi^T$, where $L = K_0^{1/2} S K_0^{1/2}$.

As a result, for spectral functions f, (13) is equivalent to (14), so using Lemma 5, (13) is equivalent to (15) with $U = U_\Phi V_\Phi^T$ and $L = K_0^{1/2} S K_0^{1/2}$. Also, note that the constraints in (15) can be simplified to:

$$g_i(\alpha\Phi^T\Phi + \Phi^T ULU^T\Phi) \le b_i \equiv g_i(\alpha K_0 + K_0^{1/2} L K_0^{1/2}) \le b_i.$$

Now, let $K = \alpha K_0 + K_0^{1/2} L K_0^{1/2} = \alpha K_0 + K_0 S K_0$, that is, $L = K_0^{-1/2}(K - \alpha K_0)K_0^{-1/2}$. Theorem 4 now follows by substituting for L in (15). ■

As a first consequence of this result, we can achieve induction over the learned kernels, analogous to (6) for the LogDet case. Given that $K = \Phi^T W \Phi$, we can see that the learned kernel function is a linear transformation kernel; that is, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T W \phi(\boldsymbol{x}_j)$. Given a pair of new data points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, we use the fact that the learned kernel is a linear transformation kernel, along with the first result of the theorem ($W = \alpha I^d + \Phi S \Phi^T$) to compute the learned kernel as:

$$\phi(\boldsymbol{z}_1)^T W \phi(\boldsymbol{z}_2) = \alpha \cdot \kappa_0(\boldsymbol{z}_1, \boldsymbol{z}_2) + \boldsymbol{k}_1^T S \boldsymbol{k}_2, \text{where } \boldsymbol{k}_i = [\kappa_0(\boldsymbol{z}_i, \boldsymbol{x}_1), ..., \kappa_0(\boldsymbol{z}_i, \boldsymbol{x}_n)]^T. \quad (18)$$

Since LogDet divergence is also a spectral function, Theorem 4 is a generalization of Theorem 1 and implies kernelization for our metric learning formulation (2). Moreover, many Mahalanobis metric learning methods can be viewed as a special case of (13), so a corollary of Theorem 4 is that we can constructively apply these metric learning methods in kernel space by solving their corresponding kernel learning problem, and then compute the learned metrics via (18). Kernelization of Mahalanobis metric learning has previously been established for some special cases; our results generalize and extend previous methods, as well as provide simpler techniques in some cases. We will further elaborate in Section 5 with several special cases.

4.3 Parameter Reduction

As noted in Section 3.6 that the size of the kernel matrices K and the parameter matrices S are $n \times n$, and thus grow quadratically with the number of data points. Similar to the special case of the LogDet divergence (see Section 3.6), we would like to have a way to restrict our general optimization problem (12) over a smaller number of parameters. So, we now discuss a generalization of (13) by introducing an additional constraint to make it possible to reduce the number of parameters to learn, permitting scalability to data sets with many training points *and* with very high dimensionality.

Theorem 4 shows that the optimal K^* is of the form $\Phi^T W^* \Phi = \alpha K_0 + K_0 S K_0$. In order to accommodate fewer parameters to learn, a natural option is to replace the unknown S matrix with a *low-rank* matrix JLJ^T, where $J \in \mathbb{R}^{n \times k}$ is a pre-specified matrix, $L \in \mathbb{R}^{k \times k}$ is unknown (we use L instead of S to emphasize that S is of size $n \times n$ whereas L is $k \times k$), and the rank k is a parameter of the algorithm. Then, we will explicitly enforce that the learned kernel is of this form.

By plugging in $K = \alpha K_0 + K_0 S K_0$ into (12) and replacing S with JLJ^T, the resulting optimization problem is given by:

$$\min_{L \succeq 0} \; f(\alpha I^n + K_0^{1/2} JLJ^T K_0^{1/2}), \quad \text{s.t. } g_i(\alpha K_0 + K_0 JLJ^T K_0) \le b_i, \; 1 \le i \le m. \quad (19)$$

Note that the above problem is a strict generalization of our LogDet function based parameter reduction approach (see Section 3.6).

While the above problem involves only $k \times k$ variables, the functions f and g_i's are applied to $n \times n$ matrices and therefore the problem may still be computationally expensive to optimize. Below, we show that for any spectral function f and linear constraints $g_i(K) = \mathrm{tr}(C_iK)$, (19) reduces to a problem that applies f and g_i's to $k \times k$ matrices only, which provides significant scalability.

Theorem 7 *Let $K_0 = \Phi^T\Phi$ and J be some matrix in $\mathbb{R}^{n\times k}$. Also, let the loss function f be a spectral function (see Definition 3) such that the corresponding strictly convex scalar function f_s has the global minimum at $\alpha > 0$. Then problem* (19) *with $g_i(K) = \mathrm{tr}(C_iK)$ is equivalent to the following alternative optimization problem:*

$$\min_{L\succeq -\alpha(K^J)^{-1}} \; f((K^J)^{-1/2}(\alpha K^J + K^J L K^J)(K^J)^{-1/2}),$$
$$s.t. \; \mathrm{tr}(LJ^TK_0C_iK_0J) \leq b_i - \mathrm{tr}(\alpha K_0 C_i), \; 1 \leq i \leq m, \tag{20}$$

where $K^J = J^TK_0J$.

Proof Let $U = K_0^{1/2}J(J^TK_0J)^{-1/2}$ and let J be a full rank matrix, then U is an orthogonal matrix. Using (17) we get,

$$f(\alpha I^n + U(J^TK_0J)^{1/2}L(J^TK_0J)^{1/2}U^T) = f(\alpha I^k + (J^TK_0J)^{1/2}L(J^TK_0J)^{1/2}).$$

Now consider a linear constraint $\mathrm{tr}(C_i(\alpha K_0 + K_0JLJ^TK_0)) \leq b_i$. This can be easily simplified to $\mathrm{tr}(LJ^TK_0C_iK_0J) \leq b_i - \mathrm{tr}(\alpha K_0C_i)$. Similar simple algebraic manipulations to the PSD constraint completes the proof. ■

Note that (20) is over $k \times k$ matrices (after initial pre-processing) and is in fact similar to the kernel learning problem (12), but with a kernel K^J of smaller size $k \times k$, $k \ll n$.

Similar to (12), we can show that (19) is also equivalent to LT kernel function learning. This enables us to naturally apply the above kernel learning problem in the inductive setting.

Theorem 8 *Consider* (19) *with $g_i(K) = \mathrm{tr}(C_iK)$ and a spectral function f whose corresponding scalar function f_s has a global minimum at $\alpha > 0$. Let $J \in \mathbb{R}^{n\times k}$. Then,* (19) *and* (20) *with $g_i(K) = \mathrm{tr}(C_iK)$ are equivalent to the following linear transformation kernel learning problem (analogous to the connection between* (12) *and* (13)*):*

$$\min_{W\succeq 0, L} f(W), \qquad s.t. \; \mathrm{tr}(\Phi^TW\Phi) \leq b_i, \; 1 \leq i \leq m, \quad W = \alpha I^d + \Phi JLJ^T\Phi^T. \tag{21}$$

Proof Consider the last constraint in (21): $W = \alpha I^d + \Phi JLJ^T\Phi^T$. Let $\Phi = U\Sigma V^T$ be the SVD of Φ. Hence, $W = \alpha I^d + UV^TV\Sigma V^TJLJ^TV\Sigma V^TVU^T = \alpha I^d + UV^TK_0^{1/2}JLJ^TK_0^{1/2}VU^T$, where we used $K_0^{1/2} = V\Sigma V^T$. For dis-ambiguity, rename L as L' and U as U'. The result now follows by using Lemma 5 where $U = U'V^T$ and $L = K_0^{1/2}JL'J^TK_0^{1/2}$. ■

Note that, in contrast to (13), where the last constraint over W is achieved automatically, (21) requires this constraint on W to be satisfied during the optimization process, which leads to a reduced number of parameters for our kernel learning problem. The above theorem shows that our reduced-parameter kernel learning method (19) also implicitly learns a linear transformation kernel function, hence we can generalize the learned kernel to unseen data points using an expression similar to (18).

5. Special Cases

In the previous section, we proved a general result showing the connections between metric and kernel learning using a wide class of loss functions and constraints. In this section, we consider a few special cases of interest: the von Neumann divergence, the squared Frobenius norm and semidefinite programming. For each of the cases, we derive the required optimization problem and mention the relevant optimization algorithms that can be used.

5.1 Von Neumann Divergence

The von Neumann divergence is a generalization of the well known KL-divergence to matrices. It is used extensively in quantum computing to compare density matrices of two different systems (Nielsen and Chuang, 2000). It is also used in the exponentiated matrix gradient method by Tsuda et al. (2005), online-PCA method by Warmuth and Kuzmin (2008) and fast SVD solver by Arora and Kale (2007). The von Neumann divergence between A and A_0 is defined to be, $D_{\mathrm{vN}}(A,A_0) = \mathrm{tr}(A\log A - A\log A_0 - A + A_0)$, where both A and A_0 are positive definite. Computing the von Neumann divergence with respect to the identity matrix, we get: $f_{\mathrm{vN}}(A) = \mathrm{tr}(A\log A - A + I)$. Note that f_{vN} is a spectral function with corresponding scalar function $f_{\mathrm{vN}}(\lambda) = \lambda\log\lambda - \lambda$ and minima at $\lambda = 1$. Now, the kernel learning problem (12) with loss function f_{vN} and linear constraints is:

$$\min_{K\succeq 0} \; f_{\mathrm{vN}}(K_0^{-1/2}KK_0^{-1/2}), \qquad \text{s.t. } \; \mathrm{tr}(KC_i) \le b_i, \quad \forall 1 \le i \le m. \tag{22}$$

As f_{vN} is an spectral function, using Theorem 4, the above kernel learning problem is equivalent to the following metric learning problem:

$$\min_{W\succeq 0} \; D_{\mathrm{vN}}(W,I), \qquad \text{s.t. } \; \mathrm{tr}(W\Phi C_i\Phi^T) \le b_i, \quad \forall 1 \le i \le m.$$

Using elementary linear algebra, we obtain the following simplified version:

$$\min_{\lambda_1,\lambda_2,\ldots,\lambda_m \ge 0} \; F(\lambda) = \mathrm{tr}(\exp(-C(\lambda)K_0)) + b(\lambda),$$

where $C(\lambda) = \sum_i \lambda_i C_i$ and $b(\lambda) = \sum_i \lambda_i b_i$. Further, $\frac{\partial F}{\partial \lambda_i} = \mathrm{tr}(\exp(-C(\lambda)K_0)C_iK_0) + b_i$, so any first-order smooth optimization method can be used to solve the above dual problem. Alternatively, similar to the method of Kulis et al. (2008), Bregman's projection method can be used to solve the primal problem (22).

5.2 Pseudo Online Metric Learning (POLA)

Shalev-Shwartz et al. (2004) proposed the following batch metric learning formulation:

$$\min_{W\succeq 0} \|W\|_F^2, \qquad \text{s.t. } \; y_{ij}(b - d_W(\boldsymbol{x}_i,\boldsymbol{x}_j)) \ge 1, \;\; \forall (i,j) \in \mathcal{P},$$

where $y_{ij} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, and $y_{ij} = -1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are dissimilar. $\mathcal{P}$ is a set of pairs of points with known distance constraints. POLA is an instantiation of (13) with $f(A) = \frac{1}{2}\|A\|_F^2$ and side-information available in the form of pair-wise distance constraints. Note that the regularizer $f(A) = \frac{1}{2}\|A\|^2$ was also employed in Schultz and Joachims (2003) and Kwok and Tsang (2003), and these methods also fall under our general formulation. In this case, f is once again a strictly convex spectral function, and its global minimum is $\alpha = 0$, so we can use (12) to solve for the learned kernel K as

$$\min_{K \succeq 0} \ \|KK_0^{-1}\|_F^2, \qquad \text{s.t. } g_i(K) \leq b_i, \ 1 \leq i \leq m,$$

The constraints g_i for this problem can be easily constructed by re-writing each of POLA's constraints as a function of $\Phi^T W \Phi$. Note that the above approach for kernelization is much simpler than the method suggested in Shalev-Shwartz et al. (2004), which involves a kernelized Gram-Schmidt procedure at each step of the algorithm.

5.3 SDPs

Weinberger et al. (2005) and Globerson and Roweis (2005) proposed metric learning formulations that can be rewritten as semi-definite programs (SDP), which is a special case of (13) with the loss function being a linear function. Consider the following general semidefinite program (SDP) to learn a linear transformation W:

$$\min_{W \succeq 0} \ \operatorname{tr}(W\Phi C_0 \Phi^T), \qquad \text{s.t. } \operatorname{tr}(W\Phi C_i \Phi^T) \leq b_i, \quad \forall 1 \leq i \leq m. \tag{23}$$

Here we show that this problem can be efficiently solved for high dimensional data in its kernel space, hence kernelizing the metric learning methods introduced by Weinberger et al. (2005) and Globerson and Roweis (2005).

Theorem 9 *Problem* (23) *is kernelizable.*

Proof (23) has a linear objective, that is, it is a non-strict convex problem that may have multiple solutions. A variety of regularizations can be considered that lead to slightly different solutions. Here, we consider the LogDet regularization, which seeks the solution with maximum determinant. To this effect, we add a log-determinant regularization:

$$\min_{W \succeq 0} \ \operatorname{tr}(W\Phi C_0 \Phi^T) - \gamma \log\det W, \qquad \text{s.t. } \operatorname{tr}(W\Phi C_i \Phi^T) \leq b_i, \quad \forall 1 \leq i \leq m. \tag{24}$$

The above regularization was also considered by Kulis et al. (2009b), who provided a fast projection algorithm for the case when each C_i is a one-rank matrix and discussed conditions for which the optimal solution to the regularized problem is an optimal solution to the original SDP. The above formulation also generalizes the metric learning formulation of RCA (Bar-Hillel et al., 2005).

Consider the following variational formulation of (24):

$$\begin{aligned} \min_t \min_{W \succeq 0} \ t - \gamma \log\det W, \qquad \text{s.t. } & \operatorname{tr}(W\Phi C_i \Phi^T) \leq b_i, \quad \forall 1 \leq i \leq m, \\ & \operatorname{tr}(W\Phi C_0 \Phi^T) \leq t. \end{aligned} \tag{25}$$

Note that the objective function of the inner optimization problem of (25) is a spectral function and hence using Theorem 4, (25), or equivalently (24), is kernelizable. ■

5.4 Trace-norm Based Inductive Semi-supervised Kernel Dimensionality Reduction (Trace-SSIKDR)

Finally, we apply our framework to semi-supervised kernel dimensionality reduction, which provides a novel and practical application of our framework. While there exists a variety of methods for kernel dimensionality reduction, most of these methods are unsupervised (e.g., kernel-PCA) or are restricted to the transductive setting. In contrast, we can use our kernel learning framework to learn a low-rank transformation of the feature vectors implicitly that in turn provides a low-dimensional embedding of the data set. Furthermore, our framework permits a variety of side-information such as pair-wise or relative distance constraints, beyond the class label information allowed by existing transductive methods.

We describe our method starting from the linear transformation problem. Our goal is to learn a low-rank linear transformation W whose corresponding low-dimensional mapped embedding of $\boldsymbol{x}_i$ is $W^{1/2}\phi(\boldsymbol{x}_i)$. Even when the dimensionality of $\phi(\boldsymbol{x}_i)$ is very large, if the rank of W is low enough, then the mapped embedding will have small dimensionality. With that in mind, a possible regularizer could be the rank, that is, $r(A) = \text{rank}(A)$; one can easily show that this satisfies the definition of a spectral function. Unfortunately, optimization is intractable in general with the non-convex rank function, so we use the trace-norm relaxation for the matrix rank function, that is, we set $f(A) = \text{tr}(A)$. This function has been extensively studied as a relaxation for the rank function in Recht et al. (2010), and it satisfies the definition of a spectral function (with $\alpha = 0$). We also add a small Frobenius norm regularization for ease of optimization (this does not affect the spectral property of the regularization function). Then using Theorem 4, the resulting relaxed kernel learning problem is:

$$\min_{K\succeq 0} \tau\,\text{tr}(K_0^{-1/2}KK_0^{-1/2}) + \|K_0^{-1/2}KK_0^{-1/2}\|_F^2, \qquad \text{s.t.} \qquad \text{tr}(C_iK) \le b_i,\ 1 \le i \le m, \tag{26}$$

where $\tau > 0$ is a parameter. The above problem can be solved using a method based on Uzawa's inexact algorithm, similar to Cai et al. (2008).

We briefly describe the steps taken by our method at each iteration. For simplicity, denote $\tilde{K} = K_0^{-1/2}KK_0^{-1/2}$; we will optimize with respect to $\tilde{K}$ instead of K. Let $\tilde{K}^t$ be the t-th iterate. Associate variable $z_i^t, 1 \le i \le m$ with each constraint at each iteration t, and let $z_i^0 = 0, \forall i$. Let δ_t be the step size at iteration t. The algorithm performs the following updates:

$$\begin{aligned} U\Sigma U^T &\leftarrow K_0^{1/2}CK_0^{1/2}, \\ \tilde{K}^t &\leftarrow U\max(\Sigma - \tau I, 0)U^T, \qquad (27) \\ z_i^t &\leftarrow z_i^{t-1} - \delta\max(\text{tr}(C_iK^{1/2}\tilde{K}^tK^{1/2}) - b_i, 0),\ \forall i, \qquad (28) \end{aligned}$$

where $C = \sum_i z_i^{t-1}C_i$. The above updates require computation of $K_0^{1/2}$ which is expensive for large high-rank matrices. However, using elementary linear algebra we can show that $\tilde{K}$ and the learned kernel function can be computed efficiently without computing $K_0^{1/2}$ by maintaining $S = K_0^{-1/2}\tilde{K}K_0^{-1/2}$ from step to step.

We first prove a technical lemma to relate eigenvectors U of $K_0^{1/2}CK_0^{1/2}$ and V of the matrix CK_0.

Lemma 10 *Let $K_0^{1/2}CK_0^{1/2} = U_k\Sigma_kU_k^T$, where U_k contains the top-k eigenvectors of $K_0^{1/2}CK_0^{1/2}$ and Σ_k contains the top-k eigenvalues of $K_0^{1/2}CK_0^{1/2}$. Similarly, let $CK_0 = V_k\Lambda_kV_k^{-1}$, where V_k contains*

Algorithm 2 Trace-SSIKDR

```
Input: $K_0, (C_i, b_i), 1 \leq i \leq m, \tau, \delta$
1: Initialize: $z_i^0 = 0, t = 0$
2: repeat
3:     $t = t + 1$
4:     Compute $V_k$ and $\Sigma_k$, the top $k$ eigenvectors and eigenvalues of $\left(\sum_i z_i^{t-1} C_i\right) K_0$, where $k = \text{argmax}_j \sigma_j > \tau$
5:     $D_k(i,i) \leftarrow 1/\boldsymbol{v}_i^T K_0 \boldsymbol{v}_i, 1 \leq i \leq k$
6:     $z_i^t \leftarrow z_i^{t-1} - \delta \max(\text{tr}(C_i K_0 V_k D_k \Sigma_k D_k V_k^T K_0) - b_i, 0), \forall i.$
7: until Convergence
8: Return $\Sigma_k, D_k, V_k$
```

the top-k right eigenvectors of CK_0 and Λ_k contains the top-k eigenvalues of CK_0. Then,

$$U_k = K_0^{1/2} V_k D_k, \qquad \Sigma_k = \Lambda_k,$$

where D_k is a diagonal matrix with i-th diagonal element $D_k(i,i) = 1/\boldsymbol{v}_i^T K_0 \boldsymbol{v}_i$. Note that eigenvalue decomposition is unique up to sign, so we assume that the sign has been set correctly.

Proof Let $\boldsymbol{v}_i$ be i-th eigenvector of CK_0. Then, $CK_0 \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$. Multiplying both sides with $K_0^{1/2}$, we get $K_0^{1/2} C K_0^{1/2} K_0^{1/2} \boldsymbol{v}_i = \lambda_i K_0^{1/2} \boldsymbol{v}_i$. After normalization we get:

$$(K_0^{1/2} C K_0^{1/2}) \frac{K_0^{1/2} \boldsymbol{v}_i}{\boldsymbol{v}_i^T K_0 \boldsymbol{v}_i} = \lambda_i \frac{K_0^{1/2} \boldsymbol{v}_i}{\boldsymbol{v}_i^T K_0 \boldsymbol{v}_i}.$$

Hence, $\frac{K_0^{1/2} \boldsymbol{v}_i}{\boldsymbol{v}_i^T K_0 \boldsymbol{v}_i} = K_0^{1/2} \boldsymbol{v}_i D_k(i,i)$ is the i-th eigenvector $\boldsymbol{u}_i$ of $K_0^{1/2} C K_0^{1/2}$. Also, $\Sigma_k(i,i) = \lambda_i$. ■

Using the above lemma and (27), we get: $\tilde{K} = K_0^{1/2} V_k D_k \lambda D_k V_k^{-1} K_0^{1/2}$. Therefore, the update for the z variables (see (28)) reduces to:

$$z_i^t \leftarrow z_i^{t-1} - \delta \max(\text{tr}(C_i K_0 V_k D_k \lambda D_k V_k^{-1} K_0) - b_i, 0), \forall i.$$

Lemma 10 also implies that if k eigenvalues of CK_0 are larger than τ then we only need the top k eigenvalues of CK_0. Since k is typically significantly smaller than n, the update should be significantly more efficient than computing the whole eigenvalue decomposition.

Algorithm 2 details an efficient method for optimizing (26) and returns matrices Σ_k, D_k and V_k, all of which contain only $O(nk)$ parameters, where k is the rank of $\tilde{K}^t$, which changes from iteration to iteration. Note that step 4 of the algorithm computes k singular vectors and requires only $O(nk^2)$ computation. Note also that the learned embedding $\boldsymbol{x}_i \rightarrow \tilde{K}^{1/2} K_0^{-1/2} \boldsymbol{k}_i$, where $\boldsymbol{k}_i$ is a vector of input kernel function values between $\boldsymbol{x}_i$ and the training data, can be computed efficiently as $\boldsymbol{x}_i \rightarrow \Sigma_k^{1/2} D_k V_k \boldsymbol{k}_i$, which does not require $K_0^{1/2}$ explicitly.

6. Experimental Results

In Section 3, we presented metric learning as a constrained LogDet optimization problem to learn a linear transformation, and we showed that the problem can be efficiently kernelized to learn linear transformation kernels. Kernel learning yields two fundamental advantages over standard non-kernelized metric learning. First, a non-linear kernel can be used to learn non-linear decision boundaries common in applications such as image analysis. Second, in Section 3.6, we showed that the kernelized problem can be learned with respect to a reduced basis of size k, admitting a learned kernel parameterized by $O(k^2)$ values. When the number of training examples n is large, this represents a substantial improvement over optimizing over the entire $O(n^2)$ matrix, both in terms of computational efficiency as well as statistical robustness. In Section 4, we generalized kernel function learning to other loss functions. A special case of our approach is the trace-norm based kernel function learning problem, which can be applied to the task of semi-supervised inductive kernel dimensionality reduction.

In this section, we present experiments from several domains: benchmark UCI data, automated software debugging, text analysis, and object recognition for computer vision. We evaluate performance of our learned distance metrics or kernel functions in the context of a) classification accuracy for the k-nearest neighbor algorithm, b) kernel dimensionality reduction. For the classification task, our k-nearest neighbor classifier uses $k = 10$ nearest neighbors (except for Section 6.3 where we use $k = 1$), breaking ties arbitrarily. We select the value of k arbitrarily and expect to get slightly better accuracies using cross-validation. Accuracy is defined as the number of correctly classified examples divided by the total number of classified examples. For the dimensionality reduction task, we visualize the two dimensional embedding of the data using our trace-norm based method with pairwise similarity/dissimilarity constraints.

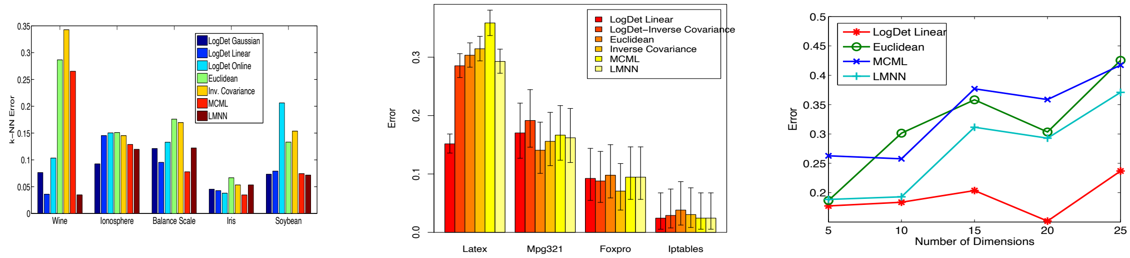
For our proposed algorithms, pairwise constraints are inferred from true class labels. For each class i, 100 pairs of points are randomly chosen from within class i and are constrained to be similar, and 100 pairs of points are drawn from classes other than i to form dissimilarity constraints. Given c classes, this results in $100c$ similarity constraints, and $100c$ dissimilarity constraints, for a total of $200c$ constraints. The upper and lower bounds for the similarity and dissimilarity constraints are determined empirically as the 1^{st} and 99^{th} percentiles of the distribution of distances computed using a baseline Mahalanobis distance parameterized by W_0. Finally, the slack penalty parameter γ used by our algorithms is cross-validated using values $\{.01, .1, 1, 10, 100, 1000\}$.

All metrics/kernels are trained using data only in the training set. Test instances are drawn from the test set and are compared to examples in the training set using the learned distance/kernel function. The test and training sets are established using a standard two-fold cross validation approach. For experiments in which a baseline distance metric/kernel is evaluated (for example, the squared Euclidean distance), nearest neighbor searches are again computed from test instances to only those instances in the training set.

For additional large-scale results, see Kulis et al. (2009a), which uses our parameter-reduction strategy to learn kernels over a data set containing nearly half a million images in 24,000 dimensional space for the problem of human-body pose estimation; we also applied our algorithms on the MNIST data set of 60,000 digits in Kulis et al. (2008).

6.1 Low-Dimensional Data Sets

First we evaluate our LogDet divergence based metric learning method (see Algorithm 1) on the standard UCI data sets in the low-dimensional (non-kernelized) setting, to directly compare with several existing metric learning methods. In Figure 1 (a), we compare LogDet Linear (K_0 equals the linear kernel) and the LogDet Gaussian (K_0 equals Gaussian kernel in kernel space) algorithms



(a)UCI Data Sets (b) Clarify Data Sets (c) Latex

Figure 1: **(a)**: Results over benchmark UCI data sets. LogDet metric learning was run with in input space (LogDet Linear) as well as in kernel space with a Gaussian kernel (LogDet Gaussian). **(b), (c)**: Classification error rates for *k*-nearest neighbor software support via different learned metrics. We see in figure (b) that LogDet Linear is the only algorithm to be optimal (within the 95% confidence intervals) across all data sets. In (c), we see that the error rate for the Latex data set stays relatively constant for LogDet Linear.

against existing metric learning methods for *k*-NN classification. We also show results of a recently-proposed online metric learning algorithm based on the LogDet divergence over this data (Jain et al., 2008), called LogDet Online. We use the squared Euclidean distance, $d(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})$ as a baseline method (i.e., $W_0 = I$). We also use a Mahalanobis distance parameterized by the inverse of the sample covariance matrix (i.e., $W_0 = \Sigma^{-1}$, where Σ is the sample covariance matrix of the data). This method is equivalent to first performing a standard PCA whitening transform over the feature space and then computing distances using the squared Euclidean distance. We compare our method to two recently proposed algorithms: Maximally Collapsing Metric Learning by Globerson and Roweis (2005) (MCML), and metric learning via Large Margin Nearest Neighbor by Weinberger et al. (2005) (LMNN). Consistent with existing work such as Globerson and Roweis (2005), we found the method of Xing et al. (2002) to be very slow and inaccurate, so the latter was not included in our experiments. As seen in Figure 1 (a), LogDet Linear and LogDet Gaussian algorithms obtain somewhat higher accuracy for most of the data sets. In addition to our evaluations on standard UCI data sets, we also apply our algorithm to the recently proposed problem of nearest neighbor software support for the Clarify system (Ha et al., 2007). The basis of the Clarify system lies in the fact that modern software design promotes modularity and abstraction. When a program terminates abnormally, it is often unclear which component should be responsible or capable of providing an error report. The system works by monitoring a set of predefined program features (the data sets presented use function counts) during program runtime which are then used by a classifier in the event of abnormal program termination. Nearest neighbor searches are particularly relevant to this problem. Ideally, the neighbors returned should not only have the correct class label, but should also represent similar program configurations or program inputs. Such a matching can be a powerful tool to help users diagnose the root cause of their problem. The four data sets we use correspond to the following software: Latex (the document compiler, 9 classes), Mpg321 (an mp3 player, 4 classes), Foxpro (a database manager, 4 classes), and Iptables (a Linux kernel application, 5 classes).

Data Set	n	d	Unsupervised	LogDet Linear	HMRF-KMeans
Ionosphere	351	34	0.314	**0.113**	0.256
Digits-389	317	16	0.226	**0.175**	0.286

Table 1: Unsupervised k-means clustering error using the baseline squared Euclidean distance, along with semi-supervised clustering error with 50 constraints.

Our experiments on the Clarify system, like the UCI data, are over fairly low-dimensional data. Ha et al. (2007) showed that high classification accuracy can be obtained by using a relatively small subset of available features. Thus, for each data set, we use a standard information gain feature selection test to obtain a reduced feature set of size 20. From this, we learn metrics for k-NN classification using the methods developed in this paper. Results are given in Figure 1(b). The LogDet Linear algorithm yields significant gains for the Latex benchmark. Note that for data sets where Euclidean distance performs better than the inverse covariance metric, the LogDet Linear algorithm that normalizes to the standard Euclidean distance yields higher accuracy than that regularized to inverse covariance (LogDet-Inverse Covariance). In general, for the Mpg321, Foxpro, and Iptables data sets, learned metrics yield only marginal gains over the baseline Euclidean distance measure.

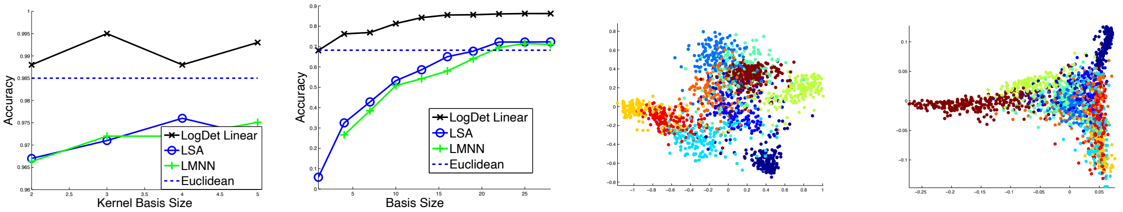
Figure 1(c) shows the error rate for the Latex data sets with a varying number of features (the feature sets are again chosen using the information gain criteria). We see here that LogDet Linear is surprisingly robust. Euclidean distance, MCML, and LMNN all achieve their best error rates for five dimensions. LogDet Linear, however, attains its lowest error rate of .15 at $d = 20$ dimensions.

We also briefly present some semi-supervised clustering results for two of the UCI data sets. Note that both MCML and LMNN are not amenable to optimization subject to pairwise distance constraints. Instead, we compare our method to the semi-supervised clustering algorithm HMRF-KMeans (Basu et al., 2004). We use a standard 2-fold cross validation approach for evaluating semi-supervised clustering results. Distances are constrained to be either similar or dissimilar, based on class values, and are drawn only from the training set. The entire data set is then clustered into c clusters using k-means (where c is the number of classes) and error is computed using only the test set. Table 1 provides results for the baseline k-means error, as well as semi-supervised clustering results with 50 constraints.

6.2 Metric Learning for Text Classification

Next we present results in the text domain. Our text data sets are created by standard bag-of-words Tf-Idf representations. Words are stemmed using a standard Porter stemmer and common stop words are removed, and the text models are limited to the 5,000 words with the largest document frequency counts. We provide experiments for two data sets: CMU 20-Newsgroups Data Set (2008), and Classic3 Data Set (2008). Classic3 is a relatively small 3 class problem with 3,891 instances. The newsgroup data set is much larger, having 20 different classes from various newsgroup categories and 20,000 instances.

Our text experiments employ a linear kernel, and we use a set of basis vectors that is constructed from the class labels via the following procedure. Let c be the number of distinct classes and let k be the size of the desired basis. If $k = c$, then each class mean r_i is computed to form the basis $R = [\boldsymbol{r}_1 \ldots \boldsymbol{r}_c]$. If $k < c$ a similar process is used but restricted to a randomly selected subset of



(a) Classic3 (b) 20-Newsgroups (c)USPS Digits (d)USPS Digits

Figure 2: **(a), (b)**: Classification accuracy for our Mahalanobis metrics learned over basis of different dimensionality. Overall, our method (LogDet Linear) significantly outperforms existing methods. **(c)**: Two-dimensional embedding of 2000 USPS digits obtained using our method Trace-SSIKDR for a training set of just 100 USPS digits. Note that we use the **inductive** setting here and the embedding is color coded according to the underlying digit. **(d)**: Embedding of the USPS digits data set obtained using kernel-PCA.

k classes. If $k > c$, instances within each class are clustered into approximately $\frac{k}{c}$ clusters. Each cluster's mean vector is then computed to form the set of low-rank basis vectors R. Figure 2 shows classification accuracy across bases of varying sizes for the Classic3 data set, along with the newsgroup data set. As baseline measures, the standard squared Euclidean distance is shown, along with Latent Semantic Analysis (LSA) (Deerwester et al., 1990), which works by projecting the data via principal components analysis (PCA), and computing distances in this projected space. Comparing our algorithm to the baseline Euclidean measure, we can see that for smaller bases, the accuracy of our algorithm is similar to the Euclidean measure. As the size of the basis increases, our method obtains significantly higher accuracy compared to the baseline Euclidean measure.

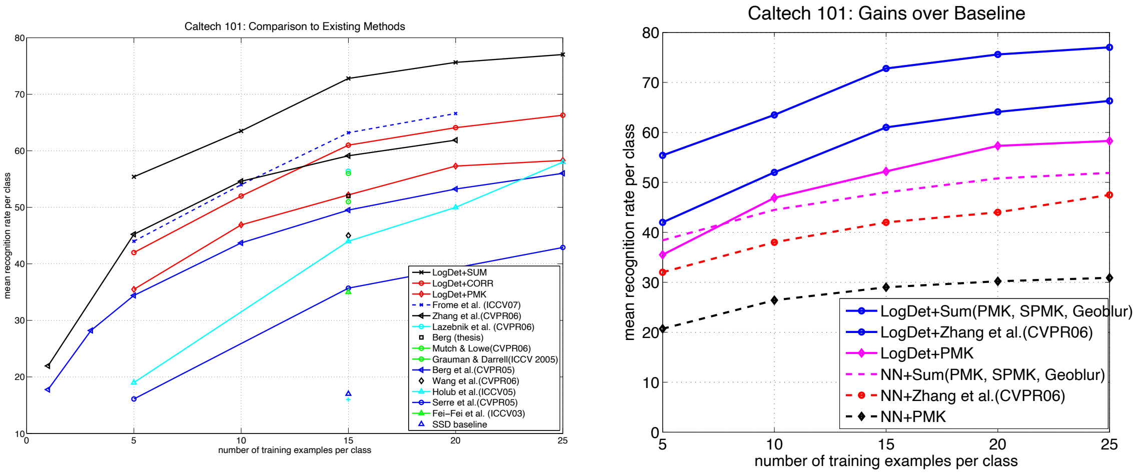
6.3 Kernel Learning for Visual Object Recognition

Next we evaluate our method over high-dimensional data applied to the object-recognition task using Caltech-101 Data Set (2004), a common benchmark for this task. The goal is to predict the category of the object in the given image using a k-NN classifier.

We compute distances between images using learning kernels with three different base image kernels: 1) PMK: Grauman and Darrell's pyramid match kernel (Grauman and Darrell, 2007) applied to SIFT features, 2) CORR: the kernel designed by Zhang et al. (2006) applied to geometric blur features , and 3) SUM: the average of four image kernels, namely, PMK (Grauman and Darrell, 2007), Spatial PMK (Lazebnik et al., 2006), and two kernels obtained via geometric blur (Berg and Malik, 2001). Note that the underlying dimensionality of these embeddings is typically in the millions of dimensions.

We evaluate the effectiveness of metric/kernel learning on this data set. We pose a k-NN classification task, and evaluate both the original (SUM, PMK or CORR) and learned kernels. We set $k = 1$ for our experiments; this value was chosen arbitrarily. We vary the number of training examples T per class for the database, using the remainder as test examples, and measure accuracy in terms of the mean recognition rate per class, as is standard practice for this data set.

Figure 3 (a) shows our results relative to several other existing techniques that have been applied to this data set. Our approach outperforms all existing single-kernel classifier methods when using



(a) (b)

Figure 3: Results on Caltech-101. LogDet+SUM refers to our learned kernel when the base kernel is the average of four kernels (PMK, SPMK, Geoblur-1, Geoblur-2), LogDet+PMK refers to the learned kernel when the base kernel is pyramid match kernel, and LogDet+CORR refers to the learned kernel when the base kernel is correspondence kernel of Zhang et al. (2006). **(a)**: Comparison of LogDet based metric learning method with other state-of-the-art object recognition methods. Our method outperforms all other single metric/kernel approaches. **(b)**: Our learned kernels significantly improve NN recognition accuracy relative to their non-learned counterparts, the SUM (average of four kernels), the CORR and PMK kernels.

the learned CORR kernel: we achieve 61.0% accuracy for $T = 15$ and 69.6% accuracy for $T = 30$. Our learned PMK achieves 52.2% accuracy for $T = 15$ and 62.1% accuracy for $T = 30$. Similarly, our learned SUM kernel achieves 73.7% accuracy for $T = 15$. Figure 3 (b) specifically shows the comparison of the original baseline kernels for NN classification. The plot reveals gains in 1-NN classification accuracy; notably, our learned kernels with simple NN classification also outperform the baseline kernels when used with SVMs (Zhang et al., 2006; Grauman and Darrell, 2007).

6.4 USPS Digits

Finally, we qualitatively evaluate our dimensionality reduction method (see Section 5.4) on the USPS digits data set. Here, we train our method using 100 examples to learn a mapping to two dimensions, that is, a rank-2 matrix W. For the baseline kernel, we use the data-dependent kernel function proposed by Sindhwani et al. (2005) that also accounts for the manifold structure of the data within the kernel function. We then embed 2000 (unseen) test examples into two dimensions using our learned low-rank transformation. Figure 2 (c) shows the embedding obtained by our Trace-SSIKDR method, while Figure 2 (d) shows the embedding obtained by the kernel-PCA algorithm. Each point is color coded according to the underlying digit. Note that our method is able to separate out seven of the digits reasonably well, while kernel-PCA is able to separate out only three of the digits.

7. Conclusions

In this paper, we considered the general problem of learning a linear transformation of the input data and applied it to the problems of metric and kernel learning, with a focus on establishing connections between the two problems. We showed that the LogDet divergence is a useful loss for learning a linear transformation over very high-dimensional data, as the algorithm can easily be generalized to work in kernel space, and proposed an algorithm based on Bregman projections to learn a kernel function over the data points efficiently under this loss. We also showed that our learned metric can be restricted to a small dimensional basis efficiently, thereby permitting scalability of our method to large data sets with high-dimensional feature spaces. Then we considered how to generalize this result to a larger class of convex loss functions for learning the metric/kernel using a linear transformation of the data. We proved that many loss functions can lead to efficient kernel *function* learning, though the resulting optimizations may be more expensive to solve than the simpler LogDet formulation. A key consequence of our analysis is that a number of existing approaches for Mahalanobis metric learning may be applied in kernel space using our kernel learning formulation. Finally, we presented several experiments on benchmark data, high-dimensional vision, and text classification problems as well as a semi-supervised kernel dimensionality reduction problem, demonstrating our method compared to several existing state-of-the-art techniques.

There are several potential directions for future work. To facilitate even larger data sets than the ones considered in this paper, online learning methods are one promising research direction; in Jain et al. (2008), an online learning algorithm was proposed based on LogDet regularization, and this remains a part of our ongoing efforts. Recently, there has been some interest in learning multiple local metrics over the data; Weinberger and Saul (2008) considered this problem. We plan to explore this setting with the LogDet divergence, with a focus on scalability to very large data sets.

Acknowledgments

This research was supported by NSF grant CCF-0728879. We would like to acknowledge Suvrit Sra for various helpful discussions and anonymous reviewers for various helpful suggestions.

References

A. Argyriou, C. A. Micchelli, and M. Pontil. On spectral learning. *Journal of Machine Learning Research (JMLR)*, 11:935–953, 2010.

S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 227–236, 2007.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research (JMLR)*, 6:937–965, 2005.

S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2004.

Y. Bengio, O. Delalleau, N. Le Roux, J. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

A. C. Berg and J. Malik. Geometric blur for template matching. In *Proccedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 607–614, 2001.

J. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. Arxiv:0810.3286, 2008.

Caltech-101 Data Set. *http://www.vision.caltech.edu/Image_Datasets/Caltech101/*, 2004.

R. Chatpatanasiri, T. Korsrilabutr, P. Tangchanachaianan, and B. Kijsirikul. A new kernelization framework for Mahalanobis distance learning algorithms. *Neurocomputing*, 73(10–12):1570–1579, 2010.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proccedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Classic3 Data Set. *ftp.cs.cornell.edu/pub/smart*, 2008.

CMU 20-Newsgroups Data Set. *http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html*, 2008.

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2001.

J. V. Davis and I. S. Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 195–203, 2008.

J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proccedings of the International Conference on Machine Learning (ICML)*, pages 209–216, 2007.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

R. Fletcher. A new variational result for quasi-Newton formulae. *SIAM Journal on Optimization*, 1 (1), 1991.

A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2005.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2004.

K. Grauman and T. Darrell. The Pyramid Match Kernel: Efficient learning with sets of features. *Journal of Machine Learning Research (JMLR)*, 8:725–760, April 2007.

M. Groschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.

J. Ha, C. J. Rossbach, J. V. Davis, I. Roy, H. E. Ramadan, D. E. Porter, D. L. Chen, and E. Witchel. Improved error reporting for software that uses black-box components. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 101–111, 2007.

T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18:607–616, 1996.

P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, pages 761–768, 2008.

P. Jain, B. Kulis, and I. S. Dhillon. Inductive regularized learning of kernel functions. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2010.

W. James and C. Stein. Estimation with quadratic loss. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 361–379. Univ. of California Press, 1961.

B. Kulis, M. Sustik, and I. S. Dhillon. Learning low-rank kernel matrices. In *Proccedings of the International Conference on Machine Learning (ICML)*, pages 505–512, 2006.

B. Kulis, M. Sustik, and I. Dhillon. Low-rank kernel learning with Bregman matrix divergences. *Journal of Machine Learning Research*, 2008.

B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(12):2143–2157, 2009a.

B. Kulis, S. Sra, and I. S. Dhillon. Convex perturbations for scalable semidefinite programming. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009b.

J. T. Kwok and I. W. Tsang. Learning with idealized kernels. In *Proccedings of the International Conference on Machine Learning (ICML)*, 2003.

G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research (JMLR)*, 5:27–72, 2004.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proccedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.

G. Lebanon. Metric learning for text documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(4):497–508, 2006. ISSN 0162-8828.

M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research (JMLR)*, 6:1043–1071, 2005.

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.

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2003.

M. Seeger. Cross-validation optimization for large scale hierarchical classification kernel methods. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1233–1240, 2006.

S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *Proccedings of the International Conference on Machine Learning (ICML)*, 2004.

V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proccedings of the International Conference on Machine Learning (ICML)*, pages 824–831, 2005.

K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research (JMLR)*, 6:995–1018, 2005.

M. K. Warmuth and D. Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9:2287–2320, 2008.

K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proccedings of the International Conference on Machine Learning (ICML)*, 2008.

K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, 2005.

E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, pages 505–512, 2002.

H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proccedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2136, 2006.

X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Proccedings of Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 1641–1648, 2005.

Journal of Machine Learning Research 13 (2012) 549-553 Submitted 8/11; Published 3/12

MULTIBOOST: A Multi-purpose Boosting Package

Djalel Benbouzid DJALEL.BENBOUZID@GMAIL.COM
Róbert Busa-Fekete[*] BUSAROBI@GMAIL.COM
Linear Accelerator Laboratory
University of Paris-Sud, CNRS
Orsay 91898, France

Norman Casagrande NORMAN@WAVII.COM
Wavii, Inc.
13-19 Bevenden Street
London, N1 6AA, United Kingdom

François-David Collin FRADAV@GMAIL.COM
Balázs Kégl[†] BALAZS.KEGL@GMAIL.COM
Linear Accelerator Laboratory
University of Paris-Sud, CNRS
Orsay 91898, France

Editor: Sören Sonnenburg

Abstract

The MULTIBOOST package provides a fast C++ implementation of multi-class/multi-label/multi-task boosting algorithms. It is based on ADABOOST.MH but it also implements popular cascade classifiers and FILTERBOOST. The package contains common multi-class base learners (stumps, trees, products, Haar filters). Further base learners and strong learners following the boosting paradigm can be easily implemented in a flexible framework.

Keywords: boosting, ADABOOST.MH, FILTERBOOST, cascade classifier

1. Introduction

ADABOOST (Freund and Schapire, 1997) is one of the best off-the-shelf learning methods developed in the last fifteen years. It constructs a classifier in an incremental fashion by adding simple classifiers to a pool, and uses their weighted "vote" to determine the final classification. ADABOOST was later extended to multi-class classification problems (Schapire and Singer, 1999). Although various other attempts have been made handle the multi-class setting, ADABOOST.MH has become the gold standard of multi-class boosting due to its simplicity and versatility.

Despite the simplicity and the practical success of the ADABOOST, there are relatively few off-the-shelf implementations available in the free software market. Whereas binary ADABOOST with decision stumps is easy to code, multi-class ADABOOST.MH and complex base learners are not straightforward to implement efficiently. The MULTIBOOST software package is intended to fill this gap. Its main boosting engine is based on the ADABOOST.MH algorithm of Schapire and

[*] R. Busa-Fekete is on leave from the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged.

[†] Also in the Computer Science Laboratory.

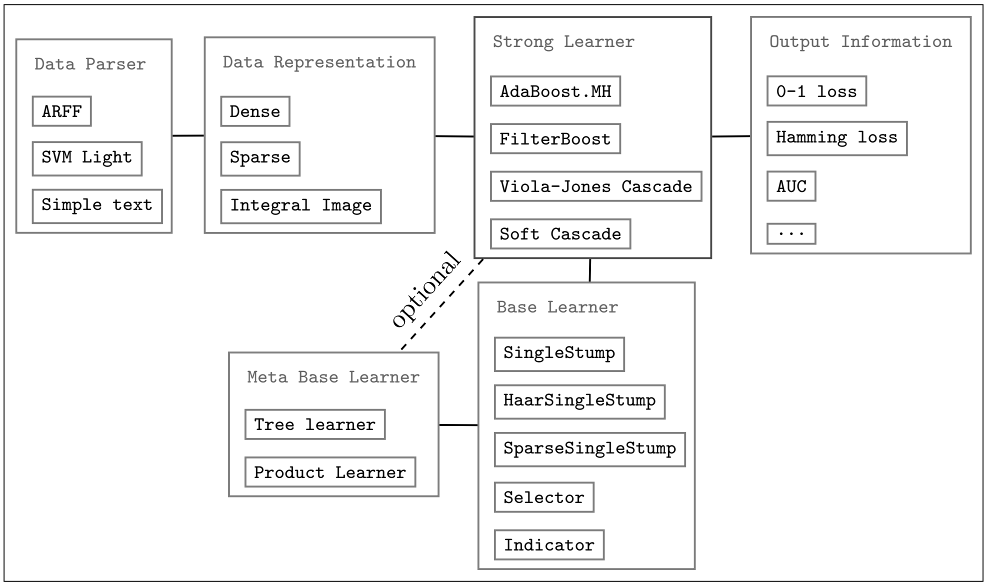


Figure 1: The architecture of MULTIBOOST

Singer (1999), but popular cascade classifiers (VJCASCADE (Viola and Jones, 2004), SOFTCASCADE (Bourdev and Brandt, 2005)) and FILTERBOOST (Bradley and Schapire, 2008) have also been implemented. The package includes common multi-class base learners (real and nominal valued decision stumps, trees, products (Kégl and Busa-Fekete, 2009), and Haar filters), but the flexible architecture makes it simple to add new base learners without interfering with the main boosting engine. MULTIBOOST was designed in the object-oriented paradigm and coded in C++, so it is fast and it provides a flexible base for implementing further modules.

The rest of this paper is organized as follows. Section 2 describes the general architecture and the modules of MULTIBOOST. Section 3 deals with practical issues (website, documentation, licence), and Section 4 describes some of our results obtained on benchmark data sets and in data mining challenges.

2. The Architecture

MULTIBOOST was implemented within the object-oriented paradigm using some design patterns. It consists of several modules which can be changed or extended more or less independently (Figure 1). For instance, an advanced user can implement a data-type/base-learner pair without any need to modify the other modules.

2.1 Strong Learners

The strong learner[1] calls the base learners iteratively, stores the learned base classifiers and their coefficients, and manages the weights of the training instances. The resulting classifier is serialized

[1] The name originally comes from the boosting (PAC learning) literature. Here, we use it in a broader sense to mean the “outer” loop of the boosting iteration.

in a human-readable XML format that allows one to resume a run after it was stopped or crashed. MULTIBOOST implements the following strong learners:

- ADABOOST.MH (Schapire and Singer, 1999): a multi-class/multi-label/multi-task version of ADABOOST that learns a "flat" linear combination of vector-valued base classifiers.
- FILTERBOOST (Bradley and Schapire, 2008): an online "filtering" booster.
- VJCASCADE (Viola and Jones, 2004): an algorithm that learns a cascade classifier tree by running ADABOOST at each node.
- SOFTCASCADE (Bourdev and Brandt, 2005): another cascade learner that starts with a set of base classifiers, reorders them, and augments them with rejection thresholds.

2.2 Base Learners

MULTIBOOST implements the following base learners.

- The STUMP learner is a one-decision two-leaf tree learned on real-valued features. It is indexed by the feature it cuts and the threshold where it cuts the feature.
- SELECTOR is a one-decision two-leaf tree learned on nominal features. It is indexed by the feature it cuts and the value of the feature it selects.
- INDICATOR is similar to SELECTOR but it can select several values for a given feature (that is, it can *indicate* a subset of the values).
- HAARSTUMP is a STUMP learned over a feature space generated using rectangular filters on images.
- TREE is a *meta* base learner that takes any base learner as input and learns a vector-valued multi-class decision tree using the input base learner as the basic cut.
- PRODUCT is another meta base learner that takes any base learner as input and learns a vector-valued multi-class decision product (Kégl and Busa-Fekete, 2009) using the input base learner as terms of the product.

2.3 The Data Representation

The multi-class data structure is a set of observation-label pairs, where each observation is a vector of feature values, and each label is a vector of binary class indicators. In binary classification, we also allow one single label that indicates the class dichotomy. In single-label multi-class classification, only one of the K labels is 1 and the others are -1, but the framework also allows multi-label classification with several positive classes per instance. In addition, multi-task classification can be encoded by letting each label column represent a different task. We implement a sparse data representation for both the observation matrix and the label matrix. In general, base learners were implemented to work with their own data representation. For example, SPARSESTUMP works on sparse observation matrices and HAARSTUMP works on an integral image data representation.

2.4 The Data Parser and the Output Information

The training and test sets can be input in the attribute-relation file format (ARFF),[2] in the SVM-LIGHT format,[3] or using a comma separated text file. We augmented the first two formats with initial label weighting, which is an important feature in the boosting framework (especially in the multi-class/multi-label setup).

In each iteration, MULTIBOOST can output several metrics (specified by the user), such as the 0-1 error, the Hamming loss, or the area under the ROC curve. New metrics can also be implemented without modifying other parts of the code.

2.5 LAZYBOOST and BANDITBOOST

When the number of features is large, *featurewise* learners (STUMP, SELECTOR, and INDICATOR) can be accelerated by searching only a subset of the features in each iteration. MULTIBOOST implements two options, namely, LAZYBOOST (Escudero et al., 2000), where features are sampled randomly, and BANDITBOOST (Busa-Fekete and Kégl, 2010), where the sampling is biased towards "good" features learned using a multi-armed bandit algorithm.

3. Documentation and License

The code of MULTIBOOST has been fully documented in Doxygen.[4] It is available under the GPL licence at multiboost.org. The website also provides documentation that contains detailed instructions and examples for using the package along with tutorials explaining how to implement new features. The documentation also contains the pseudo-code of the multi-class base learners implemented in MULTIBOOST.

4. Challenges and Benchmarks

We make available reproducible test results (validated test errors, learning curves) of MULTIBOOST on the web site as we produce them. Among other results, the boosted decision product is one of the best reported no-domain-knowledge algorithms on MNIST.[5] An early version of the program (Bergstra et al., 2006) was the best genre classifier out of 15 submissions and the second-best out of 10 submissions at recognizing artists in the MIREX 2005 international contest on music information extraction. More recently, we participated in the Yahoo! Learning to Rank Challenge[6] using a pointwise ranking approach based on hundreds of MULTIBOOST classifiers. We finished 6th in Track 1 and 11th in Track 2 out of several hundred participating teams (Busa-Fekete et al., 2011).

Acknowledgments

This work was supported by the ANR-2010-COSI-002 grant of the French National Research Agency.

[2] See www.cs.waikato.ac.nz/~ml/weka/arff.html.
[3] See svmlight.joachims.org.
[4] See www.doxygen.org.
[5] See yann.lecun.com/exdb/mnist.
[6] See learningtorankchallenge.yahoo.com.

References

J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and AdaBoost for music classification. *Machine Learning Journal*, 65(2/3):473–484, 2006.

L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 236–243. IEEE Computer Society, 2005.

J.K. Bradley and R.E. Schapire. FilterBoost: Regression and classification on large datasets. In *Advances in Neural Information Processing Systems*, volume 20. The MIT Press, 2008.

R. Busa-Fekete and B. Kégl. Fast boosting using adversarial bandits. In *International Conference on Machine Learning*, volume 27, pages 143–150, 2010.

R. Busa-Fekete, B. Kégl, Éltető T., and Gy. Szarvas. Ranking by calibrated AdaBoost. In *(JMLR W&CP)*, volume 14, pages 37–48, 2011.

G. Escudero, L. Màrquez, and G. Rigau. Boosting applied to word sense disambiguation. In *Proceedings of the 11th European Conference on Machine Learning*, pages 129–141, 2000.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

B. Kégl and R. Busa-Fekete. Boosting products of base classifiers. In *International Conference on Machine Learning*, volume 26, pages 497–504, Montreal, Canada, 2009.

R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

Journal of Machine Learning Research 13 (2012) 555-559 Submitted 6/11; Revised 2/12; Published 3/12

ML-Flex: A Flexible Toolbox for Performing Classification Analyses In Parallel

Stephen R. Piccolo STEPHEN.PICCOLO@HSC.UTAH.EDU
Department of Pharmacology and Toxicology, School of Pharmacy
University of Utah
Salt Lake City, UT 84112, USA

Lewis J. Frey LEWIS.FREY@HSC.UTAH.EDU
Huntsman Cancer Institute
Department of Biomedical Informatics, School of Medicine
University of Utah
Salt Lake City, UT 84112, USA

Editor: Geoff Holmes

Abstract

Motivated by a need to classify high-dimensional, heterogeneous data from the bioinformatics domain, we developed ML-Flex, a machine-learning toolbox that enables users to perform two-class and multi-class classification analyses in a systematic yet flexible manner. ML-Flex was written in Java but is capable of interfacing with third-party packages written in other programming languages. It can handle multiple input-data formats and supports a variety of customizations. ML-Flex provides implementations of various validation strategies, which can be executed in parallel across multiple computing cores, processors, and nodes. Additionally, ML-Flex supports aggregating evidence across multiple algorithms and data sets via ensemble learning. This open-source software package is freely available from `http://mlflex.sourceforge.net`.

Keywords: toolbox, classification, parallel, ensemble, reproducible research

1. Introduction

The machine-learning community has developed a wide array of classification algorithms, but they are implemented in diverse programming languages, have heterogeneous interfaces, and require disparate file formats. Also, because input data come in assorted formats, custom transformations often must precede classification. To address these challenges, we developed ML-Flex, a general-purpose toolbox for performing two-class and multi-class classification analyses. Via command-line interfaces, ML-Flex can invoke algorithms implemented in any programming language. Currently, ML-Flex can interface with several popular third-party packages, including *Weka* (Hall et al., 2009), *Orange* (Demsar et al., 2004), *C5.0 Decision Trees* (RuleQuest Research, 2011), and *R* (R Development Core Team, 2011). In many cases, new packages can be integrated with ML-Flex through only minor modifications to configuration files. However, via a simple extension mechanism, ML-Flex also supports a great amount of flexibility for custom integrations. ML-Flex can parse input data in delimited and ARFF formats, and it can easily be extended to parse data from custom sources.

ML-Flex can perform systematic evaluations across multiple algorithms and data sets. Furthermore, it can aggregate evidence across algorithms and data sets via ensemble learning. The following ensemble learners currently are supported: majority vote (Boland, 1989), weighted majority vote (Littlestone and Warmuth, 1994), mean probability rule, weighted mean probability rule, maximum probability rule, select-best rule, and stacked generalization (Wolpert, 1992). (When ensemble learners are applied, predictions from individual classifiers are reused from prior execution steps, thus decreasing computational overhead.)

ML-Flex provides implementations of various validation strategies, including simple train-test validation, K-fold cross validation, repeated random sampling validation, and leave-one-out cross validation. For each validation strategy, ML-Flex can also apply feature selection/ranking algorithms and perform nested cross-validation within respective training sets. To enable shorter execution times for computationally intensive validation strategies, ML-Flex can be executed in parallel across multiple computing cores/processors and multiple nodes on a network. Individual computing nodes may have heterogeneous hardware configurations so long as each node can access a shared file system. In a recent analysis of a large biomedical data set, ML-Flex was executed simultaneously across hundreds of cluster-computing cores (Piccolo, 2011).

Upon completing classification tasks, ML-Flex produces parsable text files that report performance metrics, confusion matrices, outputs from individual algorithms, and a record of all configuration settings used. A formatted HTML report with the same information is also provided. These features enable reproducibility and transparency about how a given set of results was obtained.

Available from `http://mlflex.sourceforge.net`, ML-Flex is licensed under the GNU General Public License Version 3.0. The distribution contains extensive documentation, including tutorials and sample experiments.

2. Architecture

Execution of ML-Flex revolves around the concept of an "experiment." For a given experiment, the user specifies one or more sets of independent (predictor) variables and a dependent variable (class) as well as any algorithm(s) that should be applied to the data. Various other settings (for example, number of cross-validation folds, number of iterations, random seed) can be altered optionally.

To configure an experiment, users can specify three types of settings: 1) learner templates, 2) algorithm parameters, and 3) experiment-specific preferences. For example, if a user wanted to apply the *Weka* implementation of the *One Rule* classification algorithm, with a minimum bucket size of 6, to the classic Iris data set, she would first create a learner template such as the following (simplified for brevity):

```
wekac;mlflex.learners.WekaLearner;java -classpath lib/weka.jar {ALGORITHM}
-t {INPUT_TRAINING_FILE} -T {INPUT_TEST_FILE} -p 0 -distribution
```

The above template contains three items, separated by semicolons: 1) a unique key, 2) the name of a Java class that supports interfacing with Weka, and 3) a templated shell command for invoking Weka on that system. (When ML-Flex executes a command, placeholders—for example, "{ALGORITHM}"—are replaced with relevant values.) Having specified this template, the user would specify the following in an algorithm-parameters file:

```
one_r;wekac;weka.classifiers.rules.OneR -B 6
```

This entry indicates 1) a unique key, 2) a reference to the learner template, and 3) the parameters that should be passed to Weka. Finally, the user would include the following in an experiment file:

```
CLASSIFICATION_ALGORITHMS=one_r
DATA_PROCESSORS=mlflex.dataprocessors.ArffDataProcessor("iris.arff")
```

The first line references the algorithm-parameters entry, and the second line indicates the name of a Java class that can parse the input data. (Example files and detailed explanations of all configuration settings are provided.)

At each stage of an experiment, ML-Flex can execute in parallel, using a simple, coarse-grained architecture. Independent computing tasks—for example, parsing a given input file, classifying a given combination of data set and algorithm and cross-validation fold, or outputting results—are packaged by each computing node into uniquely defined Java objects. Then thread(s) on each computing node compete to execute each task via a locking mechanism. Initially, each thread checks for a status file that would indicate whether a given task has been executed and the corresponding result has been stored. If the task has not yet been processed, the thread checks the file system for a correspondingly named "lock file" that indicates whether the task is currently being processed by another thread. If no lock file exists, the thread attempts to create the file atomically. Having successfully created the lock file, the thread executes the task, stores the result on the file system, and deletes the lock file. If a system error or outage occurs, the lock file will persist for a user-configurable period of time, after which it will be deleted and the task reattempted.

Because this parallel-processing approach requires many input/output operations and because individual computing nodes do not communicate with each other directly, minor inefficiencies may arise. However, the simplicity of the approach offers many advantages: 1) no third-party software package is necessary for interprocess communication, 2) individual computing nodes may run different operating systems and/or have different hardware configurations, 3) the number of computing nodes that can be employed simultaneously is scalable, 4) if an individual computing node goes offline, remaining nodes are unaffected, 5) additional computing nodes may be assigned after a job has already begun processing, and 6) experiments are restartable. The latter three features are particularly desirable in cluster-computing environments where server reliability may be less than optimal and where computing nodes may become available incrementally.

3. Related Work

Machine-learning toolboxes like *caret*, *Weka*, *Orange*, and *KNIME* implement a broad range of classification algorithms, but many useful algorithms are not included in these packages, perhaps due to licensing restrictions, resource constraints, or programming-language preferences. Like *SHOGUN* (Sonnenburg et al., 2010), ML-Flex provides a harness that allows developers to implement algorithms natively in the language of their choice. Because no language-specific interfaces are necessary in ML-Flex, integration can often occur with no change to the ML-Flex source code. This approach also empowers algorithm developers to take the lead in integrating with ML-Flex and thus benefit from its other features, including model evaluation and parallelization.

KNIME and *RapidMiner* (Mierswa et al., 2006) support various input-file formats, transformation procedures, and data-filtering modules. In an alternative approach, ML-Flex provides an extension mechanism, which allows users to preprocess data using custom Java code. (*Weka* supports similar functionality.) This approach may be especially useful in research settings where unusual data formats are prevalent, advanced transformations are desired, or data must be accessed remotely (for example, via Web services, including those that require authentication).

Other toolboxes support the ability to distribute workloads across multiple computers. For example, a client machine executing the *Weka* Experimenter module can distribute its workload via Java Remote Method Invocation. The *caret* R package (Kuhn, 2008) uses the *NetWorkSpaces*^{TM} technology to distribute workloads. The commercial version of *KNIME* (Berthold et al., 2007) can distribute its workload to cluster servers running *Oracle*® *Grid Engine*. And *Apache Mahout*^{TM} (Ingersol, 2009) uses the map/reduce paradigm to enable execution on cluster-computing environments. ML-Flex differentiates itself from these tools by 1) supporting heterogeneous configurations among computing nodes, 2) allowing recovery from system outages due to no single point of failure (assuming redundant disk storage), and 3) supporting restartability and incremental node allocations.

Many toolboxes—including *Weka*, *Orange*, *KNIME*, *caret*, *SHOGUN*, and *Waffles* (Gashler, 2011)—support experimental reproducibility via application programming interfaces (API), command-line interfaces (CLI), and/or visual workflow pipelines. Users can write client tools that invoke APIs and that can later be re-executed. Scripts that invoke CLIs can be repeated; and visual pipelines typically encapsulate execution logic. In ML-Flex, users encode all configuration settings in text files. With this approach, users are not required to write code nor extensive scripts. However, with a modest scripting effort, it is possible to generate configuration files dynamically, an approach that may not be feasible with visual workflows. Because a copy of relevant configuration files accompany the results of each experiment, subsequent replication of results is straightforward. Additionally, in experiments that use repeated random sampling validation, ML-Flex encapsulates sampling and summarization logic, which may be burdensome to replicate with alternative approaches.

Acknowledgments

While developing this software, SRP was funded by a U.S. National Library of Medicine training fellowship (1T15-LM007124). We gratefully acknowledge an allocation of computer time from the Fulton Supercomputing Lab at Brigham Young University. We also thank the reviewers for thoughtful, detailed feedback.

References

M. Berthold, N. Cebron, F. Dill, T. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel. KNIME: The Konstanz information miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*, Springer, 2007.

P. Boland. Majority systems and the condorcet jury theorem. *The Statistician*, 38(3):181–189, 1989.

J. Demsar, B. Zupan, G. Leban, and T. Curk. Orange: From experimental machine learning to interactive data mining. In *Knowledge Discovery in Databases: PKDD 2004*, pages 537–539, Berlin, 2004.

M. Gashler. Waffles: A machine learning toolkit. *Journal of Machine Learning Research*, 12(Jul):2383–2387, 2011.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1):10, 2009.

G. Ingersol. Introducing Apache Mahout. *IBM developerWorks Technical Library*, 2009. Available electronically at http://www.ibm.com/developerworks/java/library/j-mahout/.

M. Kuhn. Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5), 2008.

N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (August 2006)*, ACM, 2006.

S. Piccolo. *Informatics Framework For Evaluating Multivariate Prognosis Models: Application to Human Glioblastoma Multiforme*. PhD dissertation, University of Utah, Salt Lake City, Utah, 2011.

R Development Core Team. *R: A Language and Environment For Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. Available electronically at http://www.R-project.org.

Rulequest Research. Data mining tools See5 and C5.0. Available electronically at http://www.rulequest.com/see5-info.html.

S. Sonnenburg, G. Ratsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. Bona, A. Binder, C. Gehl and V. Franc. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11(Jun):1799–1802, 2010.

D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

A Primal-Dual Convergence Analysis of Boosting

Matus Telgarsky MTELGARS@CS.UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
San Diego, CA 92093-0404, USA

Editor: Yoram Singer

Abstract

Boosting combines weak learners into a predictor with low empirical risk. Its dual constructs a high entropy distribution upon which weak learners and training labels are uncorrelated. This manuscript studies this primal-dual relationship under a broad family of losses, including the exponential loss of AdaBoost and the logistic loss, revealing:

- Weak learnability aids the whole loss family: for any $\varepsilon > 0$, $O(\ln(1/\varepsilon))$ iterations suffice to produce a predictor with empirical risk ε-close to the infimum;
- The circumstances granting the existence of an empirical risk minimizer may be characterized in terms of the primal and dual problems, yielding a new proof of the known rate $O(\ln(1/\varepsilon))$;
- Arbitrary instances may be decomposed into the above two, granting rate $O(1/\varepsilon)$, with a matching lower bound provided for the logistic loss.

Keywords: boosting, convex analysis, weak learnability, coordinate descent, maximum entropy

1. Introduction

Boosting is the task of converting inaccurate *weak learners* into a single accurate predictor. The existence of any such method was unknown until the breakthrough result of Schapire (1990): under a *weak learning assumption*, it is possible to combine many carefully chosen weak learners into a majority of majorities with arbitrarily low training error. Soon after, Freund (1995) noted that a single majority is enough, and that $\Theta(\ln(1/\varepsilon))$ iterations are both necessary and sufficient to attain accuracy ε. Finally, their combined effort produced AdaBoost, which exhibits this optimal convergence rate (under the weak learning assumption), and has an astonishingly simple implementation (Freund and Schapire, 1997).

It was eventually revealed that AdaBoost was minimizing a risk functional, specifically the exponential loss (Breiman, 1999). Aiming to alleviate perceived deficiencies in the algorithm, other loss functions were proposed, foremost amongst these being the logistic loss (Friedman et al., 2000). Given the wide practical success of boosting with the logistic loss, it is perhaps surprising that no convergence rate better than $O(\exp(1/\varepsilon^2))$ was known, even under the weak learning assumption (Bickel et al., 2006). The reason for this deficiency is simple: unlike SVM, least squares, and basically any other optimization problem considered in machine learning, there might not exist a choice which attains the minimal risk! This reliance is carried over from convex optimization, where the assumption of attainability is generally made, either directly, or through stronger conditions like compact level sets or strong convexity (Luo and Tseng, 1992). But this limitation seems artificial: a function like $\exp(-x)$ has no minimizer but decays rapidly.

Convergence rate analysis provides a valuable mechanism to compare and improve of minimization algorithms. But there is a deeper significance with boosting: a convergence rate of $O(\ln(1/\varepsilon))$ means that, with a combination of just $O(\ln(1/\varepsilon))$ predictors, one can construct an ε-optimal classifier, which is crucial to both the computational efficiency and statistical stability of this predictor.

The main contribution of this manuscript is to provide a tight convergence theory for a large family of losses, including the exponential and logistic losses, which has heretofore resisted analysis. In particular, it is shown that the (disjoint) scenarios of weak learnability (Section 6.1) and attainability (Section 6.2) both exhibit the rate $O(\ln(1/\varepsilon))$. These two scenarios are in a strong sense extremal, and general instances are shown to decompose into them; but their conflicting behavior yields a degraded rate $O(1/\varepsilon)$ (Section 6.3). A matching lower bound for the logistic loss demonstrates this is no artifact.

1.1 Outline

Beyond providing these rates, this manuscript will study the rich ecology within the primal-dual interplay of boosting.

Starting with necessary background, Section 2 provides the standard view of boosting as coordinate descent of an empirical risk. This primal formulation of boosting obscures a key internal mechanism: boosting iteratively constructs distributions where the previously selected weak learner fails. This view is recovered in the dual problem; specifically, Section 3 reveals that the dual feasible set is the collection of distributions where all weak learners have no correlation to the target, and the dual objective is a max entropy rule.

The dual optimum is always attainable; since a standard mechanism in convergence analysis to control the distance to the optimum, why not overcome the unattainability of the primal optimum by working in the dual? It turns out that the classical weak learning rate was a mechanism to control distances in the dual all along; by developing a suitable generalization (Section 4), it is possible to convert the improvement due to a single step of coordinate descent into a relevant distance in the dual (Section 6). Crucially, this holds for general instances, without any assumptions.

The final puzzle piece is to relate these dual distances to the optimality gap. Section 5 lays the foundation, taking a close look at the structure of the optimization problem. The classical scenarios of attainability and weak learnability are identifiable directly from the weak learning class and training sample; moreover, they can be entirely characterized by properties of the primal and dual problems.

Section 5 will also reveal another structure: there is a subset of the training set, the *hard core*, which is the maximal support of any distribution upon which every weak learner and the training labels are uncorrelated. This set is central—for instance, the dual optimum (regardless of the loss function) places positive weight on exactly the hard core. Weak learnability corresponds to the hard core being empty, and attainability corresponds to it being the whole training set. For those instances where the hard core is a nonempty proper subset of the training set, the behavior on and off the hard core mimics attainability and weak learnability, and Section 6.3 will leverage this to produce rates using facts derived for the two constituent scenarios.

Much of the technical material is relegated to the appendices. For convenience, Section A summarizes notation, and Section B contains some important supporting results. Of perhaps practical interest, Section D provides methods to select the step size, meaning the weight with which new weak learners are included in the full predictor. These methods are sufficiently powerful to grant the convergence rates in this manuscript.

1.2 Related Work

The development of general convergence rates has a number of important milestones in the past decade. Collins et al. (2002) proved convergence for a large family of losses, albeit without any rates. Interestingly, the step size only partially modified the choice from AdaBoost to accommodate arbitrary losses, whereas the choice here follows standard optimization principles based purely on the particular loss. Next, Bickel et al. (2006) showed a general rate of $O(\exp(1/\varepsilon^2))$ for a slightly smaller family of functions: every loss has positive lower and upper bounds on its second derivative within any compact interval. This is a larger family than what is considered in the present manuscript, but Section 6.2 will discuss the role of the extra assumptions when producing fast rates.

Many extremely important cases have also been handled. The first is the original rate of $O(\ln(1/\varepsilon))$ for the exponential loss under the weak learning assumption (Freund and Schapire, 1997). Next, under the assumption that the empirical risk minimizer is attainable, Rätsch et al. (2001) demonstrated the rate $O(\ln(1/\varepsilon))$. The loss functions in that work must satisfy lower and upper bounds on the Hessian within the initial level set; equivalently, the existence of lower and upper bounding quadratic functions within this level set. This assumption may be slightly relaxed to needing just lower and upper second derivative bounds on the univariate loss function within an initial bounding interval (cf. discussion within Section 5.2), which is the same set of assumptions used by Bickel et al. (2006), and as discussed in Section 6.2, is all that is really needed by the analysis in the present manuscript under attainability.

Parallel to the present work, Mukherjee et al. (2011) established general convergence under the exponential loss, with a rate of $\Theta(1/\varepsilon)$. That work also presented bounds comparing the AdaBoost suboptimality to any l^1 bounded solution, which can be used to succinctly prove consistency properties of AdaBoost (Schapire and Freund, in preparation). In this case, the rate degrades to $O(\varepsilon^{-5})$, which although presented without lower bound, is not terribly surprising since the optimization problem minimized by boosting has no norm penalization. Finally, mirroring the development here, Mukherjee et al. (2011) used the same boosting instance (due to Schapire 2010) to produce lower bounds, and also decomposed the boosting problem into finite and infinite margin pieces (cf. Section 5.3).

It is interesting to mention that, for many variants of boosting, general convergence rates were known. Specifically, once it was revealed that boosting is trying to be not only correct but also have large margins (Schapire et al., 1997), much work was invested into methods which explicitly maximized the margin (Rätsch and Warmuth, 2002), or penalized variants focused on the inseparable case (Warmuth et al., 2007; Shalev-Shwartz and Singer, 2008). These methods generally impose some form of regularization (Shalev-Shwartz and Singer, 2008), which grants attainability of the risk minimizer, and allows standard techniques to grant general convergence rates. Interestingly, the guarantees in those works cited in this paragraph are $O(1/\varepsilon^2)$.

Hints of the dual problem may be found in many works, most notably those of Kivinen and Warmuth (1999) and Collins et al. (2002), which demonstrated that boosting is seeking a difficult distribution over training examples via iterated Bregman projections.

The notion of hard core sets is due to Impagliazzo (1995). A crucial difference is that in the present work, the hard core is unique, maximal, and every weak learner does no better than random guessing upon a family of distributions supported on this set; in this cited work, the hard core is relaxed to allow some small but constant fraction correlation to the target. This relaxation is central to the work, which provides a correspondence between the complexity (circuit size) of the weak learners, the difficulty of the target function, the size of the hard core, and the correlation permitted in the hard core.

2. Setup

A view of boosting, which pervades this manuscript, is that the action of the weak learning class upon the sample can be encoded as a matrix (Rätsch et al., 2001; Shalev-Shwartz and Singer, 2008). Let a sample $\mathcal{S} := \{(x_i, y_i)\}_1^m \subseteq (\mathcal{X} \times \mathcal{Y})^m$ and a weak learning class $\mathcal{H}$ be given. For every $h \in \mathcal{H}$, let $\mathcal{S}|_h$ denote the negated projection onto $\mathcal{S}$ induced by h; that is, $\mathcal{S}|_h$ is a vector of length m, with coordinates $(\mathcal{S}|_h)_i = -y_i h(x_i)$. If the set of all such columns $\{\mathcal{S}|_h : h \in \mathcal{H}\}$ is finite, collect them into the matrix $A \in \mathbb{R}^{m \times n}$. Let a_i denote the i^{th} row of A, corresponding to the example (x_i, y_i), and let $\{h_j\}_1^n$ index the set of weak learners corresponding to columns of A. It is assumed, for convenience, that entries of A are within $[-1, +1]$; relaxing this assumption merely scales the presented rates by a constant.

The setting considered here is that this finite matrix can be constructed. Note that this can encode infinite classes, so long as they map to only $k < \infty$ values (in which case A has at most k^m columns). As another example, if the weak learners are binary, and $\mathcal{H}$ has VC dimension d, then Sauer's lemma grants that A has at most $(m+1)^d$ columns. This matrix view of boosting is thus similar to the interpretation of boosting performing descent in functional space (Mason et al., 2000; Friedman et al., 2000), but the class complexity and finite sample have been used to reduce the function class to a finite object.

To make the connection to boosting, the missing ingredient is the loss function.

Definition 1 *$\mathbb{G}_0$ is the set of loss functions $g : \mathbb{R} \to \mathbb{R}$ satisfying: g is twice continuously differentiable, $g'' > 0$, and $\lim_{x \to -\infty} g(x) = 0$.*

For convenience, whenever $g \in \mathbb{G}_0$ and sample size m are provided, let $f : \mathbb{R}^m \to \mathbb{R}$ denote the empirical risk function $f(x) := \sum_{i=1}^m g((x)_i)$. For more properties of g and f, please see Section C.

The convergence rates of Section 6 will require a few more conditions, but $\mathbb{G}_0$ suffices for all earlier results.

Example 1 *The exponential loss* $\exp(\cdot)$ *(AdaBoost) and logistic loss* $\ln(1 + \exp(\cdot))$ *are both within $\mathbb{G}_0$ (and the eventual $\mathbb{G}$). These two losses appear in Figure 1, where the log-scale plot aims to convey their similarity for negative values.*

This definition provides a notational break from most boosting literature, which instead requires $\lim_{x \to \infty} g(x) = 0$ (i.e., the exponential loss becomes $\exp(-x)$); note that the usage here simply pushes the negation into the definition of the matrix A. The significance of this modification is that the gradient of the empirical risk, which corresponds to distributions produced by boosting, is a nonnegative measure. (Otherwise, it would be necessary to negate this (nonpositive) distribution everywhere to match the boosting literature.) Note that there is no consensus on this choice, and the form followed here can be found elsewhere (Boucheron et al., 2005).

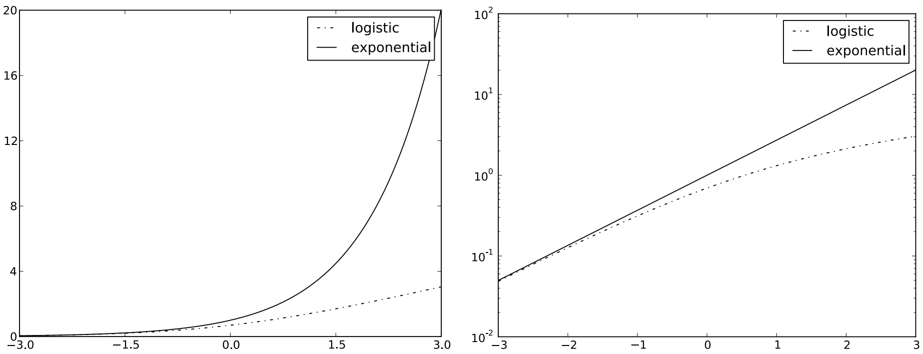


Figure 1: Exponential and logistic losses, plotted with linear and log-scale range.

Boosting determines some weighting $\lambda \in \mathbb{R}^n$ of the columns of A, which correspond to weak learners in $\mathcal{H}$. The (unnormalized) margin of example i is thus $\langle -a_i, \lambda\rangle = -\mathbf{e}_i^\top A\lambda$, where $\mathbf{e}_i$ is an indicator vector. (This negation is one notational inconvenience of making losses increasing.) Since the prediction on x_i is $\mathbb{1}[\sum_j \lambda_j h_j(x_i) \geq 0] = \mathbb{1}[y_i \langle a_i, \lambda\rangle \leq 0]$, it follows that $A\lambda < \mathbf{0}_m$ (where $\mathbf{0}_m$ is the zero vector) implies a training error of zero. As such, boosting solves the minimization problem

$$\inf_{\lambda\in\mathbb{R}^n} \sum_{i=1}^m g(\langle a_i, \lambda\rangle) = \inf_{\lambda\in\mathbb{R}^n} \sum_{i=1}^m g(\mathbf{e}_i^\top A\lambda) = \inf_{\lambda\in\mathbb{R}^n} f(A\lambda) = \inf_{\lambda\in\mathbb{R}^n} (f\circ A)(\lambda) =: \bar{f}_A; \tag{1}$$

recall $f : \mathbb{R}^m \to \mathbb{R}$ is the convenience function $f(x) = \sum_i g((x)_i)$, and in the present problem denotes the (unnormalized) empirical risk. $\bar{f}_A$ will denote the optimal objective value.

The infimum in Equation 1 may well not be attainable. Suppose there exists λ' such that $A\lambda' < \mathbf{0}_m$ (Theorem 11 will show that this is equivalent to the weak learning assumption). Then

$$0 \leq \inf_{\lambda\in\mathbb{R}^n} f(A\lambda) \leq \inf_{c>0} f(A(c\lambda')) = 0.$$

On the other hand, for any $\lambda \in \mathbb{R}^n$, $f(A\lambda) > 0$. Thus the infimum is never attainable when weak learnability holds.

The template boosting algorithm appears in Figure 2, formulated in terms of $f \circ A$ to make the connection to coordinate descent as clear as possible. To interpret the gradient terms, note that

$$(\nabla(f\circ A)(\lambda))_j = (A^\top \nabla f(A\lambda))_j = -\sum_{i=1}^m g'(\langle a_i, \lambda\rangle) h_j(x_i) y_i,$$

which is the expected negative correlation of h_j with the target labels according to an unnormalized distribution with weights $g'(\langle a_i, \lambda\rangle)$. The stopping condition $\nabla(f\circ A)(\lambda) = \mathbf{0}_m$ means: either the distribution is degenerate (it is exactly zero), or every weak learner is uncorrelated with the target.

As such, BOOST in Figure 2 represents an equivalent formulation of boosting, with one minor modification: the column (weak learner) selection has an absolute value. But note that this is the same as closing $\mathcal{H}$ under complementation (i.e., for any $h \in \mathcal{H}$, there exists $h^{(-)}$ with $h(x) = -h^{(-)}(x)$), which is assumed in many theoretical treatments of boosting.

In the case of the exponential loss and binary weak learners, the line search (when attainable) has a convenient closed form; but for other losses, and even with the exponential loss but with

```
Routine BOOST.
Input Convex function $f \circ A$.
Output Approximate primal optimum $\lambda$.

  1. Initialize $\lambda_0 := \mathbf{0}_n$.
  2. For $t = 1, 2, \ldots$, while $\nabla(f \circ A)(\lambda_{t-1}) \neq \mathbf{0}_n$:
      (a) Choose column (weak learner)
              $j_t := \underset{j}{\operatorname{argmax}} |\nabla(f \circ A)(\lambda_{t-1})^\top \mathbf{e}_j|.$
      (b) Correspondingly, set descent direction $v_t \in \{\pm \mathbf{e}_{j_t}\}$; note
              $v_t^\top \nabla(f \circ A)(\lambda_{t-1}) = -\|\nabla(f \circ A)(\lambda_{t-1})\|_\infty.$
      (c) Find $\alpha_t$ via approximate solution to the line search
              $\inf_{\alpha > 0} (f \circ A)(\lambda_{t-1} + \alpha v_t).$
      (d) Update $\lambda_t := \lambda_{t-1} + \alpha_t v_t$.
  3. Return $\lambda_{t-1}$.
```

Figure 2: l^1 steepest descent (Boyd and Vandenberghe, 2004, Algorithm 9.4) of $f \circ A$.

confidence-rated predictors, there may not be a closed form. As such, BOOST only requires an approximate line search method. Section D details two mechanisms for this: an iterative method, which requires no knowledge of the loss function, and a closed form choice, which unfortunately requires some properties of the loss, which may be difficult to bound tightly. The iterative method provides a slightly worse guarantee, but is potentially more effective in practice; thus it will be used to produce all convergence rates in Section 6.

For simplicity, it is supposed that the best weak learner j_t (or the approximation thereof encoded in A) can always be selected. Relaxing this condition is not without subtleties, but as discussed in Section E, there are ways to allow approximate selection without degrading the presented convergence rates.

As a final remark, consider the rows $\{-a_i\}_1^m$ of $-A$ as a collection of m points in $\mathbb{R}^n$. Due to the form of g, BOOST is therefore searching for a halfspace, parameterized by a vector λ, which contains all of these points. Sometimes such a halfspace may not exist, and g applies a smoothly increasing penalty to points that are farther and farther outside it.

3. Dual Problem

Applying coordinate descent to Equation 1 represents a valid interpretation of boosting, in the sense that the resulting algorithm BOOST is equivalent to the original. However this representation loses

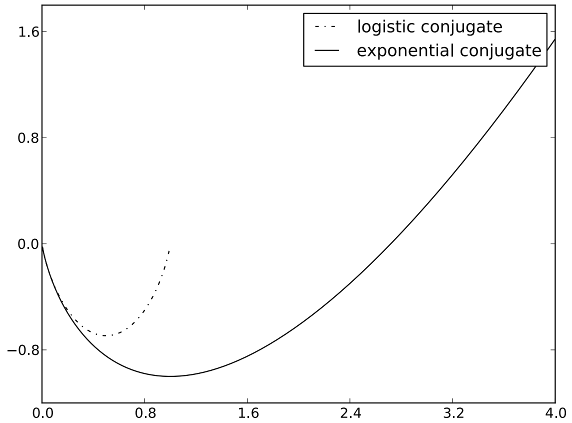


Figure 3: Fenchel conjugates of exponential and logistic losses.

the intuitive operation of boosting as generating distributions where the current predictor is highly erroneous, and requesting weak learners accurate on these tricky distributions. The dual problem will capture this.

In addition to illuminating the structure of boosting, the dual problem also possesses a major concrete contribution to the optimization behavior, and specifically the convergence rates: the dual optimum is always attainable.

The dual problem will make use of Fenchel conjugates (Hiriart-Urruty and Lemaréchal, 2001; Borwein and Lewis, 2000); for any function h, the conjugate is

$$h^*(\phi) = \sup_{x \in \text{dom}(h)} \langle x, \phi \rangle - h(x).$$

Example 2 *The exponential loss* $\exp(\cdot)$ *has Fenchel conjugate*

$$(\exp(\cdot))^*(\phi) = \begin{cases} \phi \ln(\phi) - \phi & \text{when } \phi > 0, \\ 0 & \text{when } \phi = 0, \\ \infty & \text{otherwise.} \end{cases}$$

The logistic loss $\ln(1 + \exp(\cdot))$ *has Fenchel conjugate*

$$(\ln(1 + \exp(\cdot)))^*(\phi) = \begin{cases} (1-\phi)\ln(1-\phi) + \phi \ln(\phi) & \text{when } \phi \in (0,1), \\ 0 & \text{when } \phi \in \{0,1\}, \\ \infty & \text{otherwise.} \end{cases}$$

These conjugates are known respectively as the Boltzmann-Shannon and Fermi-Dirac entropies (Borwein and Lewis, 2000, Commentary, Section 3.3). Please see Figure 3 for a depiction.

It further turns out that general members of $\mathbb{G}_0$ have a shape reminiscent of these two standard notions of entropy.

Lemma 2 *Let $g \in \mathbb{G}_0$ be given. Then g^* is continuously differentiable on $\operatorname{int}(\operatorname{dom}(g^*))$, strictly convex, and either $\operatorname{dom}(g^*) = [0,\infty)$ or $\operatorname{dom}(g^*) = [0,b]$ where $b > 0$. Furthermore, g^* has the following form:*

$$g^*(\phi) \in \begin{cases} \infty & \text{when } \phi < 0, \\ 0 & \text{when } \phi = 0, \\ (-g(0), 0) & \text{when } \phi \in (0, g'(0)), \\ -g(0) & \text{when } \phi = g'(0), \\ (-g(0), \infty] & \text{when } \phi > g'(0). \end{cases}$$

(The proof is in Section C.) There is one more object to present, the dual feasible set Φ_A.

Definition 3 *For any $A \in \mathbb{R}^{m \times n}$, define the dual feasible set*

$$\Phi_A := \operatorname{Ker}(A^\top) \cap \mathbb{R}^m_+$$

Consider any $\psi \in \Phi_A$. Since $\psi \in \operatorname{Ker}(A^\top)$, this is a weighting of examples which decorrelates all weak learners from the target: in particular, for any primal weighting $\lambda \in \mathbb{R}^n$ over weak learners, $\psi^\top A \lambda = 0$. And since $\psi \in \mathbb{R}^m_+$, all coordinates are nonnegative, so in the case that $\psi \neq \{\mathbf{0}_m\}$, this vector may be renormalized into a distribution over examples. The case $\Phi_A = \{\mathbf{0}_m\}$ is an extremely special degeneracy: it will be shown to encode the scenario of weak learnability.

Theorem 4 *For any $A \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{G}_0$ with $f(x) = \sum_i g((x)_i)$,*

$$\inf\{f(A\lambda) : \lambda \in \mathbb{R}^n\} = \sup\{-f^*(\psi) : \psi \in \Phi_A\}, \tag{2}$$

where $f^*(\phi) = \sum_{i=1}^m g^*((\phi)_i)$. The right hand side is the dual problem, and moreover the dual optimum, denoted ψ_A^f, is unique and attainable.

(The proof uses routine techniques from convex analysis, and is deferred to Section G.2.)

The definition of Φ_A does not depend on any specific $g \in \mathbb{G}_0$; this choice was made to provide general intuition on the structure of the problem for the entire family of losses. Note however that this will cause some problems later. For instance, with the logistic loss, the vector with every value two, that is, $2 \cdot \mathbf{1}_m$, has objective value $-f^*(2 \cdot \mathbf{1}_m) = -\infty$. In a sense, there are points in Φ_A which are not really candidates for certain losses, and this fact will need adjustment in some convergence rate proofs.

Remark 5 *Finishing the connection to maximum entropy, for any $g \in \mathbb{G}_0$, by Lemma 2, the optimum of the unconstrained problem is $g'(0)\mathbf{1}_m$, a rescaling of the uniform distribution. But note that $\nabla f(A\lambda_0) = \nabla f(\mathbf{0}_m) = g'(0)\mathbf{1}_m$: that is, the initial dual iterate is the unconstrained optimum! Let $\phi_t := \nabla f(A\lambda_t)$ denote the t^{th} dual iterate; since $\nabla f^*(\nabla f(x)) = x$ (cf. Section B.2), then for any $\psi \in \Phi_A \subseteq \operatorname{Ker}(A^\top)$,*

$$\langle \nabla f^*(\phi_t), \psi \rangle = \langle A\lambda_t, \psi \rangle = \left\langle \lambda_t, A^\top \psi \right\rangle = 0.$$

This allows the dual optimum to be rewritten as

$$\begin{aligned} \psi_A^f &= \operatorname*{argmin}_{\psi \in \Phi_A} f^*(\psi) \\ &= \operatorname*{argmin}_{\psi \in \Phi_A} f^*(\psi) - f^*(\phi_t) - \langle \nabla f^*(\phi_t), \psi - \phi_t \rangle; \end{aligned}$$

that is, the dual optimum ψ_A^f is the Bregman projection (according to f^*) onto Φ_A of any dual iterate $\phi_t = \nabla f(A\lambda_t)$. In particular, ψ_A^f is the Bregman projection onto the feasible set of the unconstrained optimum $\phi_0 = \nabla f(A\lambda_0)$!

The connection to Bregman divergences runs deep; in fact, mirroring the development of BOOST as "compiling out" the dual variables in the classical boosting presentation, it is possible to compile out the primal variables, producing an algorithm using only dual variables, meaning distributions over examples. This connection has been explored extensively (Kivinen and Warmuth, 1999; Collins et al., 2002).

Remark 6 *It may be tempting to use Theorem 4 to produce a stopping condition; that is, if for a supplied $\varepsilon > 0$, a primal iterate λ' and dual feasible $\psi' \in \Phi_A$ can be found satisfying $f(A\lambda') + f^*(\psi') \leq \varepsilon$,* BOOST *may terminate with the guarantee $f(A\lambda') - \bar{f}_A \leq \varepsilon$.*

Unfortunately, it is unclear how to produce dual iterates (excepting the trivial $\mathbf{0}_m$). If $\mathrm{Ker}(A^\top)$ can be computed, it suffices to l^2 project $\nabla f(A\lambda_t)$ onto this subspace. In general however, not only is $\mathrm{Ker}(A^\top)$ painfully expensive to compute, this computation does not at all fit the oracle model of boosting, where access to A is obscured. (What is $\mathrm{Ker}(A^\top)$ when the weak learning oracle learns a size-bounded decision tree?)

In fact, noting that the primal-dual relationship from Equation 2 can be written

$$\inf\{f(\Lambda) : \Lambda \in \mathrm{Im}(A)\} = \sup\left\{-f^*(\Psi) : \Psi \in \mathrm{Ker}(A^\top) = \mathrm{Im}(A)^\perp\right\}$$

(since $\mathrm{dom}(f^*) \subseteq \mathbb{R}^m_+$ encodes the orthant constraint), the standard oracle model gives elements of $\mathrm{Im}(A)$, but what is needed in the dual is an oracle for $\mathrm{Ker}(A^\top) = \mathrm{Im}(A)^\perp$.

4. Generalized Weak Learning Rate

The weak learning rate was critical to the original convergence analysis of AdaBoost, providing a handle on the progress of the algorithm. But to be useful, this value must be positive, which was precisely the condition granted by the weak learning assumption. This section will generalize the weak learning rate into a quantity which can be made positive for any boosting instance.

Note briefly that this manuscript will differ slightly from the norm in that weak learning will be a purely *sample-specific* concept. That is, the concern here is convergence in empirical risk, and all that matters is the sample $\mathcal{S} = \{(x_i, y_i)\}_1^m$, as encoded in A; it doesn't matter if there are wild points outside this sample, because the algorithm has no access to them.

This distinction has the following implication. The usual weak learning assumption states that there exists no uncorrelating distribution over the input *space*. This of course implies that any training sample $\mathcal{S}$ used by the algorithm will also have this property; however, it suffices that there is no distribution over the input *sample* $\mathcal{S}$ which uncorrelates the weak learners from the target.

Returning to task, the weak learning assumption posits the existence of a positive constant, the weak learning rate γ, which lower bounds the correlation of the best weak learner with the target for any distribution. Stated in terms of the matrix A,

$$0 < \gamma = \inf_{\substack{\phi \in \mathbb{R}^m_+ \\ \|\phi\|=1}} \max_{j \in [n]} \left| \sum_{i=1}^m (\phi)_i y_i h_j(x_i) \right| = \inf_{\phi \in \mathbb{R}^m_+ \setminus \{\mathbf{0}_m\}} \frac{\|A^\top \phi\|_\infty}{\|\phi\|_1} = \inf_{\phi \in \mathbb{R}^m_+ \setminus \{\mathbf{0}_m\}} \frac{\|A^\top \phi\|_\infty}{\|\phi - \mathbf{0}_m\|_1}. \tag{3}$$

Proposition 7 *A boosting instance is weak learnable iff* $\Phi_A = \{\mathbf{0}_m\}$.

Proof Suppose $\Phi_A = \{\mathbf{0}_m\}$; since the first infimum in Equation 3 is of a continuous function over a compact set, it has some minimizer ϕ'. But $\|\phi'\|_1 = 1$, meaning $\phi' \notin \Phi_A$, and so $\|A^\top \phi'\|_\infty > 0$. On the other hand, if $\Phi_A \neq \{\mathbf{0}_m\}$, take any $\phi'' \in \Phi_A \setminus \{\mathbf{0}_m\}$; then

$$0 \leq \gamma = \inf_{\phi \in \mathbb{R}^m_+ \setminus \{\mathbf{0}_m\}} \frac{\|A^\top \phi\|_\infty}{\|\phi\|_1} \leq \frac{\|A^\top \phi''\|_\infty}{\|\phi''\|_1} = 0.$$

■

Following this connection, the first way in which the weak learning rate is modified is to replace $\{\mathbf{0}_m\}$ with the dual feasible set $\Phi_A = \operatorname{Ker}(A^\top) \cap \mathbb{R}^m_+$. For reasons that will be sketched shortly, but fully dealt with only in Section 6, it is necessary to replace $\mathbb{R}^m_+$ with a more refined choice S.

Definition 8 *Given a matrix* $A \in \mathbb{R}^{m \times n}$ *and a set* $S \subseteq \mathbb{R}^m$, *define*

$$\gamma(A,S) := \inf \left\{ \frac{\|A^\top \phi\|_\infty}{\inf_{\psi \in S \cap \operatorname{Ker}(A^\top)} \|\phi - \psi\|_1} : \phi \in S \setminus \operatorname{Ker}(A^\top) \right\}.$$

First note that in the scenario of weak learnability (i.e., $\Phi_A = \{\mathbf{0}_m\}$ by Theorem 7), the choice $S = \mathbb{R}^m_+$ allows the new notion to exactly cover the old one: $\gamma(A, \mathbb{R}^m_+) = \gamma$.

To get a better handle on the meaning of S, first define the following projection and distance notation to a closed convex nonempty set C, where in the case of non-uniqueness (l^1 and l^∞), some arbitrary choice is made:

$$\mathsf{P}^p_C(x) \in \operatorname*{Argmin}_{y \in C} \|y - x\|_p, \qquad \mathsf{D}^p_C(x) = \|x - \mathsf{P}^p_C(x)\|_p.$$

Suppose, for some t, that $\nabla f(A\lambda_t) \in S \setminus \operatorname{Ker}(A^\top)$; then the infimum within $\gamma(A,S)$ may be instantiated with $\nabla f(A\lambda_t)$, yielding

$$\gamma(A,S) = \inf_{\phi \in S \setminus \operatorname{Ker}(A^\top)} \frac{\|A^\top \phi\|_\infty}{\|\phi - \mathsf{P}^1_{S \cap \operatorname{Ker}(A^\top)}(\phi)\|_1} \leq \frac{\|A^\top \nabla f(A\lambda_t)\|_\infty}{\|\nabla f(A\lambda_t) - \mathsf{P}^1_{S \cap \operatorname{Ker}(A^\top)}(\nabla f(A\lambda_t))\|_1}. \tag{4}$$

Rearranging this,

$$\gamma(A,S) \left\| \nabla f(A\lambda_t) - \mathsf{P}^1_{S \cap \operatorname{Ker}(A^\top)}(\nabla f(A\lambda_t)) \right\|_1 \leq \|A^\top \nabla f(A\lambda_t)\|_\infty. \tag{5}$$

This is helpful because the right hand side appears in standard guarantees for single-step progress in descent methods. Meanwhile, the left hand side has reduced the influence of A to a single number, and the normed expression is the distance to a restriction of dual feasible set, which will converge to zero if the infimum is to be approached, so long as this restriction contains the dual optimum.

This will be exactly the approach taken in this manuscript; indeed, the first step towards convergence rates, Proposition 20, will use exactly the upper bound in Equation 5. The detailed work that remains is then dealing with the distance to the dual feasible set. The choice of S will be made to facilitate the production of these bounds, and will depend on the optimization structure revealed in Section 5.

In order for these expressions to mean anything, $\gamma(A,S)$ must be positive.

Theorem 9 *Let matrix $A \in \mathbb{R}^{m\times n}$ and polyhedron $S \subseteq \mathbb{R}^m$ be given with $S \setminus \operatorname{Ker}(A^\top) \neq \emptyset$ and $S \cap \operatorname{Ker}(A^\top) \neq \emptyset$. Then $\gamma(A,S) > 0$.*

The proof, material on other generalizations of γ, and discussion on the polyhedrality of S can all be found in Section F.

As a final connection, since $A^\top \mathsf{P}^1_{S\cap\operatorname{Ker}(A^\top)}(\phi) = \mathbf{0}_n$, note that

$$\gamma(A,S) = \inf_{\phi\in S\setminus\operatorname{Ker}(A^\top)} \frac{\|A^\top\phi\|_\infty}{\|\phi - \mathsf{P}^1_{S\cap\operatorname{Ker}(A^\top)}(\phi)\|_1} = \inf_{\phi\in S\setminus\operatorname{Ker}(A^\top)} \frac{\|A^\top(\phi - \mathsf{P}^1_{S\cap\operatorname{Ker}(A^\top)}(\phi))\|_\infty}{\|\phi - \mathsf{P}^1_{S\cap\operatorname{Ker}(A^\top)}(\phi)\|_1}.$$

In this way, $\gamma(A,S)$ resembles a Lipschitz constant, reflecting the effect of A on elements of the dual, relative to the dual feasible set.

5. Optimization Structure

The scenario of weak learnability translates into a simple condition on the dual feasible set: the dual feasible set is the origin (in symbols, $\Phi_A = \operatorname{Ker}(A^\top)\cap\mathbb{R}^m_+ = \{\mathbf{0}_m\}$). And how about attainability—is there a simple way to encode this problem in terms of the optimization problem?

This section will identify the structure of the boosting optimization problem both in terms of the primal and dual problems, first studying the scenarios of weak learnability and attainability, and then showing that general instances can be decomposed into these two.

There is another behavior which will emerge through this study, motivated by the following question. The dual feasible set $\Phi_A = \operatorname{Ker}(A^\top)\cap\mathbb{R}^m_+$ is the set of nonnegative weightings of examples under which every weak learner (every column of A) has zero correlation; what is the support of these weightings?

Definition 10 *$H(A)$ denotes the* hard core *of A: the collection of examples which receive positive weight under some dual feasible point, a distribution upon which no weak learner is correlated with the target. Symbolically,*

$$H(A) := \{i \in [m] : \exists\psi \in \Phi_A, (\psi)_i > 0\}.$$

One case has already been considered; as established in Theorem 7, weak learnability is equivalent to $\Phi_A = \{\mathbf{0}_m\}$, which in turn is equivalent to $|H(A)| = 0$. But it will turn out that other possibilities for $H(A)$ also have direct relevance to the behavior of BOOST. Indeed, contrasted with the primal and dual problems and feasible sets, $H(A)$ will provide a conceptually simple, discrete object with which to comprehend the behavior of boosting.

5.1 Weak Learnability

The following theorem establishes four equivalent formulations of weak learnability.

Theorem 11 *For any $A \in \mathbb{R}^{m\times n}$ and $g \in \mathbb{G}_0$ the following conditions are equivalent:*

$$\exists\lambda \in \mathbb{R}^n \centerdot A\lambda \in \mathbb{R}^m_{--}, \tag{6}$$

$$\inf_{\lambda\in\mathbb{R}^n} f(A\lambda) = 0, \tag{7}$$

$$\psi^f_A = \mathbf{0}_m, \tag{8}$$

$$\Phi_A = \{\mathbf{0}_m\}. \tag{9}$$

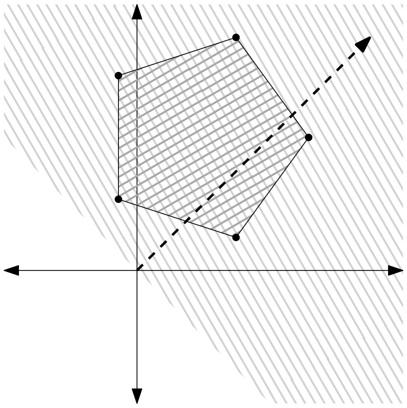


Figure 4: Geometric view of the primal and dual problem, under weak learnability. The vertices of the pentagon denote the points $\{-a_i\}_1^m$. The arrow, denoting λ in Equation 6, defines a homogeneous halfspace containing these points; on the other hand, their convex hull does not contain the origin. Please see Theorem 11 and its discussion.

First note that Equation 9 indicates (via Theorem 7) this is indeed the weak learnability setting, equivalently $|H(A)| = 0$.

Recall the earlier discussion of boosting as searching for a halfspace containing the points $\{-a_i\}_1^m = \{-\mathbf{e}_i^\top A\}_1^m$; Equation 6 encodes precisely this statement, and moreover that there exists such a halfspace with these points interior to it. Note that this statement also encodes the margin separability equivalence of weak learnability due to Shalev-Shwartz and Singer (2008); specifically, if labels are bounded away from 0 and each point $-a_i$ (row of $-A$) is replaced with $-y_i a_i$, the definition of A grants that positive examples will land on one side of the hyperplane, and negative examples on the other.

Equation 9 and Equation 6 can be interpreted geometrically, as depicted in Figure 4: the dual feasibility statement is that no convex combination of $\{-a_i\}_1^m$ will contain the origin.

Next, Equation 7 is the (error part of the) usual strong PAC guarantee (Schapire, 1990): weak learnability entails that the training error will go to zero. And, as must be the case when $\Phi_A = \{\mathbf{0}_m\}$, Equation 8 provides that $\psi_A^f = \mathbf{0}_m$.

Proof of Theorem 11 (Equation 6 $\implies$ Equation 7.) Let $\bar{\lambda} \in \mathbb{R}^n$ be given with $A\bar{\lambda} \in \mathbb{R}^m_{--}$, and let any increasing sequence $\{c_i\}_1^\infty \uparrow \infty$ be given. Then, since $f > 0$ and $\lim_{x \to -\infty} g(x) = 0$,

$$\inf_\lambda f(A\lambda) \leq \lim_{i \to \infty} f(c_i A\bar{\lambda}) = 0 \leq \inf_\lambda f(A\lambda).$$

(Equation 7 $\implies$ Equation 8.) The point $\mathbf{0}_m$ is always dual feasible, and

$$\inf_\lambda f(A\lambda) = 0 = -f^*(\mathbf{0}_m).$$

Since the dual optimum is unique (Theorem 4), $\psi_A^f = \mathbf{0}_m$.

(Equation 8 $\implies$ Equation 9.) Suppose there exists $\psi \in \Phi_A$ with $\psi \neq \mathbf{0}_m$. Since $-f^*$ is continuous and increasing along every positive direction at $\mathbf{0}_m = \psi_A^f$ (see Lemma 2 and Lemma 36), there must exist some tiny $\tau > 0$ such that $-f^*(\tau\psi) > -f^*(\psi_A^f)$, contradicting the selection of ψ_A^f as the unique optimum.

(Equation 9 $\Longrightarrow$ Equation 6.) This case is directly handled by Gordan's theorem (cf. Theorem 29). ■

5.2 Attainability

For strictly convex functions, there is a nice characterization of attainability, which will require the following definition.

Definition 12 (Hiriart-Urruty and Lemaréchal 2001, Section B.3.2) *A closed convex function h is called* 0-coercive *when all level sets are compact. (That is, for any $\alpha \in \mathbb{R}$, the set $\{x : f(x) \leq \alpha\}$ is compact.)*

Proposition 13 *Suppose h is differentiable, strictly convex, and* $\text{dom}(h) = \mathbb{R}^m$. *Then* $\inf_x h(x)$ *is attainable iff h is 0-coercive.*

Note that 0-coercivity means the domain of the infimum in Equation 1 can be restricted to a compact set, and attainability in turn follows just from properties of minimization of continuous functions on compact sets. It is the converse which requires some structure; the proof however is unilluminating and deferred to Section G.3.

Armed with this notion, it is now possible to build an attainability theory for $f \circ A$. Some care must be taken with the above concepts, however; note that while f is strictly convex, $f \circ A$ need not be (for instance, if there exist nonzero elements of $\text{Ker}(A)$, then moving along these directions does not change the objective value). Therefore, 0-coercivity statements will refer to the function

$$(f + \iota_{\text{Im}(A)})(x) = \begin{cases} f(x) & \text{when } x \in \text{Im}(A), \\ \infty & \text{otherwise.} \end{cases}$$

This function is effectively taking the epigraph of f, and intersecting it with a slice representing $\text{Im}(A) = \{A\lambda : \lambda \in \mathbb{R}^n\}$, the set of points considered by the algorithm. As such, it is merely a convenient way of dealing with $\text{Ker}(A)$ as discussed above.

Theorem 14 *For any $A \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{G}_0$, the following conditions are equivalent:*

$$\forall \lambda \in \mathbb{R}^n \centerdot A\lambda \notin \mathbb{R}^m_- \setminus \{\mathbf{0}_m\}, \tag{10}$$

$$f + \iota_{\text{Im}(A)} \text{ is 0-coercive}, \tag{11}$$

$$\psi_A^f \in \mathbb{R}^m_{++}, \tag{12}$$

$$\Phi_A \cap \mathbb{R}^m_{++} \neq \emptyset. \tag{13}$$

Following the discussion above, Equation 11 is the desired attainability statement.

Next, note that Equation 13 is equivalent to the expression $|H(A)| = m$, that is, there exists a distribution with positive weight on all examples, upon which every weak learner is uncorrelated. The forward direction is direct from the existence of a single $\psi \in \Phi_A \cap \mathbb{R}^m_{++}$. For the converse, note

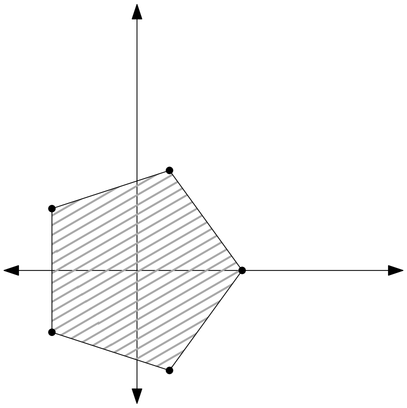


Figure 5: Geometric view of the primal and dual problem, under attainability. Once again, the $\{-a_i\}_1^m$ are the vertices of the pentagon. This time, no (closed) homogeneous halfspace containing all the points will contain one strictly, and the relative interior of the pentagon contains the origin. Please see Theorem 14 and its discussion.

that the ψ_i corresponding to each $i \in H(A)$ can be combined into $\psi = \sum_i \psi_i \in \text{Ker}(A^\top) \cap \mathbb{R}^m_{++}$ (since $\text{Ker}(A^\top)$ is a subspace).

For a geometric interpretation, consider Equation 10 and Equation 13. The first says that any halfspace containing some $-a_i$ within its interior must also fail to contain some $-a_j$ (with $i \neq j$). (Equation 10 also allows for the scenario that no valid enclosing halfspace exists, that is, $\lambda = \mathbf{0}_n$.) The latter states that the origin $\mathbf{0}_m$ is contained within a positive convex combination of $\{-a_i\}_1^m$ (alternatively, the origin is within the relative interior of these points). These two scenarios appear in Figure 5.

Finally, note Equation 12: it is not only the case that there are dual feasible points fully interior to $\mathbb{R}^m_+$, but furthermore the dual optimum is also interior. This will be crucial in the convergence rate analysis, since it will allow the dual iterates to never be too small.

Proof of Theorem 14 (Equation 10 $\Longrightarrow$ Equation 11.) Let $d \in \mathbb{R}^m \setminus \{\mathbf{0}_m\}$ and $\lambda \in \mathbb{R}^n$ be arbitrary. To show 0-coercivity, it suffices (Hiriart-Urruty and Lemaréchal, 2001, Proposition B.3.2.4.iii) to show

$$\lim_{t\to\infty} \frac{f(A\lambda + td) + \iota_{\text{Im}(A)}(A\lambda + td) - f(A\lambda)}{t} > 0. \tag{14}$$

If $d \notin \text{Im}(A)$ (and $t > 0$), then $\iota_{\text{Im}(A)}(A\lambda + td) = \infty$. Suppose $d \in \text{Im}(A)$; by Equation 10, since $d \neq \mathbf{0}_m$, then $d \notin \mathbb{R}^m_-$, meaning there is at least one positive coordinate j. But then, since $g > 0$ and g is convex,

$$\begin{aligned}\text{Eq. 14} &\geq \lim_{t\to\infty} \frac{g(\mathbf{e}_j^\top(A\lambda + td)) - f(A\lambda)}{t} \\ &\geq \lim_{t\to\infty} \frac{g(\mathbf{e}_j^\top A\lambda) + td_j g'(\mathbf{e}_j^\top A\lambda) - f(A\lambda)}{t} \\ &= d_j g'(\mathbf{e}_j^\top A\lambda),\end{aligned}$$

which is positive by the selection of d_j and since $g' > 0$.

(Equation 11 $\Longrightarrow$ Equation 12.) Since the infimum is attainable, designate any $\bar{\lambda}$ satisfying $\inf_\lambda f(A\lambda) = f(A\bar{\lambda})$ (note, although f is strictly convex, $f \circ A$ need not be, thus uniqueness is not guaranteed!). The optimality conditions of Fenchel problems may be applied, meaning $\psi_A^f = \nabla f(A\bar{\lambda})$, which is interior to $\mathbb{R}_+^m$ since $\nabla f \in \mathbb{R}_{++}^m$ everywhere (cf. Lemma 36). (For the optimality conditions, see Borwein and Lewis 2000, Exercise 3.3.9.f, with a negation inserted to match the negation inserted within the proof of Theorem 4.)

(Equation 12 $\Longrightarrow$ Equation 13.) This holds since $\Phi_A \supseteq \{\psi_A^f\}$ and $\psi_A^f \in \mathbb{R}_{++}^m$.

(Equation 13 $\Longrightarrow$ Equation 10.) This case is directly handled by Stiemke's Theorem (cf. Theorem 30). ■

5.3 General Setting

So far, the scenarios of weak learnability and attainability corresponded to the extremal hard core cases of $|H(A)| \in \{0, m\}$. The situation in the general setting $1 \le |H(A)| \le m-1$ is basically as good as one could hope for: it interpolates between the two extremal cases.

As a first step, partition A into two submatrices according to $H(A)$.

Definition 15 *Partition $A \in \mathbb{R}^{m\times n}$ by rows into two matrices $A_0 \in \mathbb{R}^{m_0\times n}$ and $A_+ \in \mathbb{R}^{m_+\times n}$, where A_+ has rows corresponding to $H(A)$, and $m_+ = |H(A)|$. For convenience, permute the examples so that*

$$A = \begin{bmatrix} A_0 \\ A_+ \end{bmatrix}.$$

(This merely relabels the coordinate axes, and does not change the optimization problem.) Note that this decomposition is unique, since $H(A)$ is uniquely specified.

As a first consequence, this partition cleanly decomposes the dual feasible set Φ_A into Φ_{A_0} and Φ_{A_+}.

Proposition 16 *For any $A \in \mathbb{R}^{m\times n}$, $\Phi_{A_0} = \{\mathbf{0}_{m_0}\}$, $\Phi_{A_+} \cap \mathbb{R}_{++}^{m_+} \neq \emptyset$, and*

$$\Phi_A = \Phi_{A_0} \times \Phi_{A_+}.$$

Furthermore, no other partition of A into $B_0 \in \mathbb{R}^{z\times n}$ and $B_+ \in \mathbb{R}^{p\times n}$ satisfies these properties.

Proof It must hold that $\Phi_{A_0} = \{\mathbf{0}_{m_0}\}$, since otherwise there would exist $\psi \in \operatorname{Ker}(A_0^\top) \cap \mathbb{R}_+^{m_0}$ with $\psi \neq \mathbf{0}_{m_0}$, which could be extended to $\psi' = \psi \times \mathbf{0}_{m_+} \in \Phi_A$ and the positive coordinate of ψ could be added to $H(A)$, contradicting the construction of $H(A)$ as including all such rows.

The property $\Phi_{A_+} \cap \mathbb{R}_{++}^{m_+} \neq \emptyset$ was proved in the discussion of Theorem 14: simply add together, for each $i \in H(A)$, the ψ_i's corresponding to positive weight on i.

For the decomposition, note first that certainly every $\psi \in \Phi_{A_0} \times \Phi_{A_+}$ satisfies $\psi \in \Phi_A$. Now suppose contradictorily that there exists $\psi' \in \Phi_A \setminus (\Phi_{A_0} \times \Phi_{A_+})$. There must exist $j \in [m] \setminus H(A)$ with $(\psi')_j > 0$, since otherwise $\psi' \in \{\mathbf{0}_z\} \times \Phi_{A_+}$; but that means j should have been included in $H(A)$, a contradiction.

For the uniqueness property, suppose some other B_0, B_+ is given, satisfying the desired properties. It is impossible that some $a_i \in B_+$ is not in $H(A)$, since any $\psi \in \Phi_{B_+}$ can be extended to

$\psi' \in \Phi_A$ with positive weight on i, and thus is included in $H(A)$ by definition. But the other case with $i \in H(A)$ but $a_i \in B_0$ is equally untenable, since the corresponding measure ψ_i is in Φ_A but not in $\Phi_{B_0} \times \Phi_{B_+}$. ■

The main result of this section will have the same two main ingredients as Proposition 16:

- The full boosting instance may be uniquely decomposed into two pieces, A_0 and A_+, each of which individually behave like the weak learnability and attainability scenarios.
- The subinstances have a somewhat independent effect on the full instance.

Theorem 17 *Let $g \in \mathbb{G}_0$ and $A \in \mathbb{R}^{m\times n}$ be given. Let $B_0 \in \mathbb{R}^{z\times n}$, $B_+ \in \mathbb{R}^{p\times n}$ be any partition of A by rows. The following conditions are equivalent:*

$$\exists \lambda \in \mathbb{R}^n \centerdot B_0\lambda \in \mathbb{R}^z_{--} \wedge B_+\lambda = \mathbf{0}_p \quad \textit{and} \quad \forall \lambda \in \mathbb{R}^n \centerdot B_+\lambda \notin \mathbb{R}^p_- \setminus \{\mathbf{0}_p\}, \tag{15}$$

$$\left\{ \begin{array}{l} \inf_{\lambda\in\mathbb{R}^n} f(A\lambda) = \inf_{\lambda\in\mathbb{R}^n} f(B_+\lambda), \quad \textit{and} \quad \inf_{\lambda\in\mathbb{R}^n} f(B_0\lambda) = 0, \\ \textit{and} \quad f + \iota_{\mathrm{Im}(B_+)} \textit{ is 0-coercive}, \end{array} \right\} \tag{16}$$

$$\psi_A^f = \begin{bmatrix} \psi_{B_0}^f \\ \psi_{B_+}^f \end{bmatrix} \quad \textit{with} \quad \psi_{B_0}^f = \mathbf{0}_z \quad \textit{and} \quad \psi_{B_+}^f \in \mathbb{R}^p_{++}, \tag{17}$$

$$\Phi_{B_0} = \{\mathbf{0}_z\}, \quad \textit{and} \quad \Phi_{B_+} \cap \mathbb{R}^p_{++} \neq \emptyset, \quad \textit{and} \quad \Phi_A = \Phi_{B_0} \times \Phi_{B_+}. \tag{18}$$

Stepping through these properties, notice that Equation 18 mirrors the expression in Proposition 16. But that Theorem also granted that this representation was unique, thus only one partition of A satisfies the above properties, namely A_0, A_+. Since this Theorem is stated as a series of equivalences, any one of these properties can in turn be used to identify the hard core set $H(A)$.

To continue with geometric interpretations, notice that Equation 15 states that there exists a halfspace strictly containing those points in $[m] \setminus H(A)$, with all points of $H(A)$ on its boundary; furthermore, trying to adjust this halfspace to contain elements of $H(A)$ will place others outside it. With regards to the geometry of the dual feasible set as provided by Equation 18, the origin is within the relative interior of the points corresponding to $H(A)$, however the convex hull of the other $m - |H(A)|$ points can not contain the origin. Furthermore, if the origin is written as a convex combination of all points, this combination must place zero weight on the points with indices $[m] \setminus H(A)$. This scenario is depicted in Figure 6.

In Equation 16 and Equation 17, B_0 mirrors the behavior of weakly learnable instances in Theorem 11, and analogously B_+ follows instances with minimizers from Theorem 14. The interesting addition, as discussed above, is the independence of these components: Equation 16 provides that the infimum of the combined problem is the sum of the infima of the subproblems, while Equation 17 provides that the full dual optimum may be obtained by concatenating the subproblems' dual optima.

Proof of Theorem 17 (Equation 15 $\implies$ Equation 16.) Let $\bar{\lambda}$ be given with $B_0\bar{\lambda} \in \mathbb{R}^z_{--}$ and $B_+\bar{\lambda} = \mathbf{0}_p$, and let $\{c_i\}_1^\infty \uparrow \infty$ be an arbitrary sequence increasing without bound. Lastly, let $\{\lambda_i\}_1^\infty$ be a minimizing sequence for $\inf_\lambda f(B_+\lambda)$. Then

$$\begin{aligned} \inf_\lambda f(B_+\lambda) &= \lim_{i\to\infty} \left(f(B_+\lambda_i) + f(c_i B_0 \bar{\lambda}) \right) \geq \inf_\lambda f(A\lambda) \\ &= \inf_\lambda (f(B_+\lambda) + f(B_0\lambda)) \geq \inf_\lambda f(B_+\lambda), \end{aligned}$$

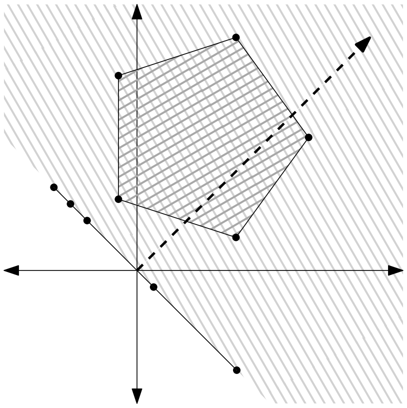


Figure 6: Geometric view of the primal and dual problem in the general case. There is a closed homogeneous halfspace containing the points $\{-a_i\}_1^m$, where the hard core lies on the halfspace boundary, and the other points are within its interior; moreover, there does not exist a closed homogeneous halfspace containing all points but with strict containment on a point in the hard core. Finally, although the origin is in the convex hull of $\{-a_i\}_1^m$, any such convex combination places zero weight on points outside the hard core. Please see Theorem 17 and its discussion.

which used the fact that $f(B_0\lambda) \geq 0$ since $f \geq 0$. And since the chain of inequalities starts and ends the same, it must be a chain of equalities, which means $\inf_\lambda f(B_0\lambda) = 0$. To show 0-coercivity of $f + \iota_{\mathrm{Im}(B_+)}$, note the second part of Equation 15 is one of the conditions of Theorem 14.

(Equation 16 $\implies$ Equation 17.) First, by Theorem 11, $\inf_\lambda f(B_0\lambda) = 0$ means $\psi^f_{B_0} = \mathbf{0}_z$ and $\Phi_{B_0} = \{\mathbf{0}_z\}$. Thus

$$\begin{aligned}
-f^*(\psi^f_A) &= \sup_{\psi \in \Phi_A} -f^*(\psi) \\
&= \sup_{\substack{\psi_z \in \mathbb{R}^z_+ \\ \psi_p \in \mathbb{R}^p_+ \\ B_0^\top \psi_z + B_+^\top \psi_p = \mathbf{0}_n}} -f^*(\psi_z) - f^*(\psi_p) \\
&\geq \sup_{\psi_z \in \Phi_{B_0}} -f^*(\psi_z) + \sup_{\psi_p \in \Phi_{B_+}} -f^*(\psi_p) \\
&= 0 - f^*(\psi^f_{B_+}) = \inf_{\lambda \in \mathbb{R}^n} f(B_+\lambda) = \inf_{\lambda \in \mathbb{R}^n} f(A\lambda) = -f^*(\psi^f_A).
\end{aligned}$$

Combining this with $f^*(x) = \sum_i g((x)_i)$ and $g^*(0) = 0$ (cf. Lemma 2, Theorem 4), $f^*(\psi^f_A) = f^*(\psi^f_{B_+}) = f^*(\begin{bmatrix} \psi^f_{B_0} \\ \psi^f_{B_+} \end{bmatrix})$. But Theorem 4 shows ψ^f_A was unique, which gives the result. And to obtain $\psi^f_{B_+} \in \mathbb{R}^p_{++}$, use Theorem 14 with the 0-coercivity of $f + \iota_{\mathrm{Im}(B_+)}$.

(Equation 17 $\implies$ Equation 18.) Since $\psi^f_{B_0} = \mathbf{0}_z$, it follows by Theorem 11 that $\Phi_{B_0} = \{\mathbf{0}_z\}$. Furthermore, since $\psi^f_{B_+} \in \mathbb{R}^p_{++}$, it follows that $\Phi_{B_+} \cap \mathbb{R}^p_{++} \neq \emptyset$. Now suppose contradictorily that $\Phi_A \neq \Phi_{B_0} \times \Phi_{B_+}$; since it always holds that $\Phi_A \supseteq \Phi_{B_0} \times \Phi_{B_+}$, this supposition grants the existence of $\psi = \left[\begin{smallmatrix} \psi_z \\ \psi_p \end{smallmatrix}\right] \in \Phi_A$ where $\psi_z \in \mathbb{R}^z_+ \setminus \{\mathbf{0}_z\}$.

Consider the element $q := \psi + \psi_A^f$, which has more nonzero entries than ψ_A^f, but still $q \in \Phi_A$ since Φ_A is a convex cone. Let I_q index the nonzero entries of q, and let A_q be the restriction of A to the rows I_q. Since $q \in \Phi_A$, meaning q is nonnegative and $q \in \operatorname{Ker}(A^\top)$, it follows that the restriction of q to its positive entries is within $\operatorname{Ker}(A_q^\top)$ (because only zeros of q and matching rows of A are removed, dot products between q with rows of $A^\top$ are the same as dot products between the restriction of q and rows of $A_q^\top$), and so $q \in \Phi_{A_q}$, meaning $\Phi_{A_q} \cap \mathbb{R}_{++}^{|I_q|}$ is nonempty. Correspondingly, by Theorem 14, the dual optimum $\psi_{A_q}^f$ of this restricted problem will have only positive entries. But by the same reasoning granting that q restricted to I_q is within Φ_{A_q}, it follows that the full optimum ψ_A^f, restricted to I_q, must also be within Φ_{A_q} (since, by q's construction, ψ_A^f's zero entries are a superset of the zero entries of q). Therefore this restriction $\hat{\psi}_A^f$ of ψ_A^f to I_q will have at least one zero entry, meaning it can not be equal to $\psi_{A_q}^f$; but Theorem 4 provided that the dual optimum is unique, thus $-f^*(\psi_{A_q}^f) > -f^*(\hat{\psi}_A^f)$. Finally, produce $\bar{\psi}_{A_q}^f$ from $\psi_{A_q}^f$ by inserting a zero for each entry of I_q; the same reasoning that allows feasibility to be maintained while removing zeros allows them to be added, and thus $\bar{\psi}_{A_q}^f \in \Phi_A$. But this is a contradiction: since $g^*(0) = 0$ (cf. Lemma 2), both $\bar{\psi}_{A_q}^f$ and the optimum ψ_A^f have zero contribution to the objective along the entries outside of I_q, and thus

$$-f^*(\bar{\psi}_{A_q}^f) = -f^*(\psi_{A_q}^f) > -f^*(\hat{\psi}_A^f) = -f^*(\psi_A^f),$$

meaning $\bar{\psi}_{A_q}^f$ is feasible and has strictly greater objective value than the optimum ψ_A^f, a contradiction.

(Equation 18 $\implies$ Equation 15.) Unwrapping the definition of Φ_A, the assumed statements imply

$$(\forall \phi_0 \in \mathbb{R}_+^z \setminus \{\mathbf{0}_z\}, \phi_+ \in \mathbb{R}_+^p \centerdot B_0^\top \phi_0 + B_+^\top \phi_+ \neq \mathbf{0}_n) \wedge (\exists \phi_+ \in \mathbb{R}_{++}^p \centerdot B_+^\top \phi_+ = \mathbf{0}_n).$$

Applying Motzkin's transposition theorem (cf. Theorem 31) to the left statement and Stiemke's theorem (cf. Theorem 30, which is implied by Motzkin's theorem) to the right yields

$$(\exists \lambda \in \mathbb{R}^n \centerdot B_0 \lambda \in \mathbb{R}_{--}^z \wedge B_+ \lambda \in \mathbb{R}_-^p) \wedge (\forall \lambda \in \mathbb{R}^n \centerdot B_+ \lambda \notin \mathbb{R}_-^p \setminus \{\mathbf{0}_p\}),$$

which implies the desired statement. ■

Remark 18 *Notice the dominant role A plays in the structure of the solution found by boosting. For every $i \in [m] \setminus H(A)$, the corresponding dual weights go to zero (i.e., $(\nabla f(A\lambda_t))_i \downarrow 0$), and the corresponding primal margins grow unboundedly (i.e., $-\mathbf{e}_i^\top A\lambda_t \uparrow \infty$, since otherwise $\inf_\lambda f(A_0\lambda) > 0$). This is completely unaffected by the choice of $g \in \mathbb{G}_0$. Furthermore, whether this instance is weak learnable, attainable, or neither is dictated purely by A (respectively $|H(A)| = 0$, $|H(A)| = m$, or $|H(A)| \in [1, m-1]$).*

Where different loss functions disagree is how they assign dual weight to the points in $H(A)$. In particular, each $g \in \mathbb{G}_0$ (and corresponding f) defines a notion of entropy via f^*. The dual optimization in Theorem 4 can then be interpreted as selecting the max entropy choice (per f^*) amongst those convex combinations of $H(A)$ equal to the origin.

6. Convergence Rates

Convergence rates will be proved for the following family of loss functions.

Definition 19 *$\mathbb{G}$ contains all functions g satisfying the following properties. First, $g \in \mathbb{G}_0$. Second, for any $x \in \mathbb{R}^m$ satisfying $f(x) \leq f(A\lambda_0) = mg(0)$, and for any coordinate $(x)_i$, there exist constants $\eta > 0$ and $\beta > 0$ such that $g''((x)_i) \leq \eta g((x)_i)$ and $g((x)_i) \leq \beta g'((x)_i)$.*

The exponential loss is in this family with $\eta = \beta = 1$ since $\exp(\cdot)$ is a fixed point with respect to the differentiation operator. Furthermore, as is verified in Remark 46, the logistic loss is also in this family, with $\eta = 2^m/(m\ln(2))$ and $\beta = 1 + 2^m$ (which may be loose). In a sense, η and β encode how similar some $g \in \mathbb{G}$ is to the exponential loss, and thus these parameters can degrade radically. However, outside the weak learnability case, the other terms in the bounds here can also incur a large penalty with the exponential loss, and there is some evidence that this is unavoidable (see the lower bounds in Mukherjee et al. 2011 or the upper bounds in Rätsch et al. 2001).

The first step towards proving convergence rates will be to lower bound the improvement due to one iteration. As discussed previously, standard techniques for analyzing descent methods provide such bounds in terms of gradients, however to overcome the difficulty of unattainability in the primal space, the key will be to convert this into distances in the dual via $\gamma(A,S)$, as in Equation 5.

Proposition 20 *For any t, $g \in \mathbb{G}$, $A \in \mathbb{R}^{m\times n}$, and $S \supseteq \{\nabla f(A\lambda_t)\}$ with $\gamma(A,S) > 0$,*

$$f(A\lambda_{t+1}) - \bar{f}_A \leq f(A\lambda_t) - \bar{f}_A - \frac{\gamma(A,S)^2 \mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))^2}{6\eta f(A\lambda_t)}.$$

Proof The stopping condition grants $\nabla f(A\lambda_t) \notin \mathrm{Ker}(A^\top)$. Proceeding as in Equation 4,

$$\gamma(A,S) = \inf_{\phi\in S\setminus\mathrm{Ker}(A^\top)} \frac{\|A^\top\phi\|_\infty}{\mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)} \leq \frac{\|A^\top\nabla f(A\lambda_t)\|_\infty}{\mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))}.$$

Combined with the approximate line search guarantee from Proposition 38,

$$f(A\lambda_t) - f(A\lambda_{t+1}) \geq \frac{\|A^\top\nabla f(A\lambda_t)\|_\infty^2}{6\eta f(A\lambda_t)} \geq \frac{\gamma(A,S)^2 \mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))^2}{6\eta f(A\lambda_t)}.$$

Subtracting $\bar{f}_A$ from both sides and rearranging yields the statement. ■

The task now is to manage the dual distance $\mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))$, specifically to produce a relation to $f(A\lambda_t) - \bar{f}_A$, the total suboptimality in the preceding iteration; from there, standard tools in convex optimization will yield convergence rates. Matching the problem structure revealed in Section 5, first the extremal cases of weak learnability and attainability will be handled, and only then the general case. The significance of this division is that the extremal cases have rate $O(\ln(1/\epsilon))$, whereas the general case has rate $O(1/\epsilon)$ (with a matching lower bound provided for the logistic loss). The reason, which will be elaborated in further sections, is straightforward: the extremal cases are fast for essentially opposing regions, and this conflict will degrade the rate in the general case.

6.1 Weak Learnability

Theorem 21 *Suppose $|H(A)| = 0$ and $g \in \mathbb{G}$; then $\gamma(A, \mathbb{R}^m_+) > 0$, and for any $t \geq 0$,*

$$f(A\lambda_t) \leq f(A\lambda_0)\left(1 - \frac{\gamma(A,\mathbb{R}^m_+)^2}{6\beta^2\eta}\right)^t.$$

Proof By Theorem 11, $\Phi_A = \{\mathbf{0}_m\}$, meaning

$$\mathsf{D}^1_{\Phi_A}(\nabla f(A\lambda_t)) = \inf_{\psi\in\Phi_A} \|\nabla f(A\lambda_t) - \psi\|_1 = \|\nabla f(A\lambda_t)\|_1 \geq f(A\lambda_t)/\beta.$$

Next, $\mathbb{R}^m_+$ is polyhedral, and Theorem 11 grants $\mathbb{R}^m_+ \cap \operatorname{Ker}(A^\top) \neq \emptyset$ and $\mathbb{R}^m_+ \setminus \operatorname{Ker}(A^\top) \neq \emptyset$, so Theorem 9 provides $\gamma(A, \mathbb{R}^m_+) > 0$. Since $\nabla f(A\lambda_t) \in \mathbb{R}^m_+$, all conditions of Proposition 20 are met, and using $\bar{f}_A = 0$ (again by Theorem 11),

$$f(A\lambda_{t+1}) \leq f(A\lambda_t) - \frac{\gamma(A,\mathbb{R}^m_+)^2 f(A\lambda_t)^2}{6\beta^2\eta f(A\lambda_t)} = f(A\lambda_t)\left(1 - \frac{\gamma(A,\mathbb{R}^m_+)^2}{6\beta^2\eta}\right), \tag{19}$$

and recursively applying this inequality yields the result. ■

As discussed in Section 4, $\gamma(A, \mathbb{R}^m_+) = \gamma$, the latter quantity being the classical weak learning rate.

Specializing this analysis to the exponential loss (where $\eta = \beta = 1$), the bound becomes $(1 - \gamma^2/6)^t$, which recovers the bound of Schapire and Singer (1999), although with vastly different analysis. (The exact expression has denominator 2 rather than 6, which can be recovered with the closed form line search; cf. Section D.)

In general, solving for t in the expression

$$\varepsilon = \frac{f(A\lambda_t) - \bar{f}_A}{f(A\lambda_0) - \bar{f}_A} \leq \left(1 - \frac{\gamma^2}{6\beta^2\eta}\right)^t \leq \exp\left(-\frac{t\gamma^2}{6\beta^2\eta}\right)$$

reveals that $t \leq \frac{6\beta^2\eta}{\gamma^2}\ln(1/\varepsilon)$ iterations suffice to reach suboptimality ε. Recall that β and η, in the case of the logistic loss, have only been bounded by quantities like 2^m. While it is unclear if this analysis of β and η was tight, note that it is plausible that the logistic loss is slower than the exponential loss in this scenario, as it works less in initial phases to correct minor margin violations.

Remark 22 *The rate $O(\ln(1/\varepsilon))$ depended crucially on both $g \leq \beta g'$ and $g'' \leq \eta g$. If for instance the second inequality were replaced with $g'' \leq C$, then Equation 19 would instead have form $f(A\lambda_{t+1}) \leq f(A\lambda_t) - f(A\lambda_t)^2 O(1)$, which by an application of Lemma 33 would grant a rate $O(1/\varepsilon)$. For functions which asymptote to zero (i.e., everything in $\mathbb{G}_0$), satisfying this milder second order condition is quite easy. The real mechanism behind producing a fast rate is $g \leq \beta g'$, which guarantees that the flattening of the objective function is concomitant with low objective values.*

6.2 Attainability

Consider now the case of attainability. Recall from Theorem 14 and Proposition 13 that attainability occurred along with a stronger property, the 0-coercivity (compact level sets) of $f + \iota_{\operatorname{Im}(A)}$ (it was not possible to work with $f \circ A$ directly, which will have unbounded level sets when $\operatorname{Ker}(A) \neq \mathbf{0}_n$).

This has an immediate consequence to the task of relating $f(A\lambda_t) - \bar{f}_A$ to the dual distance $\mathsf{D}^1_{S\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))$. f is a strictly convex function, which means it is strongly convex over any compact set. Strong convexity in the primal corresponds to upper bounds on second derivatives (occasionally termed *strong smoothness*) in the dual, which in turn can be used to relate distance and objective values. This also provides the choice of polyhedron S in $\gamma(A,S)$: unlike the case of weak learnability, where the unbounded set $\mathbb{R}^m_+$ was used, a compact set containing the initial level set will be chosen.

Theorem 23 *Suppose $|H(A)| = m$ and $g \in \mathbb{G}$. Then there exists a (compact) tightest axis-aligned rectangle $\mathcal{C}$ containing the initial level set $\{x \in \mathbb{R}^m : (f + \iota_{\mathrm{Im}(A)})(x) \le f(A\lambda_0)\}$, and f is strongly convex with modulus $c > 0$ over $\mathcal{C}$. Finally, either λ_0 is optimal, or $\gamma(A, \nabla f(\mathcal{C})) > 0$, and for all t,*

$$f(A\lambda_t) - \bar{f}_A \le (f(A\lambda_0) - \bar{f}_A)\left(1 - \frac{c\gamma(A,\nabla f(\mathcal{C}))^2}{3\eta f(A\lambda_0)}\right)^t.$$

As in Section 6.1, when λ_0 is suboptimal, this bound may be rearranged to say that $t \le \frac{3\eta f(A\lambda_0)}{c\gamma(A,\nabla f(\mathcal{C}))^2}\ln(1/\varepsilon)$ iterations suffice to reach suboptimality ε.

To make sense of this bound and its proof, the essential object is $\mathcal{C}$, whose properties are captured in the following Theorem, which is stated with some slight generality in order to allow reuse in Section 6.3.

Lemma 24 *Let $g \in \mathbb{G}$, $A \in \mathbb{R}^{m\times n}$ with $|H(A)| = m$, and any $d \ge \inf_\lambda f(A\lambda)$ be given. Then there exists a (compact nonempty) tightest axis-aligned rectangle $\mathcal{C} \supseteq \{x \in \mathbb{R}^m : (f + \iota_{\mathrm{Im}(A)})(x) \le d\}$. Furthermore, the dual image $\nabla f(\mathcal{C}) \subset \mathbb{R}^m$ is also a (compact nonempty) axis-aligned rectangle, and moreover it is strictly contained within* $\mathrm{dom}(f^*) \subseteq \mathbb{R}^m_+$. *Finally, $\nabla f(\mathcal{C})$ contains dual feasible points (i.e., $\nabla f(\mathcal{C}) \cap \Phi_A \neq \emptyset$).*

A full proof may be found in Section G.4; the principle is that $|H(A)| = m$ provides 0-coercivity of $f + \iota_{\mathrm{Im}(A)}$, and thus the initial level set is compact. To later show $\gamma(A,S) > 0$ via Theorem 9, S must be polyhedral, and to apply Proposition 20, it must contain the dual iterates $\{\nabla f(A\lambda_t)\}_{t=1}^\infty$; the easiest choice then is to take the bounding box $\mathcal{C}$ of the initial level set, and use its dual map $\nabla f(\mathcal{C})$. To exhibit dual feasible points within $\nabla f(\mathcal{C})$, note that $\mathcal{C}$ will contain a primal minimizer, and optimality conditions grant that $\nabla f(\mathcal{C})$ contains the dual optimum.

With the polyhedron in place, Proposition 20 may be applied, so what remains is to control the dual distance. Again, this result will be stated with some extra generality in order to allow reuse in Section 6.3.

Lemma 25 *Let $A \in \mathbb{R}^{m\times n}$, $g \in \mathbb{G}$, and any compact set S with $\nabla f(S) \cap \mathrm{Ker}(A^\top) \neq \emptyset$ be given. Then f is strongly convex over S, and taking $c > 0$ to be the modulus of strong convexity, for any $x \in S \cap \mathrm{Im}(A)$,*

$$f(x) - \bar{f}_A \le \frac{1}{2c}\inf_{\psi\in\nabla f(S)\cap\mathrm{Ker}(A^\top)} \|\nabla f(x) - \psi\|_1^2.$$

Before presenting the proof, it can be sketched quite easily. Using the Fenchel-Young inequality (cf. Proposition 32) and the form of the dual optimization problem (cf. Theorem 4), primal suboptimality can be converted into a Bregman divergence in the dual. If there is strong convexity in the primal, it allows this Bregman divergence to be converted into a distance via standard tools in convex optimization (cf. Lemma 34). Although f lacks strong convexity in general, it is strongly convex over any compact set.

Proof of Lemma 25 Consider the optimization problem

$$\inf_{x\in S}\inf_{\substack{\phi\in\mathbb{R}^m\\ \|\phi\|_2=1}}\left\langle\nabla^2 f(x)\phi,\phi\right\rangle=\inf_{x\in S}\inf_{\substack{\phi\in\mathbb{R}^m\\ \|\phi\|_2=1}}\sum_{i=1}^m g''(x_i)\phi_i^2;$$

since S is compact and g'' and $(\cdot)^2$ are continuous, the infimum is attainable. But $g''>0$ and $\phi\neq\mathbf{0}_m$, meaning the infimum c is nonzero, and moreover it is the modulus of strong convexity of f over S (Hiriart-Urruty and Lemaréchal, 2001, Theorem B.4.3.1.iii).

Now let any $x\in S\cap\mathrm{Im}(A)$ be given, define $D=\nabla f(S)\subset\mathbb{R}^m_+$, and for convenience set $K:=\mathrm{Ker}(A^\top)$. Consider the dual element $\mathsf{P}^2_{D\cap K}(\nabla f(x))$ (which exists since $D\cap K\neq\emptyset$); due to the projection, it is dual feasible, and thus it must follow from Theorem 4 that

$$\bar{f}_A=\sup\{-f^*(\psi):\psi\in\Phi_A\}\geq -f^*\left(\mathsf{P}^2_{D\cap K}(\nabla f(x))\right).$$

Furthermore, since $x\in\mathrm{Im}(A)$,

$$\left\langle x,\mathsf{P}^2_{D\cap K}(\nabla f(x))\right\rangle=0.$$

Combined with the Fenchel-Young inequality (cf. Proposition 32) and $x=\nabla f^*(\nabla f(x))$,

$$\begin{align}
f(x)-\bar{f}_A &\leq f(x)+f^*\left(\mathsf{P}^2_{D\cap K}(\nabla f(x))\right) \notag\\
&= f^*\left(\mathsf{P}^2_{D\cap K}(\nabla f(x))\right)+\langle\nabla f(x),x\rangle-f^*(\nabla f(x)) \notag\\
&= f^*\left(\mathsf{P}^2_{D\cap K}(\nabla f(x))\right)-f^*(\nabla f(x))-\left\langle\nabla f^*(\nabla f(x)),\mathsf{P}^2_{D\cap K}(\nabla f(x))-\nabla f(x)\right\rangle \tag{20}\\
&\leq\frac{1}{2c}\|\nabla f(x)-\mathsf{P}^2_{D\cap K}(\nabla f(x))\|_2^2, \tag{21}
\end{align}$$

where the last step follows by an application of Lemma 34, noting that both $\nabla f(x)$ and $\mathsf{P}^2_{D\cap K}(\nabla f(x))$ are in $\nabla f(S)=D$, and f is strongly convex with modulus c over S. To finish, rewrite P as an infimum and use $\|\cdot\|_2\leq\|\cdot\|_1$. ■

The desired result now follows readily.

Proof of Theorem 23 Invoking Lemma 24 with $d=f(A\lambda_0)$ immediately provides a compact tightest axis-aligned rectangle $\mathcal{C}$ containing the initial level set $S:=\{x\in\mathbb{R}^m:(f+\iota_{\mathrm{Im}(A)})(x)\leq f(A\lambda_0)\}$. Crucially, since the objective values never increase, S and $\mathcal{C}$ contain every iterate $\{A\lambda_t\}_{t=1}^\infty$.

Applying Lemma 25 to the set $\mathcal{C}$ (by Lemma 24, $\nabla f(\mathcal{C})\cap\mathrm{Ker}(A^\top)\neq\emptyset$), then for any t,

$$f(A\lambda_t)-\bar{f}_A\leq\frac{1}{2c}\|\nabla f(A\lambda_t)-\mathsf{P}^1_{\nabla f(\mathcal{C})\cap\mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))\|_1^2,$$

where $c>0$ is the modulus of strong convexity of f over $\mathcal{C}$.

Finally, if there are suboptimal iterates, then $\nabla f(\mathcal{C})\supseteq\nabla f(S)$ contains points that are not dual feasible, meaning $\nabla f(\mathcal{C})\setminus\mathrm{Ker}(A^\top)\neq\emptyset$; since Lemma 24 also provided $\nabla f(\mathcal{C})\cap\Phi_A\neq\emptyset$ and $\nabla f(\mathcal{C})$

is a hypercube, it follows by Theorem 9 that $\gamma(A, \nabla f(\mathcal{C})) > 0$. Plugging this into Proposition 20 and using $f(A\lambda_t) \leq f(A\lambda_0)$ gives

$$\begin{aligned} f(A\lambda_{t+1}) - \bar{f}_A &\leq f(A\lambda_t) - \bar{f}_A - \frac{\gamma(A, \nabla f(\mathcal{C}))^2 \mathsf{D}^1_{\nabla f(\mathcal{C}) \cap \operatorname{Ker}(A^\top)}(\nabla f(A\lambda_t))^2}{6\eta f(A\lambda_t)} \\ &\leq (f(A\lambda_t) - \bar{f}_A) \left(1 - \frac{c\gamma(A, \nabla f(\mathcal{C}))^2}{3\eta f(A\lambda_0)}\right), \end{aligned}$$

and the result again follows by recursively applying this inequality. ■

Remark 26 *The key conditions on $g \in \mathbb{G}$, namely the existence of constants granting $g \leq \beta g'$ and $g'' \leq \eta g$ within the initial level set, are much more than are needed in this setting. Inspecting the presented proofs, it entirely suffices that on any compact set in $\mathbb{R}^m$, f has quadratic upper and lower bounds (equivalently, bounds on the smallest and largest eigenvalues of the Hessian), which are precisely the weaker conditions used in previous treatments (Bickel et al., 2006; Rätsch et al., 2001).*

These quantities are therefore necessary for controlling convergence under weak learnability. To see how the proofs of this section break down in that setting, consider the central Bregman divergence expression in Equation 20. What is really granted by attainability is that every iterate lies well within the interior of $\operatorname{dom}(f^*)$*, and therefore these Bregman divergences, which depend on ∇f^*, can not become too wild. On the other hand, with weak learnability, all dual weights go to zero (cf. Theorem 11), which means that $\nabla g^* \uparrow \infty$, and thus the upper bound in Equation 21 ceases to be valid. As such, another mechanism is required to control this scenario, which is precisely the role of $g \leq \beta g'$ and $g'' \leq \eta g$.*

6.3 General Setting

The key development of Section 5.3 was that general instances may be decomposed uniquely into two smaller pieces, one satisfying attainability and the other satisfying weak learnability, and that these smaller problems behave somewhat independently. This independence is leveraged here to produce convergence rates relying upon the existing rate analysis for the attainable and weak learnable cases. The mechanism of the proof is as straightforward as one could hope for: decompose the dual distance into the two pieces, handle them separately using preceding results, and then stitch them back together.

Theorem 27 *Suppose $g \in \mathbb{G}$ and $1 \leq |H(A)| \leq m-1$. Recall from Section 5.3 the partition of the rows of A into $A_0 \in \mathbb{R}^{m_0 \times n}$ and $A_+ \in \mathbb{R}^{m_+ \times n}$, and suppose the axes of $\mathbb{R}^m$ are ordered so that $A = \left[\begin{smallmatrix} A_0 \\ A_+ \end{smallmatrix}\right]$. Set $\mathcal{C}_+$ to be the tightest axis-aligned rectangle $\mathcal{C}_+ \supseteq \{x \in \mathbb{R}^{m_+} : (f + \iota_{\operatorname{Im}(A_+)})(x) \leq f(A\lambda_0)\}$, and $w := \sup_t \|\nabla f(A_+\lambda_t) - \mathsf{P}^1_{\nabla f(\mathcal{C}_+) \cap \operatorname{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t))\|_1$. Then $\mathcal{C}_+$ is compact, $w < \infty$, f has modulus of strong convexity $c > 0$ over $\mathcal{C}_+$, and $\gamma(A, \mathbb{R}^{m_0} \times \nabla f(\mathcal{C}_+)) > 0$. Using these terms, for all t,*

$$f(A\lambda_t) - \bar{f}_A \leq \frac{2f(A\lambda_0)}{(t+1)\min\{1, \gamma(A, \mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))^2/(3\eta(\beta + w/(2c))^2)\}}.$$

The new term, w, appears when stitching together the two subproblems. For choices of $g \in \mathbb{G}$ where $\mathrm{dom}(g^*)$ is a compact set, this value is easy to bound; for instance, the logistic loss, where $\mathrm{dom}(g^*) = [0,1]$, has $w \leq \sup_{\phi \in \mathrm{dom}(f^*)} \|\phi - \mathbf{0}_m\|_1 = m$ (since $\mathbf{0}_m \in \mathrm{dom}(f^*)$). And with the exponential loss, taking $S := \{\lambda \in \mathbb{R}^n : f(A\lambda) \leq f(A\lambda_0)\}$ to denote the initial level set, since $\mathbf{0}_m$ is always dual feasible,

$$w \leq \sup_{\lambda \in S} \|\nabla f(A\lambda)\|_1 = \sup_{\lambda \in S} f(A\lambda) = f(A\lambda_0) = m.$$

Note that rearranging the rate from Theorem 27 will provide that $O(1/\varepsilon)$ iterations suffice to reach suboptimality ε, whereas the earlier scenarios needed only $O(\ln(1/\varepsilon))$ iterations. The exact location of the degradation will be pinpointed after the proof, and is related to the introduction of w.

Proof of Theorem 27 By Theorem 17, $\bar{f}_{A_+} = \bar{f}_A$, and the form of f gives $f(A\lambda_t) = f(A_0\lambda_t) + f(A_+\lambda_t)$, thus

$$f(A\lambda_t) - \bar{f}_A = f(A_0\lambda_t) + f(A_+\lambda_t) - \bar{f}_{A_+}. \tag{22}$$

For the left term, since $g(x) \leq \beta|g'(x)|$,

$$f(A_0\lambda_t) \leq \beta\|\nabla f(A_0\lambda_t)\|_1 = \beta\|\nabla f(A_0\lambda_t) - \mathsf{P}^1_{\Phi_{A_0}}(\nabla f(A_0\lambda_t))\|_1, \tag{23}$$

which used the fact (from Theorem 17) that $\Phi_{A_0} = \{\mathbf{0}_{m_0}\}$.

For the right term of Equation 22, recall from Theorem 17 that $f + \iota_{\mathrm{Im}(A_+)}$ is 0-coercive, thus the level set $S_+ := \{x \in \mathbb{R}^{m_+} : (f + \iota_{\mathrm{Im}(A_+)})(x) \leq f(A\lambda_0)\}$ is compact. For all t, since $f \geq 0$ and the objective values never increase,

$$f(A\lambda_0) \geq f(A\lambda_t) = f(A_0\lambda_t) + f(A_+\lambda_t) \geq f(A_+\lambda_t);$$

in particular, $A_+\lambda_t \in S_+$. It is crucial that the level set compares against $f(A\lambda_0)$ and not $f(A_+\lambda_0)$.

Continuing, Lemma 24 may be applied to A_+ with value $d = f(A\lambda_0)$, which grants a tightest axis-aligned rectangle $\mathcal{C}_+ \subseteq \mathbb{R}^{m_+}$ containing S_+, and moreover $\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A_+^\top) \neq \emptyset$. Applying Lemma 25 to A_+ and $\mathcal{C}_+$, f is strongly convex with modulus $c > 0$ over $\mathcal{C}_+$, and for any t,

$$f(A_+\lambda_t) - \bar{f}_{A_+} \leq \frac{1}{2c}\|\nabla f(A_+\lambda_t) - \mathsf{P}^1_{\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t))\|_1^2. \tag{24}$$

Next, set $w := \sup_t \|\nabla f(A_+\lambda_t) - \mathsf{P}^1_{\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A^\top)}(\nabla f(A_+\lambda_t))\|_1$; $w < \infty$ since S_+ is compact and $\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A^\top)$ is nonempty. By the definition of w,

$$\mathsf{D}^1_{\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t))^2 \leq w\mathsf{D}^1_{\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t)),$$

which combined with Equation 24 yields

$$f(A_+\lambda_t) - \bar{f}_{A_+} \leq \frac{w}{2c}\mathsf{D}^1_{\nabla f(\mathcal{C}_+) \cap \mathrm{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t)). \tag{25}$$

To merge the subproblem dual distance upper bounds Equation 23 and Equation 25 via Lemma 47, it must be shown that $(\mathbb{R}_+^{m_0} \times \nabla f(\mathcal{C}_+)) \cap \Phi_A \neq \emptyset$. But this follows by construction and Theorem 17, since $\{\mathbf{0}_m\} = \Phi_{A_0} \subseteq \mathbb{R}^m_+, \nabla f(\mathcal{C}_+) \cap \Phi_{A_+} \neq \emptyset$ by Lemma 24, and the decomposition $\Phi_A = \Phi_{A_0} \times \Phi_{A_+}$. Returning to the total suboptimality expression Equation 22, these dual distance bounds yield

$$\begin{aligned} f(A\lambda_t) - \bar{f}_A &\leq \beta \mathsf{D}^1_{\Phi_{A_0}}(\nabla f(A_0\lambda_t)) + w/(2c)\mathsf{D}^1_{\nabla f(\mathcal{C}_+)\cap \mathrm{Ker}(A_+^\top)}(\nabla f(A_+\lambda_t)) \\ &\leq (\beta + w/(2c))\mathsf{D}^1_{(\mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))\cap \mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t)), \end{aligned}$$

the second step using Lemma 47.

To finish, note $\mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+)$ is polyhedral, and

$$(\mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+)) \setminus \mathrm{Ker}(A^\top) \quad \supseteq \quad \{\nabla f(A\lambda_t)\}_{t=1}^\infty \setminus \mathrm{Ker}(A^\top) \quad \neq \quad \emptyset$$

since no primal iterate is optimal and thus $\nabla f(A\lambda_t)$ is not dual feasible by optimality conditions; combined with the above derivation $(\mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+)) \cap \Phi_A \neq \emptyset$, Theorem 9 may be applied, meaning $\gamma(A, \mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+)) > 0$. As such, all conditions of Proposition 20 are met, and making use of $f(A\lambda_t) \leq f(A\lambda_0)$,

$$\begin{aligned} f(A\lambda_{t+1}) - \bar{f}_A &\leq f(A\lambda_t) - \bar{f}_A - \frac{\gamma(A, \mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))^2 \mathsf{D}^1_{(\mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))\cap \mathrm{Ker}(A^\top)}(\nabla f(A\lambda_t))^2}{6\eta f(A\lambda_t)} \\ &\leq f(A\lambda_t) - \bar{f}_A - \frac{\gamma(A, \mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))^2 (f(A\lambda_t) - \bar{f}_A)^2}{6\eta f(A\lambda_0)(\beta + w/(2c))^2}. \end{aligned}$$

Applying Lemma 33 with

$$\epsilon_t := \frac{f(A\lambda_t) - \bar{f}_A}{f(A\lambda_0)} \qquad \text{and} \qquad r := \frac{1}{2}\min\left\{1, \frac{\gamma(A, \mathbb{R}^{m_0}_+ \times \nabla f(\mathcal{C}_+))^2}{3\eta(\beta + w/(2c))^2}\right\}$$

gives the result. ■

In order to produce a rate $O(\ln(1/\epsilon))$ under attainability, strong convexity related the suboptimality to a *squared* dual distance $\|\cdot\|_1^2$ (cf. Equation 21). On the other hand, the rate $O(\ln(1/\epsilon))$ under weak learnability came from a fortuitous cancellation with the denominator $f(A\lambda_t)$ (cf. Equation 19), which is equal to the total suboptimality since Theorem 11 provides $\bar{f}_A = 0$. But in order to merge the subproblem dual distances via Lemma 47, the differing properties granting fast rates must be ignored. (In the case of attainability, this process introduces w.)

This incompatibility is not merely an artifact of the analysis. Intuitively, the finite and infinite margins sought by the two pieces A_0, A_+ are in conflict. For a beautifully simple, concrete case of this, consider the following matrix, due to Schapire (2010):

$$S := \begin{bmatrix} -1 & +1 \\ +1 & -1 \\ -1 & -1 \end{bmatrix}.$$

The optimal solution here is to push both coordinates of λ unboundedly positive, with margins approaching $(0, 0, \infty)$. But pushing any coordinate $(\lambda)_i$ too quickly will increase the objective value, rather than decreasing it. In fact, this instance will provide a lower bound, and the mechanism of the proof shows that the primal weights grow extremely slowly, as $O(\ln(t))$.

Theorem 28 *Fix* $g = \ln(1 + \exp(\cdot)) \in \mathbb{G}$*, the logistic loss, and suppose the line search is exact. Then for any* $t \geq 1$*,* $f(S\lambda_t) - \bar{f}_S \geq 1/(8t)$.

(The proof, in Section G.6, is by brute force.)

Finally, note that this third setting does not always entail slow convergence. Again taking the view of the rows of S being points $\{-s_i\}_1^3$, consider the effect of rotating the entire instance around the origin by $\pi/4$. The optimization scenario is unchanged, however coordinate descent can now be arbitrarily close to the optimum in one iteration by pushing a single primal weight extremely high.

Acknowledgments

The author thanks his advisor, Sanjoy Dasgupta, for valuable discussions and support throughout this project; Daniel Hsu for many discussions, and for introducing the author to the problem; Indraneel Mukherjee and Robert Schapire for discussions, and for sharing their related work; Robert Schapire for sharing early drafts of his book with Yoav Freund; the JMLR reviewers and editor for their extremely careful reading and comments. This work was supported by the NSF under grants IIS-0713540 and IIS-0812598.

Appendix A. Common Notation

Symbol	Comment
$\mathbb{R}^m$	m-dimensional vector space over the reals.
$\mathbb{R}^m_+$	Non-negative m-dimensional real vectors.
$\text{int}(S)$	The interior of set S.
$\mathbb{R}^m_{++}$	Positive m-dimensional real vectors, that is, $\text{int}(\mathbb{R}^m_+)$.
$\mathbb{R}^m_-, \mathbb{R}^m_{--}$	Respectively $-\mathbb{R}^m_+, -\mathbb{R}^m_{++}$.
$\mathbf{0}_m, \mathbf{1}_m$	m-dimensional vectors of all zeros and all ones, respectively.
$\mathbf{e}_i$	Indicator vector: 1 at coordinate i, 0 elsewhere. Context will provide the ambient dimension.
$\text{Im}(A)$	Image of linear operator A.
$\text{Ker}(A)$	Kernel of linear operator A.
ι_S	Indicator function on a set S: $$\iota_S(x) := \begin{cases} 0 & x \in S, \\ \infty & x \notin S. \end{cases}$$
$\text{dom}(h)$	Domain of convex function h, that is, the set $\{x \in \mathbb{R}^m : h(x) < \infty\}$.
h^*	The Fenchel conjugate of h: $$h^*(\phi) = \sup_{x \in \text{dom}(h)} \langle \phi, x \rangle - h(x).$$ (Cf. Section 3 and Section B.2.)
0-coercive	A convex function with all level sets compact is called 0-coercive (cf. Section 5.2).
$\mathbb{G}_0$	Basic loss family under consideration (cf. Section 2).
$\mathbb{G}$	Refined loss family for which convergence rates are established (cf. Section 6).
η, β	Parameters corresponding to some $g \in \mathbb{G}$ (cf. Section 6).
Φ_A	The general dual feasibility set: $\Phi_A := \text{Ker}(A^\top) \cap \mathbb{R}^m_+$ (cf. Section 3).
$\gamma(A, S)$	Generalization of classical weak learning rate (cf. Section 4).
$\bar{f}_A$	The minimal objective value of $f \circ A$: $\bar{f}_A := \inf_\lambda f(A\lambda)$ (cf. Section 2).
ψ^f_A	Dual optimum (cf. Section 3).
P^p_S	l^p projection onto closed nonempty convex set S, with ties broken in some consistent manner (cf. Section 4).
D^p_S	l^p distance to closed nonempty convex set S: $\mathsf{D}^p_S(\phi) := \Vert\phi - \mathsf{P}^p_S(\phi)\Vert_p$.

Appendix B. Supporting Results from Convex Analysis, Optimization, and Linear Programming

This appendix collects various supporting results from the literature.

B.1 Theorems of the Alternative

Theorems of the alternative consider the interplay between a matrix (or a few matrices) and its transpose; they are typically stated as two alternative scenarios, exactly one of which must hold. These results usually appear in connection with linear programming, where Farkas's lemma is used to certify (or not) the existence of solutions. In the present manuscript, they are used to establish the relationship between $\mathrm{Im}(A)$ and $\mathrm{Ker}(A^\top)$, appearing as the first and fourth clauses of the various characterization theorems in Section 5.

The first such theorem, used in the setting of weak learnability, is perhaps the oldest theorem of alternatives (Dantzig and Thapa, 2003, Bibliographic Notes, Section 5 of Chapter 2). Interestingly, a streamlined presentation, using a related optimization problem (which can nearly be written as $f \circ A$ from this manuscript), can be found in Borwein and Lewis (2000, Theorem 2.2.6).

Theorem 29 (Gordan, Borwein and Lewis, 2000, Theorem 2.2.1) *For any $A \in \mathbb{R}^{m\times n}$, exactly one of the following situations holds:*

$$\exists \lambda \in \mathbb{R}^n \centerdot A\lambda \in \mathbb{R}^m_{--};$$
$$\exists \phi \in \mathbb{R}^m_+ \setminus \{\mathbf{0}_m\} \centerdot A^\top \phi = \mathbf{0}_n.$$

A geometric interpretation is as follows. Take the rows of A to be m points in $\mathbb{R}^n$. Then there are two possibilities: either there exists an open homogeneous halfspace containing all points, or their convex hull contains the origin.

Next is Stiemke's Theorem of the Alternative, used in connection with attainability.

Theorem 30 (Stiemke, Borwein and Lewis, 2000, Exercise 2.2.8) *For any $A \in \mathbb{R}^{m\times n}$, exactly one of the following situations holds:*

$$\exists \lambda \in \mathbb{R}^n \centerdot A\lambda \in \mathbb{R}^m_{-} \setminus \{\mathbf{0}_m\};$$
$$\exists \phi \in \mathbb{R}^m_{++} \centerdot A^\top \phi = \mathbf{0}_n.$$

The geometric interpretation here is that either there exists a closed homogeneous halfspace containing all m points, with at least one point interior to the halfspace, or the relative interior of the convex hull of the points contains the origin (for the connection to relative interiors, see for instance Hiriart-Urruty and Lemaréchal 2001, Remark A.2.1.4).

Finally, a version of Motzkin's Transposition Theorem, which can encode the theorems of alternatives due to Farkas, Stiemke, and Gordan (Ben-Israel, 2002).

Theorem 31 (Motzkin, Dantzig and Thapa, 2003, Theorem 2.16) *For any $B \in \mathbb{R}^{z\times n}$ and $C \in \mathbb{R}^{p\times n}$, exactly one of the following situations holds:*

$$\exists \lambda \in \mathbb{R}^n \centerdot B\lambda \in \mathbb{R}^z_{--} \wedge C\lambda \in \mathbb{R}^p_{-},$$
$$\exists \phi_B \in \mathbb{R}^z_+ \setminus \{\mathbf{0}_z\}, \phi_C \in \mathbb{R}^p_+ \centerdot B^\top \phi_B + C^\top \phi_C = \mathbf{0}_n.$$

For this geometric interpretation, take any matrix $A \in \mathbb{R}^{m\times n}$, broken into two submatrices $B \in \mathbb{R}^{z\times n}$ and $C \in \mathbb{R}^{p\times n}$, with $z + p = m$; again, consider the rows of A as m points in $\mathbb{R}^n$. The first possibility is that there exists a closed homogeneous halfspace containing all m points, the z points corresponding to B being interior to the halfspace. Otherwise, the origin can be written as a convex combination of these m points, with positive weight on at least one element of B.

B.2 Fenchel Conjugacy

The Fenchel conjugate of a function h, defined in Section 3, is

$$h^*(\phi) = \sup_{x \in \mathrm{dom}(h)} \langle x, \phi \rangle - h(x),$$

where $\mathrm{dom}(h) = \{x : h(x) < \infty\}$. The main property of the conjugate, indeed what motivated its definition, is that $\nabla h^*(\nabla h(x)) = x$ (Hiriart-Urruty and Lemaréchal, 2001, Corollary E.1.4.4). To demystify this, differentiate and set to zero the contents of the above sup: the Fenchel conjugate acts as an inverse gradient map. For a beautiful description of Fenchel conjugacy, please see Hiriart-Urruty and Lemaréchal (2001, Section E.1.2).

Another crucial property of Fenchel conjugates is the Fenchel-Young inequality, simplified here for differentiability (the "if" can be strengthened to "iff" via subgradients).

Proposition 32 (Fenchel-Young, Borwein and Lewis, 2000, Proposition 3.3.4) *For any convex function h and $x \in \mathrm{dom}(h)$, $\phi \in \mathrm{dom}(h^*)$,*

$$h(x) + h^*(\phi) \geq \langle x, \phi \rangle,$$

with equality if $\phi = \nabla h(x)$.

B.3 Convex Optimization

Two standard results from convex optimization will help produce convergence rates; note that these results can be found in many sources.

First, a lemma to convert single-step convergence results into general convergence results.

Lemma 33 (Lemma 20 from Shalev-Shwartz and Singer 2008) *Let $1 \geq \varepsilon_1 \geq \varepsilon_2 \geq \ldots$ be given with $\varepsilon_{t+1} \leq \varepsilon_t - r\varepsilon_t^2$ for some $r \in (0, 1/2]$. Then $\varepsilon_t \leq (r(t+1))^{-1}$.*

Although strong convexity in the primal grants the existence of a lower bounding quadratic, it grants upper bounds in the dual. The following result is also standard in convex analysis, see for instance Hiriart-Urruty and Lemaréchal (2001, proof of Theorem E.4.2.2).

Lemma 34 (Lemma 18 from Shalev-Shwartz and Singer 2008) *Let h be strongly convex over compact convex set S with modulus c. Then for any $\phi_1, \phi_1 + \phi_2 \in \nabla h(S)$,*

$$h^*(\phi_1 + \phi_2) - h^*(\phi_1) \leq \langle \nabla h^*(\phi_1), \phi_2 \rangle + \frac{1}{2c} \|\phi_2\|_2^2.$$

Appendix C. Basic Properties of $g \in \mathbb{G}_0$

Lemma 35 *Let any $g \in \mathbb{G}_0$ be given. Then g is strictly convex, $g > 0$, g strictly increases ($g' > 0$), and g' strictly increases. Lastly, $\lim_{x \to \infty} g(x) = \infty$.*

Proof (Strict convexity and g' strictly increases.) For any $x < y$,

$$g'(y) = g'(x) + \int_x^y g''(t)dt \geq g'(x) + (y - x) \inf_{t \in [x,y]} g''(t) > g'(x),$$

thus g' strictly increases, granting strict convexity (Hiriart-Urruty and Lemaréchal, 2001, Theorem B.4.1.4).

(g strictly increases, that is, $g' > 0$.) Suppose there exists y with $g'(y) \leq 0$, and choose any $x < y$. Since g' strictly increases, $g'(x) < 0$. But that means

$$\lim_{z \to -\infty} g(z) \geq \lim_{z \to -\infty} g(x) + (z - x)g'(x) = \infty,$$

a contradiction.

($g > 0$.) If there existed y with $g(y) \leq 0$, then the strict increasing property would invalidate $\lim_{x \to -\infty} g(x) = 0$.

($\lim_{x \to \infty} g(x) = \infty$.) Let any sequence $\{c_i\}_1^\infty \uparrow \infty$ be given; the result follows by convexity and $g' > 0$, since

$$\lim_{i \to \infty} g(c_i) \geq \lim_{i \to \infty} g(c_1) + g'(c_1)(c_i - c_1) = \infty.$$

■

Next, a deferred proof regarding properties of g^* for $g \in \mathbb{G}_0$.

Proof of Lemma 2 g^* is strictly convex because g is differentiable, and g^* is continuously differentiable on $\mathrm{int}(\mathrm{dom}(g^*))$ because g is strictly convex (Hiriart-Urruty and Lemaréchal, 2001, Theorems E.4.1.1, E.4.1.2).

Next, when $\phi < 0$: $\lim_{x \to -\infty} g(x) = 0$ grants the existence of y such that for any $x \leq y$, $g(x) \leq 1$, thus

$$g^*(\phi) = \sup_x \phi x - g(x) \geq \sup_{x \leq y} \phi x - 1 = \infty.$$

($g > 0$ precludes the possibility of $\infty - \infty$.)

Take $\phi = 0$; then

$$g^*(\phi) = \sup_x -g(x) = -\inf_x g(x) = 0.$$

When $\phi = g'(0)$, by the Fenchel-Young inequality (Proposition 32),

$$g^*(\phi) = g^*(g'(0)) = 0 \cdot g'(0) - g(0) = -g(0).$$

Moreover $\nabla g^*(g'(0)) = 0$ (Hiriart-Urruty and Lemaréchal, 2001, Corollary E.1.4.3), which combined with strict convexity of g^* means $g'(0)$ minimizes g^*. g^* is closed (Hiriart-Urruty and Lemaréchal, 2001, Theorem E.1.1.2), which combined with the above gives that $\mathrm{dom}(g^*) = [0, \infty)$ or $\mathrm{dom}(g^*) = [0, b]$ for some $b > 0$, and the rest of the form of g^*. ■

Finally, properties of the empirical risk function f and its conjugate f^*.

Lemma 36 *Let any $g \in \mathbb{G}_0$ be given. Then the corresponding f is strictly convex, twice continuously differentiable, and $\nabla f > \mathbf{0}_m$. Furthermore, $\mathrm{dom}(f^*) = \mathrm{dom}(g^*)^m \subseteq \mathbb{R}^m_+$, $f^*(\mathbf{0}_m) = 0$, f^* is strictly convex, f^* is continuously differentiable on the interior of its domain, and finally $f^*(\phi) = \sum_{i=1}^m g^*(\phi_i)$.*

Proof First,

$$f^*(\phi) = \sup_{x\in\mathbb{R}^m} \langle \phi, x\rangle - f(x) = \sup_{x\in\mathbb{R}^m} \sum_{i=1}^m x_i\phi_i - g(x_i) = \sum_{i=1}^m g^*(\phi_i).$$

Next, strict convexity of g^* (cf. Lemma 2) means, for $x \neq y$, $\langle \nabla g^*(x) - \nabla g^*(y), x-y\rangle > 0$ (Hiriart-Urruty and Lemaréchal, 2001, Theorem E.4.1.4); thus, given $\phi_1, \phi_2 \in \mathbb{R}^m$ with $\phi_1 \neq \phi_2$, strict convexity of f^* follows from

$$\langle \nabla f^*(\phi_1) - \nabla f^*(\phi_2), \phi_1 - \phi_2\rangle = \sum_{i=1}^m \langle \nabla g^*((\phi_1)_i) - \nabla g^*((\phi_2)_i), (\phi_1)_i - (\phi_2)_i\rangle > 0.$$

The remaining properties follow from properties of g and g^* (cf. Lemma 35 and Lemma 2). ■

Appendix D. Approximate Line Search

This section provides two approximate line search methods for BOOST: an iterative approach, outlined in Section D.1 and analyzed in Section D.2, and a closed form choice, outlined in Section D.3.

The iterative approach follows standard line search principles from nonlinear optimization (Bertsekas, 1999; Nocedal and Wright, 2006). It requires no parameters, only the ability to evaluate objective values and their gradients, and as such is perhaps of greater practical interest. Due to this, and the fact that its guarantee is just a constant factor worse than the closed form method, all convergence analysis will use this choice.

The closed form step size is provided for the sake of comparison to other choices from the boosting literature. The drawback, as mentioned above, is the need to know certain parameters, specifically a second derivative bound, which may be loose.

Before proceeding, note briefly that this section is the only place where boundedness of the entries of A is used. Without this assumption, the second derivative upper bounds would contain the term $\max_{i,j} A_{ij}^2$, which in turn would appear in the various convergence rates of Section 6.

D.1 The Wolfe Conditions

Consider any convex differentiable function h, a current iterate x, and a descent direction v (that is, $\nabla h(x)^\top v < 0$). By convexity, the linearization of h at x in direction v, symbolically $h(x) + \alpha\nabla h(x)^\top v$, will lie below the function. But, by continuity, it must be the case that, for any $c_1 \in (0,1)$, the ray $h(x) + \alpha c_1 \nabla h(x)^\top v$, depicted in Figure 8, must lie above h for some small region around x; this gives the first Wolfe condition, also known as the Armijo condition (cf. Nocedal and Wright 2006, Equation 3.4 and Bertsekas 1999, Exercise 1.2.16):

$$h(x+\alpha v) \leq h(x) + \alpha c_1 \nabla h(x)^\top v. \tag{26}$$

Unfortunately, this rule may grant only very limited decrease in objective value, since $\alpha > 0$ can be chosen arbitrarily small and still satisfy the rule; thus, the second Wolfe condition, also called a curvature condition, which depends on $c_2 \in (c_1, 1)$, forces the step to be farther away:

$$\nabla h(x+\alpha v)^\top v \geq c_2 \nabla h(x)^\top v. \tag{27}$$

```
Routine WOLFE.
Input Convex function $h$, iterate $x$, descent direction $v$.
Output Step size $\alpha > 0$ satisfying Equation 26 and Equation 27.

  1. Bracketing step.
     (a) Set $\alpha_{\max} := 1$.
     (b) While $\alpha_{\max}$ satisfies Equation 26:
         • Set $\alpha_{\max} := 2\alpha_{\max}$.
  2. Bisection step.
     (a) Set $\alpha_{\min} := 0$ and $\alpha := \alpha_{\max}/2$.
     (b) While $\alpha$ does not satisfy Equation 26 and Equation 27:
           i. If $\alpha$ violates Equation 26:
              • Set $\alpha_{\max} := \alpha$.
          ii. Else, $\alpha$ must violate Equation 27:
              • Set $\alpha_{\min} := \alpha$.
         iii. Set $\alpha := (\alpha_{\min} + \alpha_{\max})/2$.
     (c) Return $\alpha$.
```

Figure 7: Bracketing and bisecting search for step size satisfying Wolfe conditions.

This requires the new gradient (in direction v) to be closer to 0, mimicking first order optimality conditions for the exact line search. Note that the new gradient (in direction v) may in fact be positive; this does not affect the analysis.

In the case of boosting, with function $f \circ A$, current iterate λ_t, direction $v_{t+1} \in \{\pm \mathbf{e}_{j_{t+1}}\}$ satisfying $\nabla(f \circ A)(\lambda_t)^\top v_{t+1} = -\|\nabla(f \circ A)(\lambda_t)\|_\infty$, these conditions become

$$(f \circ A)(\lambda_t + \alpha v_{t+1}) \leq (f \circ A)(\lambda_t) - \alpha c_1 \|\nabla(f \circ A)(\lambda_t)\|_\infty, \tag{28}$$

$$\nabla(f \circ A)(\lambda_t + \alpha v_{t+1})^\top v_{t+1} \geq -c_2 \|\nabla(f \circ A)(\lambda_t)\|_\infty. \tag{29}$$

An algorithm to find a point satisfying these conditions, presented in Figure 7, is simple enough: grow α as quickly as possible, and then bisect backwards for a satisfactory point. As compared with the presentation in Nocedal and Wright (2006, Algorithm 3.5), $\alpha_{\max}$ is searched for rather than provided, and convexity removes the need for interpolation.

Proposition 37 *Given a continuously differentiable convex bounded below function h, iterate x, and direction v,* WOLFE *terminates with an $\alpha > 0$ satisfying Equation 26 and Equation 27.*

Proof The bracketing search must terminate: v is a descent direction, so the linearization at λ_{t-1} with slope $c_1 \nabla h(x)^\top v$ will eventually intersect h (since h it is bounded below).

The remainder of this proof is illustrated in Figure 8. Let α_1 be the greatest positive real satisfying Equation 26; due to convexity, every $\alpha \geq 0$ satisfying this first condition must also satisfy $\alpha \in [0, \alpha_1]$. Crucially, $\alpha_1 < \alpha_{\max}$.

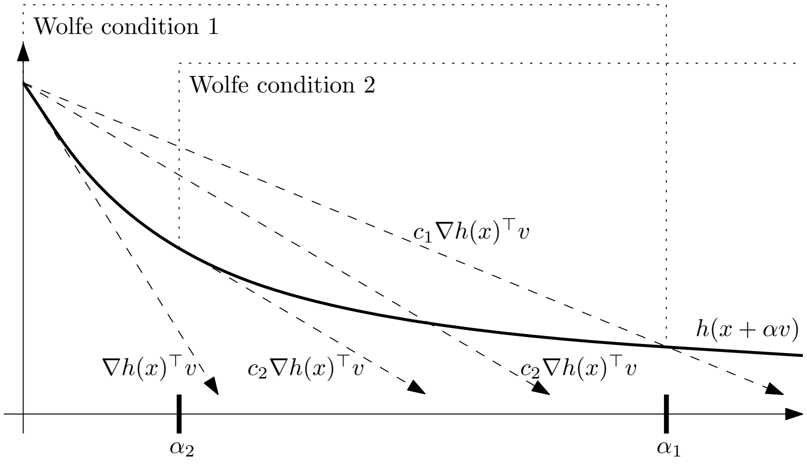


Figure 8: The mechanism behind WOLFE: the set of points satisfying Equation 26 and Equation 27 is a closed interval, and bisection will find interior points. In this figure, dashed lines denote various relevant slopes.

Next, let α_2 be the smallest positive real satisfying Equation 27; existence of such a point follows from the existence of points satisfying both Wolfe conditions (Nocedal and Wright, 2006, Lemma 3.1). By convexity,

$$\langle \nabla h(x+\alpha v) - \nabla h(x), v\rangle \geq 0,$$

and therefore every $\alpha \geq 0$ satisfying Equation 27 must satisfy $\alpha \geq \alpha_2$.

Finally, $\alpha_1 \neq \alpha_2$, since $c_1 < c_2$, meaning

$$\nabla h(x+\alpha_1 v)^\top v = c_2 \nabla h(x)^\top v < c_1 \nabla h(x)^\top v < \nabla h(x+\alpha_2 v)^\top v.$$

Combining these facts, the interval $[\alpha_2, \alpha_1]$ is precisely the set of points which satisfy Equation 28 and Equation 27. The bisection search maintains the invariants $\alpha_{\min} \leq \alpha_2$ and $\alpha_{\max} \geq \alpha_1$, meaning no valid solution is ever thrown out: $[\alpha_2, \alpha_1] \subseteq [\alpha_{\min}, \alpha_{\max}]$. $[\alpha_2, \alpha_1]$ has nonzero width (since $\alpha_1 \neq \alpha_2$), and every bisection step halves the width of $[\alpha_{\min}, \alpha_{\max}]$, thus the procedure terminates. ■

D.2 Improvement Guaranteed by WOLFE Search

The following proof, adapted from Nocedal and Wright (2006, Lemma 3.1), provides the improvement gained by a single line search step. The usual proof depends on a Lipschitz parameter on the gradient, which is furnished here by $g''(x) \leq \eta g(x)$.

Proposition 38 (See Nocedal and Wright 2006, Lemma 3.1) *Fix any $g \in \mathbb{G}$. If α_{t+1} is chosen by* WOLFE *applied to function $f \circ A$ at iterate λ_t in direction v_{t+1} with $c_1 = 1/3$ and $c_2 = 1/2$, then*

$$f(A(\lambda_t + \alpha_{t+1} v_{t+1})) \leq f(A\lambda_t) - \frac{\|A^\top \nabla f(A\lambda_t)\|_\infty^2}{6\eta f(A\lambda_t)}.$$

Proof First note that every $\alpha \in [0, \alpha_{t+1}]$ satisfies

$$f(A(\lambda_t + \alpha v_{t+1})) \leq f(A\lambda_t).$$

By the fundamental theorem of calculus,

$$\begin{aligned}
(\nabla(f \circ A)(\lambda_t + \alpha_{t+1} v_{t+1}) - \nabla(f \circ A)(\lambda_t))^\top v_{t+1} \\
&= \int_0^{\alpha_{t+1}} v_{t+1}^\top \nabla^2 (f \circ A)(\lambda_t + \alpha v_{t+1}) v_{t+1} d\alpha \\
&\leq \alpha_{t+1} \sup_{\alpha \in [0, \alpha_{t+1}]} \sum_{i=1}^m g''(\mathbf{e}_i^\top A(\lambda_t + \alpha v_{t+1}))(A_{ij_{t+1}})^2 \\
&\leq \eta \alpha_{t+1} \sup_{\alpha \in [0, \alpha_{t+1}]} \sum_{i=1}^m g(\mathbf{e}_i^\top A(\lambda_t + \alpha v_{t+1})) \\
&\leq \eta \alpha_{t+1} f(A\lambda_t),
\end{aligned}$$

which used boundedness of the entries in A.

The rest of the proof continues as in Nocedal and Wright (2006, Theorem 3.2). Specifically, subtracting $\nabla(f \circ A)(\lambda_t)^\top v_{t+1}$ from both sides of Equation 29 yields

$$(\nabla(f \circ A)(\lambda_t + \alpha_{t+1} v_{t+1}) - \nabla(f \circ A)(\lambda_t))^\top v_{t+1} \geq (c_2 - 1)\nabla(f \circ A)(\lambda_t)^\top v_{t+1}.$$

Combining these two gives

$$\alpha_{t+1} \geq \frac{(c_2 - 1)\nabla(f \circ A)(\lambda_t)^\top v_{t+1}}{\eta f(A\lambda_t)} = \frac{(1 - c_2)\|\nabla(f \circ A)(\lambda_t)\|_\infty}{\eta f(A\lambda_t)}.$$

Plugging this into Equation 28 yields

$$(f \circ A)(\lambda_t + \alpha_{t+1} v_{t+1}) \leq (f \circ A)(\lambda_t) - \frac{c_1(1 - c_2)\|\nabla(f \circ A)(\lambda_t)\|_\infty^2}{\eta f(A\lambda_t)}.$$

■

Note briefly that the simpler iterative strategy of backtracking line search is doomed to require knowledge of the sorts of parameters appearing in the closed form choice.

D.3 Non-iterative Step Selection

The same techniques from the proof of Proposition 38 can provide a closed form choice of α_t. In particular, it follows that any $\alpha \in \{\alpha \geq 0 : f(A\lambda_t) \geq f(A(\lambda_t + \alpha v_{t+1}))\}$ is upper bounded by the quadratic

$$f(A(\lambda_t + \alpha v_{t+1})) \leq f(A\lambda_t) - \alpha \|A^\top \nabla f(A\lambda_t)\|_\infty + \frac{\alpha^2 \eta f(A\lambda_t)}{2}.$$

This quadratic is minimized at

$$\alpha' := \frac{\|A^\top \nabla f(A\lambda_t)\|_\infty}{\eta f(A\lambda_t)};$$

moreover, this minimum is attained within the interval above, which in particular implies

$$f(A(\lambda_t + \alpha' v_{t+1})) \leq f(A\lambda_t) - \frac{\|A^\top \nabla f(A\lambda_t)\|_\infty^2}{2\eta f(A\lambda_t)}.$$

When η is simple and tight, this yields a pleasing expression (for instance, $\eta = 1$ when $g = \exp(\cdot)$). In general, however, η might be hard to calculate, or simply very loose, in which case performing a line search like WOLFE is preferable.

Appendix E. Approximate Coordinate Selection

Selecting a coordinate j_t translates into selecting some hypothesis $h_t \in \mathcal{H}$; this is in fact a key strength of boosting, since A need not be written down, and a weak learning oracle can select $h_t \in \mathcal{H}$. But for certain hypothesis classes $\mathcal{H}$, it may be impossible to guarantee h_t is truly the best choice.

Observe how these statements translate into gradient descent. Specifically, the choice v_{t+1} made by boosting satisfies

$$v_{t+1}^\top \nabla(f \circ A)(\lambda_t) = v_{t+1}^\top A^\top \nabla f(A\lambda_t) = -\|A^\top \nabla f(A\lambda_t)\|_\infty.$$

On the other hand, the usual choice $v = -\nabla(f \circ A)(\lambda_t)/\|A^\top \nabla f(A\lambda_t)\|_2$ of gradient descent (l^2 steepest descent) grants

$$v^\top \nabla(f \circ A)(\lambda_t) = -\|A^\top \nabla f(A\lambda_t)\|_2;$$

note that this choice of v is potentially a dense vector.

Remark 39 *Suppose the relaxed condition that the weak learner need merely have any correlation over the provided distribution; in optimization terms, the returned direction v satisfies*

$$v^\top \nabla(f \circ A)(\lambda_t) < 0.$$

This choice is not sufficient to guarantee convergence, let alone any reasonable convergence rate. As an example boosting instance, consider either of the matrices

$$A_1 := \begin{bmatrix} -1 & +1 & 0 \\ +1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \qquad A_2 := \begin{bmatrix} -1 & +1 & -1 \\ +1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix},$$

the first of which uses confidence-rated predictors, the second of which is weak learnable; note that both instances embed the matrix S due to Schapire (2010), used for lower bounds in Section 6.3.

For either instance, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1, \ldots$ *is a sequence of descent directions. But, for either matrix, to approach optimality, the weight on the third column must go to infinity.*

A first candidate fix is to choose some appropriate $c_0 > 0$, and require

$$v^\top \nabla(f \circ A)(\lambda_t) \leq -c_0 \|\nabla f(A\lambda_t)\|_1;$$

but note, by Theorem 7 and Theorem 11, that this is only possible under weak learnability. (Dropping the term $\|\nabla f(A\lambda_t)\|_1$ also fails; suppose A grants a minimizer $\bar{\lambda}$: plugging this in makes the left hand side exactly zero, and continuity thus grants arbitrarily small values.)

Instead consider requiring the weak learning oracle to return some hypothesis at least a fraction $c_0 \in (0,1]$ as good as the best weak learner in the class; written in the present framework, the direction v must satisfy

$$v^\top \nabla(f \circ A)(\lambda_t) \leq -c_0 \|A^\top \nabla f(A\lambda_t)\|_\infty.$$

Inspecting the proof of Proposition 20, it follows that this approximate selection would simply introduce the constant c_0^2 in all rates, but would not degrade their asymptotic relationship to suboptimality ε.

Appendix F. Generalizing the Weak Learning Rate

This appendix develops the generalization $\gamma(A,S)$ of the classical weak learning rate.

F.1 Choosing a Generalization to γ

Any generalization γ' of γ should satisfy the following properties.

- When weak learnability holds, $\gamma' = \gamma$.
- For any boosting instance, $\gamma' \in (0,\infty)$.
- γ' provides an expression similar to Equation 5, which allows the full gradient to be converted into a notion of suboptimality in the dual.

Taking the form of the classical weak learning rate from Equation 3 as a model, the template generalized weak learning rate is

$$\gamma'(A,S,C,D) := \inf_{\phi \in S \setminus C} \frac{\|A^\top \phi\|_\infty}{\inf_{\psi \in S \cap D} \|\phi - \psi\|_1},$$

for some sets S, C, and D (for instance, the classical weak learning rate uses $S = \mathbb{R}^m_+$ and $C = D = \{\mathbf{0}_m\}$). In order to provide an expression similar to Equation 5, the domain of the infimum must include every suboptimal dual iterate $\nabla f(A\lambda_t)$.

Any choice C which does not include all of $\operatorname{Ker}(A^\top)$ is immediately problematic: this allows $\phi \in S \cap \operatorname{Ker}(A^\top)$ to be selected, whereby $A^\top \phi = \mathbf{0}_m$ and $\gamma' = 0$. But note that without being careful about D, it is still possible to force the value 0.

Remark 40 *Another generalization is to define*

$$\gamma''(A) := \gamma'(A, \mathbb{R}^m_+, \operatorname{Ker}(A^\top), \{\psi^f_A\}) = \inf_{\phi \in \mathbb{R}^m_+ \setminus \Phi_A} \frac{\|A^\top \phi\|_\infty}{\|\phi - \psi^f_A\|_1}.$$

This form agrees with the original γ when weak learnability holds, and will lead to a very convenient analog to Equation 5.

Unfortunately, γ'' may be zero. Specifically, take the matrix S defined in Section 6.3, due to Schapire (2010), where

$$\psi^f_S = g'(0) \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Furthermore, for any $\alpha \in (0,1)$*, define*

$$\phi_a := \alpha \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \in \operatorname{Im}(S); \qquad\qquad \psi_\alpha := (1-\alpha)\begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix} + \psi_S^f \in \operatorname{Ker}(S^\top).$$

Then

$$\inf_{\phi \in \mathbb{R}^m_+ \setminus \operatorname{Ker}(S^\top)} \frac{\|S^\top \phi\|_\infty}{\|\phi - \psi_S^f\|_1} \le \inf_{\alpha\in(0,1)} \frac{\|S^\top(\phi_\alpha + \psi_\alpha)\|_\infty}{\|\phi_\alpha + \psi_\alpha - \psi_S^f\|_1} = \inf_{\alpha\in(0,1)} \frac{\|\left[\begin{smallmatrix} -\alpha \\ -\alpha \end{smallmatrix}\right]\|_\infty}{1} = 0.$$

The natural correction to these worries is to set $C = D = \operatorname{Ker}(A^\top)$. But there is still sensitivity due to S.

Remark 41 *Set* $A := \mathbf{1}_2$*, meaning* $\operatorname{Ker}(A^\top) = \{z(1,-1) : z \in \mathbb{R}\}$*, and* $S = B(\mathbf{1}_2, \sqrt{2})$*, the ball of radius* $\sqrt{2}$ *around* $\mathbf{1}_2$*; note that* $S \cap \operatorname{Ker}(A^\top) = \mathbf{0}_2$*. Consider* $\gamma'(A, S, \operatorname{Ker}(A^\top), \operatorname{Ker}(A^\top))$*, and the sequence* $\{\phi_i\}_{i=1}^\infty$ *where*

$$\phi_i = \mathbf{1}_2 - \frac{1}{\sqrt{i^2+1}} \begin{bmatrix} i+1 \\ i-1 \end{bmatrix}.$$

Note that $\|\phi_i - \mathbf{1}_2\|_2 = \sqrt{2}$*, thus* $\phi_i \in S$*. Furthermore,* $A^\top \phi_i \neq 0$*, so* $\phi_i \notin S \cap \operatorname{Ker}(A^\top)$*. As such,*

$$\begin{aligned} \gamma'(A, S, \operatorname{Ker}(A^\top), \operatorname{Ker}(A^\top)) &\le \inf_i \frac{\|A^\top \phi_i\|_\infty}{\|\phi_i - \mathsf{P}^1_{S\cap\operatorname{Ker}(A^\top)}(\phi_i)\|_1} \\ &= \frac{\|\mathbf{1}_2^\top \left(\mathbf{1}_2\sqrt{i^2+1} - \left[\begin{smallmatrix} i+1 \\ i-1 \end{smallmatrix}\right]\right)\|_\infty}{\|\mathbf{1}_2\sqrt{i^2+1} - \left[\begin{smallmatrix} i+1 \\ i-1 \end{smallmatrix}\right]\|_1}. \end{aligned} \tag{30}$$

Using $\sqrt{y} \le (1+y)/2$*, the numerator has upper bound*

$$\begin{aligned} \|\mathbf{1}_2^\top \left(\mathbf{1}_2\sqrt{i^2+1} - \left[\begin{smallmatrix} i+1 \\ i-1 \end{smallmatrix}\right]\right)\|_\infty &= |2\sqrt{i^2+1} - 2i| \\ &= 2i(\sqrt{1+i^{-2}} - 1) \\ &\le 2i((2+i^{-2})/2 - 1) = 1/i. \end{aligned}$$

The denominator is

$$\begin{aligned} \|\mathbf{1}_2\sqrt{i^2+1} - \left[\begin{smallmatrix} i+1 \\ i-1 \end{smallmatrix}\right]\|_1 &= |\sqrt{i^2+1} - (i+1)| + |\sqrt{i^2+1} - (i-1)| \\ &= ((i+1) - \sqrt{i^2+1}) + (\sqrt{i^2+1} - (i-1)) \\ &= 2. \end{aligned}$$

Thus Equation 30 is bounded above by $\inf_i (2i)^{-1} = 0$.

The difficulty here was the curvature of S, which allowed elements arbitrarily close to $\operatorname{Ker}(A^\top)$ without actually being inside this subspace. This possibility is averted in this manuscript by requiring polyhedrality of S. This choice is sufficiently rich to allow the various dual-distance upper bounds of Section 6.

F.2 Proof of Theorem 9

The proof of Theorem 9 requires a few steps, but the strategy is straightforward. First note that $\gamma(A,S)$ can be rewritten as

$$\begin{aligned}
\gamma(A,S) &= \inf_{\phi\in S\setminus\mathrm{Ker}(A^\top)} \frac{\|A^\top\phi\|_\infty}{\|\phi-\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)\|_1} \\
&= \inf_{\phi\in S\setminus\mathrm{Ker}(A^\top)} \frac{\|A^\top(\phi-\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi))\|_\infty}{\|\phi-\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)\|_1} \\
&= \inf\left\{\frac{\|A^\top v\|_\infty}{\|v\|_1} : v\in\mathbb{R}^m\setminus\{\mathbf{0}_m\}, \exists\phi\in S \centerdot v=\phi-\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)\right\} \\
&= \inf\left\{\|A^\top v\|_\infty : \|v\|_1=1, \exists\phi\in S, \exists c>0 \centerdot cv=\phi-\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)\right\}, \qquad (31)
\end{aligned}$$

where the second equivalence used $A^\top\mathsf{P}^1_{S\cap\mathrm{Ker}(A^\top)}(\phi)=\mathbf{0}_n$.

In the final form, $v\notin\mathrm{Ker}(A^\top)$, and so $A^\top v\neq\mathbf{0}_n$; that is to say, the infimand is positive for every element of its domain. The difficulty is that the domain of the infimum, written in this way, is not obviously closed; thus one can not simply assert the infimum is attainable and positive.

The goal then will be to reparameterize the infimum to have a compact domain. For technical convenience, the result will be mainly proved for the l^2 norm (where projections behave nicely), and norm equivalence will provide the final result.

Lemma 42 *Given $A\in\mathbb{R}^{m\times n}$ and a polyhedron $S\subseteq\mathbb{R}^m$ with $S\cap\mathrm{Ker}(A^\top)\neq\emptyset$ and $S\setminus\mathrm{Ker}(A^\top)\neq\emptyset$,*

$$\inf\left\{\frac{\|A^\top(\phi-\mathsf{P}^2_{S\cap\mathrm{Ker}(A^\top)}(\phi))\|_2}{\|\phi-\mathsf{P}^2_{S\cap\mathrm{Ker}(A^\top)}(\phi)\|_2} : \phi\in S\setminus\mathrm{Ker}(A^\top)\right\}>0. \qquad (32)$$

To produce the desired reparameterization of this infimum, the following characterization of polyhedral sets will be used.

Definition 43 *For any nonempty polyhedral set $S\subseteq\mathbb{R}^m$, let $\mathbb{H}_S$ index a finite (but possibly empty) collection of affine functions $g_\alpha:\mathbb{R}^m\to\mathbb{R}$ so that $S=\cap_{\alpha\in\mathbb{H}_S}\{x\in\mathbb{R}^m : g_\alpha(x)\leq 0\}$ (with the convention that $S=\mathbb{R}^m$ when $\mathbb{H}_S=\emptyset$). For any $x\in S$, let $I_S(x)$ denote the* active set *for x: $\alpha\in I_S(x)$ iff $g_\alpha(x)=0$. Lastly, define a relation $\sim_S$ over points in S: given $x,y\in S$, $x\sim_S y$ iff $I_S(x)=I_S(y)$. Observe that $\sim_S$ is an equivalence relation over points within S, and let $\mathcal{C}_S$ be the set of equivalence classes.*

The equivalence relation $\sim_S$ thus partitions S into the members of $\mathcal{C}_S$, each of which has a very convenient structure.

Lemma 44 *Let a polyhedral set $S\subseteq\mathbb{R}^m$ be given, and fix a nonempty $F\in\mathcal{C}_S$. Then F is convex, and F is equal to its relative interior (i.e., $F=\mathrm{ri}(F)$). Finally, fixing an arbitrary $z_0\in F$, the normal cone at any point $z\in F$ is orthogonal to the vector space parallel to the affine hull of F (i.e., $N_F(z)=(\mathrm{aff}(F)-\{z\})^\perp=(\mathrm{aff}(F)-\{z_0\})^\perp$).*

Throughout the remainder of this section, normal and tangent cones will be considered at points within a set $F \in \mathcal{C}_S$. As Lemma 44 establishes, any set $F \in \mathcal{C}_S$ is *relatively open* ($F = \text{ri}(F)$), however, the required properties of normal and tangent cones, as developed by Hiriart-Urruty and Lemaréchal (2001, Sections A.5.2 and A.5.3), suppose *closed* convex sets. But it is always the case that $\text{ri}(F) = \text{ri}(\text{cl}(F))$ (Hiriart-Urruty and Lemaréchal, 2001, Proposition A.2.1.8); as such, the normal and tangent cones at the desired relative interior points may just as well be constructed against $\text{cl}(F)$, and thus the aforementioned properties safely hold.

Proof If $S = \mathbb{R}^m$ (meaning $\mathbb{H}_S$ is empty) or $\dim(F) = 0$ (F is a single point), everything follows directly, thus suppose $S \neq \mathbb{R}^m$, and fix a nonempty $F \in \mathcal{C}_S$ with $\dim(F) > 0$.

Let any $x_0, x_1 \in F$ and $\beta \in [0,1]$ be given, and define $x_\beta := (1-\beta)x_0 + \beta x_1$. Since each g_α defining S is affine,

$$g_\alpha(x_\beta) = (1-\beta)g_\alpha(x_0) + \beta g_\alpha(x_1). \tag{33}$$

By construction of $\mathcal{C}_S$, $g_\alpha(x_0) = 0$ iff $g_\alpha(x_1) = 0$ and otherwise both are negative, thus $g_\alpha(x_\beta) = 0$ iff $g_\alpha(x_0) = g_\alpha(x_1) = 0$, meaning $I_S(x_\beta) = I_S(x_0) = I_S(x_1)$, so $x_\beta \in F$ and F is convex.

Now let any $y_0 \in F$ be given; $y_0 \in \text{ri}(F)$ when there exists a $\delta > 0$ so that

$$B(y_0, \delta) \cap \text{aff}(F) \subseteq F \tag{34}$$

(Hiriart-Urruty and Lemaréchal, 2001, Definition A.2.1.1). To this end, first define δ to be half the distance to the closest hyperplane defining S which is not active for y_0:

$$\delta := \frac{1}{2} \min_{\alpha \in \mathbb{H}_S \setminus I_S(y)} \min\{\|y' - y_0\|_2 : y' \in \mathbb{R}^m, g_\alpha(y') = 0\}.$$

Since there are only finitely many such hyperplanes, and the distance to each is nonzero, $\delta > 0$. Let any $y_\beta \in B(y, \delta) \cap \text{aff}(F)$ be given; by definition of $\text{aff}(F)$, there must exist $\beta \in \mathbb{R}$ and $y_1 \in F$ so that $y_\beta = (1-\beta)y_0 + \beta y_1$. By Equation 33, for any $\alpha \in I_S(y_0) = I_S(y_1)$,

$$g_\alpha(y_\beta) = (1-\beta)g_\alpha(y_0) + \beta g_\alpha(y_1) = 0.$$

On the other hand, for any $\alpha \in \mathbb{H}_S \setminus I_S(y_0)$, it must be the case that $g_\alpha(y_\beta) < 0$, since $y_\beta \in B(y_0, \delta)$, and due to the choice of δ. Returning to the definition of relative interior in Equation 34, it follows that $y_0 \in \text{ri}(F)$, and $\text{ri}(F) = F$ since $y_0 \in F$ was arbitrary.

For the final property, for any $z_0, z \in \text{ri}(F) = F$, the tangent cone $T_F(z)$ has form $(\text{aff}(F) - \{z\})$ (Hiriart-Urruty and Lemaréchal, 2001, see Proposition A.5.2.1 and discussion within Section A.5.3), and note $\text{aff}(F) - \{z\} = \text{aff}(F) + \{z_0 - z\} - \{z_0\} = \text{aff}(F) - \{z_0\}$. Lastly, $N_F(z) = T_F(z)^\perp$ (Hiriart-Urruty and Lemaréchal, 2001, Proposition A.5.2.4). ■

The relevance to Equation 32 and Equation 31 is that projections from polyhedron S onto $S \cap \text{Ker}(A^\top)$ (itself a polyhedron, as is verified in the proof of Lemma 42) must land on some equivalence class of $\mathcal{C}_{S \cap \text{Ker}(A^\top)}$, and these projections are easily characterized.

Lemma 45 *Let any nonempty polyhedra $S \subseteq \mathbb{R}^m$ and $K \subseteq \mathbb{R}^m$ be given, and fix any nonempty $F \in \mathcal{C}_{S \cap K}$ and $x_F \in F$. Define*

$$P_F := \{c(\phi - \mathsf{P}^2_{S \cap K}(\phi)) : c > 0, \phi \in S, \mathsf{P}^2_{S \cap K}(\phi) \in F\},$$
$$D_F := N_F(x_F) \cap \{y - x_F : y \in \mathbb{R}^m, \forall \alpha \in I_S(x_F) \centerdot g_\alpha(y) \leq 0\},$$

where $N_F(x_F)$ is the normal cone of F at x_F. Then $P_F = D_F$.

Note that the final active set $I_S(x_F)$ is with respect to S, not $S \cap K$.

Proof ($\subseteq$) Let any $\phi \in S$ with $\psi := \mathsf{P}^2_{S\cap K}(\phi) \in F$ be given, where the latter is well-defined since F and hence $S \cap K$ are nonempty. By Lemma 44, $\psi \in \mathrm{ri}(F)$, and $N_F(\psi) = N_F(x_F)$, meaning $\phi - \psi \in N_F(x_F)$ (Hiriart-Urruty and Lemaréchal, 2001, Proposition A.5.3.3). Since $\phi \in S$, for any $\alpha \in I_S(\psi) = I_S(x_F) \subseteq \mathbb{H}_S$, $g_\alpha(\phi) \leq 0$, so

$$\begin{aligned}\phi - \psi \in \{y \in \mathbb{R}^m : g_\alpha(y) \leq 0\} - \{\psi\} &= \big(\{y \in \mathbb{R}^m : g_\alpha(y) \leq 0\} - \{\psi - x_F\}\big) - \{x_F\}\\ &= \{y \in \mathbb{R}^m : g_\alpha(y) \leq 0\} - \{x_F\},\end{aligned}$$

the final equality following since $g_\alpha(x_F) = g_\alpha(\psi) = 0$ and g_α defines an affine hyperplane, meaning the corresponding affine halfspace is closed under translations by $\psi - x_F$. This holds for all $\alpha \in I_S(x_F)$, thus $\phi - \psi \in D_F$, and since D_F is a convex cone, for any $c > 0$, $c(\phi - \psi) \in D_F$.

($\supseteq$) Define

$$\delta := \min\{\|x_F - z\|_2 : \alpha \in \mathbb{H}_S \setminus I_S(x_F), z \in \mathbb{R}^m, g_\alpha(z) = 0\}.$$

For any fixed α, this minimum is positive since $g_\alpha(x_F) < 0$, while polyhedrality of S grants that α ranges over a finite set, together meaning $\delta > 0$. Now let any $v \in D_F$ be given, and set $\phi := x_F + \delta v/(2\|v\|_2)$. The form of D_F immediately grants $g_\alpha(\phi) \leq 0$ for $\alpha \in I_S(x_F)$, but notice for $\alpha \in \mathbb{H}_S \setminus I_S(x_F)$, it still holds that $g_\alpha(\phi) \leq 0$, since $g_\alpha(x_F) < 0$ and $\|\phi - x_F\|_2 < \delta$. So $v = (2\|v\|_2/\delta)(\phi - \mathsf{P}^2_{S\cap K}(\phi))$ where $\phi \in S$ and $\mathsf{P}^2_{S\cap K}(\phi) = x_F \in F$, meaning $v \in P_F$. ■

The result now follows by considering all elements of $\mathcal{C}_{S\cap \mathrm{Ker}(A^\top)}$.

Proof of Lemma 42 For convenience, set $K := \mathrm{Ker}(A^\top)$. Note that K (and hence $S \cap K$) is a polyhedron; indeed, it has the form

$$\begin{aligned}K = \mathrm{Ker}(A^\top) &= \{\phi \in \mathbb{R}^m : A^\top \phi = \mathbf{0}_n\}\\ &= \bigcap_{i=1}^{n} \Big(\{\phi \in \mathbb{R}^m : \mathbf{e}_i^\top A^\top \phi \leq 0\} \cap \{\phi \in \mathbb{R}^m : \mathbf{e}_i^\top A^\top \phi \geq 0\}\Big).\end{aligned}$$

Next, note $\mathcal{C}_{S\cap K}$ has at least one nonempty equivalence class, since $S \cap K$ is nonempty by assumption. Rewriting Equation 32 as in Equation 31, and fixing an x_F within each nonempty $F \in \mathcal{C}_{S\cap K}$, Lemma 45 grants

$$\begin{aligned}\text{Eq. 32} &= \inf\Big\{\|A^\top v\|_2 : \|v\|_2 = 1, \exists c > 0, \exists \phi \in S \centerdot \phi - \mathsf{P}^2_{S\cap K}(\phi) = cv\Big\}\\ &= \min_{\substack{F \in \mathcal{C}_{S\cap K}\\ F \neq \emptyset}} \inf\Big\{\|A^\top v\|_2 : \|v\|_2 = 1, \exists c > 0, \exists \phi \in S \centerdot \phi - \mathsf{P}^2_{S\cap K}(\phi) = cv, \mathsf{P}^2_{S\cap K}(\phi) \in F\Big\}\\ &= \min_{\substack{F \in \mathcal{C}_{S\cap K}\\ F \neq \emptyset}} \inf\Big\{\|A^\top v\|_2 : \|v\|_2 = 1, v \in N_F(x_F), \forall \alpha \in I_S(x_F) \centerdot g_\alpha(x_F + v) \leq 0\Big\}.\end{aligned}$$

Since $S \setminus \mathrm{Ker}(A^\top) \neq \emptyset$ and $S \cap \mathrm{Ker}(A^\top)$, at least one infimum has a nonempty domain (for the others, take the convention that their value is $+\infty$). Each infimum with a nonempty domain in this final expression is of a continuous function over a compact set (in fact, a polyhedral cone intersected with the boundary of the unit l^2 ball), and thus it has a minimizer $\bar{v}$, which corresponds to some $c(\bar{\phi} - \mathsf{P}^2_{S\cap K}(\bar{\phi})) \notin \mathrm{Ker}(A^\top)$, where $c > 0$. It follows that

$$A^\top \bar{v} = cA^\top(\bar{\phi} - \mathsf{P}^2_{S\cap K}(\bar{\phi})) \neq 0,$$

meaning each of these infima is positive. But since S is polyhedral, $\mathcal{C}_S$ has finitely many equivalence classes ($|\mathcal{C}_S| \leq 2^{|\mathbb{H}_S|}$), meaning the outer minimum is attained and positive. ■

Finally, as mentioned above, the desired result follows by norm equivalence.

Proof of Theorem 9 For the upper bound, note as in the proof of Lemma 42 that $S \cap \mathrm{Ker}(A^\top) \neq \emptyset$ and the infimand is positive for every element of the domain, so the infimum is finite. For the lower bound, by Lemma 42 and norm equivalence,

$$\begin{aligned}\gamma(A,S) &= \inf_{\phi \in S\setminus\mathrm{Ker}(A^\top)} \frac{\|A^\top\phi\|_\infty}{\inf_{\psi\in S\cap\mathrm{Ker}(A^\top)}\|\phi-\psi\|_1} \\ &\geq \left(\frac{1}{\sqrt{mn}}\right)\inf_{\phi \in S\setminus\mathrm{Ker}(A^\top)} \frac{\|A^\top\phi\|_2}{\inf_{\psi\in S\cap\mathrm{Ker}(A^\top)}\|\phi-\psi\|_2} > 0.\end{aligned}$$

■

Appendix G. Miscellaneous Technical Material

This appendix collects remaining technical material.

G.1 The Logistic Loss is within $\mathbb{G}$

Remark 46 *This remark develops bounds on the quantities η, β for the logistic loss $g = \ln(1+\exp(\cdot))$. First note that the initial level set $S_0 := \{x \in \mathbb{R}^m : f(x) \leq f(A\lambda_0)\}$ is contained within a cube $(-\infty, b]^m$, where $b \leq m\ln(2)$; this follows since $f(A\lambda_0) = f(\mathbf{0}_m) = m\ln(2)$, whereas $g(m\ln(2)) = \ln(1+\exp(m\ln(2))) \geq m\ln(2)$.*

For convenience, the analysis will be mainly written with respect to $b = m\ln(2)$. Let any $x \in (-\infty, b]$ be given, and note $g' = \exp(\cdot)/(1+\exp(\cdot))$, and $g'' = \exp(\cdot)/(1+\exp(\cdot))^2$.

To determine η, note $1 \leq 1+\exp(x) \leq 1+\exp(b)$. Since $\ln$ is concave, it follows for all $z \in [1, 1+\exp(b)]$ that the secant line through $(1,0)$ and $(1+\exp(b), \ln(1+\exp(b)))$ is a lower bound:

$$\ln(z) \geq \left(\frac{\ln(1+\exp(b)) - 0}{1+\exp(b)-1}\right)z - \frac{\ln(1+\exp(b))-0}{1+\exp(b)-1} = \ln(1+\exp(b))\exp(-b)(z-1).$$

As such, for $x \in (-\infty, b]$, $\ln(1+\exp(x)) \geq \exp(x)\ln(1+\exp(b))\exp(-b)$, so

$$\frac{g''(x)}{g(x)} = \frac{\exp(x)}{(1+\exp(x))^2\ln(1+\exp(x))} \leq \frac{\exp(b)}{(1+\exp(x))^2\ln(1+\exp(b))} \leq \frac{\exp(b)}{\ln(1+\exp(b))}.$$

Consequently, a sufficient choice is $\eta := \exp(b)/\ln(1+\exp(b)) \leq 2^m/(m\ln(2))$.

For $g(x) \leq \beta g'(x)$, using $\ln(x) \leq x-1$,

$$\frac{g(x)}{g'(x)} = \frac{\ln(1+\exp(x))}{\frac{\exp(x)}{1+\exp(x)}} \leq \frac{\exp(x)}{\frac{\exp(x)}{1+\exp(x)}} \leq 1+\exp(b).$$

That is, it suffices to set $\beta := 1+\exp(b) = 1+2^m$.

G.2 Proof of Theorem 4

Proof of Theorem 4 Writing the objective as two Fenchel problems,

$$\bar{f}_A = \inf_{\lambda} f(A\lambda) + \iota_{\mathbb{R}^n}(\lambda),$$
$$d := \sup_{\phi} -f^*(-\phi) - \iota^*_{\mathbb{R}^n}(A^\top \phi).$$

Since $\mathrm{cont}(f) = \mathbb{R}^m$ (set of points where f is continuous) and $\mathrm{dom}(\iota_{\mathbb{R}^n}) = \mathbb{R}^n$, it follows that $A\mathrm{dom}(\iota_{\mathbb{R}^n}) \cap \mathrm{cont}(f) = \mathrm{Im}(A) \neq \emptyset$, thus $d = \bar{f}_A$ (Borwein and Lewis, 2000, Theorem 3.3.5). Moreover, since $\bar{f}_A \leq f(\mathbf{0}_m)$ and $d \geq -f^*(\mathbf{0}_m) = 0$, the optimum is finite, and thus the same theorem grants that it is attainable in the dual.

To complete the dual problem, note for any $\lambda \in \mathbb{R}^n$ that

$$\iota^*_{\mathbb{R}^n}(\lambda) = \sup_{\mu \in \mathbb{R}^n} \langle \lambda, \mu \rangle - \iota_{\mathbb{R}^n}(\mu) = \iota_{\{\mathbf{0}_n\}}(\lambda).$$

From this, the term $-\iota^*_{\mathbb{R}^n}(A^\top \phi)$ allows the search in the dual to be restricted to $\phi \in \mathrm{Ker}(A^\top)$. Next, replace $\phi \in \mathrm{Ker}(A^\top)$ with $-\psi \in \mathrm{Ker}(A^\top)$, which combined with $\mathrm{dom}(f^*) \subseteq \mathbb{R}^m_+$ (from Lemma 36) means it suffices to consider $\psi \in \mathrm{Ker}(A^\top) \cap \mathbb{R}^m_+ = \Phi_A$. (Note that the negation was simply to be able to interpret feasible dual variables as nonnegative measures.)

Next, $f^*(\phi) = \sum_i g^*((\phi)_i)$ was proved in Lemma 36.

Finally, the uniqueness of ψ^f_A was established by Collins et al. (2002, Theorem 1), however a direct argument is as follows by the strict convexity of f^* (cf. Lemma 36). Specifically, if there were some other optimal $\psi' \neq \psi$, the point $(\psi + \psi')/2$ is dual feasible and has strictly larger objective value, a contradiction. ■

G.3 Proof of Proposition 13

Proof of Proposition 13 It holds in general that 0-coercivity grants attainable minima (cf. Hiriart-Urruty and Lemaréchal 2001, Proposition B.3.2.4 and Borwein and Lewis 2000, Proposition 1.1.3). Conversely, let $\bar{x}$ with $h(\bar{x}) = \inf_x h(x)$ and any direction $d \in \mathbb{R}^m$ with $\|d\|_2 = 1$ be given. To demonstrate 0-coercivity, it suffices to show

$$\lim_{t \to \infty} \frac{h(\bar{x} + td) - h(\bar{x})}{t} > 0$$

(Hiriart-Urruty and Lemaréchal, 2001, Proposition B.3.2.4.iii). To this end, first note, for any $t \in \mathbb{R}$, that convexity grants

$$h(\bar{x} + td) \geq h(\bar{x} + d) + (t - 1) \langle \nabla h(\bar{x} + d), d \rangle .$$

By strict monotonicity of gradients (Hiriart-Urruty and Lemaréchal, 2001, Section B.4.1.4) and first-order necessary conditions ($\nabla h(\bar{x}) = \mathbf{0}_m$),

$$\langle \nabla h(\bar{x} + d), d \rangle = \langle \nabla h(\bar{x} + d) - \nabla h(\bar{x}), \bar{x} + d - \bar{x} \rangle =: c > 0,$$

Combining these,

$$\lim_{t \to \infty} \frac{h(\bar{x} + td) - h(\bar{x})}{t} \geq \lim_{t \to \infty} \frac{h(\bar{x} + d) + (t-1)c - h(\bar{x})}{t} = c > 0.$$

■

G.4 Proof of Lemma 24

Proof of Lemma 24 Since $d \geq \inf_\lambda f(A\lambda)$, the level set $S_d := \{x \in \mathbb{R}^m : (f + \iota_{\text{Im}(A)})(x) \leq d\}$ is nonempty. Since $|H(A)| = m$, Theorem 14 provides $f + \iota_{\text{Im}(A)}$ is 0-coercive, meaning S_d is compact.

Now consider the rectangle $\mathcal{C}$ defined as a product of intervals $\mathcal{C} = \otimes_{i=1}^m [a_i, b_i]$, where

$$a_i := \inf\{x_i : x \in S_d\}, \qquad b_i := \sup\{x_i : x \in S_d\}.$$

By construction, $\mathcal{C} \supseteq S_d$, and furthermore any smaller axis-aligned rectangle must violate some infimum or supremum above, and so must fail to include a piece of S_d. In particular, the tightest rectangle exists, and it is $\mathcal{C}$.

Next, note that $\nabla f(x) = (g'(x_1), g'(x_2), \ldots, g'(x_m))$, thus $D = \otimes_{i=1}^m g'([a_i, b_i])$, an axis-aligned rectangle in the dual. Since g is strictly convex and $\text{dom}(g) = \mathbb{R}$, both $g'(a_i)$ and $g'(b_i)$ are within $\text{int}(\text{dom}(g^*))$ (for all i), and so $\nabla f(\mathcal{C}) \subset \text{int}(\text{dom}(f^*))$.

Finally, Proposition 13 grants that $f + \iota_{\text{Im}(A)}$ has a minimizer; thus choose any $\bar{\lambda} \in \mathbb{R}^n$ so that $f(A\bar{\lambda}) = \inf_\lambda f(A\lambda)$. By optimality conditions of Fenchel problems, $\psi_A^f = \nabla f(A\bar{\lambda})$ (cf. the optimality conditions in Borwein and Lewis (2000, Exercise 3.3.9.f), and the proof of Theorem 4, where a negation was inserted into the dual to allow dual points to be interpreted as nonnegative measures). But the dual optimum is dual feasible, and $A\bar{\lambda} \in S_d$, so

$$\nabla f(\mathcal{C}) \cap \Phi_A \supseteq \{\nabla f(A\bar{\lambda})\} \cap \Phi_A = \{\psi_A^f\} \cap \Phi_A \neq \emptyset.$$

■

G.5 Splitting Distances along A_0, A_+

Lemma 47 *Let $A = \begin{bmatrix} A_0 \\ A_+ \end{bmatrix}$ be given as in Theorem 27, and let a set $S = S_0 \times S_+$ be given with $S_0 \subseteq \mathbb{R}^{m_0}$ and $S_+ \subseteq \mathbb{R}^{m_+}$ and $S \cap \Phi_A \neq \emptyset$. Then, for any $\phi = \begin{bmatrix} \phi_0 \\ \phi_+ \end{bmatrix}$ with $\phi_0 \in \mathbb{R}^{m_0}$ and $\phi_+ \in \mathbb{R}^{m_+}$,*

$$\mathsf{D}^1_{S\cap\Phi_A}(\phi) = \mathsf{D}^1_{S_0\cap\Phi_{A_0}}(\phi_0) + \mathsf{D}^1_{S_+\cap\Phi_{A_+}}(\phi_+).$$

Proof Recall from Theorem 17 that $\Phi_A = \Phi_{A_0} \times \Phi_{A_+}$, thus

$$S \cap \Phi_A = (S_0 \cap \Phi_{A_0}) \times (S_+ \cap \Phi_{A_+}),$$

and $S \cap \Phi_A \neq \emptyset$ grants that $S_0 \cap \Phi_{A_0} \neq \emptyset$ and $S_+ \cap \Phi_{A_+} \neq \emptyset$. Define now the notation $[\cdot]_0 : \mathbb{R}^m \to \mathbb{R}^{m_0}$ and $[\cdot]_+ : \mathbb{R}^m \to \mathbb{R}^{m_+}$, which respectively select the coordinates corresponding to the rows of A_0, and the rows of A_+.

Let $\phi = \begin{bmatrix} \phi_0 \\ \phi_+ \end{bmatrix} \in \mathbb{R}^m$ be given; in the above notation, $\phi_0 = [\phi]_0$ and $\phi_+ = [\phi]_+$. By the above Cartesian product and intersection properties,

$$\begin{bmatrix} \mathsf{P}^1_{S_0\cap\Phi_{A_0}}(\phi_0) \\ \mathsf{P}^1_{S_+\cap\Phi_{A_+}}(\phi_+) \end{bmatrix} \in S \cap \Phi_A,$$

and so

$$\mathsf{D}^1_{S\cap\Phi_A}(\phi) \le \left\| \begin{bmatrix} \phi_0 \\ \phi_+ \end{bmatrix} - \begin{bmatrix} \mathsf{P}^1_{S_0\cap\Phi_{A_0}}(\phi_0) \\ \mathsf{P}^1_{S_+\cap\Phi_{A_+}}(\phi_+) \end{bmatrix} \right\|_1 = \mathsf{D}^1_{S_0\cap\Phi_{A_0}}(\phi_0) + \mathsf{D}^1_{S_+\cap\Phi_{A_+}}(\phi_+).$$

On the other hand, since $\mathsf{P}^1_{S\cap\Phi_A}(\phi) \in (S_0 \cap \phi_{A_0}) \times (S_+ \cap \phi_{A_+})$,

$$\mathsf{D}^1_{S_0\cap\Phi_{A_0}}(\phi_0) + \mathsf{D}^1_{S_+\cap\Phi_{A_+}}(\phi_+) \le \left\|\phi_0 - [\mathsf{P}^1_{S\cap\Phi_A}(\phi)]_0\right\|_1 + \left\|\phi_+ - [\mathsf{P}^1_{S\cap\Phi_A}(\phi)]_+\right\|_1 = \mathsf{D}^1_{S\cap\Phi_A}(\phi).$$

■

G.6 Proof of Theorem 28

Proof of Theorem 28 This proof proceeds in two stages: first the gap between any solution with l^1 norm B is shown to be large, and then it is shown that the l^1 norm of the BOOST solution (under logistic loss) grows slowly.

To start, $\text{Ker}(S^\top) = \{z(1,1,0) : z \in \mathbb{R}\}$, and $-g^*$ is maximized at $g'(0)$ with value $-g(0)$ (cf. Lemma 2). Thus $\psi^f_S = (g'(0), g'(0), 0)$, and $\bar{f}_S = -f^*(\psi^f_S) = 2g(0) = 2\ln(2)$.

Next, by calculus, given any B,

$$\begin{aligned}\inf_{\|\lambda\|_1 \le B} f(S\lambda) - \bar{f}_S &= f\left(S\begin{bmatrix} B/2 \\ B/2 \end{bmatrix}\right) - 2\ln(2) \\ &= (2\ln(2) + \ln(1+\exp(-B))) - 2\ln(2) \\ &= \ln(1+\exp(-B)).\end{aligned}$$

Now to bound the l^1 norm of the iterates. By the nature of exact line search, the coordinates of λ are updated in alternation (with arbitrary initial choice); thus let u_t denote the value of the coordinate updated in iteration t, and v_t be the one which is held fixed. (In particular, $v_t = u_{t-1}$.)

The objective function, written in terms of (u_t, v_t), is

$$\begin{aligned} & \ln\big(1+\exp(v_t - u_t)\big) + \ln\big(1+\exp(u_t - v_t)\big) + \ln\big(1+\exp(-u_t - v_t)\big) \\ = \; & \ln\big(2 + \exp(v_t - u_t) + \exp(u_t - v_t) + 2\exp(-u_t - v_t) + \exp(-2u_t) + \exp(-2v_t)\big).\end{aligned}$$

Due to the use of exact line search, and the fact that u_t is the new value of the updated variable, the derivative with respect to u_t of the above expression must equal zero. In particular, producing this equality and multiplying both sides by the (nonzero) denominator yields

$$-\exp(v_t - u_t) + \exp(u_t - v_t) - 2\exp(-u_t - v_t) - 2\exp(-2u_t) = 0.$$

Multiplying by $\exp(u_t + v_t)$ and rearranging, it follows that, after line search, u_t and v_t must satisfy

$$\exp(2u_t) = \exp(2v_t) + 2\exp(v_t - u_t) + 2. \tag{35}$$

First it will be shown for $t \ge 1$, by induction, that $u_t \ge v_t$. The base case follows by inspection (since $u_0 = v_0 = 0$ and so $u_1 = \ln(2)$). Now the inductive hypothesis grants $u_t \ge v_t$; the case $u_t = v_t$ can be directly handled by Equation 35, thus suppose $u_t > v_t$. But previously, it was shown that the optimal l^1 bounded choice has both coordinates equal; as such, the current iterate, with coordinates

(u_t, v_t), is worse than the iterate (u_t, u_t), and thus the line search will move in a positive direction, giving $u_{t+1} \geq v_{t+1}$.

It will now be shown by induction that, for $t \geq 1$, $u_t \leq \frac{1}{2}\ln(4t)$. The base case follows by the direct inspection above. Applying the inductive hypothesis to the update rule above, and recalling $v_{t+1} = u_t$ and that the weights increase (i.e., $u_{t+1} \geq v_{t+1} = u_t$),

$$\exp(2u_{t+1}) = \exp(2u_t) + 2\exp(u_t - u_{t+1}) + 2 \leq \exp(2u_t) + 2\exp(u_t - u_t) + 2 \leq 4t + 4 \leq 4(t+1).$$

To finish, recall by Taylor expansion that $\ln(1+q) \geq q - \frac{q^2}{2}$; consequently for $t \geq 1$

$$f(S\lambda_t) - \bar{f}_S \geq \inf_{\|\lambda\|_1 \leq \ln(4t)} f(S\lambda) - \bar{f}_S \geq \ln\left(1 + \frac{1}{4t}\right) \geq \frac{1}{4t} - \frac{1}{2}\left(\frac{1}{4t}\right)^2 \geq \frac{1}{8t}.$$

■

References

Adi Ben-Israel. Motzkin's transposition theorem, and the related theorems of Farkas, Gordan and Stiemke. In M. Hazewinkel, editor, *Encyclopaedia of Mathematics, Supplement III*. 2002.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2 edition, 1999.

Peter J. Bickel, Yaacov Ritov, and Alon Zakai. Some theory for generalized boosting algorithms. *Journal of Machine Learning Research*, 7:705–732, 2006.

Jonathan Borwein and Adrian Lewis. *Convex Analysis and Nonlinear Optimization*. Springer Publishing Company, Incorporated, 2000.

Stéphane Boucheron, Olivier Bousquet, and Gabor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Leo Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, October 1999.

Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

George B. Dantzig and Mukund N. Thapa. *Linear Programming 2: Theory and Extensions*. Springer, 2003.

Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121 (2):256–285, 1995.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer Publishing Company, Incorporated, 2001.

Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–545, 1995.

Jyrki Kivinen and Manfred K. Warmuth. Boosting as entropy projection. In *COLT*, pages 134–144, 1999.

Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72:7–35, 1992.

Llew Mason, Jonathan Baxter, Peter L. Bartlett, and Marcus R. Frean. Functional gradient techniques for combining hypotheses. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 221–246, Cambridge, MA, 2000. MIT Press.

Indraneel Mukherjee, Cynthia Rudin, and Robert Schapire. The convergence rate of AdaBoost. In *COLT*, 2011.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

Gunnar Rätsch and Manfred K. Warmuth. Maximizing the margin with boosting. In *COLT*, pages 334–350, 2002.

Gunnar Rätsch, Sebastian Mika, and Manfred K. Warmuth. On the convergence of leveraging. In *NIPS*, pages 487–494, 2001.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, July 1990.

Robert E. Schapire. The convergence rate of AdaBoost. In *COLT*, 2010.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, in preparation.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

Robert E. Schapire, Yoav Freund, Peter Barlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML*, pages 322–330, 1997.

Shai Shalev-Shwartz and Yoram Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. In *COLT*, pages 311–322, 2008.

Manfred K. Warmuth, Karen A. Glocer, and Gunnar Rätsch. Boosting algorithms for maximizing the soft margin. In *NIPS*, 2007.

Journal of Machine Learning Research 13 (2012) 607-642 Submitted 3/11; Revised 10/11; Published 3/12

Non-Sparse Multiple Kernel Fisher Discriminant Analysis

Fei Yan F.YAN@SURREY.AC.UK
Josef Kittler J.KITTLER@SURREY.AC.UK
Krystian Mikolajczyk K.MIKOLAJCZYK@SURREY.AC.UK
Atif Tahir M.TAHIR@SURREY.AC.UK
Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, Surrey, United Kingdom, GU2 7XH

Editor: Sören Sonnenburg, Francis Bach, Cheng Soon Ong

Abstract

Sparsity-inducing multiple kernel Fisher discriminant analysis (MK-FDA) has been studied in the literature. Building on recent advances in non-sparse multiple kernel learning (MKL), we propose a non-sparse version of MK-FDA, which imposes a general ℓ_p norm regularisation on the kernel weights. We formulate the associated optimisation problem as a semi-infinite program (SIP), and adapt an iterative wrapper algorithm to solve it. We then discuss, in light of latest advances in MKL optimisation techniques, several reformulations and optimisation strategies that can potentially lead to significant improvements in the efficiency and scalability of MK-FDA. We carry out extensive experiments on six datasets from various application areas, and compare closely the performance of ℓ_p MK-FDA, fixed norm MK-FDA, and several variants of SVM-based MKL (MK-SVM). Our results demonstrate that ℓ_p MK-FDA improves upon sparse MK-FDA in many practical situations. The results also show that on image categorisation problems, ℓ_p MK-FDA tends to outperform its SVM counterpart. Finally, we also discuss the connection between (MK-)FDA and (MK-)SVM, under the unified framework of regularised kernel machines.

Keywords: multiple kernel learning, kernel fisher discriminant analysis, regularised least squares, support vector machines

1. Introduction

Since their introduction in the mid-1990s, kernel methods (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) have proven successful for many machine learning problems, for example, classification, regression, dimensionality reduction, clustering. Representative methods such as support vector machine (SVM) (Vapnik, 1999; Shawe-Taylor and Cristianini, 2004), kernel Fisher discriminant analysis (kernel FDA) (Mika et al., 1999; Baudat and Anouar, 2000), kernel principal component analysis (kernel PCA) (Schölkopf et al., 1999) have been reported to produce state-of-the-art performance in numerous applications. Kernel methods work by embedding data items in an input space (vector, graph, string, etc.) into a feature space, and applying linear methods in the feature space. This embedding is defined implicitly by specifying an inner product for the feature space via a symmetric positive semidefinite (PSD) kernel function.

It is well recognised that in kernel methods, the choice of kernel function is critically important, since it completely determines the embedding of the data in the feature space. Ideally, this embedding should be learnt from training data. In practice, a relaxed version of this very challenging problem is often considered: given multiple kernels capturing different "views" of the problem, how to learn an "optimal" combination of them. Among several others (Cristianini et al., 2002; Chapelle et al., 2002; Bousquet and Herrmann, 2003; Ong et al., 2003), Lanckriet et al. (2002, 2004) are one of the pioneering works for this multiple kernel learning (MKL) problem.

Lanckriet et al. (2002, 2004) study a binary classification problem, and their key idea is to learn a linear combination of a given set of base kernels by maximising the margin between the two classes or by maximising kernel alignment. More specifically, suppose one is given n $m \times m$ symmetric PSD kernel matrices $K_j, j = 1, \cdots, n$, and m class labels $y_i \in \{1, -1\}, i = 1, \cdots, m$. A linear combination of the n kernels under an ℓ_1 norm constraint is considered:

$$K = \sum_{j=1}^{n} \beta_j K_j, \ \boldsymbol{\beta} \geq \mathbf{0}, \ \|\boldsymbol{\beta}\|_1 = 1,$$

where $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)^T \in \mathbb{R}^n$, and $\mathbf{0}$ is the m dimensional vector of zeros. Geometrically, taking the sum of kernels can be interpreted as taking the Cartesian product of the associated feature spaces. Different scalings of the feature spaces lead to different embeddings of the data in the composite feature space. The goal of MKL is then to learn the optimal scaling of the feature spaces, such that the "separability" of the two classes in the composite feature space is maximised.

Lanckriet et al. (2002, 2004) propose to use the soft margin of SVM as a measure of separability, that is, to learn $\boldsymbol{\beta}$ by maximising the soft margin between the two classes. One of the most commonly used formulations of the resulting MKL problem is the following saddle point problem:

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} -\mathbf{y}^T \boldsymbol{\alpha} + \frac{1}{2} \sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha} \\ \text{s.t.} \quad &\mathbf{1}^T \boldsymbol{\alpha} = 0, \ \mathbf{0} \leq \mathbf{y}^T \boldsymbol{\alpha} \leq C\mathbf{1}, \ \boldsymbol{\beta} \geq \mathbf{0}, \ \|\boldsymbol{\beta}\|_1 \leq 1, \end{aligned} \tag{1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$, $\mathbf{1}$ is the m dimensional vector of ones, $\mathbf{y}$ is the m dimensional vector of class labels, C is a parameter controlling the trade-off between regularisation and empirical error, and $K_j(\mathbf{x}_i, \mathbf{x}_{i'})$ is the dot product of the i^{th} and the i'^{th} training examples in the j^{th} feature space. Note that in Equation (1), we have replaced the constraint $\|\boldsymbol{\beta}\|_1 = 1$ by $\|\boldsymbol{\beta}\|_1 \leq 1$, which can be shown to have no effect on the solution of the problem, but allows for an easier generalisation.

Several alternative MKL formulations have been proposed (Lanckriet et al., 2004; Bach and Lanckriet, 2004; Sonnenburg et al., 2006; Zien and Ong, 2007; Rakotomamonjy et al., 2008). These formulations essentially solve the same problem as Equation (1), and differ only in the optimisation techniques used. The original semi-definite programming (SDP) formulation (Lanckriet et al., 2004) becomes intractable when m is in the order of thousands, while the semi-infinite linear programming (SILP) formulation (Sonnenburg et al., 2006) and the reduced gradient descent algorithm (Rakotomamonjy et al., 2008) can deal with much larger problems.

Of particular interest to this article is the SILP formulation in Sonnenburg et al. (2006). The authors propose to use a technique called column generation to solve the SILP, which involves dividing a SILP into an inner subproblem and an outer subproblem, and alternating between solving the two subproblems until convergence. A straightforward implementation of column generation leads to a conceptually very simple wrapper algorithm, where finding the optimal $\boldsymbol{\alpha}$ in the inner subproblem corresponds to solving a standard binary SVM. This means the wrapper algorithm can take advantage of existing efficient SVM solvers, and can be reasonably fast for medium-sized problems already. However, as pointed out by Sonnenburg et al. (2006), solving the whole SVM problem to a high precision is unnecessary and therefore wasteful when the variable $\boldsymbol{\beta}$ in the outer subproblem is still far from the global optimum.

To remedy this, Sonnenburg et al. (2006) propose to optimise $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in an interleaved manner, by incorporating chunking (Joachims, 1988) into the inner subproblem. The key idea of chunking, and more generally decomposition techniques for SVM, is to freeze all but a small subset of $\boldsymbol{\alpha}$, and solve only a small-sized subproblems of the SVM dual in each iteration. The resulting interleaved algorithm in Sonnenburg et al. (2006) avoids the wasteful computation of the whole SVM dual, and as a result has an improved efficiency over the wrapper algorithm. Moreover, with the interleaved algorithm, only columns of the kernel matrices that correspond to the "active" dual variables need to be loaded into memory, extending MKL's applicability to large scale problems.

The learning problem in Equation (1) imposes an ℓ_1 regularisation on the kernel weights. It has been known that ℓ_1 norm regularisation tends to produce sparse solutions (Rätsch, 2001), which means during the learning most kernels are assigned zero weights. Conventionally, sparsity is favoured mainly for two reasons: it offers a better interpretability, and the test process is more efficient with sparse kernel weights. However, sparsity is not always desirable, since the information carried in the zero-weighted kernels is lost. In Kloft et al. (2008) and Cortes et al. (2009), non-sparse versions of MKL are proposed, where an ℓ_2 norm regularisation is imposed instead of ℓ_1 norm. Kloft et al. (2009, 2011) later extended their work to use a general ℓ_p ($p \geq 1$) norm regularisation. To solve the associated optimisation problem, Kloft et al. (2011) propose extensions of the wrapper and the interleaved algorithms in Sonnenburg et al. (2006) respectively. Experiments in Kloft et al. (2008, 2009, 2011) show that the regularisation norm contributes significantly to the performance of MKL, and confirm that in general a smaller regularisation norm produces more sparse kernel weights.

Although many of the above references discuss general loss functions (Lanckriet et al., 2004; Sonnenburg et al., 2006; Kloft et al., 2011), they have mainly been focusing on the binary hinge loss. In this sense, the corresponding MKL algorithms are essentially binary multiple kernel support vector machines (MK-SVMs). In contrast to SVM, which maximises the soft margin, Fisher discriminant analysis (FDA) (Fisher, 1936) maximises the ratio of projected between and within class scatters. Since its introduction in the 1930s, FDA has stood the test of time. Equipped recently with kernelisation (Mika et al., 1999; Baudat and Anouar, 2000) and efficient implementation (Cai et al., 2007), FDA has established itself as a strong competitor of SVM. In many comparative studies, FDA is reported to offer comparable or even better performance than SVM (Mika, 2002; Cai et al., 2007; Ye et al., 2008).

In Kim et al. (2006) and Ye et al. (2008), a multiple kernel FDA (MK-FDA) is introduced, where an ℓ_1 norm is used to regularise the kernel weights. As in the case of ℓ_1 MK-SVM, ℓ_1 MK-FDA tends to produce sparse selection results, which may lead to a loss of information. In this paper, we extend the work of Kim et al. (2006) and Ye et al. (2008) to a general ℓ_p norm regularisation by bringing latest advances in non-sparse MKL to MK-FDA. Our contribution can be summarised as follows:

- We provide a SIP formulation of ℓ_p MK-FDA for both binary and multiclass problems, and adapt the wrapper algorithm in Sonnenburg et al. (2006) to solve it. By considering recent advances in large scale MKL techniques, we also discuss several strategies that could significantly improve the efficiency and scalability of the wrapper-based ℓ_p MK-FDA. (Section 2)

- We carry out extensive experiments on six datasets, including one synthetic dataset, four object and image categorisation benchmarks, and one computational biology dataset. We confirm that as in the case of ℓ_p MK-SVM, in ℓ_p MK-FDA, a smaller regularisation norm in general leads to more sparse kernel weights. We also show that by selecting the regularisation norm p on an independent validation set, the "intrinsic sparsity" of the given set of base kernels can be learnt. As a result, using the learnt optimal norm p in ℓ_p MK-FDA offers better performance than fixed norm MK-FDAs. (Section 3)

- We compare closely the performance of ℓ_p MK-FDA and that of several variants of ℓ_p MK-SVM, and show that on object and image categorisation datasets, ℓ_p MK-FDA has a small but consistent edge. In terms of efficiency, our wrapper-based ℓ_p MK-FDA is comparable to the interleaved ℓ_p MK-SVM on small/medium sized binary problems, but can be significantly faster on multiclass problems. When compared against recently proposed MKL techniques that define the state-of-the-art, such as SMO-MKL (Vishwanathan et al., 2010) and OBSCURE (Orabona et al., 2010), our MK-FDA also compares favourably or similarly. (Section 3)

- Finally, we discuss the connection between (MK-)FDA and (MK-)SVM, from the perspectives of both loss function and version space, under the unified framework of regularised kernel machines. (Section 4)

Essentially, our work builds on Sonnenburg et al. (2006), Ye et al. (2008) and Kloft et al. (2011). However, we believe the empirical findings of this paper, especially the one that (MK-)FDA tends to outperform (MK-)SVM on image categorisation datasets, is important, given that SVM and SVM based MKL are widely accepted as the state-of-the-art classifier in most image categorisation systems. Finally, note that preliminary work to this article has been published previously as conference papers (Yan et al., 2009b,a, 2010). The aim of this article is to consolidate the results into an integrated and comprehensive account and to provide more experimental results in support of the proposed methodology.

2. ℓ_p Norm Multiple Kernel FDA

In this section we first present our ℓ_p regularised MK-FDA for binary problems and then for multiclass problems. In both cases, we first give problem formulation, then solve the associated optimisation problem using a wrapper algorithm. Towards the end of this section, we also discuss several possible improvements over the wrapper algorithm in terms of time and memory complexity, in light of recent advances in MKL optimisation techniques.

2.1 Binary Classification

Given a binary classification problem with m training examples, our goal is to learn the optimal kernel weights $\boldsymbol{\beta} \in \mathbb{R}^n$ for a linear combination of n base kernels under the ℓ_p ($p \geq 1$) constraint:

$$K = \sum_{j=1}^{n} \beta_j K_j, \;\; \beta_j \geq 0, \;\; \|\boldsymbol{\beta}\|_p^p \leq 1,$$

where the $p \geq 1$ requirement is to ensure that the triangle inequality is satisfied and $\|\cdot\|_p$ is a norm. We define optimality in terms of the class separation criterion of FDA, that is, the learnt kernel weights $\boldsymbol{\beta}$ are optimal, if the ratio of the projected between and within class scatters is maximised. In this paper we assume each kernel is centred in its feature space. Centring can be performed implicitly (Schölkopf et al., 1999) by $K_j = P\tilde{K}_jP$, where P is the $m \times m$ centring matrix defined as $P = I - \frac{1}{m}\mathbf{1}\cdot\mathbf{1}^T$, $\tilde{K}_j$ is the uncentred kernel matrix, and I is the $m \times m$ identity matrix.

Let m^+ be the number of positive training examples, and $m^- = m - m^+$ be that of negative training examples. For a given kernel K, let $\phi(\mathbf{x}_i^+)$ be the i^{th} positive training point in the implicit feature space associated with K, $\phi(\mathbf{x}_i^-)$ be the i^{th} negative training point in the feature space. Here $\mathbf{x}_i^+$ and $\mathbf{x}_i^-$ can be thought of as training examples in some input space, and ϕ is the mapping to the feature space. Also let $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$ be the centroids of the positive examples and negative examples in the feature space, respectively:

$$\boldsymbol{\mu}^+ = \frac{1}{m^+}\sum_{i=1}^{m^+}\phi(\mathbf{x}_i^+), \quad \boldsymbol{\mu}^- = \frac{1}{m^-}\sum_{i=1}^{m^-}\phi(\mathbf{x}_i^-).$$

The within class covariance matrices of the two classes are:

$$C^+ = \frac{1}{m^+}\sum_{i=1}^{m^+}\left(\phi(\mathbf{x}_i^+) - \boldsymbol{\mu}^+\right)\left(\phi(\mathbf{x}_i^+) - \boldsymbol{\mu}^+\right)^T,$$
$$C^- = \frac{1}{m^-}\sum_{i=1}^{m^-}\left(\phi(\mathbf{x}_i^-) - \boldsymbol{\mu}^-\right)\left(\phi(\mathbf{x}_i^-) - \boldsymbol{\mu}^-\right)^T.$$

The between class scatter S_B and within class scatter S_w are then defined as:

$$\begin{aligned} S_B =& \frac{m^+m^-}{m}(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)^T, \\ S_W =& \, m^+C^+ + m^-C^-. \end{aligned} \tag{2}$$

The objective of single kernel FDA is to find the projection direction $\mathbf{w}$ in the feature space that maximises $\frac{\mathbf{w}^TS_B\mathbf{w}}{\mathbf{w}^TS_W\mathbf{w}}$, or equivalently, $\frac{\mathbf{w}^T\frac{m}{m^+m^-}S_B\mathbf{w}}{\mathbf{w}^TS_T\mathbf{w}}$, where $S_T = S_B + S_W$ is the total scatter matrix. In practice a regularised objective function

$$J_{FDA}(\mathbf{w}) = \frac{\mathbf{w}^T\frac{m}{m^+m^-}S_B\mathbf{w}}{\mathbf{w}^T(S_T + \lambda I)\mathbf{w}} \tag{3}$$

is maximised to improve generalisation and numerical stability (Mika, 2002), where λ is a small positive number.

From Theorem 2.1 of Ye et al. (2008), for a given kernel K, the maximal value of Equation (3) is:

$$J_{FDA}^* = \mathbf{a}^T\mathbf{a} - \mathbf{a}^T\left(I + \frac{1}{\lambda}K\right)^{-1}\mathbf{a}, \tag{4}$$

where

$$\mathbf{a} = \left(\frac{1}{m^+}, \cdots, \frac{1}{m^+}, \frac{-1}{m^-}, \cdots, \frac{-1}{m^-}\right)^T \in \mathbb{R}^m$$

contains the centred labels. On the other hand, Lemma 2.1 of Ye et al. (2008) states that the $\mathbf{w}$ that maximises Equation (3) also minimises the following regularised least squares (RLS):

$$J_{RLS}(\mathbf{w}) = \|\phi^T(X)\mathbf{w} - \mathbf{a}\|^2 + \lambda\|\mathbf{w}\|^2, \tag{5}$$

and the minimum of Equation (5) is given by:

$$J^*_{RLS} = \mathbf{a}^T \left(I + \frac{1}{\lambda} K \right)^{-1} \mathbf{a}. \tag{6}$$

In Equation (5), $\phi(X) = (\phi(\mathbf{x}_1^+), \cdots, \phi(\mathbf{x}_{m^+}^+), \phi(\mathbf{x}_1^-), \cdots, \phi(\mathbf{x}_{m^-}^-))$ are the (centred) training data in the feature space such that $\phi(X)^T \phi(X) = K$.

Due to strong duality, the minimal value of Equation (5) is equal to the maximal value of its Lagrangian dual problem, that is,

$$J^*_{RLS} = \max_{\boldsymbol{\alpha}} \mathbf{a}^T \boldsymbol{\alpha} - \frac{1}{4} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \frac{1}{4\lambda} \boldsymbol{\alpha}^T K \boldsymbol{\alpha},$$

or equivalently

$$J^*_{RLS} = -\min_{\boldsymbol{\alpha}} \left(-\mathbf{a}^T \boldsymbol{\alpha} + \frac{1}{4} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \frac{1}{4\lambda} \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right), \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$. By combining Equation (4), Equation (6) and Equation (7), it follows that the maximal value of the FDA objective in Equation (3) is given by:

$$J^*_{FDA} = \mathbf{a}^T \mathbf{a} + \min_{\boldsymbol{\alpha}} \left(-\mathbf{a}^T \boldsymbol{\alpha} + \frac{1}{4} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \frac{1}{4\lambda} \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \right). \tag{8}$$

Now instead of a fixed single kernel, consider the case where the kernel K can be chosen from linear combinations of a set of base kernels. The kernel weights must be regularised somehow to make sure Equation (8) remains meaningful and does not become arbitrarily large. In this paper, we propose to impose an ℓ_p regularisation on the kernel weights for any $p \geq 1$, following Kloft et al. (2009, 2011):

$$\tilde{\mathcal{K}} = \left\{ K = \sum_{j=1}^{n} \beta_j K_j : \boldsymbol{\beta} \geq \mathbf{0}, \|\boldsymbol{\beta}\|_p^p \leq 1 \right\}. \tag{9}$$

Combining Equation (9) and Equation (8), and dropping the unimportant constant $\mathbf{a}^T \mathbf{a}$, it can be shown that the optimal $K \in \tilde{\mathcal{K}}$ maximising Equation (4) is found by solving:

$$\begin{gathered} \max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} -\mathbf{a}^T \boldsymbol{\alpha} + \frac{1}{4} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \frac{1}{4\lambda} \sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha} \\ \text{s.t. } \boldsymbol{\beta} \geq \mathbf{0}, \ \|\boldsymbol{\beta}\|_p^p \leq 1. \end{gathered} \tag{10}$$

Note that putting an ℓ_p constraint on $\boldsymbol{\beta}$ or penalizing $\mathbf{w}$ by an $\ell_{2,r}$ block norm are equivalent with $p = r/(2-r)$ (Szafranski et al., 2008). When $p = 1$, we have the ℓ_1 MK-FDA discussed in Ye et al. (2008); while $p = \infty$ leads to $r = 2$, and MK-FDA reduces to standard single kernel FDA with unweighted concatenation of base feature spaces. In this paper, however, we are interested in the general case of any $p \geq 1$.

Equation (10) is an optimisation problem with a quadratic objective and a general p^{th} order constraint. We borrow the idea from ℓ_p MK-SVM (Kloft et al., 2009, 2011) and use second order Taylor expansion to approximate the norm constraint:

$$\|\boldsymbol{\beta}\|_p^p \approx \frac{p(p-1)}{2} \sum_{j=1}^{n} \tilde{\beta}_j^{p-2} \beta_j^2 - \sum_{j=1}^{n} p(p-2) \tilde{\beta}_j^{p-1} \beta_j + \frac{p(p-3)}{2} + 1 := \nu(\boldsymbol{\beta}), \tag{11}$$

where $\tilde{\beta}_j$ is the current estimate of β_j in an iterative process, which will be explained in more detail shortly. Substituting Equation (11) into Equation (10), we arrive at the binary ℓ_p MK-FDA saddle point problem:

$$\max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} -\mathbf{a}^T\boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{1}{4\lambda}\sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha} \tag{12}$$
$$\text{s.t. } \boldsymbol{\beta} \geq \mathbf{0}, \ \ \nu(\boldsymbol{\beta}) \leq 1.$$

In Sonnenburg et al. (2006), the authors propose to transform a saddle point problem similar to Equation (12) to a semi-infinite program (SIP). A SIP is an optimisation problem with a finite number of variables $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$ on a feasible set described by infinitely many constraints (Hettich and Kortanek, 1993):

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } g(\mathbf{x}, u) \geq 0 \ \ \forall u \in \mathcal{U},$$

where $\mathcal{U}$ is an infinite index set. Following the similar arguments as in Sonnenburg et al. (2006) and Ye et al. (2008), we show in Theorem 1 that the saddle point problem in Equation (12) can also be transformed into a SIP.

Theorem 1 *Given a set of n kernel matrices $K_1, \cdots, K_n$, the kernel weights $\boldsymbol{\beta}$ that optimise Equation* (12) *are given by solving the following SIP problem:*

$$\max_{\theta, \boldsymbol{\beta}} \ \ \theta \tag{13}$$
$$s.t. \ \ -\mathbf{a}^T\boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{1}{4\lambda}\sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha} \geq \theta \ \ \forall \boldsymbol{\alpha} \in \mathbb{R}^m, \ \ \ \boldsymbol{\beta} \geq \mathbf{0}, \ \ \ \nu(\boldsymbol{\beta}) \leq 1.$$

Proof Let $\boldsymbol{\alpha}^*$ be the optimal solution to the saddle point problem in Equation (12). By defining

$$\theta^* := -\mathbf{a}^T\boldsymbol{\alpha}^* + \frac{1}{4}\boldsymbol{\alpha}^{*T}\boldsymbol{\alpha}^* + \frac{1}{4\lambda}\sum_{j=1}^{n} \boldsymbol{\alpha}^{*T} \beta_j K_j \boldsymbol{\alpha}^*$$

as the minimum objective value achieved by $\boldsymbol{\alpha}^*$, we have

$$-\mathbf{a}^T\boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{1}{4\lambda}\sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha} \geq \theta^*$$

$\forall \boldsymbol{\alpha} \in \mathbb{R}^m$. Now define

$$\theta = \min_{\boldsymbol{\alpha}} -\mathbf{a}^T\boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{1}{4\lambda}\sum_{j=1}^{n} \boldsymbol{\alpha}^T \beta_j K_j \boldsymbol{\alpha}$$

and substitute it into Equation (12), the theorem is proved. ■

We adapt the wrapper algorithm in Sonnenburg et al. (2006) to solve the SIP in Equation (13). This algorithm is based on the column generation technique, where the basic idea is to divide a SIP into an inner subproblem and an outer subproblem. The algorithm alternates between solving the

Algorithm 1 A wrapper algorithm for solving the binary ℓ_p MK-FDA SIP in Equation (13)

```
Input: $K_1, \cdots, K_n, \mathbf{a}, \theta^{(1)} = -\infty, \beta_j^{(1)} = n^{-1/p} \forall j, \varepsilon$.
Output: Learnt kernel weights $\boldsymbol{\beta} = (\beta_1^{(t)}, \cdots, \beta_n^{(t)})^T$.
for $t = 1, \cdots$ do
    Compute $\boldsymbol{\alpha}^{(t)}$ in Equation (15);
    Compute $S^{(t)} = -\mathbf{a}^T \boldsymbol{\alpha}^{(t)} + \frac{1}{4} \boldsymbol{\alpha}^{(t)T} \boldsymbol{\alpha}^{(t)} + \frac{1}{4\lambda} \sum_{j=1}^n \boldsymbol{\alpha}^{(t)T} \beta_j^{(t)} K_j \boldsymbol{\alpha}^{(t)}$;
    if $|1 - \frac{S^{(t)}}{\theta^{(t)}}| \leq \varepsilon$ then
        break;
    end if
    Compute $\{\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\}$ in Equation (16), where $\nu(\boldsymbol{\beta})$ is defined as in Equation (11) with
    $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$;
end for
```

two subproblems until convergence. At step t, the inner subproblem ($\boldsymbol{\alpha}$ step) identifies the constraint that maximises the constraint violation for $\{\theta^{(t)}, \boldsymbol{\beta}^{(t)}\}$:

$$\boldsymbol{\alpha}^{(t)} := \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} -\mathbf{a}^T \boldsymbol{\alpha} + \frac{1}{4} \boldsymbol{\alpha}^T \boldsymbol{\alpha} + \frac{1}{4\lambda} \sum_{j=1}^n \boldsymbol{\alpha}^T \beta_j^{(t)} K_j \boldsymbol{\alpha}. \tag{14}$$

Note that the program in Equation (14) is nothing but the single kernel FDA/RLS dual problem using the current estimate $\boldsymbol{\beta}^{(t)}$ as kernel weights. Observing that Equation (14) is an unconstrained quadratic program, $\boldsymbol{\alpha}^{(t)}$ is obtained by solving the following linear system (Ye et al., 2008):

$$\left(\frac{1}{2} I + \frac{1}{2\lambda} \sum_{j=1}^n \beta_j^{(t)} K_j \right) \boldsymbol{\alpha}^{(t)} = \mathbf{a}. \tag{15}$$

If $\boldsymbol{\alpha}^{(t)}$ satisfies constraint $-\mathbf{a}^T \boldsymbol{\alpha}^{(t)} + \frac{1}{4} \boldsymbol{\alpha}^{(t)T} \boldsymbol{\alpha}^{(t)} + \frac{1}{4\lambda} \sum_{j=1}^n \boldsymbol{\alpha}^{(t)T} \beta_j^{(t)} K_j \boldsymbol{\alpha}^{(t)} \geq \theta^{(t)}$ then $\{\theta^{(t)}, \boldsymbol{\beta}^{(t)}\}$ is optimal. Otherwise, the constraint is added to the set of constraints and the algorithm proceeds to the outer subproblem of step t.

The outer subproblem ($\boldsymbol{\beta}$ step) is also called the restricted master problem. At step t, it computes the optimal $\{\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\}$ in Equation (13) for a restricted subset of constraints:

$$\{\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\} = \underset{\theta, \boldsymbol{\beta}}{\operatorname{argmax}}\, \theta \tag{16}$$

$$\text{s.t.} \;\; -\mathbf{a}^T \boldsymbol{\alpha}^{(r)} + \frac{1}{4} \boldsymbol{\alpha}^{(r)T} \boldsymbol{\alpha}^{(r)} + \frac{1}{4\lambda} \sum_{j=1}^n \boldsymbol{\alpha}^{(r)T} \beta_j K_j \boldsymbol{\alpha}^{(r)} \geq \theta \;\; \forall r = 1, \cdots, t, \;\; \boldsymbol{\beta} \geq \mathbf{0}, \;\; \nu(\boldsymbol{\beta}) \leq 1.$$

When $p = 1$, $\nu(\boldsymbol{\beta}) \leq 1$ reduces to a linear constraint. As a result, Equation (16) becomes a linear program (LP) and the ℓ_p MK-FDA reduces to the ℓ_1 MK-FDA in Ye et al. (2008). When $p > 1$, Equation (16) is a quadratically constrained linear program (QCLP) with one quadratic constraint $\nu(\boldsymbol{\beta}) \leq 1$ and $t + n$ linear constraints. This can be solved by off-the-shelf optimisation tools such as Mosek.[1] Note that at time t, $\nu(\boldsymbol{\beta})$ is defined as in Equation (11) with $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$, that is, the current estimate of $\boldsymbol{\beta}$.

[1] Mosek optimisation toolbox can be found at `http://www.mosek.com`.

Normalised maximal constraint violation is used as a convergence criterion. The algorithm stops when $|1-\frac{S^{(t)}}{\theta^{(t)}}| \leq \varepsilon$, where $S^{(t)} := -\mathbf{a}^T\boldsymbol{\alpha}^{(t)} + \frac{1}{4}\boldsymbol{\alpha}^{(t)T}\boldsymbol{\alpha}^{(t)} + \frac{1}{4\lambda}\sum_{j=1}^{n}\boldsymbol{\alpha}^{(t)T}\beta_j^{(t)}K_j\boldsymbol{\alpha}^{(t)}$ and ε is a pre-defined accuracy parameter. This iterative wrapper algorithm for solving the binary ℓ_p MK-FDA SIP is summarised in Algorithm 1. It is a special case of a set of semi-infinite programming algorithms known as exchange methods, which are guaranteed to converge (Hettich and Kortanek, 1993). Finally, note that in line 4 of Algorithm 1, $\boldsymbol{\beta}^{(t+1)}$ can also be solved using the analytical update in Kloft et al. (2011) that is adapted to FDA. However, in practice we notice that for MK-FDA, such an analytical update tends to be numerically unstable when p is close to 1.

2.2 Multiclass Classification

In this section we consider the multiclass case. Let c be the number of classes, and m_k be the number of training examples in the k^{th} class. In multiclass FDA, the following objective is commonly maximised (Ye et al., 2008):

$$J_{MC-FDA}(W) = \text{trace}\left(\left(W^T(S_T+\lambda I)W\right)^{-1}W^TS_BW\right), \tag{17}$$

where W is the projection matrix, the within class scatter S_W is defined in a similar way as in Equation (2) but with c classes, and the between class scatter is $S_B = \phi(X)HH^T\phi(X)^T$, where $\phi(X) = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_m))$ is the set of m training examples in the feature space, and $H = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_c)$ is an $m \times c$ matrix with $\mathbf{h}_k = (h_{1k}, \cdots, h_{mk})^T$ and

$$h_{ik} = \begin{cases} \sqrt{\frac{m}{m_k}} - \sqrt{\frac{m_k}{m}} & \text{if } y_i = k \\ -\sqrt{\frac{m_k}{m}} & \text{if } y_i \neq k. \end{cases} \tag{18}$$

Similar to the binary case, using duality theory and the connection between FDA and RLS, Ye et al. (2008) show that the maximal value of Equation (17) is given by (up to an additive constant determined by the labels):

$$J^*_{MC-FDA} \sim \min_{\boldsymbol{\alpha}_k}\sum_{k=1}^{c}\left(-\mathbf{h}_k^T\boldsymbol{\alpha}_k + \frac{1}{4}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k + \frac{1}{4\lambda}\boldsymbol{\alpha}_k^TK\boldsymbol{\alpha}_k\right),$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^m$ for $k = 1, \cdots, c$. When choosing from linear combinations of a set of base kernels with kernel weights regularised with an ℓ_p norm, the optimal kernel weights are given by:

$$\max_{\boldsymbol{\beta}}\min_{\boldsymbol{\alpha}_k}\sum_{k=1}^{c}\left(-\mathbf{h}_k^T\boldsymbol{\alpha}_k + \frac{1}{4}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k + \frac{1}{4\lambda}\sum_{j=1}^{n}\boldsymbol{\alpha}_k^T\beta_jK_j\boldsymbol{\alpha}_k\right) \tag{19}$$
$$\text{s.t.} \quad \boldsymbol{\beta} \geq \mathbf{0}, \quad \|\boldsymbol{\beta}\|_p^p \leq 1.$$

We use again second order Taylor expansion to approximate the norm constraint and arrive at the multiclass ℓ_p MK-FDA saddle point problem:

$$\max_{\boldsymbol{\beta}}\min_{\boldsymbol{\alpha}_k}\sum_{k=1}^{c}\left(-\mathbf{h}_k^T\boldsymbol{\alpha}_k + \frac{1}{4}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k + \frac{1}{4\lambda}\sum_{j=1}^{n}\boldsymbol{\alpha}_k^T\beta_jK_j\boldsymbol{\alpha}_k\right)$$
$$\text{s.t.} \quad \boldsymbol{\beta} \geq \mathbf{0}, \quad \nu(\boldsymbol{\beta}) \leq 1,$$

Algorithm 2 A wrapper algorithm for solving the multiclass ℓ_p MK-FDA SIP in Equation (20)

```
Input: $K_1, \cdots, K_n, \mathbf{a}, \theta^{(1)} = -\infty, \beta_j^{(1)} = n^{-1/p} \forall j, \varepsilon$.
Output: Learnt kernel weights $\boldsymbol{\beta} = (\beta_1^{(t)}, \cdots, \beta_n^{(t)})^T$.
for $t = 1, \cdots$ do
    Compute $\boldsymbol{\alpha}_k^{(t)}$ in Equation (21);
    Compute $S^{(t)} = \sum_{k=1}^{c} \big( -\mathbf{h}_k^T \boldsymbol{\alpha}_k^{(t)} + \frac{1}{4} \boldsymbol{\alpha}_k^{(t)T} \boldsymbol{\alpha}_k^{(t)} + \frac{1}{4\lambda} \sum_{j=1}^{n} \boldsymbol{\alpha}_k^{(t)T} \beta_j^{(t)} K_j \boldsymbol{\alpha}_k^{(t)} \big)$;
    if $|1 - \frac{S^{(t)}}{\theta^{(t)}}| \leq \varepsilon$ then
        break;
    end if
    Compute $\{\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\}$ in Equation (22), where $\nu(\boldsymbol{\beta})$ is defined as in Equation (11) with
    $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$;
end for
```

where $\nu(\boldsymbol{\beta})$ is defined as in Equation (11).

Again similar to the binary case, Equation (19) can be reformulated as a SIP:

$$\max_{\theta, \boldsymbol{\beta}} \quad \theta \tag{20}$$

$$\text{s.t.} \quad \sum_{k=1}^{c} \left(-\mathbf{h}_k^T \boldsymbol{\alpha}_k + \frac{1}{4} \boldsymbol{\alpha}_k^T \boldsymbol{\alpha}_k + \frac{1}{4\lambda} \sum_{j=1}^{n} \boldsymbol{\alpha}_k^T \beta_j K_j \boldsymbol{\alpha}_k \right) \geq \theta \;\; \forall \boldsymbol{\alpha}_k \in \mathbb{R}^m, \;\; \boldsymbol{\beta} \geq \mathbf{0}, \;\; \nu(\boldsymbol{\beta}) \leq 1,$$

and the SIP can be solved using a column generation algorithm that is similar to Algorithm 1. In the inner subproblem, the only difference is that instead of one linear system, here c linear systems need to be solved, one for each $\mathbf{h}_k$:

$$\left(\frac{1}{2} I + \frac{1}{2\lambda} \sum_{j=1}^{n} \beta_j^{(t)} K_j \right) \boldsymbol{\alpha}_k^{(t)} = \mathbf{h}_k. \tag{21}$$

Accordingly, the outer subproblem for computing the optimal $\{\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}\}$ is adapted to work with multiple classes:

$$(\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}) = \operatorname*{argmax}_{\theta, \boldsymbol{\beta}} \theta \tag{22}$$

$$\text{s.t.} \quad \sum_{k=1}^{c} \left(-\mathbf{h}_k^T \boldsymbol{\alpha}_k^{(r)} + \frac{1}{4} \boldsymbol{\alpha}_k^{(r)T} \boldsymbol{\alpha}_k^{(r)} + \frac{1}{4\lambda} \sum_{j=1}^{n} \boldsymbol{\alpha}_k^{(r)T} \beta_j K_j \boldsymbol{\alpha}_k^{(r)} \right) \geq \theta \quad \forall r = 1, \cdots, t$$

$$\boldsymbol{\beta} \geq \mathbf{0}, \quad \nu(\boldsymbol{\beta}) \leq 1.$$

When $p = 1$, Equation (22) reduces to an LP and our formulation reduces to that in Ye et al. (2008). For $p > 1$, Equation (22) is an QCLP with one quadratic constraint and $t + n$ linear constraints, as in the binary case. The iterative wrapper algorithm for solving the multiclass ℓ_p MK-FDA SIP is summarised in Algorithm 2.

2.3 Addressing Efficiency Issues

In this section we discuss several possible improvements over the wrapper-based ℓ_p MK-FDA method proposed in the previous sections. In particular, we address time and memory complexity issues, in light of recent advances in MKL optimisation techniques. We show that by exploiting the equivalence between kernel FDA and least squares SVM (LSSVM) (Suykens and Vandewalle, 1999), the interleaved method in Sonnenburg et al. (2006) and Kloft et al. (2011) can be applied to MK-FDA. Furthermore, we demonstrate that the formulation in Vishwanathan et al. (2010) that tackles directly the MKL dual problem can also be adapted to work with MK-FDA. Both new formulations discussed in this section are equivalent to previous ones in terms of learnt kernel weights, but can potentially lead to significant efficiency improvement. However, note that we describe these new formulations only briefly, and do not show their efficiency in the experiments section and their implementation details, since these are not in the main scope of this paper. Note also that in the following we focus only on multiclass formulations, as the corresponding binary ones can be derived in a very similar fashion, or as special cases.

2.3.1 INTERLEAVED OPTIMISATION OF THE SADDLE POINT PROBLEM

We consider the multiclass MKL problem for a general convex loss function $V(\xi_{ik}, h_{ik})$:

$$\min_{\mathbf{w}_{jk},\xi_{ik},\boldsymbol{\beta}} \sum_{k=1}^{c} \left(\frac{1}{2} \sum_{j=1}^{n} \frac{||\mathbf{w}_{jk}||^2}{\beta_j} + C \sum_{i=1}^{m} V(\xi_{ik}, h_{ik}) \right) \tag{23}$$
$$\text{s.t.} \quad \sum_{j=1}^{n} \mathbf{w}_{jk}^T \phi_j(\mathbf{x}_i) = \xi_{ik},\ \forall i,\ \forall k;\ \ \boldsymbol{\beta} \geq \mathbf{0};\ \ ||\boldsymbol{\beta}||_p^2 \leq 1,$$

where h_{ik} is as defined in Equation (18), and we have replaced the constraint $||\boldsymbol{\beta}||_p^p \leq 1$ equivalently by $||\boldsymbol{\beta}||_p^2 \leq 1$. When $V(\xi_{ik}, h_{ik})$ is the square loss $V(\xi_{ik}, h_{ik}) = \frac{1}{2}(\xi_{ik} - h_{ik})^2$, Equation (23) is essentially multiclass multiple kernel regularised least squares (MK-RLS). It can be shown (see Appendix A for details) that this multiclass MK-RLS can be reformulated as the following saddle point problem:

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}_k} \sum_{k=1}^{c} \left(\mathbf{h}_k^T \boldsymbol{\alpha}_k - \frac{1}{2C} \boldsymbol{\alpha}_k^T \boldsymbol{\alpha}_k - \frac{1}{2} \sum_{j=1}^{n} \boldsymbol{\alpha}_k^T \beta_j K_j \boldsymbol{\alpha}_k \right) \tag{24}$$
$$\text{s.t.} \quad \boldsymbol{\beta} \geq \mathbf{0};\ \ ||\boldsymbol{\beta}||_p^2 \leq 1.$$

Making substitutions $\boldsymbol{\alpha}_k \to \frac{C}{2}\boldsymbol{\alpha}_k$ and then $C \to \frac{1}{\lambda}$, it directly follows that the MK-RLS in Equation (24) is equivalent to the MK-FDA in Equation(19). In the previous sections, we proposed to use a conceptually very simple wrapper algorithm to solve it. However, as pointed out in Sonnenburg et al. (2006) and Kloft et al. (2011), such an algorithm has two disadvantages: solving the whole single kernel problem in the $\boldsymbol{\alpha}$ step is unnecessary therefore wasteful, and all kernels need to be loaded into memory. These problems, especially the second one, significantly limit the scalability of wrapper-based MKL algorithms. For example, 50 kernel matrices of size 20000 × 20000 would usually not fit into memory since they require approximately 149GB of memory (Kloft et al., 2011).

Exploiting the fact that LSSVM, RLS and kernel FDA are equivalent (Rifkin, 2002; Gestel et al., 2002; Keerthi and Shevade, 2003), sequential minimal optimisation (SMO) techniques (Joachims, 1988) developed for LSSVM (Keerthi and Shevade, 2003; Lopez and Suykens, 2011) can be employed to remedy these problems. This effectively leads to an interleaved algorithm that is similar to Algorithm 2 in Kloft et al. (2011), but applies to square loss instead of to hinge loss. Such an interleaved optimisation strategy allows for a very cheap update of a minimal subset of the dual variables $\boldsymbol{\alpha}_k$ in each $\boldsymbol{\alpha}$ step, without having to have access to the whole kernel matrices, and as a result extends the applicability of MK-FDA to large scale problems. We omit details of the resulting interleaved MK-FDA algorithm, the interested reader is referred to Keerthi and Shevade (2003) and Lopez and Suykens (2011).

2.3.2 WORKING DIRECTLY WITH THE DUAL

The MK-FDA algorithms considered so far, including the wrapper method and the interleaved method, are all based on the intermediate saddle point formulation Equation (24), or equivalently, Equation(19). Recently, a "direct" formulation of MKL was proposed in Vishwanathan et al. (2010), where the idea is to eliminate $\boldsymbol{\beta}$ from the saddle point problem, and deal directly with the dual. Consider again MKL with a general convex loss, but following Vishwanathan et al. (2010) this time we impose the norm constraint in the form of Tikhonov regularisation instead of Ivanov regularisation:

$$\min_{\mathbf{w}_{jk},\xi_{ik},\boldsymbol{\beta}} \sum_{k=1}^{c}\left(\frac{1}{2}\sum_{j=1}^{n}\frac{\|\mathbf{w}_{jk}\|^2}{\beta_j}+C\sum_{i=1}^{m}V(\xi_{ik},h_{ik})\right)+\frac{\mu}{2}\|\boldsymbol{\beta}\|_p^2 \tag{25}$$

$$\text{s.t.} \quad \sum_{j=1}^{n}\mathbf{w}_{jk}^T\phi_j(\mathbf{x}_i)=\xi_{ik},\ \forall i,\ \forall k;\ \ \boldsymbol{\beta}\geq\mathbf{0}.$$

Note that the two formulations in Equation (25) and Equation (23) are equivalent, in the sense that for any given C there exists a μ (and vice versa) such that the optimal solutions to both problems are identical (Kloft et al., 2011).

It can be shown (see Appendix B for details) that for the special case of square loss, which corresponds to MK-FDA/MK-RLS, the dual of Equation(25) is:

$$\max_{\boldsymbol{\alpha}_k}\sum_{k=1}^{c}\left(\mathbf{h}_k^T\boldsymbol{\alpha}_k-\frac{1}{2C}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k\right)-\frac{1}{8\mu}\left\|\left(\sum_{k=1}^{c}\boldsymbol{\alpha}_k^TK_j\boldsymbol{\alpha}_k\right)_{j=1}^{n}\right\|_q^2, \tag{26}$$

where $q=\frac{p}{p-1}$ is the dual norm of p, and once the optimal $\boldsymbol{\alpha}_k$ are found by solving Equation (26), the kernel weights are given by:

$$\beta_j=\frac{1}{2\mu}\left(\sum_{j=1}^{n}(\sum_{k=1}^{c}\boldsymbol{\alpha}_k^TK_j\boldsymbol{\alpha}_k)^q\right)^{\frac{1}{q}-\frac{1}{p}}(\sum_{k=1}^{c}\boldsymbol{\alpha}_k^TK_j\boldsymbol{\alpha}_k)^{\frac{q}{p}}.$$

Equation (26) can be viewed as an extension of Equation (9) in Vishwanathan et al. (2010) to multiclass problems. Another difference is that Equation (9) in Vishwanathan et al. (2010) considers a hinge loss, while Equation (26) is for square loss. Similarly as in Vishwanathan et al. (2010), for any $p>1$, Equation (26) can be solved using an SMO type of algorithm, with the update rule for the minimal subset of dual variables adapted to work with square loss (Keerthi and Shevade, 2003; Lopez and Suykens, 2011). On the other hand, observing that Equation (26) is an unconstrained optimisation problem and the objective function is differentiable everywhere for $p>1$, an alternative approach is the quasi-Newton descent methods, for example, the limited memory variant (Liu and Nocedal, 1989). In fact, Equation (26) can also be thought of as an extension of the smooth variant of group Lasso considered in Kloft et al. (2011) to multiclass case. Note however that Equation (26) has a term of ℓ_q norm squared, while the smooth group Lasso formulation in Kloft et al. (2011) has a term of ℓ_q norm. This is a direct result of the fact that the two formulations use Tikhonov regularisation and Ivanov regularisation over $\boldsymbol{\beta}$, respectively.

3. Experiments

In this section we validate the usefulness of the proposed ℓ_p MK-FDA with experimental evidence on six datasets. The experiments can be divided into four groups:

- We first demonstrate in Section 3.1 and 3.2 the different behaviour of the sparse ℓ_1 MK-FDA and a non-sparse version of MK-FDA (ℓ_2 norm) on synthetic data and the Pascal VOC2008 object recognition dataset (Everingham et al., 2008). The goal of these two experiments is to confirm that ℓ_1 and ℓ_2 regularisations indeed lead to sparse and non-sparse kernel weights respectively.

- Next in Section 3.3, 3.4 and 3.5 we carry out experiments on another three object and image categorisation benchmarks, namely, Pascal VOC2007 (Everingham et al., 2007), Caltech101 (Fei-Fei et al., 2006), and Oxford Flower17 (Nilsback and Zisserman, 2008). We show that by selecting the regularisation norm p on an independent validation set, the intrinsic sparsity of the given set of base kernels can be learnt. As a result, using the learnt optimal norm p in the proposed ℓ_p MK-FDA offers better performance than ℓ_1 or ℓ_∞ MK-FDAs. Moreover, we compare the performance of ℓ_p MK-FDA and that of several variants of ℓ_p MK-SVM, and show that on image categorisation problems ℓ_p MK-FDA tends to have a small but consistent edge over its SVM counterpart.

- In Section 3.6 we further compare ℓ_p MK-FDA and ℓ_p MK-SVM on the protein subcellular localisation problem studied in Zien and Ong (2007) and Ong and Zien (2008). On this dataset ℓ_p MK-SVM outperforms ℓ_p MK-FDA by a small margin, and the results suggest that given the same set of base kernels, the two MKL algorithms may favour slightly different norms.

- Finally, in Section 3.7, the training speed of our wrapper-based ℓ_p MK-FDA and several ℓ_p MK-SVM implementations is analysed empirically on a few small/medium sized problems, where MK-FDA compares favourably or similarly against state-of-the-art MKL techniques.

Among the six datasets used in the experiments, three of them (synthetic, VOC08, VOC07) are binary problems and the rest (Caltech101, Flower17, Protein) are multiclass ones. In our experiments the wrapper-based ℓ_p MK-FDA is implemented in Matlab with the outer-subproblem solved using the Mosek optimisation toolbox. The code of our ℓ_p MK-FDA implementation is available online.[2] Once the kernel weights have been learnt, we use a spectral regression based efficient kernel FDA implementation (Cai et al., 2007; Tahir et al., 2009) to compute the optimal projection directions, the code of which is also available online.[3] On binary problems, we compare ℓ_p MK-FDA with two implementations of binary ℓ_p MK-SVM, namely, MK-SVM Shogun (Sonnenburg et al., 2006, 2010),[4] and SMO-MKL (Vishwanathan et al., 2010);[5] while on multiclass problems, we compare ℓ_p MK-FDA with two variants of multiclass ℓ_p MK-SVM: MK-SVM Shogun and MK-SVM OBSCURE (Orabona et al., 2010; Orabona and Jie, 2011).[6] In both ℓ_p MK-FDA and ℓ_p MK-SVM

[2] The code of our ℓ_p MK-FDA is available at `http://www.featurespace.org`

[3] The code of spectral regression FDA can be found at `http://www.zjucadcg.cn/dengcai/SR/index.html`.

[4] Version 0.10.0 of the Shogun toolbox, the latest version as of the writing of this paper, can be found at `http://www.shogun-toolbox.org`.

[5] The code of SMO-MKL is available at `http://research.microsoft.com/en-us/um/people/manik/code/SMO-MKL/download.html`.

[6] The code of OBSCURE can be found at `http://dogma.sourceforge.net`.

Shogun, the stopping threshold ε is set to 10^{-4} unless stated otherwise. Parameters in MK-SVM OBSCURE and SMO-MKL are set to default values unless stated otherwise.

All kernels used in the experiments have been normalised. For the first five datasets, due to the kernel functions used, the kernel matrices are by definition spherically normalised: all data points lie on the unit hypersphere in the feature space. For the protein localisation dataset, the kernels are multiplicatively normalised following Ong and Zien (2008) and Kloft et al. (2011) to allow comparison with Kloft et al. (2011). After normalisation, the kernels are then centred in the feature spaces, as required by ℓ_p MK-FDA. Note that ℓ_p MK-SVM is not affected by centring. Kernels used in the experiments (except for those in the simulation and in training speed experiments) are also available online.[7]

3.1 Simulation

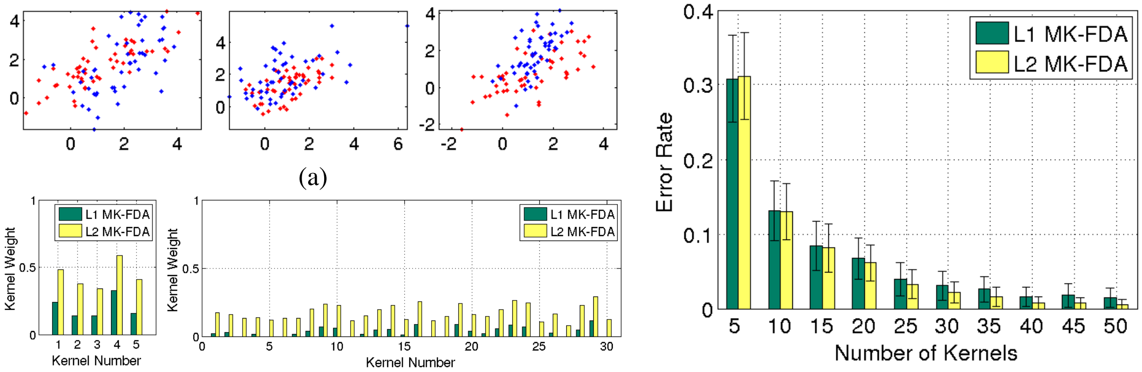
We first perform simulation to illustrate the different behaviour of ℓ_1 MK-FDA and a special case of ℓ_p MK-FDA, namely, the case of $p = 2$. We simulate two classes by sampling 100 points from two 2-dimensional Gaussian distributions, 50 points from each. The means of the two distributions in both dimensions are drawn from a uniform distribution between 1 and 2, and the covariances of the two distributions are also randomly generated. A radial basis function (RBF) kernel is then constructed using these 2-dimensional points. Similarly, 100 test points are sampled from the same distributions, 50 from each, and an RBF kernel is built for the test points. Kernel FDA is then applied to find the best projection direction in the feature space and compute the error rate on the test set. Figure 1 (a) gives 3 examples of the simulated points. It shows that due to the parameters used in the two Gaussian distributions, the two classes are heavily, but not completely, overlapping. As a result, the error rate given by single kernel FDA is around 0.43: slightly better than a random guess.

The above process of mean/covariance generation, sampling, and kernel building is repeated n times, resulting in n training kernels (and n corresponding test kernels). These n training kernels, although generated independently, can be thought of as kernels that capture different "views" of a single binary classification problem. With this interpretation in mind, we apply ℓ_1 and ℓ_2 MK-FDAs to learn optimal kernel weights for this classification problem. We vary the number n from 5 to 50 at a step size of 5. For each value of n, ℓ_1 and ℓ_2 MK-FDAs are applied and the resulting error rates are recorded. This process is repeated 100 time for each value of n to compute the mean and standard deviation of error rates. The results for various n values are plotted in Figure 1 (c).

It is clear in Figure 1 (c) that as the number of kernels increases, the error rates of both methods drop. This is expected, since more kernels bring more discriminative information. Another observation is that ℓ_1 MK-FDA slightly outperforms ℓ_2 MK-FDA when the number of kernels is 5, and vice versa when the number of kernels is 10 or 15. When there are 20 kernels, the advantage of ℓ_2 MK-FDA becomes clear. As the number of kernels keeps increasing, its advantage becomes more and more evident.

The different behaviour of ℓ_1 and ℓ_2 MK-FDAs can be explained by the different weights learnt from them. Two typical examples of such weights, learnt using $n = 5$ kernels and $n = 30$ kernels respectively, are plotted in Figure 1 (b). It has been known that ℓ_1 norm regularisation tends to produce sparse solutions (Rätsch, 2001; Kloft et al., 2008). When kernels carry complementary information, this will lead to a loss of information and hence degraded performance. When the

[7] The kernels can be downloaded at `http://www.featurespace.org`.



(a)

(b)

(c)

Figure 1: Simulation: (a) Three examples of the two Gaussian distributions. (b) Comparing the kernel weights learnt from ℓ_1 MK-FDA and ℓ_2 MK-FDA. Left: using 5 kernels. Right: using 30 kernels. (c) Mean and standard deviation of error rates of ℓ_1 MK-FDA and ℓ_2 MK-FDA using various number of kernels.

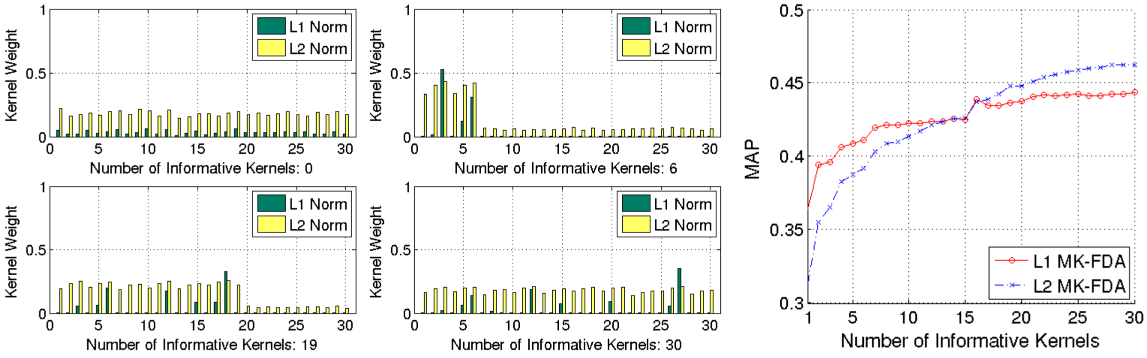
number of kernels is sufficiently small, however, this effect does not occur: as can be seen in the left plot of Figure 1 (b), when there are only 5 kernels, all of them get non-zero weights in both ℓ_1 and ℓ_2 MK-FDAs.

As the number of kernels increases, eventually there are enough of them for the over-selectiveness of ℓ_1 regularisation to exhibit itself. As the right plot of Figure 1 (b) shows, when 30 kernels are used, many of them are assigned zero weights by ℓ_1 MK-FDA. This leads to a loss of information. By contrast, the weights learnt in ℓ_2 MK-FDA are non-sparse, hence the better performance. Finally, it is worth noting that the sparsity of learnt kernel weights, which captures the sparsity of information in the kernel set, is not to be confused with the numerical sparsity of the kernel matrices. For example, when the RBF kernel function is used, the kernel matrices will not contain any zero, regardless of the sparsity of kernel weights.

3.2 Pascal VOC2008

In this section, we demonstrate again the different behaviour of ℓ_1 and ℓ_2 MK-FDAs, but this time on a real world dataset: the Pascal visual object classes (VOC) challenge 2008 development dataset. The VOC challenge provides a yearly benchmark for comparison of object classification methods, with one of the most challenging datasets in the object recognition / image classification community. The VOC2008 development dataset consists of 4332 images of 20 object classes such as aeroplane, cat, person, etc. The dataset is divided into a pre-defined training set with 2111 images and a validation set with 2221 images. In our experiments, the training set is used for training and the validation set for testing. VOC2008 test set is not used as the class labels are not publicly available.

Pascal VOC2008 is a multilabel dataset in the sense that each image can contain multiple classes of objects. To tackle this multilabel problem, the classification of the 20 object classes is treated as 20 independent binary problems. In our experiments, average precision (AP) (Snoek et al., 2006) is used to measure the performance of each binary classifier. Average precision is particularly suitable for evaluating the performance of a retrieval system, since it emphasises higher ranked relevant



(a)

(b)

Figure 2: VOC2008: (a) Learnt kernel weights in ℓ_1 MK-FDA and ℓ_2 MK-FDA. "motorbike" class. (b) MAPs of ℓ_1 MK-FDA and ℓ_2 MK-FDA with various composition of kernel set.

instances. The mean of the APs of the 20 classes in the dataset, MAP, is used as a measure of the overall performance.

The SIFT descriptor (Lowe, 2004; Mikolajczyk and Schmid, 2005) and spatial pyramid match kernel (SPMK) (Grauman and Darrell, 2007; Lazebnik et al., 2006) based on bag-of-words model (Zhang et al., 2007; Gemert et al., 2008) are used to build base kernels. The combination of two sampling strategies (dense sampling and Harris-Laplace interest point sampling), 5 colour variants of SIFT descriptors (Sande et al., 2008), and 3 ways of dividing an image into spatial location grids results in $2 \times 5 \times 3 = 30$ "informative" kernels. We also generate 30 sets of random vectors, and build 30 RBF kernels from them. These random kernels are then mixed with the informative ones, to study how the properties of kernels affect the performance of ℓ_1 and ℓ_2 MK-FDAs.

The number of kernels used in each run is fixed to 30. In the first run, only the 30 random kernels are used. In the following runs the number of informative kernels is increased and that of random kernels decreased, until the 31^{st} run, where all 30 kernels are informative. In each run, we apply both ℓ_1 and ℓ_2 MK-FDAs to the 20 binary problems, compute the MAP for each algorithm, and record the learnt kernel weights.

Figure 2 (a) plots the kernel weights learnt from ℓ_1 MK-FDA and ℓ_2 MK-FDA. In each subplot, the weights of the informative kernels are plotted towards the left end and those of random ones towards the right. We clearly observe again the "over-selective" behaviour of ℓ_1 norm: it sets the weights of most kernels, including informative kernels, to zero. By contrast, the proposed ℓ_2 MK-FDA always assigns non-zero weights to the informative kernels. However, ℓ_2 MK-FDA is "under-selective", in the sense that it assigns non-zero weights to the random kernels. It is also worth noting that the kernels that do get selected by ℓ_1 MK-FDA are usually the ones that get highest weights in ℓ_2 MK-FDA.

The MAPs of both ℓ_1 and ℓ_2 MK-FDAs are shown in Figure 2 (b). In order to improve the clarity of the interest region, in Figure 2 (b), the MAP of the first run, that is, when all kernels are random, is not plotted. In such a situation, both versions of MK-FDAs reduce to a chance classifier, which has an MAP of around 0.007. It can be seen from Figure 2 (b) that, as expected, ℓ_1 MK-FDA outperforms ℓ_2 MK-FDA when the noise level is high and vice versa when the noise level is low. Another interpretation of this observation is that when the "intrinsic" sparsity of the base kernels is high then ℓ_1 norm regularisation is appropriate, and vice versa. This suggests that if we can learn this intrinsic sparsity of base kernels on a validation set, we will be able to find the most appropriate regularisation norm p, and get improved performance over a fix norm MK-FDA. We validate this idea in the next section.

3.3 Pascal VOC2007

Similar to Pascal VOC2008, Pascal VOC2007 is a multilabel object recognition dataset consisting of the same 20 object categories. The dataset is divided into training, validation and test sets, with 2501, 2510 and 4952 images respectively. As in the case of VOC2008, the classification of the 20 object classes is treated as 20 independent binary problems, and MAP is used as a measure of overall performance.

We generate 14 base kernels by combining 7 colour variants of local descriptors (Sande et al., 2008) and two kernel functions, namely, SPMK (Lazebnik et al., 2006; Grauman and Darrell, 2007) and RBF kernel with χ^2 distance (Zhang et al., 2007). We first perform supervised dimensionality reduction on the descriptors to improve their discriminability, following Cai et al. (2011). The descriptors with reduced dimensionality are clustered with k-means to learn codewords (Csurka et al., 2004). The soft assignment scheme in Gemert et al. (2008) is then employed to generate a histogram for each image as its representation. Finally, the two kernel functions are applied to the histograms to build kernel matrices.

We investigate the idea of learning the intrinsic sparsity of the base kernels by tuning the regularisation norm p on a validation set, using both ℓ_p MK-SVM and ℓ_p MK-FDA. For both methods, we learn the parameter p on the validation set from 12 values: $\{1, 1+2^{-6}, 1+2^{-5}, 1+2^{-4}, 1+2^{-3}, 1+2^{-2}, 1+2^{-1}, 2, 3, 4, 8, 10^6\}$. For ℓ_p MK-SVM, the regularisation parameter C is learnt jointly with p from 10 values that are logarithmically spaced over 2^{-2} to 2^7. Similarly, for ℓ_p MK-FDA, the regularisation parameter λ is learnt jointly with p from a set of 10 values that are logarithmically spaced over 4^{-5} to 4^4. The sets of values of C and λ are chosen to cover the areas in the parameter spaces that give the best performance for MK-SVM and MK-FDA, respectively.

Plotted in Figure 3 are the weights learnt on the training set in ℓ_p MK-FDA and ℓ_p MK-SVM with various p values for the “aeroplane” class. For ℓ_p MK-FDA, for each p value, the weights learnt with the optimal λ value are plotted; while for ℓ_p MK-SVM, for each p value, we show the weights learnt with the optimal C value. It is clear that as p increases, in both MKL algorithms, the sparsity of the learnt weights decreases. As expected, when $p = 10^6$ (practically infinity), the kernel weights become ones, that is, ℓ_∞ MK-FDA/MK-SVM produces uniform kernel weights. Note that for the same norm p, the weights learnt in ℓ_p MK-FDA and ℓ_p MK-SVM can be different. This is especially evident when p is small. Note also that results reported in this section are obtained using the Shogun implementation of MK-SVM, which is based on the saddle point formulation of the problem. The recently proposed SMO-MKL works directly with the dual and can be more efficient, especially on large scale problems. However, as discussed in Section 2.3, these two formulations are equivalent and produce identical kernel weights. Considering this, we only present the results of SMO-MKL in terms of training speed in Section 3.7.

Next, we plot in Figure 4 top-left the APs on the validation and test sets for the “bird” class with various p values, using ℓ_p MK-FDA, where again for each p value, the APs with the λ value that gives the best AP on the validation set are plotted. It is clear that the two curves match well, which implies that learning p in addition to λ should help. Shown in the middle and right columns

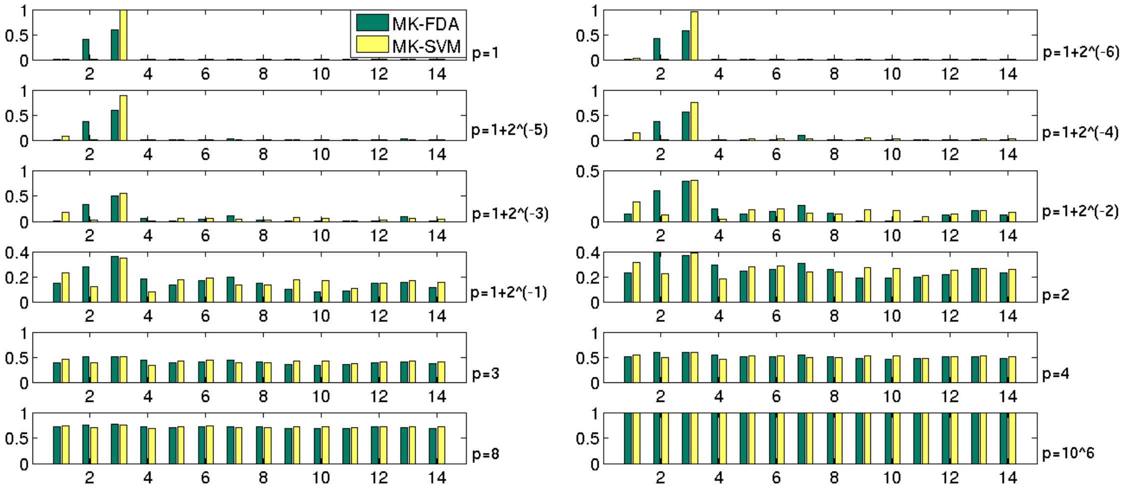


Figure 3: VOC2007: Kernel weights learnt on the training set in ℓ_p MK-FDA and ℓ_p MK-SVM with various p values. “aeroplane” class.

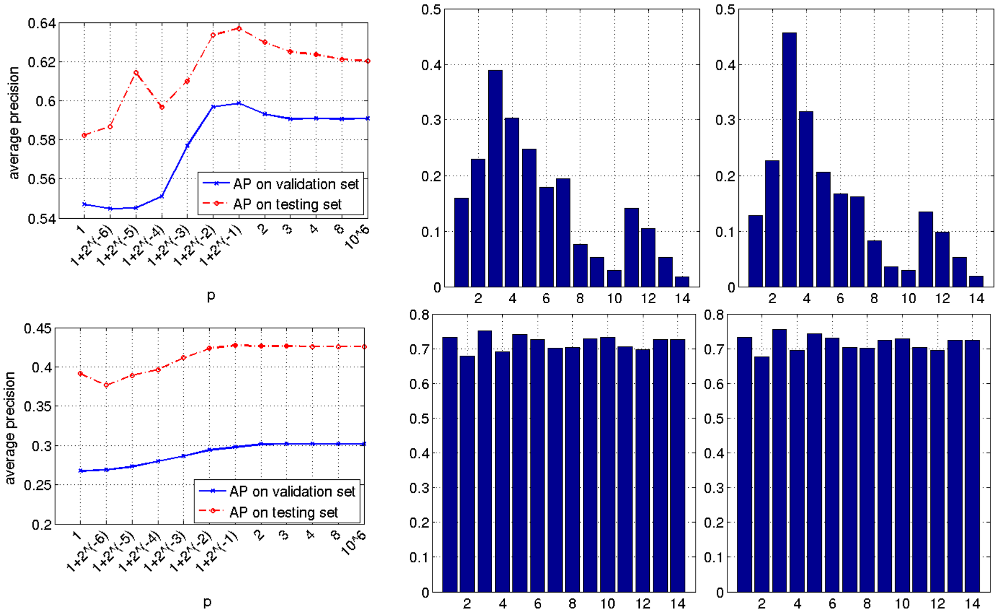


Figure 4: VOC2007: Learning the norm p for MK-FDA on the validation set. Top row: “bird” class. Bottom row: “pottedplant” class; left column: APs on the validation set and test set with various p values; middle column: kernel weights learnt on the training set with the optimal $\{p,\lambda\}$ combination; right column: kernel weights learnt on the training+validation set with the same $\{p,\lambda\}$ combination.

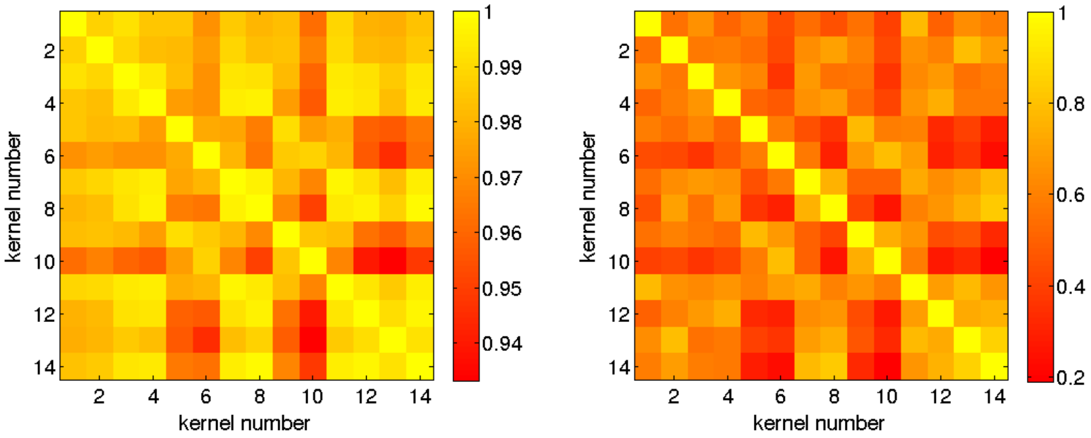
	MK-SVM			MK-FDA		
	ℓ_1	ℓ_∞	ℓ_p	ℓ_1	ℓ_∞	ℓ_p
aeroplane	78.8	79.6	79.6	79.9	79.5	**80.9**
bicycle	63.4	65.0	64.7	64.7	67.6	**67.8**
bird	57.3	61.0	61.0	57.1	62.0	**63.7**
boat	**71.1**	70.1	**71.1**	70.9	70.1	70.8
bottle	29.1	**29.9**	29.7	27.5	29.7	29.4
bus	62.9	64.9	**65.5**	63.4	66.1	66.1
car	77.9	78.8	78.8	79.1	79.5	**80.9**
cat	56.7	56.4	57.1	57.1	56.9	**58.3**
chair	52.3	**53.0**	**53.0**	51.9	52.5	52.9
cow	38.7	41.4	41.4	42.3	41.5	**43.4**
din. table	52.4	57.3	56.6	57.2	59.2	**61.4**
dog	42.8	45.8	44.6	44.2	**46.1**	45.1
horse	78.9	80.6	80.6	80.0	**81.1**	81.0
moterbike	66.3	66.8	66.8	67.8	67.8	**68.8**
person	86.7	88.0	88.0	86.8	88.1	**88.8**
pot. plant	31.8	41.0	40.5	32.5	**42.6**	42.5
sheep	40.2	**46.0**	**46.0**	39.0	44.4	43.9
sofa	44.0	43.8	44.0	43.5	43.7	**45.9**
train	81.3	82.4	82.4	83.2	84.2	**85.1**
tvmonitor	53.3	53.7	53.7	52.5	54.1	**56.9**
MAP	58.3	60.3	60.3	59.0	60.8	**61.7**

Table 1: VOC2007: Average precisions of six MKL methods

of the top row of Figure 4 are the learnt kernel weights with the optimal $\{p,\lambda\}$ combination on the training set and on the training + validation set, respectively. Since for the "bird" class the optimal p found on the validation set is $1+2^{-1}$, both sets of weights are non-sparse. For this particular binary problem, the intrinsic sparsity of the set of base kernels is medium. Similarly, the bottom row of Figure 4 shows the results for the "pottedplant" class. We again observe that the AP on the validation set and that on the test set show similar patterns. However, for the "pottedplant" class, the optimal p on the validation set is found to be 8, which implies that the intrinsic sparsity of the kernels is low.

When keeping the norm p fixed at 1, 10^6 and learning only the C/λ parameter, the ℓ_p MK-SVM/MK-FDA reduces to ℓ_1 and ℓ_∞ MK-SVM/MK-FDA, respectively. The APs and MAPs of the six MKL methods are shown in Table 1. The results in Table 1 demonstrate that learning the regularisation norm p indeed improves the performance of MK-FDA. However, it is worth noting that this improvement is achieved at a computational price of cross validating for an additional parameter, the regularisation norm p. In the case of MK-SVM, the learnt optimal p yields the same MAP as ℓ_∞ MK-SVM. However, this does not mean learning p is not bringing anything, because a priori we would not know that ℓ_∞ is the most appropriate norm. Instead, the conclusion we can draw from the MK-SVM results is that the sparsity of the base kernels, according to MK-SVM, is very low. Another observation from Table 1 is that in all three cases: ℓ_1, ℓ_∞ and ℓ_p tuned, MK-FDA outperforms MK-SVM on the majority of classes.

The pairwise alignment of the 14 kernel matrices w.r.t. the Frobenius dot product (Golub and van Loan, 1996), $\mathcal{A}(i,j) = \frac{<K_i,K_j>_F}{\|K_i\|_F\|K_j\|_F}$, is plotted in Figure 5, where subplot (a) shows the alignment of uncentred kernels and subplot (b) shows that of centred kernels. Kernel alignment has been used to analyse the property of a given kernel set (Nakajima et al., 2009; Kloft et al., 2011). We argue, however, that kernel alignment by itself cannot reveal completely the sparsity of a kernel set. First of all, as shown in Figure 5 (a) and (b), centring the kernel matrices changes significantly the alignment of the kernels. On the other hand, it is well known that centring does not change the effective weights learnt in MKL, since the shape of the data in the feature space is translation invariant. Second, kernel alignment does not take into account label information. For a multilabel dataset such as VOC07, all object classes share the same set of images (hence the same kernels),



(a)

(b)

Figure 5: VOC07: Alignment of the 14 kernels. (a) Spherically normalised kernels. (b) Spherically normalised and centred kernels. Note the scale difference between the two plots as indicated by the colorbars.

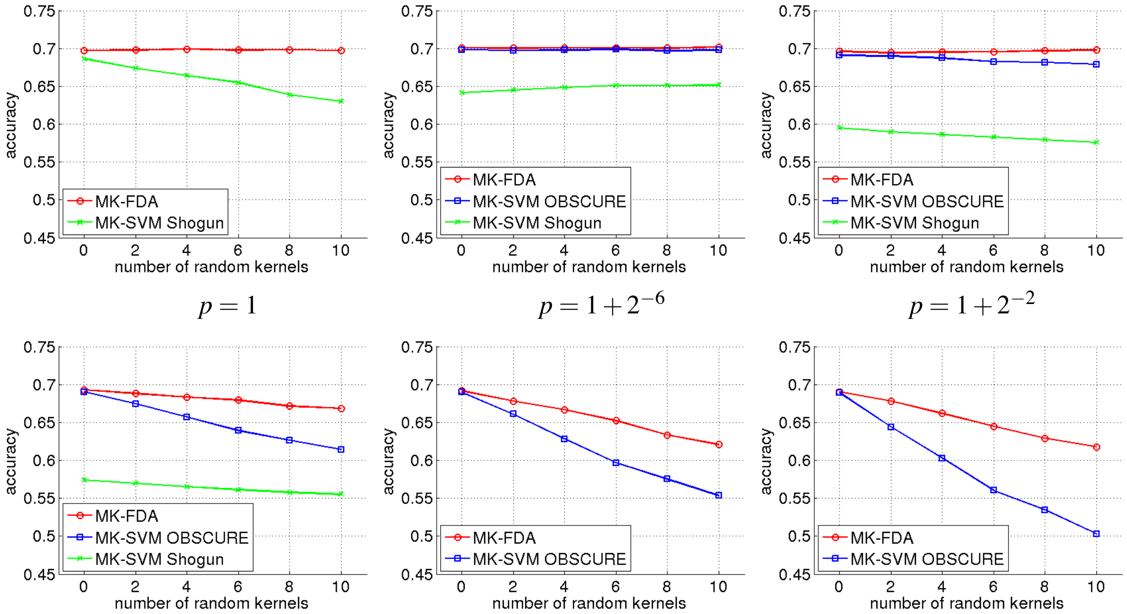
and the labels are different depending on which object class (i.e., which binary problem) is being considered. It is clear from Table 1 that for both ℓ_p MK-FDA and ℓ_p MK-SVM, the sparsity of the kernel set is class dependent. This means kernel alignment, which is class independent, by itself cannot be expected to identify the kernel set sparsity for all classes. Instead, we hypothesise that correlation analysis using projected labels (Braun et al., 2008) is probably more appropriate.

Finally, note that due to different parameter sets and different normalisation methods used (spherical normalisation in this paper while unit trace normalisation in Yan et al., 2010), the results on VOC07, Caltech101 and Flower17 reported in this paper are slightly different from those in Yan et al. (2010). However, the trends in the results remain the same, and all conclusions drawn from the results remain unchanged.

3.4 Caltech101

In the following three sections, we compare the proposed ℓ_p MK-FDA with several variants of ℓ_p MK-SVM on multiclass problems. We start in this section with the Caltech101 object recognition dataset. Caltech101 is a multiclass object recognition benchmark with 101 object categories. We follow the popular practice of using 15 randomly selected images per class for training, up to 50 randomly selected images per class for testing, and compute the average accuracy over all classes. This process is repeated 3 times, and we report the mean of the average accuracies on the test set that is achieved with the optimal parameter (C for MK-SVM and λ for MK-FDA). Validation is omitted, as the training of multiclass MK-SVM Shogun on this dataset can be very time consuming.

We generate 10 kernels in a similar way as in the VOC2007 experiments. In addition to these "informative" kernels, we also construct 10 RBF kernels from 10 sets of random vectors. To test the robustness of the MKL methods, we repeat the experiment 6 times. We start with only the informative kernels, and add two more random kernels in each subsequent run.



$p = 1$ $\quad$ $p = 1 + 2^{-6}$ $\quad$ $p = 1 + 2^{-2}$

$p = 2$ $\quad$ $p = 4$ $\quad$ $p = 10^6$

Figure 6: Caltech101: Accuracy comparison of three multiclass MKL methods.

Two multiclass MK-SVM implementations are compared against multiclass MK-FDA, namely, MK-SVM Shogun, and the recently proposed online MK-SVM algorithm OBSCURE (Orabona et al., 2010). For OBSCURE, the parameters are set to default values, except for the MKL norm p and the regularisation parameter C. In our experiments, C and λ are chosen from the same set of values that are logarithmically spaced over 4^{-5} to 4^4. We use the same set of 12 p values as in the VOC07 experiments. Note however that in OBSCURE, the MKL norm p is specified equivalently through the block norm r, where $r = 2p/(p+1)$. Moreover, OBSCURE requires that $r > 1$, so $p = r = 1$ in the set of p values is not used for OBSCURE.

The performance of the three MKL methods with various numbers of random kernels is illustrated in Figure 6, where we show results for six p values, covering the spectrum from highly sparsity-inducing norm, to uniform weighting. When p is large, MK-SVM Shogun does not converge within 24 hours, so its performance is not plotted for $p = 4$ and $p = 10^6$. We can see from Figure 6 that, when p is small, both MK-SVM OBSCURE and MK-FDA are robust to the added noise, and MK-FDA has a marginal advantage over OBSCURE (e.g., ~0.003 when $p = 1 + 2^{-6}$). When p is large, as expected, the performance of all three methods in general degrades as the number of random kernels increases. However, MK-FDA does so more gracefully than OBSCURE. On the other hand, both MK-FDA and MK-SVM OBSCURE outperform MK-SVM Shogun by a large margin on this multiclass problem.

3.5 Oxford Flower17

Oxford Flower17 is a multiclass dataset consisting of 17 categories of flowers with 80 images per category. This dataset comes with 3 predefined splits into training (17 × 40 images), validation (17 × 20 images) and test (17 × 20 images) sets. Moreover, Nilsback and Zisserman (2008) precomputed

method	accuracy and std. dev.	parameters tuned on val. set
product	85.5 ± 1.2	C
averaging	84.9 ± 1.9	C
MKL (SILP)	85.2 ± 1.5	C
MKL (Simple)	85.2 ± 1.5	C
CG-Boost	84.8 ± 2.2	C
LP-β	85.5 ± 3.0	$C_j, j = 1, \cdots, n$ and $\delta \in (0,1)$
LP-B	85.4 ± 2.4	$C_j, j = 1, \cdots, n$ and $\delta \in (0,1)$
ℓ_p MK-SVM Shogun	86.0 ± 2.4	p and C jointly
ℓ_p MK-SVM OBSCURE	85.6 ± 0.0	p and C jointly
ℓ_p MK-FDA	**87.2 ± 1.6**	p and λ jointly

Table 2: Flower17: Comparison of ten kernel fusion methods.

7 distance matrices using various features, and the matrices are available online.[8] We use these distance matrices and follow the same procedure as in Gehler and Nowozin (2009) to compute 7 kernels: $K_j(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp(-D_j(\mathbf{x}_i, \mathbf{x}_{i'})/\eta_j)$, where η_j is the mean of the pairwise distances for the j^{th} feature.

Table 2 compares ℓ_p MK-SVM Shogun, ℓ_p MK-SVM OBSCURE, ℓ_p MK-FDA, and 7 kernel combination techniques discussed in Gehler and Nowozin (2009). Note that these methods are directly comparable since they share the same kernel matrices and the same splits. For ℓ_p MK-SVM Shogun, ℓ_p MK-SVM OBSCURE and ℓ_p MK-FDA, the parameters p, C and λ are tuned on the validation set from the same sets of values as in the Caltech101 experiments. For the other seven methods, the corresponding entries in the table are taken directly from Gehler and Nowozin (2009), where: "product" and "sum" refer to the two simplest kernel combination methods, namely, taking the element-wise geometric mean and arithmetic mean of the kernels, respectively; "MKL (SILP)" and "MKL (Simple)" are essentially ℓ_1 MK-SVM; while "CG-Boost", "LP-β" and "LP-B" are three boosting based kernel combination methods.

We can see from Table 2 that the boosting based methods, although performing well on other datasets in Gehler and Nowozin (2009), fail to outperform the baseline methods "product" and "averaging". On the other hand, ℓ_p MK-FDA not only shows a considerable improvement over all the methods discussed in Gehler and Nowozin (2009), but also outperforms both ℓ_p MK-SVM Shogun and ℓ_p MK-SVM OBSCURE. Note that the optimal test accuracy over all combinations of parameters achieved by OBSCURE is comparable to that by MK-FDA. However, the performance on the validation set and that on the test set do not match as well for OBSCURE as for MK-FDA,[9] resulting in the lower test accuracy of OBSCURE. Parameters that need to be tuned on the validation set in these methods are also compared in Table 2.

3.6 Protein Subsellular Localisation

In the previous three sections, we have shown that on both binary and multiclass object recognition problems, ℓ_p MK-FDA tends to outperform ℓ_p MK-SVM by a small margin. In this section, we further compare ℓ_p MK-FDA and ℓ_p MK-SVM on a computational biology problem, namely, the prediction of subcellular localisation of proteins (Zien and Ong, 2007; Ong and Zien, 2008).

[8] The distance matrices can be found at `http://www.robots.ox.ac.uk/~vgg/research/flowers/index.html`.

[9] This is indicated by, for example, a lower Spearman or Kendall rank correlation coefficient.

norm p		1	32/31	16/15	8/7	4/3	2	4	8	16	∞
plant	MK-SVM	**8.18** ± 0.47	8.22 ± 0.45	8.20 ± 0.43	8.21 ± 0.42	8.43 ± 0.42	9.47 ± 0.43	11.00 ± 0.47	11.61 ± 0.49	11.91 ± 0.55	11.85 ± 0.60
	MK-FDA	10.86 ± 0.42	11.02 ± 0.43	10.96 ± 0.46	11.07 ± 0.43	10.85 ± 0.43	**10.69** ± 0.37	11.28 ± 0.45	11.28 ± 0.45	11.04 ± 0.43	11.35 ± 0.46
nonpl	MK-SVM	**8.97** ± 0.26	9.01 ± 0.25	9.08 ± 0.26	9.19 ± 0.27	9.24 ± 0.29	9.43 ± 0.32	9.77 ± 0.32	10.05 ± 0.32	10.23 ± 0.32	10.33 ± 0.31
	MK-FDA	10.93 ± 0.31	10.59 ± 0.33	10.91 ± 0.31	10.89 ± 0.32	**10.84** ± 0.31	11.00 ± 0.33	12.12 ± 0.41	12.12 ± 0.41	11.81 ± 0.38	12.15 ± 0.41
psortNeg	MK-SVM	9.99 ± 0.35	9.91 ± 0.34	**9.87** ± 0.34	10.01 ± 0.34	10.13 ± 0.33	11.01 ± 0.32	12.20 ± 0.32	12.73 ± 0.34	13.04 ± 0.33	13.33 ± 0.35
	MK-FDA	9.89 ± 0.34	10.07 ± 0.36	9.95 ± 0.35	9.87 ± 0.37	9.75 ± 0.39	**9.74** ± 0.37	11.39 ± 0.35	11.25 ± 0.34	11.27 ± 0.35	11.50 ± 0.35
psortPos	MK-SVM	13.07 ± 0.66	**13.01** ± 0.63	13.41 ± 0.67	13.17 ± 0.62	13.25 ± 0.61	14.68 ± 0.67	15.55 ± 0.72	16.43 ± 0.81	17.36 ± 0.83	17.63 ± 0.80
	MK-FDA	**12.59** ± 0.75	13.16 ± 0.80	13.07 ± 0.80	13.34 ± 0.80	13.45 ± 0.74	13.63 ± 0.70	16.86 ± 0.85	16.37 ± 0.89	16.56 ± 0.87	16.94 ± 0.84

Table 3: Protein Subcellular Localisation: comparing ℓ_p MK-FDA and ℓ_p MK-SVM w.r.t. prediction error and its standard error. Prediction error is measured as $1 -$ average MCC in percentage.

The protein subcellular localisation problem contains 4 datasets, corresponding to 4 different sets of organisms: plant (*plant*), non-plant eukaryotes (*nonpl*), Gram-positive (*psortPos*) and Gram-negative bacteria (*psortNeg*). Each of the 4 datasets can be considered as a multiclass classification problem, with the number of classes ranging between 3 and 5. For each dataset, 69 kernels that capture diverse aspects of protein sequences are available online.[10] We download the kernel matrices and follow the experimental setup in Kloft et al. (2011) to enable a direct comparison. More specifically, for each dataset, we first multiplicatively normalise the kernel matrices. Then for each of the 30 predefined splits, we use the first 20% of examples for testing and the rest for training.

In Kloft et al. (2011), the multiclass problem associated with each dataset is decomposed into binary problems using the one-vs-rest strategy. This is not necessary in the case of FDA, since FDA by its natures handles both binary and multiclass problems in a principled fashion. For each dataset, we consider the same set of values for the norm p as in Kloft et al. (2011): $\{1, 32/31, 16/15, 8/7, 4/3, 2, 4, 8, \infty\}$. In Kloft et al. (2011), the regularisation constant C for MK-SVM is taken from a set of 9 values: $\{1/32, 1/8, 1/2, 1, 2, 4, 8, 32, 128\}$. In our experiments, the regularisation constant λ for MK-FDA is also taken from a set of 9 values, and the values are logarithmically spaced over 10^{-8} to 10^{0}.

Again following Kloft et al. (2011), for each p/λ combination, we evaluate the performance of ℓ_p MK-FDA w.r.t. average (over the classes) Matthews correlation coefficient (MCC), and report in Table 3 the average of $1 - MCC$ over 30 splits and its standard error. For ease of comparison, we also show in Table 3 the performance of ℓ_p MK-SVM as reported in Kloft et al. (2011).

[10] The kernels can be downloaded at `http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc`.

Table 3 demonstrates that overall the performance of ℓ_p MK-FDA lags behind that of ℓ_p MK-SVM, except on *psortNeg* and on *psortPos*, where it has a small edge. Another observation is that the optimal norm p identified by MK-SVM does not necessarily agree with that by MK-FDA. On *psortPos* they are in close agreement and both methods favour sparsity-inducing norms. On *plant*, *nonpl* and *psortNeg*, on the other hand, the norms picked by MK-FDA are larger than those picked by MK-SVM. Note, however, that this observation can be slightly misleading, because on the latter three datasets, the performance curve of ℓ_p MK-FDA is quite "flat" in the area of optimal performance. As a result, the optimal norm estimated may not be stable.

3.7 Training Speed

In this section we provide an empirical analysis of the efficiency of the wrapper-based ℓ_p MK-FDA and various implementations of ℓ_p MK-SVM. We use $p = 1$ (or in some cases $1+2^{-5}$, $1+2^{-6}$) and $p = 2$ as examples of sparse/non-sparse MKL respectively,[11] and study the scalability of MK-FDA and MK-SVM w.r.t. the number of examples and the number of kernels, on both binary and multiclass problems.

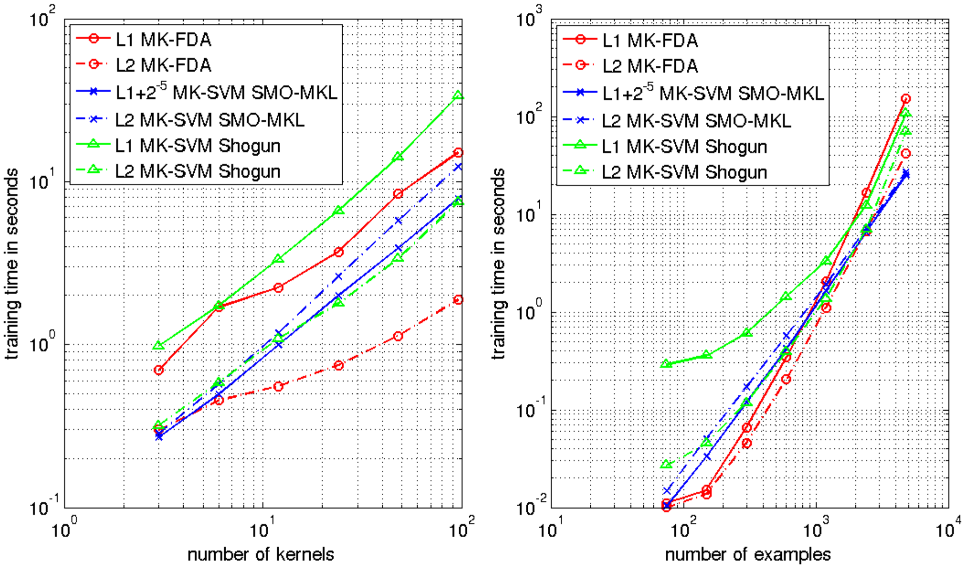
3.7.1 BINARY CASE: VOC2007

We first compare on the VOC2007 dataset the training speed of three binary MKL methods: the wrapper based binary ℓ_p MK-FDA in Section 2.1, the binary ℓ_p MK-SVM Shogun implementation (Sonnenburg et al., 2006, 2010), and the SMO-MKL in Vishwanathan et al. (2010). In the experiments, interleaved optimisation and analytical update of $\boldsymbol{\beta}$ are used for MK-SVM Shogun.

We first fix the number of training examples to 1000, and vary the number of kernels from 3 to 96. We record the time taken to learn the kernel weights, and average over the 20 binary problems. Figure 7 (a) shows the training time of the six MKL algorithms as functions of the number of kernels. Next, we fix the number of kernels to 14, and vary the number of examples from 75 to 4800. Similarly, in Figure 7 (b) we plot the average training time as functions of the number of examples.

Figure 7 (a) demonstrates that for small/medium sized problems, when a sparsity-inducing norm is used, SMO-MKL is the most efficient; while when $p = 2$, MK-FDA can be significantly faster than the competing methods. On the other hand, when training efficiency is measured as a function of the number of examples, there is no clear winner, as indicated in Figure 7 (b). However, the trends in Figure 7 (b) suggest that on large scale problems, SMO-MKL is likely to be more efficient than MK-FDA and MK-SVM Shogun. In both cases, MK-FDA has a comparable or better efficiency than MK-SVM Shogun, despite the fact that MK-SVM Shogun uses the interleaved algorithm while MK-FDA employs the somewhat wasteful wrapper-based implementation. Again, this trend is likely to flip over on large scale problems. For such problems, one can adopt either the square loss counterpart of the interleaved algorithm, or the square loss counterpart of the SMO-MKL algorithm, or the limited memory quasi-Newton method, to improve the efficiency of ℓ_p MK-FDA, as discussed in Section 2.3.

[11] Both SMO-MKL and OBSCURE require that $p > 1$. Moreover, SMO-MKL is numerically unstable when $p = 1+2^{-6}$. As a result, we use $p = 1+2^{-5}$ and $p = 1+2^{-6}$ as sparsity-inducing norms for SMO-MKL and OBSCURE, respectively.



(a) (b)

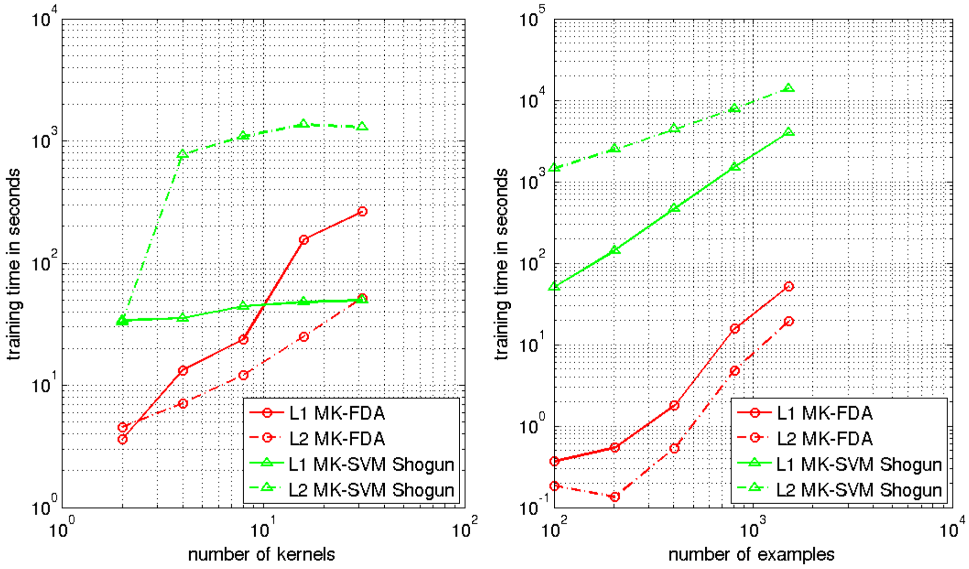
Figure 7: Training speed on a binary problem: VOC2007. (a) Training time vs. number of kernels, where number of examples is fixed at 1000. (b) Training time vs. number of examples, where number of kernels is fixed at 14. $\lambda = 1$ for MK-FDA, and $C = 1$ for MK-SVM Shogun and MK-SVM OBSCURE.

3.7.2 MULTICLASS CASE: CALTECH101

Next we compare three multiclass ℓ_p MKL algorithms on the Caltech101 dataset, namely, the wrapper-based multiclass ℓ_p MK-FDA in Section 2.2, multiclass ℓ_p MK-SVM Shogun, and MK-SVM OBSCURE. We compare the first two methods following similar protocols as in the binary case. In Figure 8 (a) we show the average training time over the 3 splits as functions of the number of kernels (from 2 to 31) when the number of examples is fixed to 101 (one example per class); while plotted in Figure 8 (b) is the average training time as functions of the number of examples (from 101 to 1515, that is, from one example per class to 15 examples per class) when the number of kernels is fixed to 10.

Figure 8 shows that on small/medium sized multiclass problems, MK-FDA is in most cases one or two orders of magnitude faster than MK-SVM Shogun. The only exception is that as the number of kernels increases, the efficiency of ℓ_1 MK-SVM Shogun degrades more gracefully than ℓ_1 MK-FDA, and eventually overtakes. Another observation from both Figure 7 and Figure 8 is that, ℓ_2 MK-FDA tends to be more efficient than ℓ_1 MK-FDA, despite the fact that in the outer subproblem, the LP solver employed in ℓ_1 MK-FDA is slightly faster than the QCLP solver in ℓ_2 MK-FDA. This is because ℓ_1 MK-FDA usually takes a few tens of iterations to converge, while the ℓ_2 version typically takes less than 5. This difference in the number of iterations reverses the efficiency advantage of LP over QCLP.

Due to its online nature, the efficiency of OBSCURE has to be measured differently to allow a fair comparison. The OBSCURE algorithm is a two-stage algorithm, and each stage involves an iterative process with a parameter $T1/T2$ controlling the number of iterations. In general the



(a) (b)

Figure 8: Training speed on a multiclass problem: Caltech101. MK-FDA vs. MK-SVM Shogun. (a) Training time vs. number of kernels, where number of examples is fixed at 101. (b) Training time vs. number of examples, where number of kernels is fixed at 10. $\lambda = 1$ for MK-FDA, and $C = 1$ for MK-SVM Shogun.

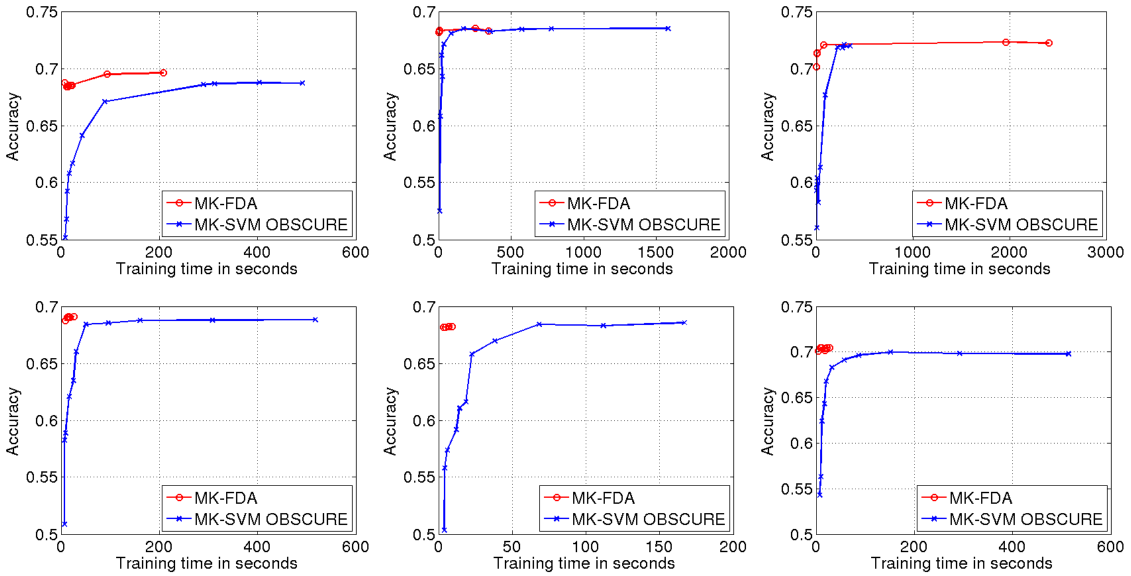


Figure 9: Training speed on a multiclass problem: Caltech101. MK-FDA vs. MK-SVM OBSCURE. Top row: $p = 1 + 2^{-6}$. Bottom row: $p = 2$. The three columns correspond to the three splits. 10 kernels and $101 \times 15 = 1515$ training examples.

larger the values of $T1$ and $T2$, the longer it takes to train, but the more accurate the learnt model. We set $T1 = T2 = T$ and vary T in a set of 11 values from 2^0 to 2^{10}. This allows us to plot a curve of model accuracy against training time. For MK-FDA, the similar curve can be plotted by varying the convergence threshold ε in a set of 11 values: $\{2^0, \cdots, 2^{-6}, 10^{-2}, \cdots, 10^{-5}\}$. Note that the regularisation parameters (λ for MK-FDA and C for OBSCURE) are set to values that yield the highest classification accuracy.

The resulting time-accuracy curves for all 3 splits of the dataset are presented in Figure 9, where the top row corresponds to the case of $p = 1 + 2^{-6}$ and the bottom row to $p = 2$, and each column corresponds to one split. It is evident that MK-FDA typically reaches its optimum faster than OBSCURE, especially in the case of $p = 2$. Moreover, the optimum achieved by MK-FDA is at least as accurate as that by OBSCURE, confirming our findings in Section 3.4. All the training time reported in this section is measured on a single core of an Intel Xeon E5520 2.27GHz processor.

4. Discussion: FDA vs. SVM

The empirical observation that MK-FDA tends to outperform MK-SVM on image categorisation datasets matches well with our experience with single kernel FDA and single kernel SVM on several other object/image/video classification benchmarks, including VOC2008, VOC2009, VOC2010,[12] Trecvid2008, Trecvid2009,[13] and ImageCLEF2010.[14] In this section, we discuss the connection between (MK-)SVM and (MK-)FDA from perspectives of both loss function and version space, and attempt to explain their different performance.

It is well known that many machine learning problems essentially boil down to function learning. In the supervised scenario, it is intuitive to learn the function by minimising the empirical loss for the given set of labelled input/output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^m$, with respect to some loss function. However, such an empirical risk minimisation principle is ill-posed and therefore does not generalise (Tikhonov and Arsenin, 1977; Vapnik, 1999). Regularisation tries to restore well-posedness of the learning problem, by restricting the complexity of the function set over which the empirical loss is minimised. By (implicitly) mapping the data into a high dimensional feature space, this can be conveniently done in the form of Tikhonov regularisation:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} V(f(\phi(\mathbf{x}_i)), y_i), \tag{27}$$

where $\phi(\mathbf{x}_i)$ is the mapping to the feature space, $f(\phi(\mathbf{x}_i)) = \mathbf{w}^T\phi(\mathbf{x}_i)$ is the linear function to be learnt, the complexity of the function set is regularised by $\frac{1}{2}||\mathbf{w}||^2$, and $V(\cdot,\cdot)$ measures the empirical loss. Learning machines with the form of Equation (27) are collectively termed regularised kernel machines, a name capturing the two key aspects of them: regularisation, and kernel mapping. Note that in the formulation above, the unregularised bias term b in standard SVM is absent from the linear function. As shown in Keerthi and Shevade (2003); Poggio et al. (2004), the two formulations, with and without b, can be made equivalent by transforming the kernel function.

The setting in Equation (27) is very general, in the sense that many state-of-the-art machine learning techniques can be realised by plugging in different loss functions. For example, the hinge loss $V(f(\phi(\mathbf{x})), y) = (1 - yf(\phi(\mathbf{x})))_+$, where $(\cdot)_+ = \max(\cdot, 0)$, gives rise to the well known SVM,

[12] More information on VOC can be found at `http://pascallin.ecs.soton.ac.uk/challenges/VOC`.

[13] More information on Trecvid can be found at `http://www-nlpir.nist.gov/projects/trecvid`.

[14] More information on ImageCLEF can be found at `http://www.imageclef.org/2010`.

probably the most popular learning machine in the past ten years. On the other hand, along with the success of SVM, regularised kernel machines using the square loss $V(f(\phi(\mathbf{x})),y) = (y - f(\phi(\mathbf{x})))^2$ have emerged several times under various names, including: regularised networks (RN) (Girosi et al., 1995; Evgeniou et al., 2000), regularised least squares (RLS) (Rifkin, 2002), kernel ridge regression (KRR) (Saunders et al., 1998; Hastie et al., 2002), least squares support vector machines (LSSVM) (Suykens and Vandewalle, 1999; Gestel et al., 2002), proximal support vector machines (PSVM) (Fung and Mangasarian, 2001). In particular, shortly after the proposal of kernel FDA (Mika et al., 1999; Baudat and Anouar, 2000), its regularised version was shown to be yet another equivalent formulation (Duda et al., 2000; Rifkin, 2002; Gestel et al., 2002).

There is a long list of literature which compares the performance of FDA and SVM, for example, Mika (2002), Rifkin (2002), Cai et al. (2007) and Ye et al. (2008), with most of them reporting both methods yield virtually identical performance, and the rest claiming there is a small advantage towards one method or the other. It is speculated in Mika et al. (1999) that the superior performance of FDA over SVM in their experiments is due to the fact that FDA uses all training examples in the test stage while SVM uses only the support vectors. However, a more elegant way of explaining the different performance of SVM and FDA is probably from the perspective of version space. Version space is the space of all consistent hypotheses, that is, all $\mathbf{w}$'s that correspond to hyperplanes with zero training error (Rujan, 1997). Note that with a full rank kernel matrix, linear separability in the feature space and therefore the existence of version space is guaranteed. It is shown in Rujan (1997) that the optimal hyperplane in the Bayes sense, which requires the knowledge of the joint distribution on $\mathcal{X} \times \mathcal{Y}$ thus not obtainable in practice, is arbitrarily close (with increasing training sample size) to the centre of mass of the version space.

Algorithms that explicitly approximate the Bayes point were later termed Bayes point machine (BPM) in Herbrich et al. (2001). Herbrich et al. (2001) also prove that the hyperplane found by SVM corresponds to the centre of the largest inscribed ball of the version space. In this light, SVM can be viewed as an approximation to BPM. This approximation is reasonable if the version space is regularly shaped, but can be weak otherwise (Rujan, 1997; Herbrich et al., 2001; Mika, 2002). For example, experiments in Herbrich et al. (2001) show that BPM consistently outperforms SVM. Recently, an ellipsoid SVM was proposed (Momma et al., 2010), where the idea is to improve the approximation to the Bayes point by using the centre of the largest inscribed ellipsoid, instead of that of the ball. We conjecture that for certain kernels (e.g., kernels generated using local descriptors and bag-of-words model, as those used in image categorisation problems), due to the different loss functions used, the hyperplane given by FDA is closer to the Bayes point than that given by SVM, resulting in the superior performance of (MK-)FDA in our experiments. How to decide without a validation process whether (MK-)FDA or (MK-SVM) is more suitable for a given kernel (set), and how to incorporate explicit BPM approximation into MKL, are interesting research directions for the future.

5. Conclusions

In this paper we have incorporated latest advances in both non-sparse MKL formulation and MKL optimisation techniques into MK-FDA. We have presented a non-sparse version of MK-FDA based on an ℓ_p norm regularisation of kernel weights, and have discussed several of its reformulations and associated optimisation strategies, including wrapper and interleaved algorithms for its saddle point formulation, and an SMO-based scheme for its dual formulation.

We carried out extensive evaluation on six datasets from various application areas. Our results indicate that the optimal norm p, and therefore the "intrinsic sparsity" of the base kernels, can be estimated on an independent validation set. This estimation can be exploited in many practical applications where there is no prior knowledge on how informative the channels are. We have also compared closely the performance of ℓ_p MK-FDA and that of several variants of ℓ_p MK-SVM. On object and image categorisation problems, MK-FDA tends to have a small advantage. This observation is consistent with our findings elsewhere regarding the performance of single kernel FDA/SVM. In terms of training time, the wrapper-based MK-FDA implementation has similar or favourable efficiency on small to medium sized problems when compared against state-of-the-art MKL techniques. On large scale problems, alternative optimisation strategies discussed in the paper should be employed to improve the efficiency and scalability of MK-FDA.

Finally, we have provided a discussion on the connection between (MK-)FDA and (MK-)SVM from the perspectives of both loss function and version space, under the unified framework of regularised kernel machines.

Acknowledgments

We would like to thank Sören Sonnenburg, Gunnar Rätsch, Alexander Binder for their help on the Shogun toolbox, Jianhui Chen and Jieping Ye for help on their discriminant MKL package, and Marius Kloft for helpful discussions. We would also like to thank the anonymous reviewers and the editors for their valuable comments and suggestions. The work presented in this paper is supported by the EPSRC/UK grant EP/F069626/1 ACASVA Project.

Appendix A. Multiclass ℓ_p MK-FDA Saddle Point Formulation

In this appendix, we first derive the saddle point formulation of multiclass MKL for a general convex loss. Multiclass MK-FDA saddle point problem is then derived as a special case of it. Using the output encoding scheme in Equation (18), multiclass MKL for a general convex loss function $V(\xi_{ik}, h_{ik})$ can be stated as:

$$\begin{aligned} \min_{\mathbf{w}_{jk},\xi_{ik},\boldsymbol{\beta}} & \sum_{k=1}^{c}\left(\frac{1}{2}\sum_{j=1}^{n}\frac{||\mathbf{w}_{jk}||^2}{\beta_j}+C\sum_{i=1}^{m}V(\xi_{ik},h_{ik})\right) \\ \text{s.t.} \quad & \sum_{j=1}^{n}\mathbf{w}_{jk}^T\phi_j(\mathbf{x}_i)=\xi_{ik},\ \forall i,\ \forall k;\ \ \boldsymbol{\beta}\geq\mathbf{0};\ \ ||\boldsymbol{\beta}||_p^2\leq 1. \end{aligned} \tag{28}$$

We build the Lagrangian of Equation (28):

$$\begin{aligned} \mathcal{L} = & \sum_{k=1}^{c}\left(\frac{1}{2}\sum_{j=1}^{n}\frac{||\mathbf{w}_{jk}||^2}{\beta_j}+C\sum_{i=1}^{m}V(\xi_{ik},h_{ik})\right)+\zeta(\frac{1}{2}||\boldsymbol{\beta}||_p^2-\frac{1}{2}) \\ & -\sum_{k=1}^{c}\sum_{i=1}^{m}\alpha_{ik}\left(\sum_{j=1}^{n}\mathbf{w}_{jk}^T\phi_j(x_i)-\xi_{ik}\right), \end{aligned}$$

set to zero the derivatives of the Lagrangian w.r.t. $\mathbf{w}_{jk}$, and substitute back. After some rearrangements we have:

$$\mathcal{L} = \sum_{k=1}^{c}\left(C\sum_{i=1}^{m}V(\xi_{ik},h_{ik}) + \sum_{i=1}^{m}\alpha_{ik}\xi_{ik} - \frac{1}{2}\sum_{j=1}^{n}\boldsymbol{\alpha}_k^T\beta_j K_j\boldsymbol{\alpha}_k\right) + \zeta(\frac{1}{2}||\boldsymbol{\beta}||_p^2 - \frac{1}{2}),$$

where $\boldsymbol{\alpha}_k = (\alpha_{1k},\cdots,\alpha_{mk})^T$. Following Theorem 1 of Kloft et al. (2011) it can be shown that at the optimum $||\boldsymbol{\beta}||_p^2 = 1$. Using this fact we arrive at the multiclass MKL saddle point problem for a general loss function:

$$\min_{\xi_{ik},\boldsymbol{\beta}}\max_{\alpha_{ik}}\sum_{k=1}^{c}\left(C\sum_{i=1}^{m}V(\xi_{ik},h_{ik}) + \sum_{i=1}^{m}\alpha_{ik}\xi_{ik} - \frac{1}{2}\sum_{j=1}^{n}\boldsymbol{\alpha}_k^T\beta_j K_j\boldsymbol{\alpha}_k\right) \tag{29}$$
$$\text{s.t.}\quad \boldsymbol{\beta}\geq\mathbf{0};\ \ ||\boldsymbol{\beta}||_p^2\leq 1.$$

At this point any convex loss function can be plugged into Equation (29). Take the square loss $V(\xi_{ik},h_{ik}) = \frac{1}{2}(\xi_{ik}-h_{ik})^2$ as an example. Setting to zero the derivatives of $\mathcal{L}$ w.r.t. ξ_{ik} we have $\xi_{ik} = h_{ik} - \alpha_{ik}/C$. Plugging this into Equation (29) and rearranging we arrive at the multiclass MKL saddle point problem for square loss, that is, multiclass multiple kernel regularised least squares (MK-RLS):

$$\min_{\boldsymbol{\beta}}\max_{\boldsymbol{\alpha}_k}\sum_{k=1}^{c}\left(\mathbf{h}_k^T\boldsymbol{\alpha}_k - \frac{1}{2C}\boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k - \frac{1}{2}\sum_{j=1}^{n}\boldsymbol{\alpha}_k^T\beta_j K_j\boldsymbol{\alpha}_k\right) \tag{30}$$
$$\text{s.t.}\quad \boldsymbol{\beta}\geq\mathbf{0};\ \ ||\boldsymbol{\beta}||_p^2\leq 1,$$

where the c classes are coupled through the common set of kernel weights $\boldsymbol{\beta}$. By making substitutions $\boldsymbol{\alpha}_k \to \frac{C}{2}\boldsymbol{\alpha}_k$ and then $C \to \frac{1}{\lambda}$, it directly follows that the MK-RLS in Equation (30) is equivalent to the MK-FDA in Equation(19).

Appendix B. Multiclass ℓ_p MK-FDA Dual Formulation

In this appendix, we derive the dual formulation of multiclass MK-FDA. We again consider multiclass MKL with a general convex loss, but following Vishwanathan et al. (2010) this time we impose the norm constraint in the form of Tikhonov regularisation instead of Ivanov regularisation:

$$\min_{\mathbf{w}_{jk},\xi_{ik},\boldsymbol{\beta}}\sum_{k=1}^{c}\left(\frac{1}{2}\sum_{j=1}^{n}\frac{||\mathbf{w}_{jk}||^2}{\beta_j} + C\sum_{i=1}^{m}V(\xi_{ik},h_{ik})\right) + \frac{\mu}{2}||\boldsymbol{\beta}||_p^2 \tag{31}$$
$$\text{s.t.}\quad \sum_{j=1}^{n}\mathbf{w}_{jk}^T\phi_j(\mathbf{x}_i) = \xi_{ik},\ \forall i,\ \forall k;\ \ \boldsymbol{\beta}\geq\mathbf{0}.$$

Note however that the switching from Ivanov to Tikhonov regularisation is not essential for the derivation in the following. The dual program for Ivanov regularisation in Equation (28) can be derived in a similar way.

Building the Lagrangian of Equation (31):

$$\mathcal{L} = \sum_{k=1}^{c}\left(\frac{1}{2}\sum_{j=1}^{n}\frac{||\mathbf{w}_{jk}||^2}{\beta_j} + C\sum_{i=1}^{m}V(\xi_{ik},h_{ik})\right) + \frac{\mu}{2}||\boldsymbol{\beta}||_p^2 - \sum_{j=1}^{n}\gamma_j\beta_j$$
$$- \sum_{k=1}^{c}\sum_{i=1}^{m}\alpha_{ik}\left(\sum_{j=1}^{n}\mathbf{w}_{jk}^T\phi_j(x_i) - \xi_{ik}\right),$$

and setting to zero the derivatives w.r.t. β_j, we have:

$$\mu(\sum_{j=1}^{n} \beta_j^p)^{\frac{2}{p}-1} \beta_j^{p-1} = \gamma_j + \frac{1}{2}\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k. \tag{32}$$

Multiplying both sides of Equation (32) by β_j and then taking summation over j gives us:

$$\mu\|\boldsymbol{\beta}\|_p^2 = \sum_{j=1}^{n} \beta_j(\gamma_j + \frac{1}{2}\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k),$$

or equivalently:

$$\sum_{j=1}^{n} \gamma_j \beta_j = -\frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T \beta_j K_j \boldsymbol{\alpha}_k + \mu\|\boldsymbol{\beta}\|_p^2. \tag{33}$$

On the other hand, raise both sides of Equation (32) to power $\frac{p}{p-1}$ and then take summation over j, we have:

$$\mu\|\boldsymbol{\beta}\|_p^2 = \frac{1}{\mu}\left\|\left(\gamma_j + \frac{1}{2}\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k\right)_{j=1}^{n}\right\|_q^2, \tag{34}$$

where $q = \frac{p}{p-1}$ is the dual norm of p.

Now let us set the derivatives of $\mathcal{L}$ w.r.t. $\mathbf{w}_{jk}$ also to zero, and substitute the result and Equation (33), Equation (34) back into $\mathcal{L}$. Using the fact that $\gamma_j = 0$ at the optimum (Vishwanathan et al., 2010), and after some rearrangements we arrive at:

$$\mathcal{L} = \sum_{k=1}^{c}\left(C\sum_{i=1}^{m} V(\xi_{ik}, h_{ik}) + \sum_{i=1}^{m} \alpha_{ik}\xi_{ik}\right) - \frac{1}{8\mu}\left\|\left(\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k\right)_{j=1}^{n}\right\|_q^2. \tag{35}$$

At this point any convex loss function can be plugged into Equation (35) to recover the corresponding multiclass MKL dual. We again take the square loss $V(\xi_{ik}, h_{ik}) = \frac{1}{2}(\xi_{ik} - h_{ik})^2$ as an example. Setting to zero the derivatives of $\mathcal{L}$ w.r.t. ξ_{ik} we have $\xi_{ik} = h_{ik} - \alpha_{ik}/C$. Plugging this into Equation (35) and rearranging we arrive at the multiclass MK-RLS dual problem:

$$\max_{\boldsymbol{\alpha}_k} \sum_{k=1}^{c}\left(\mathbf{h}_k^T \boldsymbol{\alpha}_k - \frac{1}{2C}\boldsymbol{\alpha}_k^T \boldsymbol{\alpha}_k\right) - \frac{1}{8\mu}\left\|\left(\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k\right)_{j=1}^{n}\right\|_q^2. \tag{36}$$

Unlike the saddle point formulation in Equation (30), the kernel weights $\boldsymbol{\beta}$ have been eliminated from Equation (36). Despite this, Equation (30) and Equation (36) are equivalent, in the sense that for any given C there exist a μ (and vice versa) such that the optimal solutions to both problems are identical (Kloft et al., 2011).

Finally, substituting Equation (34) and $\gamma_j = 0$ into Equation (32), we show that once the optimal $\boldsymbol{\alpha}_k$ are found by solving Equation (36), the kernel weights $\boldsymbol{\beta}$ are given by:

$$\beta_j = \frac{1}{2\mu}\left(\sum_{j=1}^{n}(\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k)^q\right)^{\frac{1}{q}-\frac{1}{p}}(\sum_{k=1}^{c} \boldsymbol{\alpha}_k^T K_j \boldsymbol{\alpha}_k)^{\frac{q}{p}}.$$

References

F. Bach and G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *International Conference on Machine Learning*, 2004.

G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

O. Bousquet and D. Herrmann. On the complexity of learning the kernel matrix. In *Advances in Neural Information Processing Systems*, 2003.

M. Braun, J. Buhmann, and K. Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9:1875–1908, 2008.

D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *International Conference on Data Mining*, 2007.

H. Cai, K. Mikolajczyk, and J. Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):338–352, 2011.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

C. Cortes, M . Mohri, and A. Rostamizadeh. L2 regularization for learning kernels. In *Uncertainty in Artificial Intelligence*, 2009.

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, 2002.

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C Bray. Visual categorization with bags of keypoints. In *ECCV workshop on Statistical Learning in Computer Vision*, 2004.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2000.

M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

G. Fung and O. L. Mangasarian. Proximal support vector machine classifier. In *International Conference on Knowledge Discovery and Data Mining*, 2001.

P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision*, 2009.

J. Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, 2008.

T. Gestel, J. Suykens, G. Lanckriet, A. Lambrechts, B. Moor, and J. Vandewalle. Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis. *Machine Learning*, 14(5):1115–1147, 2002.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

G. Golub and C. van Loan. *Matrix Computations*. John Hopkins University Press, third edition, 1996.

K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2002.

R. Herbrich, T. Graeple, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

R. Hettich and K. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

T. Joachims. *Making Large-Scale Support Vector Machine Learning Practical*. MIT Press, Cambridge, MA, 1988.

S. Keerthi and S. Shevade. Smo algorithm for least squares svm formulations. *Neural Computation*, 15(2):487–507, 2003.

S. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *International Conference on Machine Learning*, 2006.

M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. In *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Efficient and accurate lp-norm mkl. In *Advances in Neural Information Processing Systems*, 2009.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning teh kernel matrix with semi-definite programming. In *International Conference on Machine Learning*, 2002.

G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *International Conference on Computer Vision and Pattern Recognition*, 2006.

D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

J. Lopez and J. Suykens. First and second order smo algorithms for ls-svm classifiers. *Neural Processing Letters*, 33(1):31–44, 2011.

D. Lowe. Distincetive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

S. Mika. Kernel fisher discriminants. PhD Thesis, University of Technology, Berlin, Germany, 2002.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller. Fisher discriminant analysis with kernels. In *IEEE Signal Processing Society Workshop: Neural Networks for Signal Processing*, 1999.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

M. Momma, K. Hatano, and H. Nakayama. Ellipsoidal support vector machines. In *Asian Conference on Machine Learning*, 2010.

S. Nakajima, A. Binder, C. Muller, W. Wojcikiewicz, M. Kloft, U. Brefeld, K. Müller, and M. Kawanabe. Multiple kernel learning for object classification. Technical Report on Information-Based Induction Sciences, 2009.

M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

C. Ong and A. Zien. An automated combination of kernels for predicting protein subcellular localization. In *Workshop on Algorithms in Bioinformatics*, 2008.

C. Ong, A. Smola, and R. C. Williamson. Hyperkernels. In *Advances in Neural Information Processing Systems*, 2003.

F. Orabona and L. Jie. Ultra-fast optimization algorithm for sparse multi kernel learning. In *International Conference on Machine Learning*, 2011.

F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kerenl learning. In *International Conference on Computer Vision and Pattern Recognition*, 2010.

T. Poggio, S. Mukherjee, R. Rifkin, A. Rakhlin, and A. Verri. B. In *Conference on Uncertainty in Geometric Computations*, 2004.

A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

G. Rätsch. Robust boosting via convex optimization. PhD Thesis, University of Potsdam, Potsdam, Germany, 2001.

R. Rifkin. Everything old is new again: a fresh look at historical approaches in machine learning. PhD Thesis, Massachusetts Institute of Technology, Boston, USA, 2002.

P. Rujan. Playing billiard in version space. *Neural Computation*, 9:99–122, 1997.

K. Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *International Conference on Machine Learning*, 1998.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

B. Schölkopf, A. Smola, and K. Müller. Kernel principal component analysis. *Advances in Kernel Methods: Support Vector Learning*, pages 327–352, 1999.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

C. Snoek, M. Worring, J. Gemert, J. Geusebroek, and A. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia Conference*, 2006.

S. Sonnenburg, G. Rätsch, C. Schafer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien adn F. Bona, A. Binder, C. Gehl, and V. Franc. The shogun machine learning toolbox. *Journal of Machine Learning Research*, 11: 1799–1802, 2010.

J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. In *International Conference on Machine Learning*, 2008.

A. Tahir, J. Kittler, K. Mikolajczyk, F. Yan, K. Sande, and T. Gevers. Visual category recognition using spectral regression and kernel discriminant analysis. In *International Workshop on Subspace Methods*, 2009.

A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, Washington DC, 1977.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.

S. Vishwanathan, Z. Sun, and N. Theera-Ampornpunt. Multiple kernel learning and the smo algorithm. In *Advances in Neural Information Processing Systems*, 2010.

F. Yan, J. Kittler, K. Mikolajczyk, and A. Tahir. Non-sparse multiple kernel learning for fisher discriminant analysis. In *International Conference on Data Mining*, 2009a.

F. Yan, K. Mikolajczyk, J. Kittler, and A. Tahir. A comparison of l1 norm and l2 norm multiple kernel svms in image and video classification. In *International Workshop on Content-Based Multimedia Indexing*, 2009b.

F. Yan, K. Mikolajczyk, M. Barnard, H. Cai, and J. Kittler. Lp norm multiple kernel fisher discriminant analysis for object and image categorisation. In *International Conference on Computer Vision and Pattern Recognition*, 2010.

J. Ye, S. Ji, and J. Chen. Multi-class discriminant kernel learning via convex programming. *Journal of Machine Learning Research*, 9:719–758, 2008.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

A. Zien and C. Ong. Multiclass multiple kernel learning. In *International Conference on Machine Learning*, 2007.

Journal of Machine Learning Research 13 (2012) 643-669 Submitted 6/11; Revised 11/11; Published 3/12

Learning Algorithms for the Classification Restricted Boltzmann Machine

Hugo Larochelle HUGO.LAROCHELLE@USHERBROOKE.CA
Université de Sherbrooke
2500, boul. de l'Université
Sherbrooke, Québec, Canada, J1K 2R1

Michael Mandel MANDELM@IRO.UMONTREAL.CA
Razvan Pascanu PASCANUR@IRO.UMONTREAL.CA
Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Département d'informatique et de recherche opérationnelle
Université de Montréal
2920, chemin de la Tour
Montréal, Québec, Canada, H3T 1J8

Editor: Daniel Lee

Abstract

Recent developments have demonstrated the capacity of restricted Boltzmann machines (RBM) to be powerful generative models, able to extract useful features from input data or construct deep artificial neural networks. In such settings, the RBM only yields a preprocessing or an initialization for some other model, instead of acting as a complete supervised model in its own right. In this paper, we argue that RBMs can provide a self-contained framework for developing competitive classifiers. We study the Classification RBM (ClassRBM), a variant on the RBM adapted to the classification setting. We study different strategies for training the ClassRBM and show that competitive classification performances can be reached when appropriately combining discriminative and generative training objectives. Since training according to the generative objective requires the computation of a generally intractable gradient, we also compare different approaches to estimating this gradient and address the issue of obtaining such a gradient for problems with very high dimensional inputs. Finally, we describe how to adapt the ClassRBM to two special cases of classification problems, namely semi-supervised and multitask learning.

Keywords: restricted Boltzmann machine, classification, discriminative learning, generative learning

1. Introduction

The restricted Boltzmann machine (RBM) is a probabilistic model that uses a layer of hidden binary variables or units to model the distribution of a visible layer of variables. It has been successfully applied to problems involving high dimensional data such as images (Hinton et al., 2006; Larochelle et al., 2007) and text (Welling et al., 2005; Salakhutdinov and Hinton, 2007; Mnih and Hinton, 2007). In this context, two approaches are usually followed. First, an RBM is trained in an unsupervised manner to model the distribution of the inputs (possibly more than one RBM could be trained, stacking them on top of each other (Hinton et al., 2006)). Then, the RBM is used in one of two ways: either its hidden layer is used to preprocess the input data by replacing it with the representation given by the hidden layer, or the parameters of the RBM are used to initialize a feedforward neural network. In both cases, the RBM is paired with some other learning algorithm (the classifier using the preprocessed inputs or the neural network) to solve the supervised learning problem at hand. This approach unfortunately requires one to tune both sets of hyper-parameters (those of the RBM and of the other learning algorithm) at the same time. Moreover, since the RBM is trained in an unsupervised manner, it is blind to the nature of the supervised task that needs to be solved and provides no guarantees that the information extracted by its hidden layer will be useful.

In this paper, we argue that RBMs can provide a self-contained and competitive framework for solving supervised learning problems. Based on the Classification Restricted Boltzmann Machine (ClassRBM), the proposed approach and learning algorithms address both aforementioned issues. Indeed, by relying only on the RBM, the number of hyper-parameters that one needs to tune will be relatively smaller, and by modelling the *joint* distribution of the input and target, the ClassRBM will be encouraged to allocate some of its capacity at modelling their relationship as well as the relationships between the input variables. Using experiments on character recognition and text classification problems, we show that the classification performance that the ClassRBM can obtain is competitive with respect to other "black box" classifiers such as standard neural networks and Support Vector Machines (SVM). We compare different training strategies for the ClassRBM, which rely on discriminative and/or generative learning objectives. As we will see, the best approach tends to be an appropriately tuned combination of both learning objectives. Moreover, since the generative learning objective doesn't allow for the exact computation of the gradient with respect to the ClassRBM's parameters, we compare the use of different approximations of that gradient. We also address the issue of generative learning on very high dimensional inputs and propose an approach to reduce the computational cost per example of training. Finally, we describe how the ClassRBM can be used to tackle semi-supervised and multitask learning problems.

2. Classification Restricted Boltzmann Machines

The Classification Restricted Boltzmann Machine (ClassRBM) (Hinton et al., 2006) models the joint distribution of an input $\mathbf{x} = (x_1, \ldots, x_D)$ and target class $y \in \{1, \ldots, C\}$ using a hidden layer of binary stochastic units $\mathbf{h} = (h_1, \ldots, h_H)$. This is done by first defining an energy function

$$E(y, \mathbf{x}, \mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{x} - \mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{d}^T\mathbf{e}_y - \mathbf{h}^T\mathbf{U}\mathbf{e}_y$$

with parameters $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U})$ and where $\mathbf{e}_y = (1_{i=y})_{i=1}^{C}$ is the "one out of C" representation of y. From the energy function, we assign probabilities to values of y, $\mathbf{x}$ and $\mathbf{h}$ as follows:

$$p(y, \mathbf{x}, \mathbf{h}) = \frac{\exp(-E(y, \mathbf{x}, \mathbf{h}))}{Z} \tag{1}$$

where Z is a normalization constant (also called partition function) which ensures that Equation 1 is a valid probability distribution. We will assume that the elements of $\mathbf{x}$ are binary, but extensions to real-valued units on bounded or unbounded intervals are straightforward (Welling et al., 2005). An illustration of the ClassRBM is given in Figure 1.

Unfortunately, computing $p(y, \mathbf{x}, \mathbf{h})$ or $p(y, \mathbf{x})$ is typically intractable. However, it is possible to sample from the ClassRBM, using Gibbs sampling, that is, alternating between sampling a value for the hidden layer given the current value of the visible layer (made of variables $\mathbf{x}$ and the $\mathbf{e}_y$

representation of y), and vice versa. All the required conditional distributions are very simple. When conditioning on the visible layer, we have

$$p(\mathbf{h}|y,\mathbf{x}) = \prod_j p(h_j|y,\mathbf{x}), \text{ with } p(h_j=1|y,\mathbf{x}) = \mathrm{sigm}(c_j + U_{jy} + \sum_i W_{ji}x_i)$$

where $\mathrm{sigm}(a) = 1/(1+\exp(-a))$ is the logistic sigmoid function. When conditioning on the hidden layer, we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{h}) &= \prod_i p(x_i|\mathbf{h}), \text{ with } p(x_i=1|\mathbf{h}) = \mathrm{sigm}(b_i + \sum_j W_{ji}h_j) \ , \\ p(y|\mathbf{h}) &= \frac{\exp(d_y + \sum_j U_{jy}h_j)}{\sum_{y^*} \exp(d_{y^*} + \sum_j U_{jy^*}h_j)} \ . \end{aligned}$$

It is also possible to compute $p(y|\mathbf{x})$ exactly and hence perform classification. Indeed, noticing that

$$\begin{aligned} &\sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp(\mathbf{h}^T\mathbf{W}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c}^T\mathbf{h} + \mathbf{d}^T\mathbf{e}_y + \mathbf{h}^T\mathbf{U}\mathbf{e}_y) \\ &= \exp(d_y) \sum_{h_1\in\{0,1\}} \exp(h_1(c_1 + U_{1y} + \sum_i W_{1i}x_i)) \ \cdots \sum_{h_H\in\{0,1\}} \exp(h_H(c_H + U_{Hy} + \sum_i W_{Hi}x_i)) \\ &= \exp(d_y) \left(1 + \exp(c_1 + U_{1y} + \sum_i W_{1i}x_i)\right) \ldots \left(1 + \exp(c_n + U_{Hy} + \sum_i W_{ni}x_i)\right) \\ &= \exp(d_y + \sum_j \log(1 + \exp(c_j + U_{jy} + \sum_i W_{ji}x_i))) \\ &= \exp(d_y + \sum_j \mathrm{softplus}(c_j + U_{jy} + \sum_i W_{ji}x_i)) \end{aligned}$$

where $\mathrm{softplus}(a) = \log(1+\exp(a))$, then we can write

$$\begin{aligned} p(y|\mathbf{x}) &= \frac{\sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp(\mathbf{h}^T\mathbf{W}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c}^T\mathbf{h} + \mathbf{d}^T\mathbf{e}_y + \mathbf{h}^T\mathbf{U}\mathbf{e}_y)}{\sum_{y^*\in\{1,\ldots,C\}} \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp(\mathbf{h}^T\mathbf{W}\mathbf{x} + \mathbf{b}^T\mathbf{x} + \mathbf{c}^T\mathbf{h} + \mathbf{d}^T\mathbf{e}_{y^*} + \mathbf{h}^T\mathbf{U}\mathbf{e}_{y^*})} \\ &= \frac{\exp(d_y + \sum_j \mathrm{softplus}(c_j + U_{jy} + \sum_i W_{ji}x_i))}{\sum_{y^*\in\{1,\ldots,C\}} \exp(d_y^* + \sum_j \mathrm{softplus}(c_j + U_{jy^*} + \sum_i W_{ji}x_i))} \qquad (2) \\ &= \frac{\exp(-F(y,\mathbf{x}))}{\sum_{y^*\in\{1,\ldots,C\}} \exp(-F(y,\mathbf{x}))} \ . \end{aligned}$$

where $F(y,\mathbf{x})$ is referred to as the free energy. Precomputing the terms $c_j + \sum_i W_{ji}x_i$ and reusing them when computing $\mathrm{softplus}(c_j + U_{jy^*} + \sum_i W_{ji}x_i)$ for all classes y^* yields a procedure for computing this conditional distribution in time $O(HD + HC)$.

One way of interpreting Equation 2 is that, when assigning probabilities to a particular class y for some input $\mathbf{x}$, the ClassRBM looks at how well the input $\mathbf{x}$ fits or aligns with the different filters associated with the rows $\mathbf{W}_{j\cdot}$ of $\mathbf{W}$. These filters are shared across the different classes, but different classes will make comparisons with different filters by controlling the class-dependent biases U_{jy} in the $\mathrm{softplus}(c_j + U_{jy} + \sum_i W_{ji}x_i)$ terms. Notice also that two similar classes could share some of the filters in $\mathbf{W}$, that is, both could simultaneously have large positive values of U_{jy} for some of the rows $\mathbf{W}_{j\cdot}$.

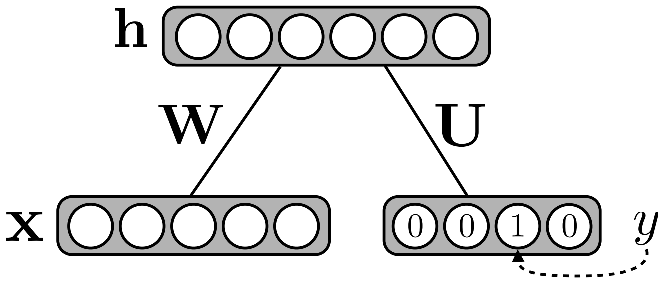


Figure 1: Illustration of a Classification Restricted Boltzmann Machine

3. Training Objectives

In order to train a ClassRBM to solve a particular classification problem, we can simply define an objective to minimize for all examples in the training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_t, y_t)\}$. The next sections describe the different training objectives which will be considered here, starting with the one most commonly used, that is, the generative training objective.

3.1 Generative Training Objective

Given that we have a model which defines a value for the joint probability $p(y, \mathbf{x})$, a natural choice for a training objective is the generative training objective

$$\mathcal{L}_{\text{gen}}(\mathcal{D}_{\text{train}}) = -\sum_{t=1}^{|\mathcal{D}_{\text{train}}|} \log p(y_t, \mathbf{x}_t) \ . \tag{3}$$

This is the most popular training objective for RBMs, for which a lot of effort has been put to obtain better estimates for its gradient (Hinton, 2002; Tieleman, 2008; Tieleman and Hinton, 2009). Indeed, as mentioned previously, computing $p(y_t, \mathbf{x}_t)$ for some example $(\mathbf{x}_t, y_t)$ is generally intractable, as is computing $\log p(y_t, \mathbf{x}_t)$ and its gradient with respect to any ClassRBM parameter θ

$$\frac{\partial \log p(y_t, \mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\left[\frac{\partial}{\partial \theta} E(y_t, \mathbf{x}_t, \mathbf{h})\right] + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\left[\frac{\partial}{\partial \theta} E(y, \mathbf{x}, \mathbf{h})\right].$$

Specifically, though the first expectation is tractable, the second is not. Different approaches have been proposed to estimate this second expectation. One which is known to work well in practice is the contrastive divergence estimator (Hinton, 2002). This approximation replaces the expectation by a point estimate at a sample generated after a limited number of Gibbs sampling steps, with the sampler's initial state for the visible variables initialized at the training example $(\mathbf{x}_t, y_t)$. A single Gibbs sampling iteration is often used and is usually found to be sufficient to learn a meaningful representation of the data. Then, this gradient estimate can be used in a stochastic gradient descent procedure for training. A pseudocode of the procedure is given by Algorithm 1, where the learning rate is controlled by λ. We will consider this procedure for now and postpone the consideration of other estimates of the generative gradient to Section 7.3.

Algorithm 1 Generative training update of the ClassRBM using Contrastive Divergence

```
**Input:** training pair $(y_t, \mathbf{x}_t)$ and learning rate $\lambda$
# Notation: $a \leftarrow b$ means $a$ is set to value $b$
#             $a \sim p$ means $a$ is sampled from $p$

# Positive phase
$y^0 \leftarrow y_t$, $\mathbf{x}^0 \leftarrow \mathbf{x}_t$, $\widehat{\mathbf{h}}^0 \leftarrow \mathrm{sigm}(\mathbf{c} + W\mathbf{x}^0 + \mathbf{U}\mathbf{e}_{y^0})$

# Negative phase
$\mathbf{h}^0 \sim p(\mathbf{h}|y^0, \mathbf{x}^0)$, $y^1 \sim p(y|\mathbf{h}^0)$, $\mathbf{x}^1 \sim p(\mathbf{x}|\mathbf{h}^0)$
$\widehat{\mathbf{h}}^1 \leftarrow \mathrm{sigm}(\mathbf{c} + W\mathbf{x}^1 + \mathbf{U}\mathbf{e}_{y^1})$

# Update
**for** $\theta \in \Theta$ **do**
    $\theta \leftarrow \theta - \lambda \left( \frac{\partial}{\partial\theta} E(y^0, \mathbf{x}^0, \widehat{\mathbf{h}}^0) - \frac{\partial}{\partial\theta} E(y^1, \mathbf{x}^1, \widehat{\mathbf{h}}^1) \right)$
**end for**
```

4. Discriminative Training Objective

The generative training objective can be decomposed as follows:

$$\mathcal{L}_{\mathrm{gen}}(\mathcal{D}_{\mathrm{train}}) = -\sum_{t=1}^{|\mathcal{D}_{\mathrm{train}}|} \left(\log p(y_t|\mathbf{x}_t) + \log p(\mathbf{x}_t)\right) = -\sum_{t=1}^{|\mathcal{D}_{\mathrm{train}}|} \log p(y_t|\mathbf{x}_t) - \sum_{t=1}^{|\mathcal{D}_{\mathrm{train}}|} \log p(\mathbf{x}_t) \qquad (4)$$

hinting that the ClassRBM will dedicate some of its capacity at modelling the marginal distribution of the input only. Since we are in a supervised learning setting and are only interested in obtaining a good prediction of the target given the input, it might be more appropriate to ignore this unsupervised part of the generative objective and focus on the supervised part.

Doing so is referred to as discriminative training, where the following training objective is used:

$$\mathcal{L}_{\mathrm{disc}}(\mathcal{D}_{\mathrm{train}}) = -\sum_{t=1}^{|\mathcal{D}_{\mathrm{train}}|} \log p(y_t|\mathbf{x}_t) \ . \qquad (5)$$

This objective is also similar to the one used by feedforward neural networks whose outputs can be interpreted as an estimate of $p(y|\mathbf{x})$. Moreover, just like neural networks, ClassRBMs trained this way are universal approximators for distributions $p(y|\mathbf{x})$ with binary inputs, since RBMs are universal approximators of distributions over binary inputs (Le Roux and Bengio, 2010).

An important advantage of the discriminative training objective is that it is possible to compute its gradient with respect to the ClassRBM's parameters exactly. The general form of the gradient for a single example $(\mathbf{x}_t, y_t)$ is

$$\frac{\partial \log p(y_t|\mathbf{x}_t)}{\partial\theta} = -\mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\left[\frac{\partial}{\partial\theta} E(y_t, \mathbf{x}_t, \mathbf{h})\right] + \mathbb{E}_{y,\mathbf{h}|\mathbf{x}}\left[\frac{\partial}{\partial\theta} E(y, \mathbf{x}, \mathbf{h})\right]$$

and, more specifically, we obtain

$$\frac{\partial \log p(y_t|\mathbf{x}_t)}{\partial d_y} = 1_{y=y_t} - p(y|\mathbf{x}_t), \quad \forall y \in \{1, \ldots, C\}$$

for the target biases **d** and

$$\frac{\partial \log p(y_t|\mathbf{x}_t)}{\partial \theta} = \sum_j \mathrm{sigm}(o_{y_t j}(\mathbf{x}_t)) \frac{\partial o_{y_t j}(\mathbf{x}_t)}{\partial \theta} - \sum_{j, y^*} \mathrm{sigm}(o_{y^* j}(\mathbf{x}_t)) p(y^*|\mathbf{x}_t) \frac{\partial o_{y^* j}(\mathbf{x}_t)}{\partial \theta}$$

for the other parameters $\theta \in \{\mathbf{c}, \mathbf{U}, \mathbf{W}\}$, where $o_{yj}(\mathbf{x}) = c_j + \sum_k W_{jk} x_k + U_{jy}$. Notice that the gradient with respect to **b** is 0, since the input biases are not involved in the computation of $p(y|\mathbf{x})$. This discriminative approach has been used previously for fine-tuning the top RBM of a Deep Belief Network (Hinton, 2007).

5. Hybrid Training Objective

In order to get an idea of when and why generative training can be better than discriminative training or vice versa, we can look at some of the known theoretical properties of both approaches.

In Ng and Jordan (2002), an analysis of naive Bayes and logistic regression classifiers (which can be seen as the same parametrization but trained according to a generative or discriminative objective respectively) indicates that the generative training objective yields models that can reach more rapidly (with training set size) their best (asymptotic) generalization performance, than models trained discriminatively. However, when the model is misspecified, the discriminative training objective allows the model to reach a better performance for sufficiently large training sets (i.e., has better asymptotic performance). In Liang and Jordan (2008), it is also shown that for models from the general exponential family, parameter estimates based on the generative training objective have smaller variance than discriminative estimates. However, if the model is misspecified, generative estimates will asymptotically yield models with worse performances. These results suggest interpreting the model trained with the generative training objective as being more regularized than the model trained with the discriminative objective.

When the ultimate task is classification, adding the generative training objective to the discriminative training objective can be seen as a way to regularize the discriminative training objective. It was already found that unsupervised pre-training of RBMs can be seen as a form of regularization (Erhan et al., 2010). Since we might want to adapt the amount of regularization to the problem at hand, we could consider interpolating between the generative and discriminative objectives as in Bouchard and Triggs (2004) or, similarly, use the following hybrid objective:

$$\mathcal{L}_{\text{hybrid}}(\mathcal{D}_{\text{train}}) = \mathcal{L}_{\text{disc}}(\mathcal{D}_{\text{train}}) + \alpha \mathcal{L}_{\text{gen}}(\mathcal{D}_{\text{train}}) \tag{6}$$

where the weight α of the generative criterion can be adjusted based on the performance of the model on a validation set. As in Equation 4, we can separate the $\log p(y_t, \mathbf{x}_t)$ terms in two and rewrite Equation 6 as

$$\mathcal{L}_{\text{hybrid}}(\mathcal{D}_{\text{train}}) = -(1+\alpha) \sum_{t=1}^{|\mathcal{D}_{\text{train}}|} \log p(y_t|\mathbf{x}_t) - \alpha \sum_{t=1}^{|\mathcal{D}_{\text{train}}|} \log p(\mathbf{x}_t) \ .$$

This different expression for $\mathcal{L}_{\text{hybrid}}(\mathcal{D}_{\text{train}})$ highlights the nature of the regularization that is imposed: among all configurations of the parameters of the ClassRBMs that can solve the supervised problem well, we will favor those that also assign high probability to the inputs and hence have extracted some of the structure present in the input's distribution.

6. Related Work

As mentioned previously, RBMs—sometimes also referred to as harmoniums (Welling et al., 2005), usually when the hidden layer's units are not binary—have been popular feature extractors in classification applications. Most of the time however, the features are learned while ignoring the target label information (Gehler et al., 2006; Xing et al., 2005). This implies that some label-related information may be thrown away and McCallum et al. (2006) have shown that incorporating labels in a feature learning procedure can be beneficial, in their work on Multi-Conditional Learning (MCL).[1] However, this latter work still required that the relationship between the hidden features and the target be learned a posteriori, by a separate classifier. One of the main points we wish to make in this paper is that RBMs provide a self-contained framework for classification, which does not need to rely on the availability of a separate classifier. This approach has two advantages: model selection is facilitated since no additional hyper-parameters from the separate classifier must be tuned, and no additional classifier training phase is required, making it possible to employ the ClassRBM in an online learning setting or to track the discriminative performance of the latent representation on a validation set. Another frequent use of RBMs is as an initializing or pretraining algorithm for deep neural networks (Hinton, 2007), but this approach shares the same disadvantages of having two training phases.

Schmah et al. (2009) proposed a different approach to discriminative training of RBMs, where each class is associated with its own individual RBM, as in a Bayes classifier. However, this approach does not rely on a global hidden representation (with shared parameters) for all classes and hence cannot model directly the latent similarity between classes, which should be advantageous for classification problems with large number of classes. From this perspective, the ClassRBM can be seen as a form of *multi-task* training, since the input to hidden weights are shared across all classes.

Yang et al. (2007) developed variants of harmoniums for video classification that can model several modalities and class information jointly. One variant uses a separate harmonium for each class as in Schmah et al. (2009), while a second is based on a shared hidden representation across classes like in the ClassRBM. However, they proposed training these models generatively, which is often not the optimal training strategy, as discussed in Section 7.

There are also several similarities between classification RBMs and ordinary multi-layer neural networks. In particular, the computation of $p(y|\mathbf{x})$ could be implemented by a single layer neural network with softplus and softmax activation functions in its hidden and output layers respectively, as well as with a special structure in the output and hidden weights where the value of the output weights is fixed and many of the hidden layer weights are shared. Glorot et al. (2011) highlight that softplus hidden activation functions tend to yield better performances than logistic-shaped functions (including the hyperbolic tangent). This might be one explanation behind the slightly superior performance of discriminatively trained ClassRBMs, compared to neural networks with hyperbolic tangent hidden units (see Sections 7.1 and 7.2). However, the main advantage of working in the framework of RBMs is that it provides a natural way to introduce generative learning, which can provide data set-dependent regularization and, as we will see, can be used to extend learning in the semi-supervised setting. As mentioned earlier, a form of generative learning can be introduced in standard neural networks with unsupervised pre-training, simply by using RBMs to initialize the hidden layer weights. However, the extent to which the final solution for the parameters of

[1] We experimented with a version of MCL for the ClassRBM, however the results did not improve on those of hybrid training.

the neural network is influenced by generative learning is not well controlled, while the hybrid objective provides an explicit handle on the role played by generative learning. One could argue that the advantage of unsupervised pre-training is that it allows to build deeper models. However, consider that it is always possible to use the ClassRBM as the top layer of a stack of RBMs, in the spirit already suggested in Hinton et al. (2006).

7. Evaluation of the Training Objectives

To evaluate the performance of the different training objectives described thus far, we present experiments on two classification problems: character recognition and text classification. Such problems are particularly interesting as they are known to benefit from the extraction of non-linear features and hence are well suited for the ClassRBM. In all experiments, for the ClassRBM variants and for all baselines, we performed model selection based on the validation set performance. For the different RBM models, model selection was done with a grid-like search over the learning rate λ (between 0.0005 and 0.1, on a log scale), H (50 to 6000), the generative learning weight α for hybrid training (0 to 0.5, on a log scale) and the weight β for semi-supervised learning $(0, 0.01$ or $0.1)$. In general, bigger values of H were found to be more appropriate with more generative learning. If no local minima was apparent, the grid was extended. The biases **b**, **c** and **d** were initialized to 0 and the initial values for the elements of the weight matrices **U** and **W** were each taken from uniform samples in $\left[-m^{-0.5}, m^{-0.5}\right]$, where m is the maximum between the number of rows and columns of the matrix. The number of iterations over the training set was determined using early stopping according to the validation set classification error, with a look ahead of 15 iterations.

7.1 Character Recognition

We first evaluate the different training objectives for the ClassRBM on the problem of classifying images of digits. The images were taken from the MNIST data set, where we separated the original training set into training and validation sets of 50000 and 10000 examples and used the standard test set of 10000 examples. The results are given in Table 1. The RBM+NNet approach is simply an unsupervised RBM used to initialize a one-hidden layer supervised neural net (as in Bengio et al. 2007). We give as a comparison the results of a Gaussian kernel SVM, a random forest classifier[2] and of a regular neural network (with random initialization, one hidden layer and hyperbolic tangent hidden activation functions).

First, we observe that discriminative training of the ClassRBM outperforms generative training. However, hybrid training appears able to make the best out of both worlds and outperforms the other approaches.

We also experimented with a sparse version of the hybrid ClassRBM, since sparsity is known to be a good characteristic for features of images. Sparse RBMs were developed by Lee et al. (2008) in the context of deep neural networks. They suggest to introduce sparsity in the hidden layer of an RBM by setting the biases **c** in the hidden layer to a value that maintains the average of the conditional expected value of these neurons to an arbitrarily small value, and so after each iteration through the whole training set. This procedure tends to make the biases negative and large. We followed a different approach by simply subtracting a small constant δ value, considered as an hyper-parameter, from the biases after each update, which is more appropriate in an online setting

[2] We used the implementation provided by the TreeLearn library: `https://github.com/capitalk/treelearn`.

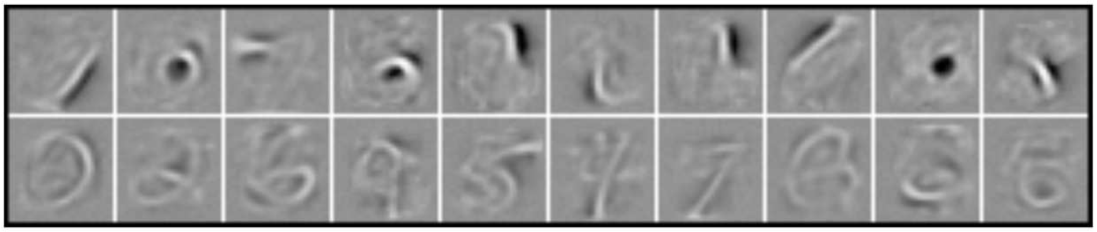


Figure 2: Filters learned by the ClassRBM on the MNIST data set. The top row shows filters that act as spatially localized stroke detectors, and the bottom shows filters more specific to a particular shape of digit.

or for large data sets. To chose δ, given the selected values for λ and α for the "non sparse" hybrid ClassRBM, we performed a second grid-search over δ (between 10^{-5} and 0.1, on a log scale) and the hidden layer size, testing bigger hidden layer sizes than previously selected.

This sparse version of the hybrid ClassRBM outperforms all the other RBM approaches, and yields significantly lower classification error than the SVM, the random forest and the standard neural network classifiers. The performance achieved by the sparse ClassRBM is particularly impressive when compared to reported performances for Deep Belief Networks (1.25% in Hinton et al. 2006) or of a deep neural network initialized using RBMs (around 1.2% in Bengio et al. 2007 and Hinton 2007) for the MNIST data set with 50000 training examples.

The discriminative power of the hybrid ClassRBM can be better understood by looking a the rows of the weight matrix $\mathbf{W}$, which act as filter features. Figure 2 displays some of these learned filters. Some of them are spatially localized stroke detectors which can possibly be active for a wide variety of digit images, and others are much more specific to a particular shape of digit.

In practice, we find that the most influential hyper-parameters are the learning rate and the generative learning weight. Conveniently, we also find that that the best learning rate value is the same for each values of the generative learning weight we tested. In other words, finding a good learning rate does not require having found the best value for the generative learning weight. Once these two hyper-parameters are set to good values, we also find that a wide range of hidden layer sizes (between 750 to 3000) yield a competitive performance, that is, under 1.4% classification error.

7.2 Document Classification

We also evaluated the RBM models on the problem of classifying documents into their corresponding newsgroup topic. We used a version of the 20 Newsgroups data set[3] for which the training and test sets contain documents collected at different times, a setting that is more reflective of a practical application. The original training set was divided into a smaller training set and a validation set, with 9578 and 1691 examples respectively. The test set contains 7505 examples. We used the 5000 most frequent words for the binary input features. The results are given in Table 2. Again, we

[3] This data set is available in Matlab format here:
`http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate-matlab.tgz`.

Model	Objective	Error
ClassRBM	Generative ($\lambda = 0.005, H = 6000$)	3.39%
	Discriminative ($\lambda = 0.05, H = 500$)	1.81%
	Hybrid ($\alpha = 0.01, \lambda = 0.05, H = 1500$)	1.28%
	Sparse Hybrid (idem + $H = 3000, \delta = 10^{-4}$)	**1.16%**
SVM	-	1.40%
Random Forest		2.94%
NNet		1.93%
RBM+NNet		1.41%

Table 1: Comparison of the classification performances on the MNIST data set. SVM results for MNIST were taken from `http://yann.lecun.com/exdb/mnist/`. On this data set, differences of 0.2% in classification error are statistically significant.

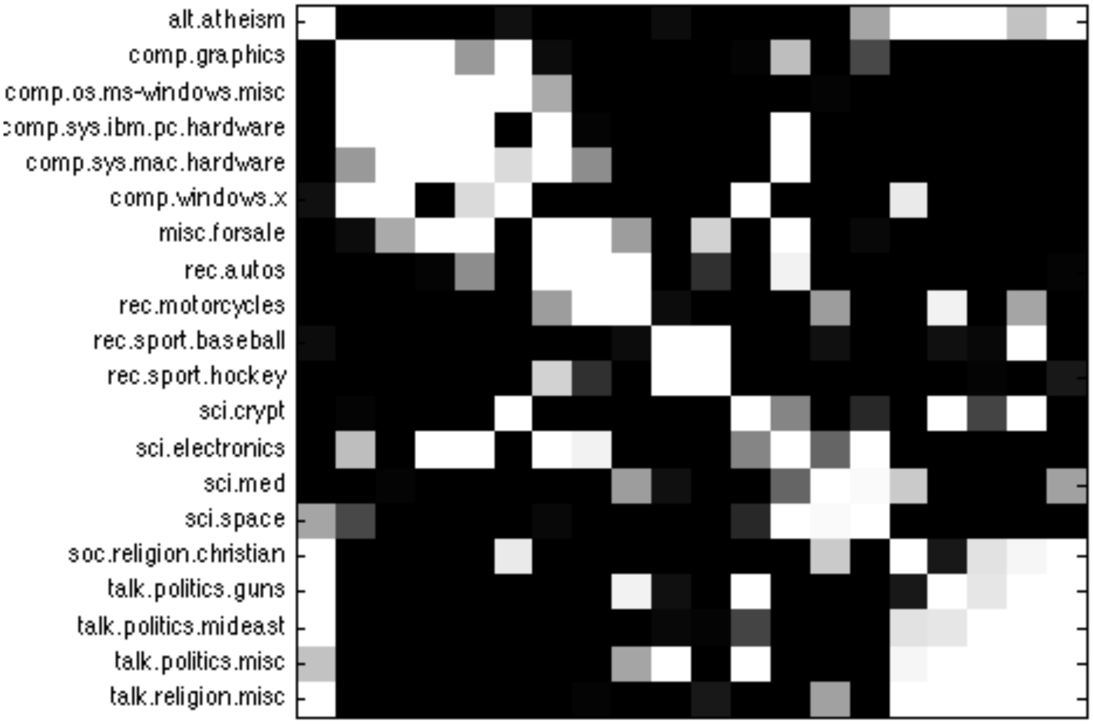


Figure 3: Similarity matrix of the newsgroup weights vectors $U_{\cdot y}$.

also provide the results of a Gaussian kernel SVM,[4] a random forest classifier and a regular neural network for comparison.

Once again, hybrid training of the ClassRBM outperforms the other approaches, including the SVM and neural network classifiers. Notice that here generative training performs better than discriminative training.

[4] We used `libSVM` v2.85 to train the SVM model.

Model	Objective	Error
ClassRBM	Generative ($\lambda = 0.0005, H = 1000$)	24.9%
	Discriminative ($\lambda = 0.0005, H = 50$)	27.6%
	Hybrid ($\alpha = 0.005, \lambda = 0.1, H = 1000$)	**23.8%**
SVM	-	32.8%
Random Forest		29.0%
NNet		28.2%
RBM+NNet		26.8%

Table 2: Classification performances on 20 Newsgroups data set for the different models. The error differences between the hybrid ClassRBM and other approaches is statistically significant.

Much like for the character recognition experiment of the previous section, we find that the learning rate and generative learning weight are the most crucial hyper-parameters to tune, and that the performance is quite stable across hidden layer size as long as it is large enough (500 or greater for this problem).

In order to get a better understanding of how the hybrid ClassRBM solves this classification problem, we looked at the weights connecting each of the classes to the hidden neurons. This corresponds to the columns $\mathbf{U}_{\cdot y}$ of the weight matrix $\mathbf{U}$. Figure 3 shows a similarity matrix $\mathbf{M}(\mathbf{U})$ for the weights of the different newsgroups, where $\mathbf{M}(\mathbf{U})_{y_1 y_2} = \mathrm{sigm}(\mathbf{U}_{\cdot y_1}^T \mathbf{U}_{\cdot y_2})$. We see that the ClassRBM does not use strictly non-overlapping sets of neurons for different newsgroups, but shares some of those neurons for newsgroups that are semantically related. We see that the ClassRBM tends to share neurons for topics such as computer (`comp.*`), science (`sci.*`) and politics (`talk.politics.*`), or secondary topics such as sports (`rec.sports.*`) and other recreational activities (`rec.autos` and `rec.motorcycles`).

Table 3 also gives the set of words used by the ClassRBM to recognize some of the newsgroups. To obtain this table we proceeded as follows: for each newsgroup y, we looked at the 20 neurons with the largest weight among $\mathbf{U}_{\cdot y}$, aggregated (by summing) the associated input-to-hidden weight vectors, sorted the words in decreasing order of their associated aggregated weights and picked the first few words according to that order. This procedure attempts to approximate the positive contribution of the words to the conditional probability of each newsgroup.

7.3 Variations on the Generative Learning Gradient Estimator

In the previous sections, we considered contrastive divergence for estimating the generative learning gradient. However, other alternatives have also been proposed. An interesting question is how much impact does the choice between these different gradient estimators has on the classification performance of the ClassRBM?

A first alternative is based on the concept of pseudolikelihood (PL) (Besag, 1975), which aims at replacing the regular likelihood objective with one more tractable, that is, for which gradients can be computed exactly. The negative log-likelihood objective on a $(\mathbf{x}, y)$ pair is then replaced by

$$-\log p(y|\mathbf{x}) - \sum_{k=1}^{D} \log p(x_k|\mathbf{x}_{\backslash k}, y) \tag{7}$$

Class	Words
alt.atheism	bible, atheists, benedikt, atheism, religion, scholars, biblical
comp.graphics	tiff, ftp, window, gif, images, pixel, rgb, viewer, image, color
comp.os.ms-windows.misc	windows, cica, bmp, window, win, installed, toronto, dos, nt
comp.sys.ibm.pc.hardware	dos, ide, adaptec, pc, config, irq, vlb, bios, scsi, esdi, dma
comp.sys.mac.hardware	apple, mac, quadra, powerbook, lc, pds, centris, fpu, power, lciii
comp.windows.x	xlib, man, motif, widget, openwindows, xterm, colormap, xdm
misc.forsale	sell, condition, floppy, week, am, obo, shipping, company, wpi
rec.autos	cars, ford, autos, sho, toyota, roads, vw, callison, sc, drive
rec.motorcycles	bikes, motorcycle, ride, bike, dod, rider, bmw, honda
rec.sport.baseball	pitching, braves, hitter, ryan, pitchers, so, rbi, yankees, teams
rec.sport.hockey	playoffs, penguins, didn, playoff, game, out, play, cup, stanley
sci.crypt	sternlight, bontchev, nsa, escrow, hamburg, encryption, rm
sci.electronics	amp, cco, together, voltage, circuits, detector, connectors
sci.med	drug, syndrome, dyer, diet, foods, physician, medicine, disease
sci.space	orbit, spacecraft, speed, safety, known, lunar, then, rockets
soc.religion.christian	rutgers, athos, jesus, christ, geneva, clh, christians, sin, paul
talk.politics.guns	firearms, handgun, firearm, gun, rkba, concealed, second, nra
talk.politics.mideast	armenia, serdar, turkish, turks, cs, argic, stated, armenians, uci
talk.politics.misc	having, laws, clinton, time, koresh, president, federal, choose
talk.religion.misc	christians, christian, bible, weiss, religion, she, latter, dwyer

Table 3: Most influential words in the hybrid ClassRBM for predicting some of the document classes

where $\mathbf{x}_{\backslash k}$ is a vector made of all elements of $\mathbf{x}$ except x_k. Hence, the model is trained to maximize the likelihood of each observed variable *given* all other observed variables. Notice that the first term corresponds to discriminative training. Hence, to obtain hybrid training we can simply weight the second summation term by α.

Equation 7 as well as its gradient with respect to the ClassRBM's parameters can be computed exactly by backpropagation. In the ClassRBM, with a development similar to the one for $p(y|\mathbf{x})$, we can show that:

$$\begin{aligned} p(x_k|\mathbf{x}_{\backslash k},y) &= \frac{p(\mathbf{x}|y)}{p(\mathbf{x}|y)+p(\bar{\mathbf{x}}_k|y)} \\ &= \frac{\exp(x_k b_k + \sum_j \mathrm{softplus}(c_j + U_{jy} + W_{jk}x_k + \sum_{i\neq k} W_{ji}x_i))}{\sum_{x'_k\in\{0,1\}} \exp(x'_k b_k + \sum_j \mathrm{softplus}(c_j + U_{jy^*} + W_{jk}x'_k + \sum_{i\neq k} W_{ji}x_i))} \end{aligned}$$

where $\bar{\mathbf{x}}_k$ corresponds to $\mathbf{x}$ but where the input's k^{th} bit has been flipped. So the terms $\log p(x_k|\mathbf{x}_{\backslash k},y)$ can be computed in $O(HD)$. A naive computation of Equation 7, which would compute the H terms $\log p(x_k|\mathbf{x}_{\backslash k},y)$ separately, would then scale in $O(HD^2+HC)$. However, by computing $\sum_i W_{ji}x_i$ for all j only once and reusing those terms to obtain the terms $\sum_{i\neq k} W_{ji}x_i$ for any k, we can obtain a procedure that is still linear in D, as is the CD gradient estimator. In practice, pseudolikelihood training still has some computational overhead compared to CD. Indeed, pseudolikelihood training requires $O(HD)$ computations of the exponential function, whereas CD only requires $O(H+D)$ such computations.

Generative gradient estimator	Error	
	MNIST	20News
Contrastive Divergence - 1 Gibbs sampling step	1.16%	23.8%
Contrastive Divergence - 10 Gibbs sampling step	1.15%	24.8%
Persistent Contrastive Divergence	1.41%	24.9%
Pseudolikelihood	1.21%	24.7%

Table 4: Comparison of the classification performances using different generative gradient estimators.

To perform stochastic gradient descent, a gradient step update is made according to the objective of Equation 7 every time a training example is visited. Because of the higher computational cost of PL, we use a sampling trick to estimate the gradient on the second summation term of Equation 7. Indeed, before every update, we randomly select a subset of the input variables x_k and sum only over those in the second term. This trick was necessary to scale down training to a reasonable time. We used a subset of size 100 and 500 for the MNIST and 20 Newsgroups data sets respectively.

Another generative gradient estimator for RBMs that has been recently proposed is the Persistent CD (PCD) estimator (Tieleman, 2008). PCD improves on CD by running a set of Gibbs sampling chains which persist through training, instead of always being reinitialized at each training example. Tieleman (2008) has shown that this new estimator can sometimes improve the rate of training as well as the quality of the solution that is found. As proposed by Tieleman (2008), we used 100 parallel chains for Gibbs sampling. Since we use stochastic gradient descent (instead of mini-batch gradient descent), only one chain was updated per update. The chains were updated sequentially by cycling through the set of chains.

Finally, an even simpler way of improving the gradient estimate that CD computes is to increase the number of Gibbs sampling steps that is used in the negative phase. In their experiments, Tieleman (2008) have found that CD with 10 Gibbs sampling steps often compares quite well to PCD.

Is the choice of the generative gradient estimator in the hybrid objective crucial for obtaining good classification performances? To answer this question, we have trained ClassRBMs using the hybrid objective on the MNIST and 20 Newsgroup data sets, with the different generative gradient estimators. Hyper-parameters were tuned separately for each variant, as in the previous sections. For the MNIST data set, we used the sparse training variant. The results of this experiment are given in Table 4. We see that in general, none of the alternative estimators provided significant performance improvements. On 20 Newsgroups, the performance even worsened. We notice that the performance obtained with PCD tends to be the particularly bad. This is explained by the fact that PCD requires smaller learning rates to work well, so that the model doesn't change faster than the rate in which the parallel Gibbs chains mix. However, using a small learning rate does not correspond to the regime at which the ClassRBM performs best in terms of classification error for these problems. This is particularly true for MNIST where the optimal learning rate is between 0.05 and 0.1.

7.4 Scaling Up to Large Input Spaces

Computing the generative gradient is typically much more computationally expensive than the discriminative gradient. This is particularly true on problems were the input data is very high-dimensional and sparse, such as the text classification problem. For instance, though we have restricted its dimensionality to 5000, the 20 Newsgroup data set could be made much more high dimensional by including more words as input features. While the computations for estimating the discriminative gradient can take advantage of the sparsity of the input (mainly when multiplying the input with the filters), estimating the generative gradient for all of the estimators in Section 7.3 requires an explicit loop over all inputs.

It would hence be beneficial to derive a more general generative gradient estimator that would allow us to control more directly its computational cost, and perhaps let us trade a little bit of accuracy for more computational efficiency. This would particularly be useful in an online learning setting, where a stream of training examples is available, with examples being presented at some given rate. In such a setting, we might want to reduce the computational time required by the generative learning objective so that updating the parameters of the ClassRBM for a training example can be done before the next sample is given.

As mentioned, the computational expense of training is closely related to the number of variables who's distribution is being modelled. At one extreme, discriminative learning is very efficient since we are only modelling the (conditional) distribution of the target variable while, at the other extreme, generative learning is much more expensive because the distribution of the target and all input variables is being modelled. Hence, a good handle over the computational complexity of an estimator would be the total number of variables involved in the conditional distribution on which the training objective is based.

Following this idea, let $I = \{1, \ldots, D\}$ be the set of input variable indices and let $\mathcal{P}_{=L}(I)$ be all the subsets of I of cardinality L, we could define the following as our new computation-aware generative training objective

$$-\sum_{t=1}^{|\mathcal{D}_{\text{train}}|} \mathbb{E}_{\mathcal{S}\in\mathcal{P}_{=L}(I)} \left[\log p(y_t, \mathbf{x}_{\mathcal{S}} | \mathbf{x}_{\backslash\mathcal{S}})\right] = -\sum_{t=1}^{|\mathcal{D}_{\text{train}}|} \sum_{\mathcal{S}\in\mathcal{P}_{=L}(I)} \frac{1}{|\mathcal{P}_{=L}(I)|} \log p(y_t, \mathbf{x}_{\mathcal{S}} | \mathbf{x}_{\backslash\mathcal{S}}) \tag{8}$$

where $\mathbf{x}_{\mathcal{S}}$ is the vector of input variables with index in $\mathcal{S}$ and $\mathbf{x}_{\backslash\mathcal{S}}$ is the vector of all other variables. Put briefly, this objective aims at maximizing the conditional likelihood of the target and all subsets of input variables of size L given the other variables, and with a uniform distribution or weight on all such possible partitions of the inputs. Since the expectation over $\mathcal{S}$ is intractable even for relatively small values of L, in practice we approximate it by sampling a single value from the associated uniform distribution over $\mathcal{S}$, and so for every parameter update.

This training objective actually corresponds to a particular type of composite likelihood estimator (Lindsay, 1988; Liang and Jordan, 2008). Here, we in addition propose to approximate the gradients of the $\log p(y_t, \mathbf{x}_{\mathcal{S}} | \mathbf{x}_{\backslash\mathcal{S}})$ terms

$$\frac{\partial \log p(y_t, \mathbf{x}_{\mathcal{S}} | \mathbf{x}_{\backslash\mathcal{S}})}{\partial\theta} = -\mathbb{E}_{\mathbf{h}|y_t,\mathbf{x}_t}\left[\frac{\partial}{\partial\theta}E(y_t, \mathbf{x}_t, \mathbf{h})\right] + \mathbb{E}_{y,\mathbf{x}_{\mathcal{S}},\mathbf{h}|\mathbf{x}_{\backslash\mathcal{S}}}\left[\frac{\partial}{\partial\theta}E(y, \mathbf{x}, \mathbf{h})\right]$$

by using contrastive divergence with one step of Gibbs sampling. This approximation requires that only the L variables in $\mathbf{x}_{\mathcal{S}}$ be sampled, making this procedure efficient for small L.

Model	Hybrid training (for varying L)				Discriminative
	250	500	5000	10000	training
Error	22.9%	22.2%	21.9%	21.9%	26.9%

Table 5: Evaluation of the composite likelihood variant of contrastive divergence on the 20 Newsgroups data set, with an input dimensionality of 25247.

We experimentally investigated how varying L impacts the performance of ClassRBMs trained using a hybrid objective based on the generative objective of Equation 8. We took the 20 Newsgroups data set and, instead of only using the 5000 most frequent words as features, we considered all words appearing at least 5 times, adding up to 25247 words. The results are given in Table 5. We observe a big improvement on the classification error obtained by restricting the input to only 5000 words, as in Table 2. The performance of purely discriminative training in the large vocabulary setting, which is essentially equivalent to setting $L = 0$, is also improved on. We see that the composite likelihood variant still allows for better generalization performance to be achieved, even for relatively small values of L. Interestingly, we also observe a fairly rapid diminishing return in the improvement of generalization error as L increases.

The idea of combining composite likelihood objectives and contrastive divergence has also been combined previously by Asuncion et al. (2010), but in a different way. Asuncion et al. (2010) focused on models for which standard contrastive divergence with Gibbs sampling corresponds to sampling only a single randomly selected variable at each step. In this case, contrastive divergence with one sampling step actually corresponds to a stochastic version of pseudolikelihood (Hyvärinen, 2006). They propose instead to use block-Gibbs sampling on randomly selected blocks of variables of limited size L at each step. L must be small however since, in general, computing the associated conditionals is exponential in L. Using a single sampling step then corresponds to a stochastic version of composite likelihood. They show that increasing L and using a single Gibbs step can be more advantageous than using $L = 1$ and increasing the number of iterations. Their work can be understood as an investigation of how to improve contrastive divergence using ideas from composite likelihood objectives.

However, for RBMs, block-Gibbs sampling is actually the standard, where we first sample all hidden units and then all input variables in one iteration. Hence, the approach of Asuncion et al. (2010) is not directly applicable here. What we propose instead, is to apply contrastive divergence to a composite likelihood objective, such that we approximate the gradients on the $\log p(y_t, \mathbf{x}_{\mathcal{S}}|\mathbf{x}_{\backslash\mathcal{S}})$ terms. Crucially, this approach is linear in L, as opposed to exponential.

8. Semi-supervised Learning

In certain situations, in addition to a (possibly small) set of labeled training examples $\mathcal{D}_{\text{train}}$, even more data can be obtained in the form of an unlabeled training set $\mathcal{D}_{\text{unlab}} = \{(\mathbf{x}_t)\}$. This is particularly true for data such as images and text documents, for which the Internet is an almost infinite source. Semi-supervised learning algorithms (Chapelle et al., 2006) address this situation by leveraging the unlabeled data to bias learning towards solutions that are also "consistent" with the unlabeled data. Different algorithms can then be seen as defining different notions of consistency.

Because a ClassRBM is a proper generative model, a very natural notion of consistency in this context is that unlabeled training data have high likelihood under it. To achieve this, one can optimize the following negative log-likelihood

$$\mathcal{L}_{\text{unsup}}(\mathcal{D}_{\text{unlab}}) = -\sum_{t=1}^{|\mathcal{D}_{\text{unlab}}|} \log p(\mathbf{x}_t) \tag{9}$$

which requires computing the gradients

$$\frac{\partial \log p(\mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{y,\mathbf{h}|\mathbf{x}_t}\left[\frac{\partial}{\partial\theta}E(y_t,\mathbf{x}_t,\mathbf{h})\right] + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\left[\frac{\partial}{\partial\theta}E(y,\mathbf{x},\mathbf{h})\right] .$$

The contrastive divergence approximation proceeds slightly differently here. The first term can be computed in time $O(HD+HC)$, by noticing that

$$\mathbb{E}_{y,\mathbf{h}|\mathbf{x}_t}\left[\frac{\partial}{\partial\theta}E(y_t,\mathbf{x}_t,\mathbf{h})\right] = \mathbb{E}_{y|\mathbf{x}_t}\left[\mathbb{E}_{\mathbf{h}|y,\mathbf{x}_t}\left[\frac{\partial}{\partial\theta}E(y_t,\mathbf{x}_t,\mathbf{h})\right]\right]$$

and then either average the usual RBM gradient $\frac{\partial}{\partial\theta}E(y_t,\mathbf{x}_t,\mathbf{h})$ for each class y (weighted by $p(y|\mathbf{x}_t)$), or sample from $p(y|\mathbf{x}_t)$ and only collect the gradient for the sampled value of y. In the latter sampling version, the online training update for this objective can be described as replacing the statement $y^0 \leftarrow y_t$ with $y^0 \sim p(y|\mathbf{x}_t)$ in Algorithm 1. We used this version in our experiments.

In order to perform semi-supervised learning, we can weight and combine the objective of Equation 9 with those of Equations 3, 5 or 6 as follows:

$$\mathcal{L}_{\text{semi-sup}}(\mathcal{D}_{\text{train}},\mathcal{D}_{\text{unlab}}) = \mathcal{L}_{\text{TYPE}}(\mathcal{D}_{\text{train}}) + \beta\mathcal{L}_{\text{unsup}}(\mathcal{D}_{\text{unlab}}) \tag{10}$$

where TYPE $\in \{gen, disc, hybrid\}$. Online training by stochastic gradient descent then corresponds to applying two gradients updates: one for the objective $\mathcal{L}_{\text{TYPE}}$ and one for the unlabeled data objective $\mathcal{L}_{\text{unsup}}$.

We evaluated our semi-supervised learning algorithm for the hybrid ClassRBM on both previous digit recognition and document classification problems. We also experimented with a version (noted MNIST-BI) of the MNIST data set proposed by Larochelle et al. (2007) where background images have been added to MNIST digit images. This version corresponds to a much harder problem and it will help to illustrate the advantage brought by semi-supervised learning in ClassRBMs. The ClassRBM trained on this data used truncated exponential input units (see Bengio et al., 2007).

In this semi-supervised setting, we reduced the size of the labeled training set to 800 examples, and used some of the remaining data to form an unlabeled data set $\mathcal{D}_{\text{unlab}}$. The validation set was also reduced to 200 labeled examples. Model selection covered all the parameters of the hybrid ClassRBM as well as the unsupervised objective weight β of Equation 10, with $\beta = 0.1$ for MNIST and 20 Newsgroups, and $\beta = 0.01$ for MNIST-BI performing best. For comparison purposes, we also provide the performance of a standard non-parametric semi-supervised learning algorithm based on function induction (Bengio et al., 2006a), which is very similar to other non-parametric semi-supervised learning algorithms such as Zhu et al. (2003). We provide results for the use of a Gaussian kernel (NP-Gauss) and a data-dependent truncated Gaussian kernel (NP-Trunc-Gauss) used in Bengio et al. (2006a), which essentially outputs zero for pairs of inputs that are not near neighbors. The experiments on the MNIST and MNIST-BI (with background images) data

Model	Objective	MNIST	MNIST-BI	20News
ClassRBM	Hybrid	9.73%	42.4%	40.5%
	Semi-supervised + Hybrid	8.04%	**37.5%**	**31.8%**
NP-Gauss	-	10.60%	66.5%	85.0%
NP-Trunc-Gauss		**7.49%**	61.3%	82.6%

Table 6: Comparison of the classification errors in semi-supervised learning setting. The errors in bold are statistically significantly better.

sets used 5000 unlabeled examples and the experiment on 20 Newsgroups used 8778. The results are given in Table 6, where we observe that semi-supervised learning consistently improves the performance of the ClassRBM trained based on the hybrid objective.

The usefulness of non-parametric semi-supervised learning algorithms has been demonstrated many times in the past, but usually so on problems where the dimensionality of the inputs is low or the data lies on a much lower dimensional manifold. This is reflected in the result on MNIST for the non-parametric methods. However, for high dimensional data with many factors of variation, these methods can quickly suffer from the curse of dimensionality, as argued by Bengio et al. (2006b). This is also reflected in the results for the MNIST-BI data set which contains many factors of variation, and for the 20 Newsgroups data set where the input is very high dimensional. Finally, it is important to notice that semi-supervised learning in ClassRBMs proceeds in an online fashion and hence could scale to very large data sets, unlike most non-parametric methods.

We mention that, in the context of log-linear models, Druck et al. (2007) introduced semi-supervised learning in hybrid generative/discriminative models using a similar approach to the one presented in here. While log-linear models depend much more on the discriminative quality of the features that are fed as input, the ClassRBM can learn useful features through its hidden layer and model non-linear decision boundaries.

9. Multitask Learning

The classification problems considered so far had in common that a given input could only belong to a single class, that is, classes were mutually exclusive. For certain problems, this assumption is too restrictive and inputs can be simultaneously associated with multiple classes or labels. One example is online collections of images, documents or music augmented with social tags (see Lamere 2008 for an example), which are short descriptions applied by users to items and can be used by users to search and browse through a collection. One approach to this problem would be to train a separate classifier for each tag. However, a better approach is to perform multitask learning (Caruana, 1997), where a single model is trained to perform all tasks simultaneously. This allows for the model to leverage the similarity between certain tasks and improve generalization.

We describe here how multitask learning can also be performed within a ClassRBM. In this context, the target's representation in the energy function of the ClassRBM does not follow the "one out of C" constraint and is an unconstrained binary vector $\mathbf{y}$. The conditional distribution of $\mathbf{y}$ given

h then becomes:

$$p(\mathbf{y}|\mathbf{h}) = \prod_c p(y_c|\mathbf{h}), \text{ with } p(y_c = 1|\mathbf{h}) = \operatorname{sigm}(d_c + \sum_i U_{jc} h_j).$$

Another important implication of this change is that the predictive posterior $p(\mathbf{y}|\mathbf{x})$ is no longer tractable, since $\mathbf{y}$ now has 2^C possible values. At test time, we are particularly interested in estimating $p(y_c = 1|\mathbf{x})$ for each label, in order to make a prediction of the binary value of each individual label. Fortunately, there exist several message-passing approximate inference procedures for general graphical models that can be employed here. The two most popular are mean field and loopy belief propagation.

The mean field (MF) approach tries to approximate the joint posterior $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ by a factorial distribution $q(\mathbf{y}, \mathbf{h}) = \prod_{c=1}^{C} \mu_c^{y_c} (1-\mu_c)^{1-y_c} \prod_{j=1}^{n} \tau_j^{h_j} (1-\tau_j)^{1-h_j}$ that minimizes the Kullback-Leibler (KL) divergence with the true posterior. Running the following message passing procedure to convergence

$$\begin{aligned} \mu_c &\leftarrow \operatorname{sigm}\left(d_c + \sum_j U_{jc} \tau_j\right) \quad \forall\, c \in \{1, \ldots, C\}, \\ \tau_j &\leftarrow \operatorname{sigm}\left(c_j + \sum_c U_{jc} \mu_c + \sum_i W_{ji} x_i\right) \quad \forall\, j \in \{1, \ldots, n\} \end{aligned}$$

we can reach a saddle point of the KL divergence, at which point μ_c serves as the estimate for $p(y_c = 1|\mathbf{x})$ and τ_j can be used to estimate $p(h_j = 1|\mathbf{x})$. In our experiments, we initialized the messages to 0. Moreover, we treat the number of message passing iterations as an hyper-parameter, so as to control the computational cost of inference.

Loopy belief propagation (Pearl, 1988) (LBP) also relies on a message passing procedure between variables. LBP is more complex than MF in that the number of distinct messages to be maintained scales in $O(HC)$, that is, the number of connections between $\mathbf{y}$ and $\mathbf{h}$, instead of in $O(H+C)$ as in MF. It also provides a direct estimate of the pair-wise probabilities $p(y_c = 1, h_j = 1|\mathbf{x})$. LBP tends to give estimates of the true marginals that are more accurate than the iterative MF procedure (Weiss, 2001). While not guaranteed to converge it frequently does in practice. One method that has been shown to be useful in aiding convergence is message damped belief propagation (Pretti, 2005). In this case the normal updates computed by belief propagation are mixed with the previous updates in order to smooth them, the damping factor being a parameter of the algorithm. Algorithm 2 details the procedure.

As for learning, the discriminative gradient expression, which is now

$$\frac{\partial \log p(\mathbf{y}_t|\mathbf{x}_t)}{\partial \theta} = -\mathbb{E}_{\mathbf{h}|\mathbf{y}_t, \mathbf{x}_t}\left[\frac{\partial}{\partial \theta} E(\mathbf{y}_t, \mathbf{x}_t, \mathbf{h})\right] + \mathbb{E}_{\mathbf{y}, \mathbf{h}|\mathbf{x}}\left[\frac{\partial}{\partial \theta} E(\mathbf{y}, \mathbf{x}, \mathbf{h})\right]$$

must also be approximated, specifically the second expectation over $\mathbf{y}$ and $\mathbf{h}$. Contrastive divergence is a natural approach to estimating this expectation, using K iterations of Gibbs sampling alternating between sampling $\mathbf{h}$ and $\mathbf{y}$.

However, MF or LBP can also be used to approximate the expectation. Because the energy function decomposes into sums of either unary or pairwise terms, only the marginals $p(y_c = 1|\mathbf{x})$, $p(h_j = 1|\mathbf{x})$ and $p(y_c = 1, h_j = 1|\mathbf{x})$ are required. The assumption of a factorial distribution behind

Algorithm 2 Loopy Belief Propagation algorithm for inference in the multilabel ClassRBM

```
Input: training pair $(\mathbf{y}, \mathbf{x})$, number of iterations $K$ and damping factor $\beta$
$m^{\uparrow}_{jc} \leftarrow 0,\ \ m^{\downarrow}_{jc} \leftarrow 0 \quad \forall\, c, j$
$\mathbf{c}^{\text{data}} \leftarrow \mathbf{c} + \mathbf{W}\mathbf{x}$

# Update downwards (towards $\mathbf{y}$) and upwards (towards $\mathbf{h}$) messages
for $K$ iterations do
    $m^{\downarrow}_{jc} \leftarrow \beta m^{\downarrow}_{jc} + (1-\beta)\log\left(1 + (\exp(U_{jc}) - 1)\ \mathrm{sigm}(c^{\text{data}}_j + \sum_{c^* \neq c} m^{\uparrow}_{jc^*})\right), \quad \forall\, c, j$
    $m^{\uparrow}_{jc} \leftarrow \beta m^{\uparrow}_{jc} + (1-\beta)\log\left(1 + (\exp(U_{jc}) - 1)\ \mathrm{sigm}(d_c + \sum_{j^* \neq j} m^{\downarrow}_{j^*c})\right), \quad \forall\, c, j$
end for

# Compute estimated marginals
$p^{\text{LBP}}(y_c = 1|\mathbf{x}) \leftarrow \mathrm{sigm}(d_c + \sum_j m^{\downarrow}_{jc}), \quad \forall\, c$
$p^{\text{LBP}}(h_j = 1|\mathbf{x}) \leftarrow \mathrm{sigm}(c^{\text{data}}_j + \sum_c m^{\uparrow}_{jc}), \quad \forall\, j$

$\mathrm{num}^{01}_{jc} \leftarrow d_c + \sum_{j^* \neq j} m^{\downarrow}_{j^*c},\ \ \mathrm{num}^{10}_{jc} \leftarrow c^{\text{data}}_j + \sum_{c^* \neq c} m^{\uparrow}_{jc^*}, \quad \forall\, c, j$
$\mathrm{num}^{11}_{jc} \leftarrow U_{jc} + \mathrm{num}^{10}_{jc} + \mathrm{num}^{01}_{jc}, \quad \forall\, c, j$
$p^{\text{LBP}}(y_c = 1, h_j = 1|\mathbf{x}) = \exp(\mathrm{num}^{11}_{jc}) / (\exp(\mathrm{num}^{11}_{jc}) + \exp(\mathrm{num}^{01}_{jc}) + \exp(\mathrm{num}^{10}_{jc})), \quad \forall\, c, j$
```

MF means that $p(y_c = 1, h_j = 1|\mathbf{x})$ is simply estimated as the product of its estimates for $p(y_c = 1|\mathbf{x})$ and $p(h_j = 1|\mathbf{x})$, while LBP provides a more sophisticated estimate. The MF gradient estimates can also be improved by initializing the μ_c message to the value of the associated training target y_k. This approach was first described by Welling and Hinton (2002) and is known as mean field contrastive divergence. It was also extended to general variational approximations in Welling and Sutton (2005). When making predictions at test time however, we still must initialize μ_c to 0.

Finally, as in Section 7.3, the intractability of discriminative maximum likelihood training can be avoided by using a pseudolikelihood objective $-\sum_{c=1}^{C} \log p(y_c|\mathbf{y}_{\backslash c}\mathbf{x})$ for which exact gradients can be computed.

Given all of these possible ways of approximating the marginal posteriors $p(y_c = 1|\mathbf{x})$ at test time and of performing discriminative training, we performed an extensive comparison of all possible combinations of such choices. We used three different music social tags data sets based on databases of 10-second song clips. The first data set, was collected from Amazon.com's Mechanical Turk service and is described in Mandel et al. (2010). The second data set was collected from the MajorMiner music labeling game and is described in Mandel and Ellis (2008). The final data set was collected from Last.fm's website and is described in Schifanella et al. (2010). We will refer to these data sets as MTurk, MajMin and Last.fm respectively.

All of these data sets were in the form of (user, item, tag) triples, where the items were either 10-second clips of tracks or whole tracks. These data were condensed into (item, tag, count) triples by summing across users. Converting (item, tag, count) triples to binary matrices for training and evaluation purposes required some care. In the MajorMiner and Last.fm data, the counts were high enough that we could require the verification of an (item, tag) pair by at least two people, meaning that the count had to be at least 2 to be considered as a positive example. The Mechanical Turk data set did not have high enough counts to allow this, so we had to count every (item, tag) pair.

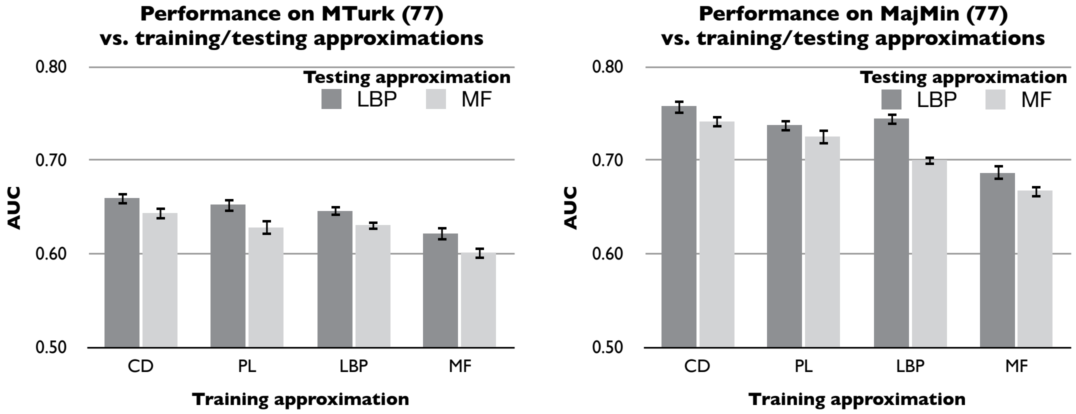


Figure 4: Results of the multilabel ClassRBM (discriminative training) on the Mechanical Turk and MajorMiner data sets, comparing the performance of different approximation combinations for training and testing. The approximations used during training are represented on the x-axis, while the approximations used during testing are represented through the color of the bar. The error bars correspond to the standard error across folds.

In the MajorMiner and Last.fm data sets, (item, tag) pairs with only a single count were not used as negative examples because we assumed that they had higher potential relevance than (item, tag) pairs that never occurred, which served as stronger negative examples.

The timbral and rhythmic features of Mandel and Ellis (2008) were used to characterize the audio of 10-second song clips. Each dimension of both sets of features was normalized across the database to have zero-mean and unit-variance, and then each feature vector was normalized to be unit norm to reduce the effect of outliers. The timbral features were 189-dimensional and the rhythmic features were 200-dimensional, making the combined feature vector 389-dimensional.

In order to asses the impact of different approximations (of the gradients or $p(\mathbf{y}|\mathbf{x})$) on the solution found by the model we only considered discriminative learning. We also augmented the number of data sets by changing the number of tags, to see how this factor influences the results. Next to a data set name, the number in parenthesis thus indicates the number of tags considered. The tags were selected by sorting them by popularity and picking the leading tags. For all data sets we select the hyper-parameters of the model using a 5-fold cross-validation. In order to increase the accuracy of our procedure, for each fold we computed the score as an average across 4 sub-folds. Each run used a different fold (from the remaining 4 folds) as the validation set and the other 3 as the training set. From this validation procedure, 50, 100 and 200 hidden units were selected respectively for the MTurk, MajMin and Last.fm data sets and a learning rate of 0.01 for all data sets. We also fixed a priori the number of iterations for approximating the gradients (for CD, MF or LBP) to 10, and the number of MF or LBP iterations for approximating $p(\mathbf{y}|\mathbf{x})$ to 20, to limit

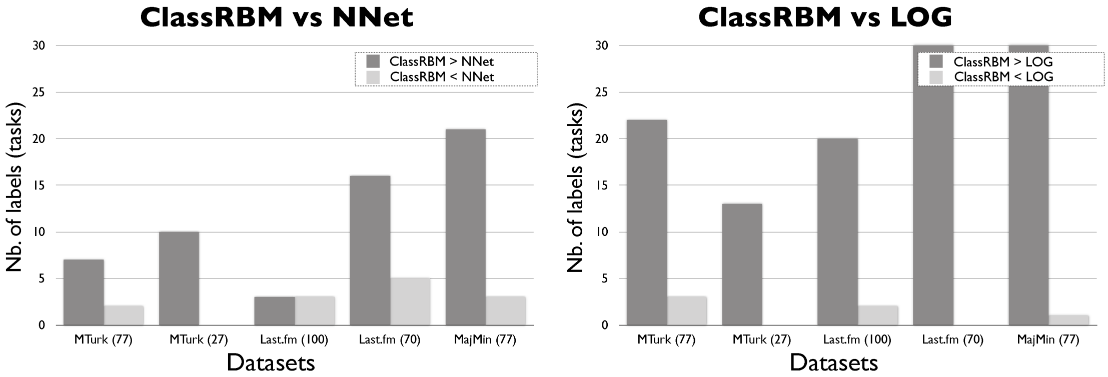


Figure 5: Comparison of the multilabel ClassRBM and with a multitask neural network (NNet) and with single task logistic regression classifiers (LOG). Bars show the number of labels (tags) on which the ClassRBM is significantly better (>) or worse (<) than the baseline in a two-sided paired t-test.

the hyper-parameter search space.[5] Finally, we set to 0.9 the damping factor for LBP inference, but other values were found to yield similar performances.

Figure 4 provides the performance of all possible combinations of approximations at training and test time, on two data sets. The performance is evaluated in terms of retrieval performance using the area under the ROC curve (AROC) (Cortes and Mohri, 2004).[6] We measure the AROC for each tag separately and use the average across tags and folds as an overall measure of performance. As we see, contrastive divergence tends to outperform other approaches for training the ClassRBM, either when mean field or loopy belief propagation is used at test time. Using the same deterministic inference at training and test time hence appears not to be optimal, with mean field being the worst option.

We also compared the performance of the ClassRBM with two baselines. The first is a multitask neural network (Caruana, 1997), which is among the best baselines for multitask learning. Moreover, a neural network makes for an interesting comparison because its prediction for the marginals $p(y_c = 1|\mathbf{x})$ is also non-linear, but feedforward and non-recursive, unlike in the ClassRBM. The second baseline is a set of single task logistic regression classifiers (one for each task). Though previous work on these multitask data sets has instead considered single task SVMs as a baseline (Mandel et al., 2011a), we have found logistic regression classifiers to outperform SVMs, hence we use those here as the single task baseline.

The same model selection procedure was used to select the baselines' hyper-parameters, namely the learning rate (both baselines) and hidden layer size (neural network baseline only). Contrastive divergence and loopy belief propagation was used in this comparison, for discriminative training.

[5] We validated this choice for these hyper-parameters afterwards, based on the best learning rate and hidden layer size found, and observed that while the performance increases with the number of iterations, the increase is not considerable, especially when we account for the increase in training time.

[6] This metric scores the ability of an algorithm to rank relevant examples in a collection above irrelevant examples. A random ranking will achieve an AROC of approximately 0.5, while a perfect ranking will achieve an AROC of 1.0.

Model	MTurk (77)	MTurk (27)	Last.fm (100)	Last.fm (70)	MajMin (77)
ClassRBM	65.9	68.8	72.4	72.2	76.1
NNet	65.8	65.4	72.4	72.0	75.3
LOG	63.4	65.7	70.2	70.3	70.7

Table 7: Average AROC across labels as a percentage for each model on all multitask data sets.

We compared the ClassRBM in a head to head fashion with each baseline, and computed a two-sided paired t-test across folds, per tag, to count the number of tags for which either model performs significantly better than the other. As illustrated in Figure 5, the ClassRBM is a better classifier for strictly more tags on all data sets when compared to the logistic regression approach and on 4 out of 5 data sets when compared to the neural network (with a tie on the remaining data set). Finally, Table 7 gives the absolute performance of the ClassRBM and the baselines.

10. Conclusion

We argued that RBMs can and should be used as stand-alone non-linear classifiers alongside other standard and more popular classifiers, instead of merely being considered as simple feature extractors. We considered different training strategies for the Classification RBM and evaluated them. In particular, we highlighted the importance of combining generative and discriminative training and we explored the impact of using different generative gradient estimators on the classification performance of the ClassRBM. We also extended the range of situations where the ClassRBM can be employed, by presenting learning algorithms tailored to settings where unlabeled data are available, where the input is sparse and very high-dimensional, as well as when multiple classification problems must be solved.

By describing and establishing the ClassRBM as a “black box” classifier in its own right, we hope to make its use more accessible and stimulate research in how to adapt it to even more application settings. As an illustration of this potential, we end by mentioning extensions of the ClassRBM that have already been developed, since the first conference publication of this work (Larochelle and Bengio, 2008). Gelfand et al. (2010) explored a different way of using the ClassRBM energy function to perform classification, using a conditional herding learning algorithm. Memisevic et al. (2010) investigated a variant of the ClassRBM with third-order (as opposed to pair-wise) interactions between the input, target and hidden units. van der Maaten et al. (2011) developed an extension for sequential classification problems with linear-chain interactions between the sequence of targets, while Mnih et al. (2011) considered other structured output prediction problems such as denoising. Finally Louradour and Larochelle (2011) adapted the ClassRBM to problems where the input $\mathbf{x}$ is a set containing an arbitrary number of input vectors.

Acknowledgments

We thank Dumitru Erhan for discussions about sparse RBMs. Earlier versions of some of the material presented here appeared in the ICML 2008 conference proceedings (Larochelle and Bengio, 2008) and on arXiv (Mandel et al., 2011b).

References

Arthur Asuncion, Qiang Liu, Alexander Ihler, and Padhraic Smyth. Learning with blocks: Composite likelihood and contrastive divergence. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, pages 33–40, 2010.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006a. URL `http://www.iro.umontreal.ca/~lisa/pointeurs/bengio_ssl.pdf`.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. The curse of highly variable functions for local kernel machines. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18 (NIPS'05)*, pages 107–114. MIT Press, Cambridge, MA, 2006b.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 153–160. MIT Press, 2007.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers. In *IASC International Symposium on Computational Statistics (COMPSTAT)*, pages 721–728, Prague, August 2004.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

Olivier Chapelle, Bernhard. Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

Corinna Cortes and Mohri Mohri. Auc optimization vs. error rate minimization. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 (NIPS'03)*, volume 16, Cambridge, MA, 2004. MIT Press.

Gregory Druck, Chris Pal, Andrew Mccallum, and Xiaojin Zhu. Semi-supervised classification with hybrid generative/discriminative methods. In *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, pages 280–289, New York, NY, USA, 2007. ACM.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, February 2010.

Peter V. Gehler, Alex D. Holub, and Max Welling. The rate adapting poisson model for information retrieval and object recognition. In William W. Cohen and Andrew Moore, editors, *Proceedings of the Twenty-three International Conference on Machine Learning (ICML'06)*, pages 337–344, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: http://doi.acm.org/10.1145/1143844.1143887.

Andrew Gelfand, Yutian Chen, Laurens van der Maaten, and Max Welling. On herding and the perceptron cycling theorem. 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'10)*, pages 694–702. Curran Associates, 2010.

Xavier Glorot, Antoire Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS'11)*, April 2011.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

Geoffrey E. Hinton. To recognize shapes, first learn to generate images. In Paul Cisek, Trevor Drew, and John Kalaska, editors, *Computational Neuroscience: Theoretical Insights into Brain Function*. Elsevier, 2007.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

Aapo Hyvärinen. Consistency of Pseudolikelihood Estimation of Fully Visible Boltzmann Machines. *Neural Computation*, 18:2283–2292, 2006.

Paul Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37 (2):101–114, 2008.

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, pages 536–543. ACM, 2008.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In Zoubin Ghahramani, editor, *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML'07)*, pages 473–480. ACM, 2007.

Nicolas Le Roux and Yoshua Bengio. Deep belief networks are compact universal approximators. *Neural Computation*, 22(8):2192–2207, August 2010. ISSN 0899-7667.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area V2. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS'07)*, pages 873–880. MIT Press, Cambridge, MA, 2008.

Percy Liang and Michael I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, pages 584–591, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390230.

Bruce G. Lindsay. Composite likelihood methods. *Contemporary Mathematics*, 80:221–239, 1988.

Jérôme Louradour and Hugo Larochelle. Classification of sets using restricted Boltzmann machines. In *Proceedings of the Twenty-seventh Conference on Uncertainty in Artificial Intelligence (UAI'11) (to appear)*. AUAI Press, 2011.

Michael I. Mandel and Daniel P. W. Ellis. A web-based game for collecting music metadata. *Journal of New Music Research*, 37(2):151–165, 2008.

Michael I. Mandel, Douglas Eck, and Yoshua Bengio. Learning tags that vary within a song. In *Proceedings of the Eleventh International Conference on Music Information Retrieval (ISMIR)*, pages 399–404, August 2010.

Michael I. Mandel, Razvan Pascanu, Douglas Eck, Yoshua Bengio, Luca M. Aiello, Rossano Schifanella, and Filippo Menczer. Contextual tag inference. *ACM Transactions on Multimedia Computing, Communications and Applications*, 7S(1):32:1–32:18, October 2011a.

Michael I. Mandel, Razvan Pascanu, Hugo Larochelle, and Yoshua Bengio. Autotagging music with conditional restricted Boltzmann machines. *ArXiv e-prints*, March 2011b.

Andrew McCallum, Chris Pal, Gregory Druck, and Xuerui Wang. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Twenty-first National Conference on Artificial Intelligence (AAAI'06)*. AAAI Press, 2006.

Roland Memisevic, Christopher Zach, Geoffrey Hinton, and Marc Pollefeys. Gated softmax classification. 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'10)*, pages 1603–1611. Curran Associates, 2010.

Andriy Mnih and Geoffrey E. Hinton. Three new graphical models for statistical language modelling. In Zoubin Ghahramani, editor, *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML'07)*, pages 641–648. ACM, 2007.

Volodymyr Mnih, Hugo Larochelle, and Geoffrey E. Hinton. Conditional restricted boltzmann machines for structured output prediction. In *Proceedings of the Twenty-seventh Conference on Uncertainty in Artificial Intelligence (UAI'11) (to appear)*. AUAI Press, 2011.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS'01)*, pages 841–848, 2002.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems:Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Marco Pretti. A message-passing algorithm with damping. *Journal of Statistical Mechanics: Theory and Experiment*, page P11008, 2005.

Ruslan Salakhutdinov and Geoffrey E. Hinton. Semantic hashing. In *Proceedings of the 2007 Workshop on Information Retrieval and applications of Graphical Models (SIGIR'07)*, Amsterdam, 2007. Elsevier.

Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Folks in folksonomies: Social link prediction from shared metadata. In *Proceedings of the Third International Conference on Web Search and Data Mining (WSDM'10)*, pages 271–280. ACM, Mar 2010.

Tanya Schmah, Geoffrey E. Hinton, Richard Zemel, Steven L. Small, and Stephen Strother. Generative versus discriminative training of RBMs for classification of fMRI images. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Leon Bottou, editors, *Advances in Neural Information Processing Systems 21 (NIPS'08)*, pages 1409–1416. Curran Associates, 2009.

Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, pages 1064–1071. ACM, 2008.

Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In Léon Bottou and Michael Littman, editors, *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML'09)*, pages 1033–1040. ACM, 2009. ISBN 978-1-60558-516-1. doi: http://doi.acm.org/10.1145/1553374.1553506.

Laurens van der Maaten, Max Welling, and Lawrence K. Saul. Hidden-unit conditional random fields. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS'11)*, volume 15 of JMLR: W&CP, 2011.

Yair Weiss. Comparing the mean field method and belief propagation for approximate inference in MRFs. In *Advanced Mean Field Methods - Theory and Practice*. MIT Press, 2001.

Max Welling and Geoffrey E. Hinton. A new learning algorithm for mean field Boltzmann machines. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*, pages 351–357, London, UK, 2002. Springer-Verlag. ISBN 3-540-44074-7.

Max Welling and Charles Sutton. Learning in markov random fields with contrastive free energies. In *In Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS'05)*, pages 397–404, 2005.

Max Welling, Michal Rosen-Zvi, and Geoffrey 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)*, volume 17, Cambridge, MA, 2005. MIT Press.

Eric P. Xing, Rong Yan, and Alexander G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proceedings of the Twenty-first Conference in Uncertainty in Artificial Intelligence (UAI'05)*, pages 633–641. AUAI Press, 2005. ISBN 0-9749039-1-4.

Jun Yang, Yan Liu, Eric P. Xing, and Alexander G. Hauptmann. Harmonium models for semantic video representation and classification. In *Proceedings of the Seventh SIAM International Conference on Data Mining (SDM'07)*. SIAM, 2007.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the Twenty International Conference on Machine Learning (ICML'03)*, pages 912–919. AAAI Press, 2003.

Submitted 9/11; Revised 2/12; Published 3/12

Structured Sparsity and Generalization

Andreas Maurer AM@ANDREAS-MAURER.EU
Adalbertstr. 55
D-80799, München
GERMANY

Massimiliano Pontil M.PONTIL@CS.UCL.AC.UK
Department of Computer Science
University College London
Gower St.
London, UK

Editor: Gabor Lugosi

Abstract

We present a data dependent generalization bound for a large class of regularized algorithms which implement structured sparsity constraints. The bound can be applied to standard squared-norm regularization, the Lasso, the group Lasso, some versions of the group Lasso with overlapping groups, multiple kernel learning and other regularization schemes. In all these cases competitive results are obtained. A novel feature of our bound is that it can be applied in an infinite dimensional setting such as the Lasso in a separable Hilbert space or multiple kernel learning with a countable number of kernels.

Keywords: empirical processes, Rademacher average, sparse estimation.

1. Introduction

We study a class of regularization methods used to learn a linear function from a finite set of examples. The regularizer is expressed as an infimum convolution which involves a set $\mathcal{M}$ of linear transformations (see Equation (1) below). As we shall see, this regularizer generalizes, depending on the choice of the set $\mathcal{M}$, the regularizers used by several learning algorithms, such as ridge regression, the Lasso, the group Lasso (Yuan and Lin, 2006), multiple kernel learning (Lanckriet et al., 2004; Bach et al., 2004), the group Lasso with overlap (Obozinski et al., 2009), and the regularizers in Micchelli et al. (2010).

We give a bound on the Rademacher average of the linear function class associated with this regularizer. The result matches existing bounds in the above mentioned cases but also admits a novel, dimension free interpretation. In particular, the bound applies to the Lasso in a separable Hilbert space or to multiple kernel learning with a countable number of kernels, under certain finite second-moment conditions.

We now introduce some necessary notation and state our main results. Let H be a real Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. Let $\mathcal{M}$ be an at most countable set of symmetric bounded linear operators on H such that for every $x \in H, x \neq 0$, there is some linear operator $M \in \mathcal{M}$ with $Mx \neq 0$ and that $\sup_{M \in \mathcal{M}} |||M||| < \infty$, where $||| \cdot |||$ is the operator norm. Define the function $\|\cdot\|_{\mathcal{M}} : H \to \mathbb{R}_+ \cup \{\infty\}$ by

$$\|\beta\|_{\mathcal{M}} = \inf\left\{ \sum_{M\in\mathcal{M}} \|v_M\| : v_M \in H,\ \sum_{M\in\mathcal{M}} M v_M = \beta \right\}. \tag{1}$$

It is shown in Section 3.2 that the chosen notation is justified, because $\|\cdot\|_{\mathcal{M}}$ is indeed a norm on the subspace of H where it is finite, and the dual norm is, for every $z \in H$, given by

$$\|z\|_{\mathcal{M}^*} = \sup_{M\in\mathcal{M}} \|Mz\|.$$

The somewhat complicated definition of $\|\cdot\|_{\mathcal{M}}$ is contrasted by the simple form of the dual norm.

As an example, if $H = \mathbb{R}^d$ and $M = \{P_1, \ldots, P_d\}$, where P_i is the orthogonal projection on the i-th coordinate, then the function (1) reduces to the ℓ_1 norm.

Using well known techniques, as described in Koltchinskii and Panchenko (2002) and Bartlett and Mendelson (2002), our study of generalization reduces to the search for a good bound on the empirical Rademacher complexity of a set of linear functionals with $\|\cdot\|_{\mathcal{M}}$-bounded weight vectors

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x}) = \frac{2}{n}\mathbb{E} \sup_{\beta:\ \|\beta\|_{\mathcal{M}}\leq 1} \sum_{i=1}^{n} \varepsilon_i \langle \beta, x_i \rangle, \tag{2}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in H^n$ is a sample vector representing observations, and $\varepsilon_1, \ldots, \varepsilon_n$ are Rademacher variables, mutually independent and each uniformly distributed on $\{-1, 1\}$.[1] Given a bound on $\mathcal{R}_{\mathcal{M}}(\mathbf{x})$ we obtain uniform bounds on the estimation error, for example using the following standard result (adapted from Bartlett and Mendelson 2002), where the Lipschitz function ϕ is to be interpreted as a loss function.

Theorem 1 *Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of iid random variables with values in H, let X be iid to X_1, let $\phi : \mathbb{R} \to [0,1]$ have Lipschitz constant L and $\delta \in (0,1)$. Then with probability at least $1-\delta$ in the draw of $\mathbf{X}$ it holds, for every $\beta \in \mathbb{R}^d$ with $\|\beta\|_{\mathcal{M}} \leq 1$, that*

$$\mathbb{E}\phi(\langle \beta, X\rangle) \leq \frac{1}{n}\sum_{i=1}^{n} \phi(\langle \beta, X_i\rangle) + L\, \mathcal{R}_{\mathcal{M}}(\mathbf{X}) + \sqrt{\frac{9 \ln 2/\delta}{2n}}.$$

A similar (slightly better) bound is obtained if $\mathcal{R}_{\mathcal{M}}(\mathbf{X})$ is replaced by its expectation $\mathcal{R}_{\mathcal{M}} = \mathbb{E}\mathcal{R}_{\mathcal{M}}(\mathbf{X})$ (see Bartlett and Mendelson 2002).

The following is the main result of this paper and leads to consistency proofs and finite sample generalization guarantees for all algorithms which use a regularizer of the form (1). A proof is given in Section 3.3.

Theorem 2 *Let $\mathbf{x} = (x_1, \ldots, x_n) \in H^n$ and $\mathcal{R}_{\mathcal{M}}(\mathbf{x})$ be defined as in (2). Then*

$$\begin{aligned}
\mathcal{R}_{\mathcal{M}}(\mathbf{x}) &\leq \frac{2^{3/2}}{n}\sqrt{\sup_{M\in\mathcal{M}} \sum_{i=1}^{n} \|Mx_i\|^2}\left(2 + \sqrt{\ln\left(\sum_{M\in\mathcal{M}} \frac{\sum_i \|Mx_i\|^2}{\sup_{N\in\mathcal{M}} \sum_j \|Nx_j\|^2}\right)}\right) \\
&\leq \frac{2^{3/2}}{n}\sqrt{\sum_{i=1}^{n} \|x_i\|_{\mathcal{M}^*}^2}\left(2 + \sqrt{\ln|\mathcal{M}|}\right).
\end{aligned}$$

[1] Our definition coincides with the one in Bartlett and Mendelson (2002), while other authors omit the factor of 2. This is relevant when comparing the constants in different bounds.

The second inequality follows from the first one, the inequality

$$\sup_{M\in\mathcal{M}}\sum_{i=1}^{n}\|Mx_i\|^2\leq\sum_{i=1}^{n}\|x_i\|^2_{\mathcal{M}^*},$$

a fact which will be tacitly used in the sequel, and the observation that every summand in the logarithm appearing in the first inequality is bounded by 1. Of course the second inequality is relevant only if $\mathcal{M}$ is finite. In this case we can draw the following conclusion: If we have an a priori bound on $\|X\|_{\mathcal{M}^*}$ for some data distribution, say $\|X\|_{\mathcal{M}^*}\leq C$, and $\mathbf{X}=(X_1,\dots,X_n)$, with X_i iid to X, then

$$\mathcal{R}_{\mathcal{M}}(\mathbf{X})\leq\frac{2^{3/2}C}{\sqrt{n}}\left(2+\sqrt{\ln|\mathcal{M}|}\right),$$

thus passing from a data-dependent to a distribution dependent bound. In Section 2 we show that this recovers existing results (Cortes et al., 2010; Kakade et al., 2010; Kloft et al., 2011; Meir and Zhang, 2003; Ying and Campbell, 2009) for many regularization schemes.[2]

But the first bound in Theorem 2 can be considerably smaller than the second and may be finite even if $\mathcal{M}$ is infinite. This gives rise to some novel features, even in the well studied case of the Lasso, when there is a (finite but potentially large) ℓ_2-bound on the data.

Corollary 3 *Under the conditions of Theorem 2 we have*

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x})\leq\frac{2^{3/2}}{n}\sqrt{\sup_{M\in\mathcal{M}}\sum_i\|Mx_i\|^2}\left(2+\sqrt{\ln\frac{1}{n}\sum_i\sum_{M\in\mathcal{M}}\|Mx_i\|^2}\right)+\frac{2}{\sqrt{n}}.$$

A proof is given in Section 3.3. To obtain a distribution dependent bound we retain the condition $\|X\|_{\mathcal{M}^*}\leq C$ and replace finiteness of $\mathcal{M}$ by the condition that

$$R^2:=\mathbb{E}\sum_{M\in\mathcal{M}}\|MX\|^2<\infty. \tag{3}$$

Taking the expectation in Corollary 3 and using Jensen's inequality then gives a bound on the expected Rademacher complexity

$$\mathcal{R}_{\mathcal{M}}\leq\frac{2^{3/2}C}{\sqrt{n}}\left(2+\sqrt{\ln R^2}\right)+\frac{2}{\sqrt{n}}. \tag{4}$$

The key features of this result are the dimension-independence and the only logarithmic dependence on R^2, which in many applications turns out to be simply $R^2=\mathbb{E}\|X\|^2$.

The rest of the paper is organized as follows. In the next section, we specialize our results to different regularizers. In Section 3, we present the proof of Theorem 2 as well as the proof of other results mentioned above. In Section 4, we discuss the extension of these results to the ℓ_q case. Finally, in Section 5, we draw our conclusions and comment on future work.

[2] We note that the numerical implementation and practical application of specific cases of the regularizer described here have been addressed in detail in a number of papers. We recommend Baldassarre et al. (2012), Obozinski et al. (2009) and Jenatton et al. (2011) and references therein for detailed information on such matters. We also refer to Baraniuk et al. (2010) and Huang et al. (2009) for related work using greedy methods.

2. Examples

Before giving the examples we mention a great simplification in the definition of the norm $\|\cdot\|_{\mathcal{M}}$ which occurs when the members of $\mathcal{M}$ have mutually orthogonal ranges. A simple argument, given in Proposition 8 below shows that in this case

$$\|\beta\|_{\mathcal{M}} = \sum_{M \in \mathcal{M}} \left\| M^{+}\beta \right\|,$$

where M^{+} is the pseudoinverse of M. If, *in addition*, every member of $\mathcal{M}$ is an orthogonal projection P, the norm further simplifies to

$$\|\beta\|_{\mathcal{M}} = \sum_{P \in \mathcal{M}} \|P\beta\|,$$

and the quantity R^2 occurring in the second moment condition (3) simplifies to

$$R^2 = \mathbb{E} \sum_{P \in \mathcal{M}} \|PX\|^2 = \mathbb{E}\|X\|^2 .$$

For the remainder of this section $\mathbf{X} = (X_1, \dots, X_n)$ will be a generic iid random vector of data points, $X_i \in H$, and X will be a generic data variable, iid to X_i. If $H = \mathbb{R}^d$ we write $(X)_k$ for the k-th coordinate of X, not to be confused with X_k, which would be the k-th member of the vector $\mathbf{X}$.

2.1 The Euclidean Regularizer

In this simplest case we set $\mathcal{M} = \{I\}$, where I is the identity operator on the Hilbert space H. Then $\|\beta\|_{\mathcal{M}} = \|\beta\|$, $\|z\|_{\mathcal{M}^*} = \|z\|$, and the bound on the empirical Rademacher complexity becomes

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x}) \leq \frac{2^{5/2}}{n} \sqrt{\sum_i \|x_i\|^2},$$

worse by a constant factor of $2^{3/2}$ than the corresponding result in Bartlett and Mendelson (2002), a tribute paid to the generality of our result.

2.2 The Lasso

Let us first assume that $H = \mathbb{R}^d$ is finite dimensional and set $\mathcal{M} = \{P_1, \dots, P_d\}$ where P_k is the orthogonal projection onto the 1-dimensional subspace generated by the basis vector e_k. All the above mentioned simplifications apply and we have $\|\beta\|_{\mathcal{M}} = \|\beta\|_1$ and $\|z\|_{\mathcal{M}^*} = \|z\|_\infty$. The bound on $\mathcal{R}_{\mathcal{M}}(\mathbf{x})$ now reads

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x}) \leq \frac{2^{3/2}}{n} \sqrt{\sum_i \|x_i\|_\infty^2} \left(2 + \sqrt{\ln d}\right).$$

If $\|X\|_\infty \leq 1$ almost surely we obtain

$$\mathcal{R}_{\mathcal{M}}(\mathbf{X}) \leq \frac{2^{3/2}}{\sqrt{n}} \left(2 + \sqrt{\ln d}\right),$$

which agrees with the bound in Kakade et al. (2010) on the dominant term (see also Bartlett and Mendelson 2002 and Meir and Zhang 2003).

Our last bound is useless if $d \geq e^n$ or if d is infinite. But whenever the norm of the data has finite second moments we can use Corollary 3 and inequality (4) to obtain

$$\mathcal{R}_{\mathcal{M}}(\mathbf{X}) \leq \frac{2^{3/2}}{\sqrt{n}}\left(2+\sqrt{\ln \mathbb{E}\,\|X\|_2^2}\right)+\frac{2}{\sqrt{n}}.$$

For nontrivial results $\mathbb{E}\,\|X\|^2$ only needs to be subexponential in n.

We remark that a similar condition to Equation (3) for the Lasso, replacing the expectation with the supremum over X, has been considered within the context of elastic net regularization (De Mol et al., 2009).

2.3 The Weighted Lasso

The Lasso assigns an equal penalty to all regression coefficients, while there may be a priori information on the respective significance of the different coordinates. For this reason different weightings have been proposed (see, for example, Shimamura et al. 2007). In our framework an appropriate set of operators is $\mathcal{M} = \{\alpha_1 P_1, \ldots, \alpha_k P_k, \ldots\}$, with $\alpha_k > 0$ where α_k^{-1} is the penalty weight associated with the k-th coordinate. Then

$$\|\beta\|_{\mathcal{M}} = \sum_k \alpha_k^{-1} |\beta_k|$$

and

$$\|z\|_{\mathcal{M}^*} = \sup_k \alpha_k |z_k|\,.$$

To further illustrate the use of Corollary 3 let us assume that the underlying space H is infinite dimensional (that is, $H = \ell_2(\mathbb{N})$), and make the compensating assumption that $\alpha \in H$, that is $\sum_k \alpha_k^2 = R^2 < \infty$. For simplicity we also assume that $\sup_k \alpha_k \leq 1$. Then, if $\|X\|_\infty \leq 1$ almost surely, we have both $\|X\|_{\mathcal{M}^*} \leq 1$ and $\sum_k \alpha_k^2 (X)_k^2 \leq R^2$. Again we obtain

$$\mathcal{R}_{\mathcal{M}}(\mathbf{X}) \leq \frac{2^{3/2}}{\sqrt{n}}\left(2+\sqrt{\ln R^2}\right)+\frac{2}{\sqrt{n}}.$$

So in this case the second moment bound is enforced by the weighting sequence.

2.4 The Group Lasso

Let $H = \mathbb{R}^d$ and let $\{J_1, \ldots, J_r\}$ be a partition of the index set $\{1, \ldots, d\}$. We take $\mathcal{M} = \{P_{J_1}, \ldots, P_{J_r}\}$ where $P_{J_\ell} = \sum_{i \in J_\ell} P_i$ is the projection onto the subspace spanned by the basis vector e_i. The ranges of the P_{J_ℓ} then provide an orthogonal decomposition of $\mathbb{R}^d$ and the above mentioned simplifications also apply. We get

$$\|\beta\|_{\mathcal{M}} = \sum_{\ell=1}^r \|P_{J_\ell}\beta\|$$

and

$$\|z\|_{\mathcal{M}^*} = \max_{\ell=1}^r \|P_{J_\ell} z\|\,.$$

The algorithm which uses $\|\beta\|_{\mathcal{M}}$ as a regularizer is called the group Lasso (see, for example, Yuan and Lin 2006). It encourages vectors β whose support lies the union of a small number of groups J_ℓ

of coordinate indices. If we know that $\|P_{J_\ell}X\| \leq 1$ almost surely for all $\ell \in \{1, \dots, r\}$ then we get

$$\mathcal{R}_{\mathcal{M}}(\mathbf{X}) \leq \frac{2^{3/2}}{\sqrt{n}} \left(2 + \sqrt{\ln r}\right), \tag{5}$$

in complete symmetry with the Lasso and essentially the same as given in Kakade et al. (2010). If r is prohibitively large or if different penalties are desired for different groups, the same remarks apply as in the previous two sections. Just as in the case of the Lasso the second moment condition (3) translates to the simple form $\mathbb{E}\|X\|_2^2 < \infty$.

2.5 Overlapping Groups

In the previous examples the members of $\mathcal{M}$ always had mutually orthogonal ranges, which gave a simple appearance to the norm $\|\beta\|_{\mathcal{M}}$. If the ranges are not mutually orthogonal, the norm has a more complicated form. For example, in the group Lasso setting, if the groups J_ℓ cover $\{1, \dots, d\}$, but are not disjoint, we obtain the regularizer of Obozinski et al. (2009), given by

$$\Omega_{\text{overlap}}(\beta) = \inf \left\{ \sum_{\ell=1}^{r} \|v_\ell\| : (v_\ell)_{jk} = 0 \text{ if } k \notin J_\ell \text{ and } \sum_{\ell=1}^{r} v_\ell = \beta \right\}.$$

If $\|P_{J_\ell}X_i\| \leq 1$ almost surely for all $\ell \in \{1, \dots, r\}$ then the Rademacher complexity of the set of linear functionals with $\Omega_{\text{overlap}}(\beta) \leq 1$ is bounded as in (5), in complete equivalence to the bound for the group Lasso.

The same bound also holds for the class satisfying $\Omega_{\text{group}}(\beta) \leq 1$, where the function Ω_{group} is defined, for every $\beta \in \mathbb{R}^d$, as

$$\Omega_{\text{group}}(\beta) = \sum_{\ell=1}^{r} \|P_{J_\ell}\beta\|$$

which has been proposed by Jenatton et al. (2011) and Zhao et al. (2009). To see this we only have to show that $\Omega_{\text{overlap}} \leq \Omega_{\text{group}}$ which is accomplished by generating a disjoint partition $\{J'_\ell\}_{\ell=1}^{r}$ where $J'_\ell \subseteq J_\ell$, writing $\beta = \sum_{\ell=1}^{r} P_{J'_\ell}\beta$ and realizing that $\left\|P_{J'_\ell}\beta\right\| \leq \|P_{J_\ell}\beta\|$. The bound obtained from this simple comparison may however be quite loose.

2.6 Regularizers Generated from Cones

Our next example considers structured sparsity regularizers as in Micchelli et al. (2010). Let Λ be a nonempty subset of the open positive orthant in $\mathbb{R}^d$ and define a function $\Omega_\Lambda : \mathbb{R}^d \to \mathbb{R}$ by

$$\Omega_\Lambda(\beta) = \frac{1}{2} \inf_{\lambda \in \Lambda} \sum_{j=1}^{d} \left(\frac{\beta_j^2}{\lambda_j} + \lambda_j \right).$$

If Λ is a convex cone, then it is shown in Micchelli et al. (2011) that Ω_Λ is a norm and that the dual norm is given by

$$\|z\|_{\Lambda^*} = \sup \left\{ \left(\sum_{j=1}^{d} \mu_j z_j^2 \right)^{1/2} : \mu_j = \lambda / \|\lambda\|_1 \text{ with } \lambda \in \Lambda \right\}.$$

The supremum in this formula is evidently attained on the set $\mathcal{E}(\Lambda)$ of extreme points of the closure of $\{\lambda/\|\lambda\|_1 : \lambda \in \Lambda\}$. For $\mu \in \mathcal{E}(\Lambda)$ let M_μ be the diagonal matrix whose diagonal entries are those of the vector μ_j and let $\mathcal{M}_\Lambda$ be the collection of matrices $\mathcal{M}_\Lambda = \left\{M_\mu : \mu \in \mathcal{E}(\Lambda)\right\}$. Then

$$\|z\|_{\Lambda^*} = \sup_{M\in\mathcal{M}_\Lambda} \|Mz\|.$$

Clearly $\mathcal{M}_\Lambda$ is uniformly bounded in the operator norm, so if Λ is a cone and $\mathcal{E}(\Lambda)$ is at most countable, then $\|\cdot\|_{\Lambda^*} = \|\cdot\|_{\mathcal{M}^*}$, $\Omega_\Lambda = \|\cdot\|_{\mathcal{M}^*}$ and our bounds apply. If $\mathcal{E}(\Lambda)$ is finite and $\mathbf{x}$ is a sample then the Rademacher complexity of the class with $\Omega_\Lambda(\beta) \leq 1$ is bounded by

$$\frac{2^{3/2}}{n}\sqrt{\sum_{i=1}^{n} \|x_i\|_{\Lambda^*}^2}\left(2+\sqrt{\ln|\mathcal{E}(\Lambda)|}\right).$$

2.7 Kernel Learning

This is the most general case to which the simplification applies: Suppose that H is the direct sum $H = \oplus_{j\in\mathcal{J}} H_j$ of an at most countable number of Hilbert spaces H_j. We set $\mathcal{M} = \left\{P_j\right\}_{j\in\mathcal{J}}$, where $P_j : H \to H$ is the projection on H_j. Then

$$\|\beta\|_{\mathcal{M}} = \sum_{j\in\mathcal{J}} \left\|P_j\beta\right\|$$

and

$$\|z\|_{\mathcal{M}^*} = \sup_{j\in\mathcal{J}} \left\|P_j z\right\|.$$

Such a situation arises in multiple kernel learning (Bach et al., 2004; Lanckriet et al., 2004) or the nonparametric group Lasso (Meier et al., 2009) in the following way: One has an input space $\mathcal{X}$ and a collection $\left\{K_j\right\}_{j\in\mathcal{J}}$ of positive definite kernels $K_j : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$. Let $\phi_j : \mathcal{X} \to H_j$ be the feature map representation associated with kernel K_j, so that, for every $x,t \in \mathcal{X}$ $K_j(x,t) = \langle\phi_j(x), \phi_j(t)\rangle$ (for background on kernel methods see, for example, Shawe-Taylor and Cristianini 2004).

Suppose that $\mathbf{x} = (x_1,\dots,x_n) \in \mathcal{X}^n$ is a sample. Define the kernel matrix $\mathbf{K}_j = (K_j(x_i,x_k))_{i,k=1}^n$. Using this notation the bound in Theorem 2 reads

$$\mathcal{R}((\phi(x_1),\dots,\phi(x_n))) \leq \frac{2^{3/2}}{n}\sqrt{\sup_{j\in\mathcal{J}} \operatorname{tr}\mathbf{K}_j}\left(2+\sqrt{\ln\frac{\sum_{j\in\mathcal{J}}\operatorname{tr}\mathbf{K}_j}{\sup_{j\in\mathcal{J}}\operatorname{tr}\mathbf{K}_j}}\right).$$

In particular, if $\mathcal{J}$ is finite and $K_j(x,x) \leq 1$ for every $x \in \mathcal{X}$ and $j \in \mathcal{J}$, then the the bound reduces to

$$\frac{2^{3/2}}{\sqrt{n}}\left(2+\sqrt{\ln|\mathcal{J}|}\right),$$

essentially in agreement with Cortes et al. (2010), Kakade et al. (2010) and Ying and Campbell (2009). Our leading constant of $2\sqrt{2}$ is slightly better than the constant of $2\sqrt{\frac{23}{22}e}$, given by Cortes et al. (2010).

For infinite or prohibitively large $\mathcal{J}$ the second moment condition now becomes

$$\mathbb{E}\sum_{j\in\mathcal{J}} K_j(X,X) < \infty.$$

We conclude this section by noting that, for every set $\mathcal{M}$ we may choose a set of kernels such that empirical risk minimization with the norm $\|\cdot\|_{\mathcal{M}}$ is equivalent to multiple kernel learning with kernels $K_M(x,t) = \langle Mx, Mt\rangle$, $M \in \mathcal{M}$. To see this, choose, for every $M \in \mathcal{M}$, $\phi_M(x) = Mx$. Note however, that this may yield an overparameterization of the problem. For example, the regularizers in Section 2.6 can be reformulated as a multiple kernel learning problem, but this requires $d|\mathcal{E}(\Lambda)|$ parameters instead of d.

3. Proofs

We first give some notation and auxiliary results, then we prove the results announced in the introduction.

3.1 Notation and Auxiliary Results

The Hilbert space H and the collection M are fixed throughout the following, as is the sample size $n \in \mathbb{N}$.

Recall that $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ denote the norm and inner product in H, respectively. For a linear transformation $M : \mathbb{R}^n \to H$ the Hilbert-Schmidt norm is defined as

$$\|M\|_{HS} = \left(\sum_{i=1}^{n} \|Me_i\|^2\right)^{1/2}$$

where $\{e_i : i \in \mathbb{N}\}$ is the canonical basis of $\mathbb{R}^n$.

We use bold letters ($\mathbf{x}$, $\mathbf{X}$, $\boldsymbol{\varepsilon}$, ...) to denote n-tuples of objects, such as vectors or random variables.

Let $\mathcal{X}$ be any space. For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, $1 \leq k \leq n$ and $y \in \mathcal{X}$ we use $\mathbf{x}_{k\leftarrow y}$ to denote the object obtained from $\mathbf{x}$ by replacing the k-th coordinate of $\mathbf{x}$ with y. That is

$$\mathbf{x}_{k\leftarrow y} = (x_1, \ldots, x_{k-1}, y, x_{k+1}, \ldots, x_n).$$

The following concentration inequality, known as the bounded difference inequality (see McDiarmid 1998), goes back to the work of Hoeffding (1963). We only need it in the weak form stated below.

Theorem 4 *Let $F : \mathcal{X}^n \to \mathbb{R}$ and write*

$$B^2 = \sum_{k=1}^{n} \sup_{y_1, y_2 \in \mathcal{X},\, \mathbf{x} \in \mathcal{X}^n} \left(F\left(\mathbf{x}_{k\leftarrow y_1}\right) - F\left(\mathbf{x}_{k\leftarrow y_2}\right)\right)^2.$$

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of independent random variables with values in $\mathcal{X}$, and let $\mathbf{X}'$ be iid to $\mathbf{X}$. Then for any $t > 0$

$$\Pr\left\{F(\mathbf{X}) > \mathbb{E}F\left(\mathbf{X}'\right) + t\right\} \leq e^{-2t^2/B^2}.$$

Finally we need a simple lemma on the normal approximation:

Lemma 5 *Let $a, \delta > 0$. Then*

$$\int_{\delta}^{\infty} \exp\left(\frac{-t^2}{2a^2}\right) dt \leq \frac{a^2}{\delta} \exp\left(\frac{-\delta^2}{2a^2}\right).$$

Proof For $t \geq \delta/a$ we have $1 \leq at/\delta$. Thus

$$\int_{\delta}^{\infty} \exp\left(\frac{-t^2}{2a^2}\right) dt = a \int_{\delta/a}^{\infty} e^{-t^2/2} dt \leq \frac{a^2}{\delta} \int_{\delta/a}^{\infty} t e^{-t^2/2} dt = \frac{a^2}{\delta} \exp\left(\frac{-\delta^2}{2a^2}\right).$$

■

3.2 Properties of the Regularizer

In this section, we show that the regularizer in Equation (1) is indeed a norm and we derive the associated dual norm. In parallel we treat an entire class of regularizers, which relates to $\|\cdot\|_{\mathcal{M}}$ as the ℓ_q-norm relates to the ℓ_1-norm. To this end, we fix an exponent $q \in [1, \infty]$. The conjugate exponent is denoted p, with $1/q + 1/p = 1$.

Recall that $|||\cdot|||$ denotes the operator norm. We first state the general conditions on the set $\mathcal{M}$ of operators.

Condition 6 *$\mathcal{M}$ is an at most countable set of symmetric bounded linear operators on a real separable Hilbert space H such that*

(a) *For every $x \in H$ with $x \neq 0$, there exists $M \in \mathcal{M}$ such that $Mx \neq 0$*

(b) $\sup_{M \in \mathcal{M}} |||M||| < \infty$ *if $q = 1$ and* $\sum_{M \in \mathcal{M}} |||M|||^p < \infty$ *if $q > 1$.*

Now we define $\ell_q(\mathcal{M})$ to be the set of those vectors $\beta \in H$ for which the quantity

$$\|\beta\|_{\mathcal{M}_q} = \inf \left\{ \left(\sum_{M \in \mathcal{M}} \|v_M\|^q \right)^{1/q} : v_M \in H \text{ and } \sum_{M \in \mathcal{M}} M v_M = \beta \right\}$$

is finite. If $q = 1$ we drop the subscript in $\|\cdot\|_{\mathcal{M}_q}$ to lighten notation. Observe that the case $q = 1$ coincides with the definition given in the introduction.

Theorem 7 *$\ell_q(\mathcal{M})$ is a Banach space with norm $\|\cdot\|_{\mathcal{M}_q}$, and $\ell_q(\mathcal{M})$ is dense in H. If $\mathcal{M}$ is finite or H is finite-dimensional, then $\ell_q(\mathcal{M}) = H$. For $z \in H$ the norm of the linear functional $\beta \in \ell_q(\mathcal{M}) \mapsto \langle \beta, z \rangle$ is*

$$\|z\|_{\mathcal{M}_{q^*}} = \begin{cases} \sup_{M \in \mathcal{M}} \|Mz\|, & \text{if } q = 1, \\ \left(\sum_{M \in \mathcal{M}} \|Mz\|^p \right)^{1/p}, & \text{if } q > 1. \end{cases}$$

Proof Let $\mathcal{V}_q(\mathcal{M}) = \{v : v = (v_M)_{M \in \mathcal{M}},\ v_M \in H\}$ be the set of those H-valued sequences indexed by $\mathcal{M}$, for which the function

$$v \mapsto \|v\|_{\mathcal{V}_q(\mathcal{M})} = \left(\sum_{M \in \mathcal{M}} \|v_M\|^q \right)^{1/q}$$

is finite. Then $\|\cdot\|_{\mathcal{V}_q(\mathcal{M})}$ defines a complete norm on $\mathcal{V}_q(\mathcal{M})$, making $\mathcal{V}_q(\mathcal{M})$ a Banach space. If $w = (w_M)_{M\in\mathcal{M}}$ is an H-valued sequence indexed by $\mathcal{M}$, then the linear functional

$$v \in \mathcal{V}_q(\mathcal{M}) \mapsto \sum_{M\in\mathcal{M}} \langle v_M, w_M\rangle$$

has norm

$$\|w\|_{\mathcal{V}_q(\mathcal{M})^*} = \begin{cases} \sup_{M\in\mathcal{M}} \|Mw_M\|, & \text{if } q = 1, \\ \left(\sum_{M\in\mathcal{M}} \|v_M\|^p\right)^{1/p}, & \text{if } q > 1. \end{cases}$$

The verification of these claims parallels that of the standard results on Lebesgue spaces.

Now define a map

$$A : v \in \mathcal{V}_q(\mathcal{M}) \mapsto \sum_{M\in\mathcal{M}} Mv_M \in H.$$

We have

$$\|Av\| \le \sum_{M\in\mathcal{M}} |||M|||\, \|v_M\|.$$

By Condition 6(b) and Hölder's inequality A is a bounded linear transformation whose kernel $\mathcal{K}$ is therefore closed, making the quotient space $\mathcal{V}_q(\mathcal{M})/\mathcal{K}$ into a Banach space with quotient norm $\|w+\mathcal{K}\|_Q = \inf\left\{\|v\|_{\mathcal{V}_q(\mathcal{M})} : w - v \in \mathcal{K}\right\}$. The map A induces an isomorphism

$$\hat{A} : w + \mathcal{K} \in \mathcal{V}_q(\mathcal{M})/\mathcal{K} \mapsto Aw \in H.$$

The range of $\hat{A}$ is $\ell_q(\mathcal{M})$ and becomes a Banach space with the norm $\left\|\hat{A}^{-1}(\beta)\right\|_Q$. But

$$\begin{aligned} \left\|\hat{A}^{-1}(\beta)\right\|_Q &= \inf\left\{\|v\|_{\mathcal{V}_q(\mathcal{M})} : \hat{A}^{-1}(\beta) - v \in \mathcal{K}\right\} \\ &= \inf\left\{\|v\|_{\mathcal{V}_q(\mathcal{M})} : \beta = Av\right\} = \|\beta\|_{\mathcal{M}_q}, \end{aligned}$$

so $\|.\|_{\mathcal{M}_q}$ is a norm making $\ell_q(\mathcal{M})$ into a Banach space.

Suppose that $w \in H$ is orthogonal to $\ell_q(\mathcal{M})$. Let $M_0 \in \mathcal{M}$ be arbitrary and define $v = (v_M)$ by $v_{M_0} = M_0 w$ and $v_M = 0$ for all other M. Then

$$0 = \langle w, Av\rangle = \left\langle w, M_0^2 w\right\rangle = \|M_0 w\|^2,$$

so $M_0 = 0$. This holds for any $M_0 \in \mathcal{M}$, so Condition 6(a) implies that $w = 0$. By the Hahn Banach Theorem $\ell_q(\mathcal{M})$ is therefore dense in H. If $\mathcal{M}$ is finite or H is finite-dimensional, then $\ell_q(\mathcal{M})$ is also finite-dimensional and closed and thus $\ell_q(\mathcal{M}) = H$.

For the last assertion let $z \in H$. Then

$$\begin{aligned} \|z\|_{\mathcal{M}_q^*} &= \sup\left\{\langle z, \beta\rangle : \|\beta\|_{\mathcal{M}_q} \le 1\right\} \\ &= \sup\left\{\langle z, Av\rangle : \|v\|_{\mathcal{V}_q(\mathcal{M})} \le 1\right\} \\ &= \sup\left\{\langle A^* z, v\rangle : \|v\|_{\mathcal{V}_q(\mathcal{M})} \le 1\right\} \\ &= \|A^* z\|_{\mathcal{V}_q(\mathcal{M})^*} \\ &= \sup_{M\in\mathcal{M}} \|Mz\| \text{ if } q = 1 \text{ or } \left(\sum_{M\in\mathcal{M}} \|Mz\|^p\right)^{1/p} \text{ if } q > 1. \end{aligned}$$

■

Proposition 8 *If the ranges of the members of $\mathcal{M}$ are mutually orthogonal then for $\beta \in \ell_1(\mathcal{M})$*

$$\|\beta\|_{\mathcal{M}} = \sum_{M \in \mathcal{M}} \left\|M^+\beta\right\|,$$

where M^+ is the pseudoinverse of M.

Proof The ranges of the members of $\mathcal{M}$ provide an orthogonal decomposition of H, so

$$\beta = \sum_{M \in \mathcal{M}} M\left(M^+\beta\right),$$

where we used the fact that MM^+ is the orthogonal projection onto the range of M. Taking $v_M = M^+\beta$ this implies that $\|\beta\|_{\mathcal{M}} \leq \sum_{M \in \mathcal{M}} \|M^+\beta\|$. On the other hand, if $\beta = \sum_{N \in \mathcal{M}} N v_N$, then, applying M^+ to this identity we see that $M^+ M v_M = M^+\beta$ for all M, so

$$\sum_{M \in \mathcal{M}} \|v_M\| \geq \sum_{M \in \mathcal{M}} \left\|M^+ M v_M\right\| = \sum_{M \in \mathcal{M}} \left\|M^+\beta\right\|,$$

which shows the reverse inequality. ■

3.3 Bounds for the $\ell_1(\mathcal{M})$-Norm Regularizer

We use the bounded difference inequality to derive a concentration inequality for linearly transformed random vectors.

Lemma 9 *Let $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ be a vector of independent real random variables with $-1 \leq \varepsilon_i \leq 1$, and ε' iid to ε. Suppose that M is a linear transformation $M : \mathbb{R}^n \to H$.*

(i) *Then for $t > 0$ we have*

$$\Pr\left\{\|M\varepsilon\| \geq \mathbb{E}\left\|M\varepsilon'\right\| + t\right\} \leq \exp\left(\frac{-t^2}{2\|M\|_{HS}^2}\right).$$

(ii) *If ε is orthonormal (satisfying $\mathbb{E}\varepsilon_i\varepsilon_j = \delta_{ij}$), then*

$$\mathbb{E}\|M\varepsilon\| \leq \|M\|_{HS}. \tag{6}$$

and, for every $r > 0$,

$$\Pr\{\|M\varepsilon\| > t\} \leq e^{1/r} \exp\left(\frac{-t^2}{(2+r)\|M\|_{HS}^2}\right).$$

Proof (i) Define $F : [-1,1]^n \to \mathbb{R}$ by $F(\mathbf{x}) = \|M\mathbf{x}\|$. By the triangle inequality

$$\begin{aligned} & \sum_{k=1}^{n} \sup_{y_1,y_2\in[-1,1],\, \mathbf{x}\in[-1,1]^n} \left(F\left(\mathbf{x}_{k\leftarrow y_1}\right) - F\left(\mathbf{x}_{k\leftarrow y_2}\right)\right)^2 \\ \leq & \sum_{k=1}^{n} \sup_{y_1,y_2\in[-1,1],\, \mathbf{x}\in[-1,1]^n} \left\|M\left(\mathbf{x}_{k\leftarrow y_1} - \mathbf{x}_{k\leftarrow y_2}\right)\right\|^2 \\ = & \sum_{k=1}^{n} \sup_{y_1,y_2\in[-1,1]} (y_1-y_2)^2 \|Me_k\|^2 \\ \leq & \; 4\|M\|_{HS}^2 . \end{aligned}$$

The result now follows from the bounded difference inequality (Theorem 4).

(ii) If ε is orthonormal then it follows from Jensen's inequality that

$$\mathbb{E}\|M\varepsilon\| \leq \left(\mathbb{E}\left\|\sum_{i=1}^{n}\varepsilon_i Me_i\right\|^2\right)^{1/2} = \left(\sum_{i=1}^{n}\|Me_i\|^2\right)^{1/2} = \|M\|_{HS} .$$

For the second assertion of (ii) first note that from calculus we get $(t-1)^2/2 - t^2/(2+r) \geq -1/r$ for all $t\in\mathbb{R}$. This implies that

$$e^{-(t-1)^2/2} \leq e^{1/r}e^{-t^2/(2+r)}. \qquad (7)$$

Since $1/r \geq 1/(2+r)$ the inequality to be proved is trivial for $t \leq \|M\|_{HS}$. If $t > \|M\|_{HS}$ then, using $\mathbb{E}\|M\varepsilon\| \leq \|M\|_{HS}$, we have $t - E\|M\varepsilon\| \geq t - \|M\|_{HS} > 0$, so by part (i) and (7) we obtain

$$\begin{aligned} \Pr\{\|M\varepsilon\| \geq t\} &= \Pr\{\|M\varepsilon\| \geq E\|M\varepsilon\| + (t - E\|M\varepsilon\|)\} \\ &\leq \exp\left(\frac{-(t-E\|M\varepsilon\|)^2}{2\|M\|_{HS}^2}\right) \leq \exp\left(\frac{-(t-\|M\|_{HS})^2}{2\|M\|_{HS}^2}\right) \\ &= \exp\left(\frac{-(t/\|M\|_{HS}-1)^2}{2}\right) \leq e^{1/r}e^{-(t/\|M\|_{HS})^2/(2+r)} \\ &= e^{1/r}\exp\left(\frac{-t^2}{(2+r)\|M\|_{HS}^2}\right). \end{aligned}$$

■

We now use integration by parts, a union bound and the above concentration inequality to derive a bound on the expectation of the supremum of the norms $\|M\varepsilon\|$. This is the essential step in the proof of Theorem 2. It is by no means a new technique, in fact it appears many times in the book by Ledoux and Talagrand (1991), but compared to the combinatorial approach by Cortes et al. (2010) it seems more suited to the study of the problem at hand, and gives insights into the fine structure of the logarithmic factor appearing in bounds for Lasso-like methods.

Lemma 10 *Let $\mathcal{M}$ be an at most countable set of linear transformations $M : \mathbb{R}^n \to H$ and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ a vector of orthonormal random variables (satisfying $\mathbb{E}\varepsilon_i\varepsilon_j = \delta_{ij}$) with values in $[-1,1]$. Then*

$$\mathbb{E} \sup_{M\in\mathcal{M}} \|M\varepsilon\| \le \sqrt{2} \sup_{M\in\mathcal{M}} \|M\|_{HS} \left(2 + \sqrt{\ln \frac{\sum_{M\in\mathcal{M}} \|M\|_{HS}^2}{\sup_{M\in\mathcal{M}} \|M\|_{HS}^2}} \right).$$

Proof To lighten notation we abbreviate $\mathcal{M}_\infty := \sup_{M\in\mathcal{M}} \|M\|_{HS}$ below. We now use integration by parts

$$\begin{aligned}
\mathbb{E} \sup_{M\in\mathcal{M}} \|M\varepsilon\| &= \int_0^\infty \Pr\left\{ \sup_{M\in\mathcal{M}} \|M\varepsilon\| > t \right\} dt \\
&\le \mathcal{M}_\infty + \delta + \int_{\mathcal{M}_\infty+\delta}^\infty \Pr\left\{ \sup_{M\in\mathcal{M}} \|M\varepsilon\| > t \right\} dt \\
&\le \mathcal{M}_\infty + \delta + \sum_{M\in\mathcal{M}} \int_{\mathcal{M}_\infty+\delta}^\infty \Pr\{\|M\varepsilon\| > t\} dt,
\end{aligned}$$

where we have introduced a parameter $\delta \ge 0$. The first inequality above follows from the fact that probabilities never exceed 1, and the second from a union bound. Now for any $M \in \mathcal{M}$ we can make a change of variables and use (6), which gives $\mathbb{E}\|M\varepsilon\| \le \|M\|_{HS} \le \mathcal{M}_\infty$, so that

$$\begin{aligned}
\int_{\mathcal{M}_\infty+\delta}^\infty \Pr\{\|M\varepsilon\| > t\} dt &\le \int_\delta^\infty \Pr\{\|M\varepsilon\| > \mathbb{E}\|M\varepsilon\| + t\} dt \\
&\le \int_\delta^\infty \exp\left(\frac{-t^2}{2\|M\|_{HS}^2} \right) dt \\
&\le \frac{\|M\|_{HS}^2}{\delta} \exp\left(\frac{-\delta^2}{2\|M\|_{HS}^2} \right),
\end{aligned}$$

where the second inequality follows from Lemma 9-(i), and the third from Lemma 5. Substitution in the previous chain of inequalities and using Hoelder's inequality (in the ℓ_1/ℓ_∞-version) give

$$\mathbb{E} \sup_{M\in\mathcal{M}} \|M\varepsilon\| \le \mathcal{M}_\infty + \delta + \frac{1}{\delta} \left(\sum_{M\in\mathcal{M}} \|M\|_{HS}^2 \right) \exp\left(\frac{-\delta^2}{2\mathcal{M}_\infty^2} \right). \tag{8}$$

We now set

$$\delta = \mathcal{M}_\infty \sqrt{2 \ln \left(e \frac{\sum_{M\in\mathcal{M}} \|M\|_{HS}^2}{\mathcal{M}_\infty^2} \right)}.$$

Then $\delta \ge 0$ as required. The substitution makes the last term in (8) smaller than $\mathcal{M}_\infty / \left(e\sqrt{2} \right)$, and since $1 + 1/\left(e\sqrt{2} \right) < \sqrt{2}$, we obtain

$$\mathbb{E} \sup_{M\in\mathcal{M}} \|M\varepsilon\| \le \sqrt{2}\mathcal{M}_\infty \left(1 + \sqrt{\ln \left(\frac{e \sum_{M\in\mathcal{M}} \|M\|_{HS}^2}{\mathcal{M}_\infty^2} \right)} \right).$$

Finally we use $\sqrt{\ln es} \leq 1 + \sqrt{\ln s}$ for $s \geq 1$. ■

Proof of Theorem 2 Let $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ be a vector of iid Rademacher variables. For $M \in \mathcal{M}$ we use $M\mathbf{x}$ to denote the linear transformation $M\mathbf{x} : \mathbb{R}^n \to H$ given by $(M\mathbf{x})\,\mathbf{y} = \sum_{i=1}^n (Mx_i)\, y_i$. We have

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x}) = \frac{2}{n}\mathbb{E} \sup_{\beta : \|\beta\|_{\mathcal{M}} \leq 1} \left\langle \beta, \sum_{i=1}^n \varepsilon_i x_i \right\rangle \leq \frac{2}{n}\mathbb{E} \left\| \sum_{i=1}^n \varepsilon_i x_i \right\|_{\mathcal{M}^*} = \frac{2}{n}\mathbb{E} \sup_{M \in \mathcal{M}} \|M\mathbf{x}\varepsilon\| .$$

Applying Lemma 10 to the set of transformations $\mathcal{M}\mathbf{x} = \left\{ M\mathbf{x} : M \in \mathcal{M} \right\}$ gives

$$\mathcal{R}_{\mathcal{M}}(\mathbf{x}) \leq \frac{2^{3/2} \sup_{M \in \mathcal{M}} \|M\mathbf{x}\|_{HS}}{n} \left(2 + \sqrt{\ln \frac{\sum_{M \in \mathcal{M}} \|M\mathbf{x}\|_{HS}^2}{\sup_{M \in \mathcal{M}} \|M\mathbf{x}\|_{HS}^2}} \right).$$

Substitution of $\|M\mathbf{x}\|_{HS}^2 = \sum_{i=1}^n \|Mx_i\|^2$ gives the first inequality of Theorem 2 and

$$\sup_{M \in \mathcal{M}} \|M\mathbf{x}\|_{HS}^2 \leq \sum_{i=1}^n \sup_{M \in \mathcal{M}} \|Mx_i\|^2 = \sum_{i=1}^n \|x_i\|_{\mathcal{M}^*}^2$$

gives the second inequality. ■

Proof of Corollary 3 From calculus we find that $t \ln t \geq -1/e$ for all $t > 0$. For $A, B > 0$ and $n \in \mathbb{N}$ this implies that

$$A \ln \frac{B}{A} = n\left[(A/n)\ln(B/n) - (A/n)\ln(A/n)\right] \leq A \ln(B/n) + n/e. \tag{9}$$

Now multiply out the first inequality of Theorem 2 and use (9) with

$$A = \sup_{M \in \mathcal{M}} \sum_{i=1}^n \|Mx_i\|^2 \text{ and } B = \sum_{M \in \mathcal{M}} \sum_{i=1}^n \|Mx_i\|^2 .$$

Finally use $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b > 0$ and the fact that $2^{3/2}/\sqrt{e} \leq 2$. ■

4. The $\ell_q(\mathcal{M})$ Case

In this section we give bounds for the $\ell_q(\mathcal{M})$-norm regularizers, with $q > 1$.

We give two results, which can be applied to cases analogous to those in Section 2. The first result is essentially equivalent to Cortes et al. (2010), Kakade et al. (2010) and Kloft et al. (2011) and is presented for completeness. The second result is not dimension free, but it approaches the bound in Theorem 2 for arbitrarily large dimensions. The proofs are analogous to the proof of Theorem 2.

Theorem 11 *Let* $\mathbf{x}$ *be a sample and* $\mathcal{R}_{\mathcal{M}_q}(\mathbf{x})$ *the empirical Rademacher complexity of the class of linear functions parameterized by* β *with* $\|\beta\|_{\mathcal{M}_q} \leq 1$*. Then for* $1 < q \leq 2$

$$\mathcal{R}_{\mathcal{M}_q}(\mathbf{x}) \leq \frac{2}{n} \left(1 + \left(\frac{\pi}{2} \right)^{\frac{1}{2p}} \sqrt{p} \right) \sqrt{\sum_{i=1}^n \|x_i\|_{\mathcal{M}_q^*}^2}.$$

The proof is based on the following

Lemma 12 *Let $\mathcal{M}$ be an at most countable set of linear transformations $M : \mathbb{R}^n \to H$ and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ a vector of orthonormal random variables (satisfying $\mathbb{E}\varepsilon_i\varepsilon_j = \delta_{ij}$) with values in $[-1,1]$. Then for $p \geq 2$*

$$\mathbb{E}\left[\left(\sum_{M\in\mathcal{M}} \|M\varepsilon\|^p\right)^{1/p}\right] \leq \left(1+\left(\frac{\pi}{2}\right)^{\frac{1}{2p}}\sqrt{p}\right)\left(\sum_{M\in\mathcal{M}} \|M\|_{HS}^p\right)^{1/p}.$$

Proof We first note, by Jensen inequality, that

$$\mathbb{E}\left[\left(\sum_{M\in\mathcal{M}} \|M\varepsilon\|^p\right)^{1/p}\right] \leq \left(\mathbb{E}\left[\sum_{M\in\mathcal{M}} \|M\varepsilon\|^p\right]\right)^{1/p}. \tag{10}$$

We rewrite the expectation appearing in the right hand side using integration by parts and a change of variable as

$$\mathbb{E}\left[\|M\varepsilon\|^p\right] = \int_0^\infty \Pr\{\|M\varepsilon\|^p > t\}\,dt = A^p + p\int_0^\infty \Pr\{\|M\varepsilon\|^p > s^p + A^p\}\,s^{p-1}ds \tag{11}$$

where $A \geq 0$. Next, we use convexity of the function $x \mapsto x^p, x \geq 0$, which gives for $\lambda \in (0,1)$

$$\left(\lambda^{\frac{p-1}{p}}s + (1-\lambda)^{\frac{p-1}{p}}A\right)^p \leq \lambda\left(\frac{s}{\lambda^{1/p}}\right)^p + (1-\lambda)\left(\frac{A}{(1-\lambda)^{1/p}}\right)^p = s^p + A^p.$$

This allows us to bound

$$\begin{aligned}\Pr\{\|M\varepsilon\|^p > s^p + A^p\} &\leq \Pr\left\{\|M\varepsilon\|^p > \left(\lambda^{\frac{p-1}{p}}s + (1-\lambda)^{\frac{p-1}{p}}A\right)^p\right\} \\ &= \Pr\left\{\|M\varepsilon\| > \lambda^{\frac{p-1}{p}}s + (1-\lambda)^{\frac{p-1}{p}}A\right\}.\end{aligned} \tag{12}$$

Combining Equations (11) and (12), choosing $A = (1-\lambda)^{\frac{1-p}{p}}\|M\|_{HS}$ and making the change of variable $t = \lambda^{\frac{p-1}{p}}s$, gives

$$\begin{aligned}\int_0^\infty \Pr\{\|M\varepsilon\|^p > t\}\,dt &\leq (1-\lambda)^{1-p}\|M\|_{HS}^p + p\lambda^{1-p}\int_0^\infty \Pr\{\|M\varepsilon\| > \|M\|_{HS} + t\}\,t^{p-1}dt \\ &\leq (1-\lambda)^{1-p}\|M\|_{HS}^p + p\lambda^{1-p}\int_0^\infty t^{1-p}e^{-t^2/2\|M\|_{HS}^2}dt \\ &\leq (1-\lambda)^{1-p}\|M\|_{HS}^p + p\lambda^{1-p}\|M\|_{HS}^p\sqrt{\frac{\pi}{2}}p^{p/2-1} \\ &= \|M\|_{HS}^p\left((1-\lambda)^{1-p} + \lambda^{1-p}\sqrt{\frac{\pi}{2}}p^{p/2}\right)\end{aligned}$$

where the second inequality follows by Lemma 9-(i) and the third inequality follows by a standard result on the moments of the normal distribution, namely

$$\int_0^\infty t^{p-1}\exp\left(\frac{-t^2}{2}\right)dt \leq \sqrt{\frac{\pi}{2}}(p-2)!! \leq \sqrt{\frac{\pi}{2}}(1\cdot 3\cdot \ldots \cdot p-2) \leq \sqrt{\frac{\pi}{2}}p^{p/2-1}.$$

Summing both sides of Equation (13) over M we obtain that

$$\mathbb{E}\left[\sum_{M\in\mathcal{M}}\|M\varepsilon\|^p\right]\le\sum_M\|M\|_{HS}^p\left((1-\lambda)^{1-p}+\lambda^{1-p}\sqrt{\frac{\pi}{2}}p^{p/2}\right).$$

A direct computation gives that the right hand side of the above equation attains its minimum at

$$\lambda=\frac{\left(\frac{\pi}{2}\right)^{\frac{1}{2p}}p^{\frac{1}{2}}}{1+\left(\frac{\pi}{2}\right)^{\frac{1}{2p}}p^{\frac{1}{2}}}.$$

The result now follows by Equation (10). ■

Proof of Theorem 11 Let $\alpha=1+\left(\frac{\pi}{2}\right)^{\frac{1}{2p}}\sqrt{p}$. As in the proof of Theorem 2 we proceed using duality and apply Lemma 12 to the set of transformations $\mathcal{M}\mathbf{x}=\left\{M\mathbf{x}:M\in\mathcal{M}\right\}$,

$$\begin{aligned}
\mathcal{R}_{\mathcal{M}_q}(\mathbf{x}) &\le \frac{2}{n}\mathbb{E}\left\|\sum_{i=1}^n\varepsilon_i x_i\right\|_{\mathcal{M}_q^*}=\frac{2}{n}\mathbb{E}\left[\left(\sum_{M\in\mathcal{M}}\|M\mathbf{x}\varepsilon\|^p\right)^{1/p}\right]\\
&\le \frac{2\alpha}{n}\left(\sum_{M\in\mathcal{M}}\|M\mathbf{x}\|_{HS}^p\right)^{1/p}=\frac{2\alpha}{n}\sqrt{\left(\sum_{M\in\mathcal{M}}\left(\sum_{i=1}^n\|Mx_i\|^2\right)^{p/2}\right)^{2/p}}\\
&\le \frac{2\alpha}{n}\sqrt{\sum_{i=1}^n\left(\sum_{M\in\mathcal{M}}\left(\|Mx_i\|^2\right)^{p/2}\right)^{2/p}}=\frac{2\alpha}{n}\sqrt{\sum_{i=1}^n\|x_i\|^2_{\mathcal{M}_{q^*}}},
\end{aligned}$$

where the last inequality is just the triangle inequality in $\ell_{p/2}$. ■

One can verify that the leading constant in our bound is smaller than the one in Cortes et al. (2010) for $p>12$. Note that the bound in Theorem 11 diverges for q going to 1 since in this case p grows to infinity.

We conclude this section with a result, which shows that the bound in Theorem 2 has a stability property in the following sense: If $\mathcal{M}$ is finite, then we can give a bound on the Rademacher complexity of the unit ball in $\ell_q(\mathcal{M})$ which converges to the bound in Theorem 2 as $q\to1$, regardless of the size of $\mathcal{M}$. Only the rate of convergence is dimension dependent.

Theorem 13 *Under the conditions of Theorem 11*

$$\mathcal{R}_{\mathcal{M}_q}(\mathbf{x})\le\frac{4\left|\mathcal{M}\right|^{1/p}}{n}\sqrt{\sup_{M\in\mathcal{M}}\sum_i\|Mx_i\|^2}\left(2+\sqrt{\ln\sum_M\frac{\sum_i\|Mx_i\|^2}{\sup_{N\in\mathcal{M}}\sum_i\|Nx_i\|^2}}\right).$$

So, as q goes to 1, $p\to\infty$ and we recover the bound in Theorem 2 up to a small multiplicative constant. The key step in the proof of Theorem 13 is the following

Lemma 14 *Let $\mathcal{M}$ be a finite set of linear transformations $M : \mathbb{R}^n \to H$ and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ a vector of orthonormal random variables with values in $[-1,1]$. Then*

$$\mathbb{E}\left[\left(\sum_M \|M\varepsilon\|^p\right)^{1/p}\right] \le 2\left|\mathcal{M}\right|^{1/p} \sup_{M\in\mathcal{M}} \|M\|_{HS}\left(2+\sqrt{\ln\frac{\sum_M \|M\|_{HS}^2}{\sup_{N\in\mathcal{M}}\|N\|_{HS}^2}}\right).$$

Proof If $t \ge 0$ and $\sum_M \|M\varepsilon\|^p > t^p$, then there must exist some $M \in \mathcal{M}$ such that $\|M\varepsilon\|^p > t^p/\left|\mathcal{M}\right|$, which in turn implies that $\|M\varepsilon\| > t/\left|\mathcal{M}\right|^{1/p}$. It then follows from a union bound that

$$\Pr\left\{\sum_M \|M\varepsilon\|^p > t^p\right\} \le \sum_M \Pr\left\{\|M\varepsilon\| > t/\left|\mathcal{M}\right|^{1/p}\right\} \le \exp\left(\frac{-t^2}{4\left|\mathcal{M}\right|^{2/p}\|M\|_{HS}^2}\right),$$

where we used the subgaussian concentration inequality Lemma 9-(ii) with $r = 2$. Using integration by parts we have with $\delta \ge 0$ that

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_M \|M\varepsilon\|^p\right)^{1/p}\right] &\le \delta + \int_\delta^\infty \Pr\left\{\sum_M \|M\varepsilon\|^p > t^p\right\}dt \\
&\le \delta + 2\sum_M \int_\delta^\infty \exp\left(\frac{-t^2}{4\left|\mathcal{M}\right|^{2/p}\|M\|_{HS}^2}\right)dt \\
&\le \delta + \frac{4\left|\mathcal{M}\right|^{2/p}}{\delta}\sum_M \|M\|_{HS}^2 \exp\left(\frac{-\delta^2}{4\left|\mathcal{M}\right|^{2/p}\|M\|_{HS}^2}\right) \\
&\le \delta + \frac{4\left|\mathcal{M}\right|^{2/p}}{\delta}\left(\sum_M \|M\|_{HS}^2\right)\exp\left(\frac{-\delta^2}{4\left|\mathcal{M}\right|^{2/p}\sup_{M\in\mathcal{M}}\|M\|_{HS}^2}\right),
\end{aligned}$$

where the third inequality follows from Lemma 5 and the fourth from Hölder's inequality. We now substitute

$$\delta = 2\left|\mathcal{M}\right|^{1/p}\sup_{M\in\mathcal{M}}\|M\|_{HS}\sqrt{\ln\frac{e\sum_M \|M\|_{HS}^2}{\sup_{N\in\mathcal{M}}\|N\|_{HS}^2}}$$

and use $1 + 1/e \le 2$ to arrive at the conclusion. ■

Proof of Theorem 13 Apply Lemma 12 to the set of transformations $\mathcal{M}\mathbf{x} = \left\{M\mathbf{x} : M \in \mathcal{M}\right\}$. This gives

$$\mathbb{E}\left[\left(\sum_M \|M\mathbf{x}\varepsilon\|^p\right)^{1/p}\right] \le 2\left|\mathcal{M}\right|^{1/p}\sup_{M\in\mathcal{M}}\|M\mathbf{x}\|_{HS}\left(2+\sqrt{\ln\frac{\sum_M \|M\mathbf{x}\|_{HS}^2}{\sup_{N\in\mathcal{M}}\|N\mathbf{x}\|_{HS}^2}}\right).$$

We now proceed as in the proof of Theorem 11 to obtain the result. ■

5. Conclusion and Future Work

We have presented a bound on the Rademacher average for linear function classes described by infimum convolution norms which are associated with a class of bounded linear operators on a Hilbert space. We highlighted the generality of the approach and its dimension independent features.

When the bound is applied to specific cases (ℓ_2, ℓ_1, mixed ℓ_1/ℓ_2 norms) it recovers existing bounds (up to small changes in the constants). The bound is however more general and allows for the possibility to remove the "$\log d$" factor which appears in previous bounds. Specifically, we have shown that the bound can be applied in infinite dimensional settings, provided that the moment condition (3) is satisfied. We have also applied the bound to multiple kernel learning. While in the standard case the bound is only slightly worse in the constants, the bound is potentially smaller and applies to the more general case in which there is a countable set of kernels, provided the expectation of the sum of the kernels is bounded.

An interesting question is whether the bound presented is tight. As noted in Cortes et al. (2010) the "$\log d$" is unavoidable in the case of the Lasso. This result immediately implies that our bound is also tight, since we may choose $R^2 = d$ in Equation (3).

A potential future direction of research is the application of our results in the context of sparsity oracle inequalities. In particular, it would be interesting to modify the analysis in Lounici et al. (2011), in order to derive dimension independent bounds. Another interesting scenario is the combination of our analysis with metric entropy.

Acknowledgments

We wish to thank Andreas Argyriou, Bharath Sriperumbudur, Alexandre Tsybakov and Sara van de Geer, for useful comments. A valuable contribution was made by one of the anonymous referees. Part of this work was supported by EPSRC Grant EP/H027203/1 and Royal Society International Joint Project Grant 2012/R2.

References

F.R. Bach, G.R.G. Lanckriet and M.I. Jordan. Multiple kernels learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)*, pages 6–13, 2004.

L. Baldassarre, J. Morales, A. Argyriou, M. Pontil. A general framework for structured sparsity via proximal optimization. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, forthcoming.

R. Baraniuk, V. Cevher, M.F. Duarte, C. Hedge. Model based compressed sensing. *IEEE Trans. on Information Theory*, 56:1982–2001, 2010.

P.L. Bartlett and S. Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3:463–482, 2002.

C. Cortes, M. Mohri, A. Rostamizadeh. Generalization bounds for learning kernels. In *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML 2010*, pages 247–254, 2010.

C. De Mol, E. De Vito, L. Rosasco. Elastic-net regularization in learning theory. *Journal of Complexity*, 25(2):201–230, 2009.

W. Hoeffding, Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association*, 58:13–30, 1963.

J. Huang, T. Zhang, D. Metaxa. Learning with structured sparsity. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML 2009)*, pages 417–424, 2009.

L. Jacob, G. Obozinski, J.-P. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML 2009)*, pages 433–440, 2009.

R. Jenatton, J.-Y. Audibert, F.R. Bach. Structured variable selection with sparsity inducing norms. *Journal of Machine Learning Research*, 12:2777-2824, 2011.

S.M. Kakade, S. Shalev-Shwartz, A. Tewari. Regularization techniques for learning with matrices. *ArXiv preprint arXiv0910.0610*, 2010.

M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien. ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

V. Koltchinskii and D. Panchenko, Empirical margin distributions and bounding the generalization error of combined classifiers, *Annals of Statistics*, 30(1):1–50, 2002.

G.R.G. Lanckriet, N. Cristianini, P.L. Bartlett, L. El Ghaoui, M.I. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

M. Ledoux, M. Talagrand. *Probability in Banach Spaces*, Springer, 1991.

K. Lounici, M. Pontil, A.B. Tsybakov and S. van de Geer. Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics*, 39(4):2164–2204, 2011.

C. McDiarmid. Concentration. In *Probabilistic Methods of Algorithmic Discrete Mathematics*", pages 195–248, Springer, 1998.

L. Meier, S.A. van de Geer, and P. Bühlmann. High-dimensional additive modeling. *Annals of Statistics*, 37(6B):3779–3821, 2009.

R. Meir and T. Zhang. Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4:839–860, 2003.

C.A. Micchelli, J.M. Morales, M. Pontil. A family of penalty functions for structured sparsity. In *Advances in Neural Information Processing Systems 23*, pages 1612–1623, 2010.

C.A. Micchelli, J.M. Morales, M. Pontil. Regularizers for structured sparsity. *Advances in Computational Mathematics*, forthcoming.

C.A. Micchelli and M. Pontil. Feature space perspectives for learning the kernel. *Machine Learning*, 66:297–319, 2007.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

T. Shimamura, S. Imoto, R. Yamaguchi and S. Miyano. Weighted Lasso in graphical Gaussian modeling for large gene network estimation based on microarray data. *Genome Informatics*, 19:142–153, 2007.

Y. Ying and C. Campbell. Generalization bounds for learning the kernel problem. In *Proceedings of the 23rd Conference on Learning Theory (COLT 2009)*, pages 407–416, 2009.

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(1):49–67, 2006.

P. Zhao and G. Rocha and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, 37(6A):3468–3497, 2009.

A Case Study on Meta-Generalising: A Gaussian Processes Approach

Grigorios Skolidis G.SKOLIDIS@SMS.ED.AC.UK
Guido Sanguinetti GSANGUIN@INF.ED.AC.UK
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK

Editors: Sören Sonnenburg, Francis Bach and Cheng Soon Ong

Abstract

We propose a novel model for meta-generalisation, that is, performing prediction on novel tasks based on information from multiple different but related tasks. The model is based on two coupled Gaussian processes with structured covariance function; one model performs predictions by learning a constrained covariance function encapsulating the relations between the various training tasks, while the second model determines the similarity of new tasks to previously seen tasks. We demonstrate empirically on several real and synthetic data sets both the strengths of the approach and its limitations due to the distributional assumptions underpinning it.

Keywords: transfer learning, meta-generalising, multi-task learning, Gaussian processes, mixture of experts

1. Introduction

The central problem of supervised learning is *generalisation*, learning input/ output relations from training data that, when applied to unseen test data, will give good performance (in terms of an appropriate loss function). A common assumption underlying many supervised learning algorithms is that the training and testing data distribution are the same, which allows them to make predictions of future instances of the problem at hand. On the other hand, in the complex world that we live in we are usually faced with unseen but similar problems, situations which human intelligence handles by adaptively taking decisions on the new tasks using knowledge from similar tasks. In this direction, *Transfer learning* (TL) has emerged as a framework to handle situations where there are multiple but related problems to be solved. The term TL is used here in its broader sense, to cover more specific areas of research such as domain adaptation, co-variate shift, sample selection bias, self-taught learning, and multi-task learning. One of the main differences between these subfields of TL lies in the availability of outputs (labels) for input data in the various tasks, no matter if it is a regression or classification problem (Arnold et al., 2007). For example, the situation where labels are available for all tasks is tackled by multi-task learning, which synergistically solves the learning problem in all tasks simultaneously (Caruana, 1997; Bakker and Heskes, 2003; Ando and Zhang, 2005). Domain adaptation (Daumé III and Marcu, 2006; Daumé, 2007; Crammer et al., 2008; Mansour et al., 2009; Pan et al., 2009), co-variate shift (Sugiyama et al., 2007; Storkey and Sugiyama, 2007; Bickel et al., 2009), and sample selection bias (Huang et al., 2007) are settings appropriate for problems where labels are only available for a task that is similar to the task that we wish to make predictions in (target task). Contrary to domain adaptation, and sample selection bias, self-taught learning (Raina et al., 2007) is a setting where labeled data are available for the target task, but the learning algorithm wishes to also use unlabelled data from a source task to improve performance. In its own right, self-taught learning is distinguishable from semi-supervised learning (Chapelle et al., 2006), where labelled and unlabelled data are assumed to come from the same task. The purpose of all these TL approaches is to enhance the generalisation power of a specific algorithm by leveraging related (but different) knowledge from multiple tasks. In particular, it is generally assumed that at least the input data for the target task will be available *during the learning*, so that a measure of similarity between the training and target tasks can be estimated.

The question that we wish to raise in this work is whether the notion of generalisation can be extended to the level of tasks as a form of *meta-generalisation*. Meta-generalisation is a concept introduced in Baxter (2000), where the author argues whether a transfer learning algorithm can generalise well on totally *unseen* tasks after seeing sufficiently many *source* (or training) tasks. We emphasize that this is much more than a theoretically interesting question. Our motivating example is a strongly applied one: we wish to create an automated diagnosis tool that can accommodate variability among patients, so that, once trained on a sufficient number of patients, it can generalise to new patients. In his work Baxter (2000) derives bounds on the generalisation error of this problem in terms of a generalised VC-dimension parameter, as well as comments that the number of source tasks and examples per task required to ensure good performance on novel tasks has to be sufficiently large. While Baxter (2000) derives an algorithm to select a subset of features to perform multi-task learning based on Neural Networks (NN), his work is more on the theoretical side as no experimental results are presented. Besides that, the model proposed in this work needs to be retrained in case a new target task arrives in order to learn a small number of task dependent parameters.

One way to approach meta-generalising is through domain adaptation, by training a model on the data of the source and the target set of tasks (Ben-David et al., 2007). This type of approach, as well as the model proposed in Baxter (2000), are essentially trained in a transductive way, as the algorithm is able to make predictions only on the test tasks that is trained on, or needs to be retrained in case a new task arrives. Obviously, the performance and the success of domain adaptation algorithms depends strongly on certain assumptions, with most important the similarity between the target and the source distribution (Ben-David et al., 2010). Clearly, if these assumptions are violated then the success of these algorithms is doubtful.

The problem of sampling the space of tasks to make predictions on totally unseen tasks in the inductive setting, which is the exact analog of generalising in the level of tasks, to the best of our knowledge has not been specifically addressed. As we mentioned before, TL is separated into different sub-categories based on the level of supervision on the target task. Hence, multi-task learning can be seen as an *Inductive* TL algorithm since input data and labels are available for all the tasks that we wish to make predictions. On the other end, settings like to Domain adaptation, Covariate shift or Sample selection bias, can be viewed as a form of *Transductive* TL since the algorithm can exploit only the input distribution of the target task they want to make predictions (Arnold et al., 2007). On this basis, meta-generalising can be considered as a form of *Unsupervised* TL, since the learning algorithm does not have any exploitable information about the target tasks during training . Note, that this classification of TL algorithms is different from the one employed in Pan and Yang (2010), where unsupervised TL encapsulates problems like dimensionality reduction, density estimation, or clustering but in situations where multiple tasks are involved, but is in agreement with the taxonomy of TL algorithms introduced in Arnold et al. (2007).

In this paper we investigate the use of coupled Gaussian process models to address this problem. The model uses a multi-class Gaussian process for assigning probabilistically unseen tasks to source tasks (determining task responsibilities), and then uses a multi-task Gaussian process (Bonilla et al., 2008) to perform prediction in individual tasks. Extensive testing on real and simulated data shows the promise of the model, as well as giving insight on the underlying assumptions.

The rest of the paper is organised as follows: in Section 2 we formally define the meta-generalising problem, emphasizing the main assumptions and highlighting the important special case of *fully observed tasks*. In Section 3 and 4 we present our model and the inference methodology used. We present our empirical results in Section 5, and we finish in Section 6 by discussing the merits of our model in the context of the wider literature in transfer learning and meta-generalisation.

2. Meta-generalising

In this section, we formally state the problem of meta-generalising, while we introduce the notation that will be used throughout this paper unless specified otherwise. For simplicity, we concentrate on binary classification problems within each task, while we note that the same formalism applies to regression and multi-class classification problems.

In a meta-generalising scenario the learner is provided with a set of source tasks $\mathcal{T}_S = \{\mathcal{T}_1^s, \ldots, \mathcal{T}_M^s\}$ which are used for training the model; testing is then performed on a set of target tasks $\mathcal{T}_T = \{\mathcal{T}_1^t, \ldots, \mathcal{T}_H^t\}$. Each of the M source tasks will contain a training set of input/ output pairs (x, y), while data from any of the H target tasks are hidden. For later convenience, we will define the whole training set across tasks as a set of triples $T^s = \{x_i^s, y_i^{st}, y_i^{sx}\}_{i=1}^{N^s}$, where $x_i^s \in \mathbb{R}^d$ is the input feature vector, $y_i^{sx} \in \{-1, +1\}$ are the class labels, and $y_i^{st} \in \{1, \ldots, M\}$ is the source task label indicating to which task the input/ output pair pertains, and $N^s = \sum_{j=1}^M n_j^s$ is the total number of training pairs where n_j^s is number of data points from the j^{th} source task. Moreover, we will write $X_j^s = \{x_{ij}^s\}_{i=1}^{n_j^s}$ to denote the total item set of the j^{th} source task, while $\mathbf{y}_j^{sx} = \{y_{ij}^{sx}\}_{i=1}^{n_j^s}$ and $\mathbf{y}_j^{st} = \{y_{ij}^{st}\}_{i=1}^{n_j^s}$ will be used to denote all class and task labels from the j^{th} source task. In the rest of the paper subscript j will be used to refer to tasks, and subscript i to data points.

Each of the H target tasks $\mathcal{T}_j^t$ will consist of a set $X_j^t = \{x_{ij}^t\}_{i=1}^{n_j^t}$ of input points, where n_j^t is number of data points from the j^{th} target task and both types of labels are missing. Likewise, the total number of test points will be denoted by $N^t = \sum_{j=1}^M n_j^t$. For reasons that will become clear later on, it is further assumed that for each target task data point x_j^t there is information that it comes from the j^{th} target task, but there is no knowledge with which of the source tasks is more similar. Note that each source task training input x_i^s is assigned two types of labels. This implies supervision in both the levels of the tasks and the data, through y^{st} and y^{sx} respectively; task labels y^{st} indicate from which of the source task a specific data point comes from, as a form of *meta-level information*, and class labels y^{sx} indicate to which class inside the task the data point belongs to, as a form of *inter-task information*.

Meta-generalisation, as all machine learning methods, relies on certain assumptions. We concentrate on two basic assumptions; the first one is the *similarity of the distribution* of the target task with at least one of the source tasks, while the second one is the agreement between the labels of the distributions termed as *low-error joint prediction* (Ben-David et al., 2010). Differently from Ben-David et al. (2010), we will define the *low-error joint prediction* between a source and a target task as the error λ_e between their predictive functions f_s and f_t respectively, evaluated at the union of the source and the target sets $X = X^s \cup X^t$, with $N = N^s + N^t$. Hence, the error λ_e will be given by,

$$\lambda_e = \sum_{i=1}^{N} |f_t(x_i) - f_s(x_i)|,$$

where $x_i \in X$. Intuitively, if the error λ_e is large then there is a disagreement between the labels of the source and target tasks distribution. Also note that, in a multi-task scenario the parameter λ_e can be computed by training two separate models under the same learning framework (e.g., NN, GPs, etc) since labeled data are available for both the source and target task. Thus, the predictive functions of the source and target task can be estimated separately and λ_e can also be used as an empirical measure of the relatedness of the two tasks. Conversely, in the scenarios of meta-generalising and domain adaptation one has to *assume* that the error λ_e will be low, since labels are available only for the source tasks. If one of these assumptions is not valid, then meta-generalisation can not be expected to guarantee success.

We now give a formal definition of meta-generalising.

Definition 1 *Given a set of source tasks $\mathcal{T}_S$ and a set of target tasks $\mathcal{T}_T$, meta-generalising is an inductive inference method that aims at making predictions on the set of target tasks by sampling the space of source tasks .*

We further define two possible scenarios: in the *fully observed tasks* case, we assume that the similarity of the distribution assumption is perfectly met, so that the data generating distribution of the target task is the same as that of one of the source tasks (but we do not know which one). This assumption is relaxed in the *partially observed tasks* scenario, where we still assume similarity of the distribution but we do not necessarily have identity.

The meta-generalising setting implies that there is hierarchical structure in the problem. The data of each task are on the base level and the distribution of the tasks is on the meta level. Hence, it is intuitive that mechanisms are required to

1. Model the distribution of the data of each task, and the distribution of the source tasks (correlation between tasks).

2. Infer the level of correlation between the target task and the source tasks.

The first prerequisite leads us to multi-task learning, as many approaches offer mechanisms to model both the data and the task distribution (Bakker and Heskes, 2003; Yu et al., 2005; Ando and Zhang, 2005; Xue et al., 2007; Argyriou et al., 2008; Bonilla et al., 2008; Daumé III, 2009). Following the multi-task route, informally speaking, the second prerequisite can be translated as the problem of which of the M outputs of the multi-task classifier to select to make predictions for the target task. In some cases, task-descriptor features may be available, giving a direct measure of task similarity. In this work, we are interested in the general case where no reliable task descriptor features are available; we will then learn similarities between tasks through a *distribution matching* pursuit.

Another way of approaching the problem of meta-generalisation is through the framework of *mixtures of experts* (ME) (Jacobs et al., 1991; Waterhouse, 1997), under which a bigger learning problem is broken down to smaller subproblems that are handled by individual experts. The underlying assumption of this framework is that the data are generated by different processes (Waterhouse, 1997, Ch. 2), an assumption that can also be made in the multi-task setting about the data generating mechanism of each task; under the ME framework each expert is used to model the data generating process of each subproblem. These experts are then combined through a gating network that models the responsibilities of the experts on each data partition. Hence, attacking the meta-generalisation problem through the ME framework can be seen as an unsupervised alternative method to that problem, that does not use the information about the origins of each task (the source task labels) but instead allows the algorithm to automatically infer the data partitions and the regions of expertise of each expert. Therefore the ME approach is in direct connection to multi-task learning and meta-generalisation in which cases the experts are equivalent to the tasks, and this framework could be used as a rough lower bound on the performance of a multi-task classifier. Note though that in principle it would be desirable to be able to automatically infer the number of experts as in Rasmussen and Ghahramani (2001) which can be seen as a similar mechanism of finding cluster of tasks, in contrast with the method of ME with GPs in Tresp (2000) where the number of experts had to be known *a priori*.

3. A Model for Meta-generalisation

Having identified the nature of the problem, we now propose a model for meta-generalising. The model builds upon the multi-task learning framework of Bonilla et al. (2008) which is able to capture the dependencies between the data and the tasks. In addition, we employ a classifier over the tasks to learn the task labels (from which task each data point comes from). Both of those two learning mechanisms, multi-task setting and classification of the tasks, are modeled by Gaussian Processes (GPs), which are coupled by sharing a common hyper-prior. In the rest of this section, we first give a short introduction to GPs and we review multi-task learning with GPs of Bonilla et al. (2008), we then present the model for meta-generalising, and finally we describe how to make predictions on new tasks.

3.1 Multi-task Learning with Gaussian Processes

Gaussian processes (Rasmussen and Williams, 2005) provide a flexible modelling framework for supervised learning which has become increasingly popular in recent years. A Gaussian Process is a probability distribution over functions f, where the joint distribution of function evaluations over a finite set of inputs is a multivariate Gaussian distribution. At core of the GP prediction is the *covariance function* or *kernel*, parameterised by θ^x, that models the output covariance at different pairs of input points, and in essence acts as a measure of similarity between different input locations. In order for a covariance function to be valid it has to be positive semidefinite, and has to satisfy Mercer's theorem (Rasmussen and Williams, 2005).

In a multi-task scenario the interest lies in learning M related functions $\mathbf{f}_j,\ j = 1,\ldots,M$, from training data $x_{ij},\ y_{ij},\quad i = 1,\ldots,n_j$, with $x \in\ \mathbb{R}^d$, and $n_1 + \ldots + n_M = N$. In the following of this section, data points from task j will be denoted by $X_j = [x_{1j},\ldots,x_{n_j j}]$ and $\mathbf{X} = [X_1,\ldots,X_M]$ will be used to denote the set of all data points. Focussing on a regression problem for simplicity, the noise model will be given by

$$y_{ij} = f_j(x_{ij}) + \varepsilon_j,\ \text{with}\ \varepsilon_j \sim \mathcal{N}(0,\sigma_j^2), \tag{1}$$

where y_{ij} (x_{ij}) denotes the i^{th} output (input) of the j^{th} task. We note that each input point has M function values associated with it (one per task); this *complete set of responses* will rarely be observed in practice, but function values corresponding to unobserved values can easily be marginalised using the consistency of GPs

The multi-task model of Bonilla et al. (2008), which has been known in the geo-statistics community as the "*Intrinsic Model of Coregionalization*" (IMC) (Cressie, 1993), can be elegantly recovered from the theory of matrix variate distributions (Gupta and Nagar, 2000). Define the vector $\mathbf{f}$ by stacking the columns of $\mathbf{F} = [\mathbf{f}_1 \ldots \mathbf{f}_M]$ into a single vector, $\mathbf{f} = \text{vec}(\mathbf{F})$, where $\mathbf{f}_j \in \mathbb{R}^{N\times 1}$ is the column vector of all latent functions evaluations of task j. Then the *probability density function* of matrix $\mathbf{F}$ will be given by:

$$(2\pi)^{-\frac{1}{2}NM}|\mathbf{K}^t|^{-\frac{1}{2}N}|\mathbf{K}^x|^{-\frac{1}{2}M}\exp\left\{-\frac{1}{2}\text{trace}\left(\left(\mathbf{K}^t\right)^{-1}\mathbf{F}\left(\mathbf{K}^x\right)^{-1}\mathbf{F}^T\right)\right\}, \qquad (2)$$

where $\mathbf{K}^t \in \mathbb{R}^{M\times M}$ and $\mathbf{K}^x \in \mathbb{R}^{N\times N}$ (Gupta and Nagar, 2000). This configuration implies that the matrix $\mathbf{K}^t$ models the correlations between the vectors $\mathbf{f}_j$, that is, the tasks in the multi-task view, and $\mathbf{K}^x$ models the correlations between each element of vectors $\mathbf{f}_j$. In the GP framework, this correlation between function evaluations at different input points is captured by the covariance function. Then, by using some matrix algebra involving the vec and Kronecker operator, Equation (2) can be written in the form Bonilla et al. (2008) proposed,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{GP}(0, \mathbf{K}^t \otimes \mathbf{K}^x).$$

Employing this type of prior for the latent functions $\mathbf{f}$ the noise model for the regression problem stated in equation (1) becomes, $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \mathbf{D}\otimes\mathbf{I})$, where $\mathbf{D} \in \mathbf{R}^{M\times M}$ is diagonal with $D_{jj} = \sigma_j^2$ and $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix.

The key element of this formulation is the task covariance matrix $\mathbf{K}^t$ which reflects the task correlations. For example, if $\mathbf{K}^t$ was fixed to the identity matrix, then all tasks would be independent but they would still share the same hyperparameters of the covariance function. Of course, one of the main goals of multi-task learning is to learn these task dependencies. Bonilla et al. (2008) approached this problem by parameterizing the task covariance matrix, with parameters θ^t, always retaining positive definite restrictions, and treating these parameters as hyperparameters to be learned. Positive definite guarantees were achieved, by parameterizing a lower triangular matrix L to employ the Cholesky factorization $\mathbf{K}^t = LL^T$. Most importantly, parameters related to the data covariance function or the task covariance matrix can be learned in the standard GP formulation, by maximizing the marginal likelihood $p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X})\text{d}\mathbf{f}$.

3.2 Model

In this section we describe the Coupled Multi-Task Multi-Class (CMTMC) model we propose for meta-generalisation. The objectives of the model are first to model the dependencies between the tasks, and second to assign unseen tasks to source tasks by finding task similarities. The first objective is met through the Multi-task part of the model, while the second is achieved through the Multi-class classifier. Figure 1 shows the graphical model of the CMTMC classifier. In this subsection we use x, y^t, and y^x to refer to x^s, y^{st}, and y^{sx} to keep the notation light, since in the learning phase only source tasks are involved. Therefore, notation introduced in Section 3.1 applies here. Moreover, from Section 2 we have that $y^t \in \{1,\ldots,M\}$ and $y^x \in \{-1,+1\}$ as the task and class labels respectively. Since both class and task prediction are effectively classification models, we choose the probit and multinomial probit models as noise models respectively. Following Albert

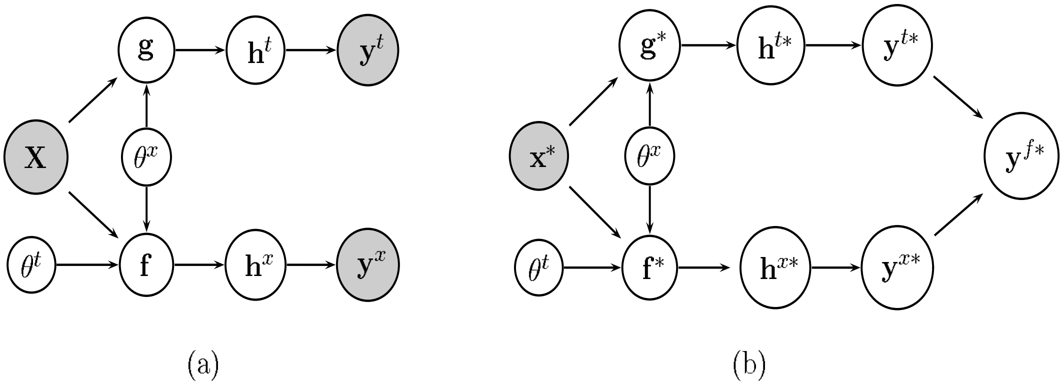


Figure 1: Coupled Multi-Task Multi-Class (CMTMC) model. Variables **f** and **g** are the two sets of GPs for the multi-task and multi-class classifiers respectively, whereas variables $\mathbf{h}^x$ and $\mathbf{h}^t$ denote the auxiliary variables of the two classifiers; (a) graphical representation of the training phase, (b) graphical representation of Meta-generalising.

and Chib (1993), we define two sets of auxiliary variables $\mathbf{h}^t = \text{vec}(\mathbf{H}^t)$, and $\mathbf{h}^x = \text{vec}(\mathbf{H}^x)$, which as shown later on enables the multinomial and the binary probit model respectively. For later convenience, we will be using $\mathbf{h}^t_j$ and $\mathbf{h}^t_n$ to denote the j^{th} column and n^{th} row of matrix $\mathbf{H}^t$.

Figure 1 shows that there are two directed channels of variables. The upper channel, with variables $C^t = \{\mathbf{g}, \mathbf{h}^t, \mathbf{y}^t\}$, is responsible for learning the task labels, thus from which task each data point comes from, while the lower channel, with variables $C^x = \{\mathbf{f}, \mathbf{h}^x, \mathbf{y}^x\}$, learns to classify the data points inside every task and to find task correlations, through the standard multi-task classifier.

Thus, there are two sets of Gaussian Processes. The first one is responsible for the classification over the tasks $\mathbf{g}|\mathbf{X}, \theta^x \sim \mathcal{GP}(0, \mathbf{I} \otimes \mathbf{K}^x)$, where $\mathbf{g} = \text{vec}(\mathbf{G})$, $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_M]$, and $\mathbf{g}_j \in \mathbb{R}^{N \times 1}$. The second one is responsible for the multi-task classification $\mathbf{f}|\mathbf{X}, \theta^x, \theta^t \sim \mathcal{GP}(0, \mathbf{K}^t \otimes \mathbf{K}^x)$, where as stated before variables θ^x and θ^t are used to denote the hyperparameters of the data covariance function and task matrix respectively. As in the multi-class case we will have that $\mathbf{f} = \text{vec}(\mathbf{F})$, where $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_M]$ and $\mathbf{f}_j \in \mathbb{R}^{N \times 1}$. In the rest of the paper we will write $\mathbf{K}^x$ to denote the covariance matrix between all data points $\mathbf{X}$, unless specified otherwise. Moreover, $\mathbf{I}$ and $\mathbf{K}^t$ will be $M \times M$, where the identity matrix in the multi-class case implies independence between the classes, thus $\mathbf{g}_j|\mathbf{X}, \theta^x \sim \mathcal{GP}(0, \mathbf{K}^x)$. The key objective is to learn M functions $\mathbf{g}_j$ for the multi-class classifier and M related functions $\mathbf{f}_j$ for the multi-task classifier.

Note that the data covariance matrix $\mathbf{K}^x$ is shared by both sets of processes **g** and **f**. This is graphically illustrated by the fact that the node of hyperparameters θ^x is connected to both latent functions; thus, the multi-class and the multi-task classifier share the same hyperparameter space for θ^x. The multi-class classifier is restricted to have the same covariance function across the classes in contrast with the standard model for multi-class classification with GPs, which in principle allows you to use different covariance functions across classes. In fact, the CMTMC model could be decoupled into two separate classifiers with different sets of hyperparameters θ^x between the two processes **f** and **g**. Seemingly, this decoupling would result in a more flexible model, but preliminary experiments with both models, the CMTMC and the decoupled model, has shown that this restriction does not affect the performance. In contrast, it reduces dramatically the computational cost since the hyperparameters of the data covariance function need to be estimated only one time.

The probit model is enabled in both channels by a standardized normal noise model over the auxiliary variables, $h^t_{ij}|g_{ij} \sim \mathcal{N}(g_{ij}, 1)$, and $h^x_i|f_i \sim \mathcal{N}(f_i, 1)$ (Albert and Chib, 1993; Csató et al., 2000; Girolami and Rogers, 2006; Skolidis and Sanguinetti, 2011). The relationship between outputs y^t and y^x and auxiliary variables h^t, and h^x is deterministic and will be given by:

$$y^t_i = j \ \text{ if } \ h^t_{ji} = \max_{1\leq k\leq M} \{h^t_{ki}\},$$

$$p(y^x_i|h^x_i) = \begin{cases} \delta(h^x_i)\delta(y^x_i) & \text{if } \ y^x_i = +1 \\ \delta(-h^x_i)\delta(-y^x_i) & \text{if } \ y^x_i = -1 \end{cases},$$

where δ is one if its argument is positive and zero otherwise, which completes the specification of the model.

3.2.1 INFERENCE

Classification problems imply non-Gaussian noise models, which make inference intractable. To address this intractability, we adopt a variational approximate treatment to the problem, as it is computationally more efficient than sampling-based methods while retaining a reasonable accuracy in empirically approximating posterior marginals.[1] For a comprehensive comparison between these approximations for GP multi-class classification, and on the multinomial probit model the interested reader in referred to Girolami and Rogers (2006). The dependencies of the random variables $\Theta = \{\mathbf{g}, \mathbf{h}^t, \mathbf{f}, \mathbf{h}^x\}$ are depicted graphically in Figure 1.a and are summarized in the joint likelihood of the CMTMC model as:

$$p(\mathbf{y}^t, \mathbf{y}^x, \Theta|\theta^x, \theta^t, \mathbf{X}) = p(\mathbf{y}^t|\mathbf{h}^t)p(\mathbf{h}^t|\mathbf{g})p(\mathbf{g}|\theta^x, \mathbf{X})p(\mathbf{y}^x|\mathbf{h}^x)p(\mathbf{h}^x|\mathbf{f})p(\mathbf{f}|\theta^x, \theta^t, \mathbf{X}).$$

Variational methods approach this problem by approximating the joint posterior of the latent variables Θ within a family of tractable distributions; in our case, we will approximate the joint posterior as a factored distribution $p(\Theta|\mathbf{y}^t, \mathbf{y}^x, \mathbf{X}, \theta^t, \theta^x) \approx Q(\Theta) = \prod_{i=1} Q(\Theta_i) = Q(\mathbf{g})Q(\mathbf{h}^t)Q(\mathbf{f})Q(\mathbf{h}^x)$. Minimizing the Kullback-Leibler divergence between the approximating and the true distribution is equivalent to maximizing the following lower bound on the marginal likelihood

$$\log p(\mathbf{y}^t, \mathbf{y}^x|\mathbf{X}, \theta^x, \theta^t) \geq \int Q(\Theta) \log \frac{p(\mathbf{y}^t, \mathbf{y}^x, \Theta|\mathbf{X}, \theta^x, \theta^t)}{Q(\Theta)} \mathrm{d}\Theta, \tag{3}$$

which is found by applying Jensen's inequality (MacKay, 2003). Standard results show that the distributions that maximize the lower bound are given by

$$Q(\Theta_i) = \frac{\exp(\mathbb{E}_{Q(\Theta\setminus\Theta_i)}\{\log\, p(\mathbf{y}^t, \mathbf{y}^x, \Theta|\mathbf{X}, \theta^t, \theta^x)\})}{\int \exp(\mathbb{E}_{Q(\Theta\setminus\Theta_i)}\{\log\, p(\mathbf{y}^t, \mathbf{y}^x, \Theta|\mathbf{X}, \theta^t, \theta^x)\})\mathrm{d}\Theta_i}$$

where $Q(\Theta \setminus \Theta_i)$ denotes the factorized distribution with the i^{th} component removed. Inference and learning are performed in a variational EM algorithm: the E-step computes the variational posteriors on the variables Θ, and the M-step optimizes the hyperparameters θ^t, θ^x given the expectations

[1] Another setting for approximate inference producing comparable results with the Variational approach that could have been employed is the EP approximation (Opper and Winther, 2000; Minka, 2001; Rasmussen and Williams, 2005); this has also been extended to the multi-class classification scenario in Girolami and Zhong (2007).

computed in the previous step. At each (E or M) iteration the variational lower bound, $\mathcal{L}(Q)$ (given in Appendix B Equation (15)), provably increases (or at worst remains unchanged), and these two steps are repeated until convergence.[2] We now briefly summarize the calculations needed to perform the E and M steps. The pseudo-algorithm of the training of the CMTMC model is given in Algorithm 3.2.1. We omit any details and emphasize only the occurrence of the special form covariance function we employ; fuller details can be found in Appendices A, and B.

E-step. The approximate posteriors for the multi-class classifier will be given by,

$$Q(\mathbf{g}) = \prod_{j=1}^{M} \mathcal{N}_{\mathbf{g}_j}(\tilde{\mathbf{g}}_j, \mathbf{\Sigma}^g), \tag{4}$$

$$Q(\mathbf{h}^t) = \prod_{n=1}^{N} \mathcal{N}_{h^t_n}^{y^t_n}(\tilde{\mathbf{g}}_n, \mathbf{I}), \tag{5}$$

where $\mathbf{\Sigma}^g = \left(\mathbf{I} + (\mathbf{K}^x)^{-1}\right)^{-1} = \mathbf{K}^x\,(\mathbf{I} + \mathbf{K}^x)^{-1}$, $\tilde{\mathbf{g}}_j = \mathbf{\Sigma}^g \tilde{\mathbf{h}}^t_j$, *and* $\mathcal{N}_{h^t_n}^{y^t_n}(\tilde{\mathbf{g}}_n, \mathbf{I})$ *denotes an M-dimensional Gaussian distribution truncated such that* j^{th} *dimension has the largest value if* $y^t_n = j$. *In the lower channel, the approximate posteriors for the multi-task classifier will be given by,*

$$Q(\mathbf{f}) = \mathcal{N}_{\mathbf{f}}(\tilde{\mathbf{f}}, \mathbf{\Sigma}^f), \tag{6}$$

$$Q(\mathbf{h}^x) = \prod_{i=1}^{NM} \left(\tilde{f}_i + y^x_i \frac{\mathcal{N}_{\tilde{f}_i}(0,1)}{\Phi(y^x_i \tilde{f}_i)} \right), \tag{7}$$

where $\tilde{\mathbf{f}} = \mathbf{\Sigma}^f \tilde{\mathbf{h}}^x$, *and* $\mathbf{\Sigma}^f = \mathbf{K}^t \otimes \mathbf{K}^x (\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x)^{-1}$ *and the tilde notation in the above random variables denotes posterior expectation, that is,* $\tilde{t}(\alpha) = \mathbb{E}_{Q(\alpha)}\{t(\alpha)\}$; *more details can be found in Appendix A.*

M-step. The M-step optimises the lower bound with respect to the hyperparameters θ^x and θ^t. This is performed by gradient descent; computation of the gradients of the lower bound given in Equation (3) are somewhat intricate and are given in Appendix B.

Algorithm 1 CMTMC model - Training

```
Inputs : $X^s_j$, $\mathbf{y}^{sx}_j$, $\mathbf{y}^{st}_j$ for $j = 1, \dots, M$
Sample parameters $\mathbf{g}, \mathbf{h}^t, \mathbf{f}, \mathbf{h}^x$ from prior
Initialise hyper-parameters $\theta^x, \theta^t$
repeat
    E-step
        Compute $Q(\mathbf{g})$ and $Q(\mathbf{h}^t)$ for MC-classifier, Equations (4),(5)
        Compute $Q(\mathbf{f})$ and $Q(\mathbf{h}^x)$ for MT-classifier, Equations (6),(7)
    M-step
        Optimize hyperparameters $\theta^x, \theta^t$, Equations (16), (16)
    Compute Lower-bound on log-marginal likelihood, Equation (15)
until convergence
```

[2] In practice, estimation of the convergence of the EM algorithm was inferred when the increase between iterations was zero or smaller than a very small constant.

3.3 Prediction on Novel Tasks

While in the previous section we described how to train the model on training data from the source tasks, we now describe how to perform predictions on unseen target tasks. We adopt a mixture of experts type approach; in these networks, multiple outputs are combined and weighted according to the responsibilities they have on a certain prediction task. In a similar manner, the multi-task classifier of the CMTMC model can be seen as a multi-output predictor, and the classifier over the task labels (multi-class) can be used to infer the responsibilities of the outputs of the multi-task classifier, since it produces posterior probabilities of task memberships. Then predictions on novel tasks are computed according to

$$p(y^{f*} = +1|x^*, \mathbf{X}, \mathbf{y}^t, \mathbf{y}^x) = \sum_{j=1}^{M} p(y_j^{x*} = +1|x^*, y^{t*}, \mathbf{X}, \mathbf{y}^x) p(y_j^{t*}|x^*, \mathbf{X}, \mathbf{y}^t), \tag{8}$$

where $p(y_j^{x*} = +1|x^*, y^{t*}, \mathbf{X}, \mathbf{y}^x) = p(y_j^{x*} = +1|x^*, \mathbf{X}, \mathbf{y}^x)$ is the posterior of the j^{th} task belonging to class “+1” from the multi-task classifier, and $p(y_j^{t*}|x^*, \mathbf{X}, \mathbf{y}^t)$ is the posterior of x^* coming from the j^{th} task, or the test point task responsibility from the multi-class classifier. A graphical representation of this process is given in Figure 1.b, where it is shown that nodes y^{t*}, and y^{x*} are combined to give the final predictions y^{f*}.

However, the meta-generalisation scenario presents some additional challenges which are not found in classical mixture of experts models. In many cases, a target task consists of a *batch* of input points, and the simple fact that they all come from the same task contains valuable information about the correlations between the associated outputs. Another closely related issue is that of the correlation between the target task and the source tasks. In many multi-task problems it is a usual phenomenon to observe groups of highly correlated tasks (e.g., Figure 3.b), while other times tasks are correlated but in a more random fashion (e.g., Figure 6.b, 7.b). As we will see in the experimental sections, this can have important consequences in terms of predictive accuracy, and in terms of choosing an appropriate prediction model.

In the following, we present two distinct scenarios for inferring the task responsibilities. Given a target task with n^t data points $\mathbf{x}^{t*} = \{x_1^{t*}, x_2^{t*}, \ldots, x_{n^t}^{t*}\}$, in the first scenario we treat each data point from the target task individually to infer its task responsibilities, which we will refer to as *Point to Point Gating* (P2PGat). This approach neglects the information that all target points come from the same task, and as we will see in the experimental section, is more appropriate when inter-task correlations are weaker. In the second scenario we wish to combine the information from all n^t test points to infer the overall task responsibilities for the target task, which we will refer to as *Batch* predictions.

3.3.1 POINT TO POINT GATING

Given a new input point which lacks both class and target labels, the CMTMC model combines the predictions of a multi-task classifier using task responsibilities obtained from the multi-class classifier channel. Thus, two sets of quantities need to be computed. The first set are the posterior probabilities of the M outputs $p(y_j^{x*} = +1|x^*, \mathbf{X}, \mathbf{y}^x)$ of the multi-task classifier, as

$$\begin{aligned} p(y_j^{x*} = +1|x^*, \mathbf{X}, \mathbf{y}^x) &= \int p(y_j^{x*} = 1|h^{x*}) p(h^{x*}|x^*, \mathbf{X}, \mathbf{y}^x) \mathrm{d}h^{x*}, \\ &\equiv \int_0^{+\infty} \mathcal{N}_{h_j^{x*}}(\lambda_j^*, \upsilon_j^{*2}) \mathrm{d}h^{x*} = \Phi\left(\frac{\lambda_j^*}{\upsilon_j^*}\right) \end{aligned} \tag{9}$$

where we have used that $\upsilon_j^{*2} = 1 + k_{jj}^t k_{x^* x^*}^x - \left(\mathbf{k}_j^t \otimes \mathbf{k}_{\mathbf{x},x_*}^x\right)^T (\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x)^{-1} \left(\mathbf{k}_j^t \otimes \mathbf{k}_{\mathbf{x},x_*}^x\right)$, and $\lambda_j^* = \mathbf{k}_j^t \otimes \mathbf{k}_{\mathbf{x},x^*}^x (\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x)^{-1} \tilde{\mathbf{h}}^x$. Additionally, $\mathbf{k}_j^t$, k_{jj}^t are used to denote the j^{th} column and the jj^{th} element of $\mathbf{K}^t$ respectively, $\mathbf{k}_{\mathbf{x},x_*}^x$ is used to denote the covariance vector between $\mathbf{X}$ and x^*, and Φ is the probit function.

The second set of quantities are the task responsibilities which are computed from Girolami and Rogers (2006)

$$\begin{aligned} p(y^{t*} = k|x^*, \mathbf{X}, \mathbf{y}^t) &= \int p(y^{t*} = k|h^{t*}) p(h^{t*}|x^*, \mathbf{X}, \mathbf{y}^t) \mathrm{d}h^{t*} \\ &\equiv \int_{-\infty}^{+\infty} \mathcal{N}_{h_k^{t*}}(\mu_k^*, \nu_k^*) \prod_{m \neq k} \int_{-\infty}^{h_k^{t*}} \mathcal{N}_{h_m^{t*}}(\mu_m^*, \nu_m^*)\ \mathrm{d}h_m^{t*}\ \mathrm{d}h_k^{t*}, \end{aligned} \tag{10}$$

which can be evaluated using numerical integration as:

$$p(y^{t*} = k|x^*, \mathbf{X}, \mathbf{y}^t) = \mathbb{E}_{p(u)} \left\{ \prod_{j \neq k} \Phi\left(\frac{1}{\nu_j^*} \left[u\nu_k^* + \mu_k^* - \mu_j^* \right] \right) \right\}, \tag{11}$$

where $u \sim \mathcal{N}_u(0,1)$, $\nu_m^* = 1 + k_{x^*,x^*}^x - \mathbf{k}_{\mathbf{x},x^*}^{x^T} (\mathbf{I} + \mathbf{K}^x)^{-1} \mathbf{k}_{\mathbf{x},x^*}^x$, and $\mu_m^* = \mathbf{k}_{\mathbf{x},x^*}^{x^T} (\mathbf{I} + \mathbf{K}^x)^{-1} \tilde{\mathbf{h}}_m^t$.

In the P2PGat scenario, the novel input points are not assumed to share a common task label. Therefore, class prediction is performed straightforwardly on every new input by inserting the posterior probabilities obtained in Equations (9,10) in the gating network given by Equation (8).

3.3.2 BATCH

In a Bayesian way using all test points $\mathbf{x}^{t*}$ to infer the overall task responsibility is performed by replacing the univariate distributions from Equation (10) with the appropriate multivariate. As a result the second integral of Equation (10) becomes the multivariate cumulative distribution function $\int_{-\infty}^{\mathbf{h}_k^{t*}} \mathcal{N}_{\mathbf{h}_m^{t*}}(\mathbf{M}_m^{g*}, \boldsymbol{\Upsilon}^*)\ \mathrm{d}\mathbf{h}_m^{t*}$. Specifically the mean and the variance of the auxiliary variables $\mathbf{h}_m^{t*}$ on the batch of test points $\mathbf{x}^*$ will be given by:

$$\mathbf{M}_m^{g*} = \mathbb{E}[\mathbf{h}_m^{t*}|\mathbf{x}^*] = \mathbf{K}_{\mathbf{x},\mathbf{x}^*}^{x^T} \left(\mathbf{I} + \mathbf{K}_{\mathbf{x},\mathbf{x}}^x\right)^{-1} \tilde{\mathbf{h}}_m^t \tag{12}$$

$$\boldsymbol{\Upsilon}^* = \mathrm{cov}[\mathbf{h}_m^{t*}|\mathbf{x}^*] = \mathbf{I} + \mathbf{K}_{\mathbf{x}^*,\mathbf{x}^*}^x - \mathbf{K}_{\mathbf{x},\mathbf{x}^*}^{x^T} \left(\mathbf{I} + \mathbf{K}_{\mathbf{x},\mathbf{x}}^x\right)^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}^*}^x, \tag{13}$$

where $\mathbf{K}_{\mathbf{x},\mathbf{x}^*}^{x^T}$ is the $N \times n^t$ covariance matrix of all training points $\mathbf{X}$, and all test task data points $\mathbf{x}^*$, and $\mathbf{K}_{\mathbf{x}^*,\mathbf{x}^*}^x$ is the $n^t \times n^t$ full covariance matrix of $\mathbf{x}^*$. Equations (12) and (13), indicate that inferring the tasks responsibilities on a set of points depends not only on the correlations between the test points and the train points but also on the correlations between the test points themselves.

Algorithm 2 CMTMC model - Meta-generalising

```
 1: Inputs : $\mathbf{x}^{t*} = [x_1^{t*},\dots,x_{n^t}^{t*}]$, $Q(\mathbf{g})$, $Q(\mathbf{h}^t)$, $Q(\mathbf{f})$, $Q(\mathbf{h}^x)$, $\mathbf{X}$
 2: for $i = 1$ to $n^t$ do
 3:     for $j = 1$ to $M$ do
 4:         Compute MC posterior probabilities $p(y_{ij}^{t*} = j|x_i^{t*}, \mathbf{X}, \mathbf{y}^t)$, Equation (11)
 5:         Compute MT posterior probabilities $p(y_{ij}^{x*} = +1|x_i^{t*}, \mathbf{X}, \mathbf{y}^x)$, Equation (9)
 6:     end for
 7: end for
 8: P2PGat predictions
 9: for $i = 1$ to $n^t$ do
10:     Compute $p(y^{f*} = +1|x^*, \mathbf{X}, \mathbf{y}^t, \mathbf{y}^x)$, Equation (8) based on steps 4 and 5
11: end for
12: BATCH predictions
13: for $j = 1$ to M do
14:         Compute overall task posterior probabilities $p(\mathbf{y}^{t*} = k|\mathbf{x}^*, \mathbf{X}, \mathbf{y}^t)$, Equation (14)
15: end for
16: for $i = 1$ to $n^t$ do
17:     Compute $p(y^{f*} = +1|x^*, \mathbf{X}, \mathbf{y}^t, \mathbf{y}^x)$, Equation (8) based on steps 5 and 14
18: end for
```

On the other hand, truncated multivariate Gaussian distributions are hard to deal with, and usually approximations are applied (Deak, 1980; Genz, 1992; Gassmann et al., 2002). The dimensions of the multivariate distribution function in the batch prediction problem depend on the number of data points n_* of the target task, which can be several thousands depending the application. To the best of our knowledge no method can tackle very high dimensional c.d.f. , and even approximations can become extremely computationally intensive when n_* is more than a few dozens (these estimations would be carried out within the inner loop of a VBEM algorithm, which would obviously further aggravate the problem). A solution to this problem is to assume that data points from the test task are i.i.d. from the unknown data generating distribution, and approximate it by:

$$p(\mathbf{y}^{t*} = k|\mathbf{x}^*, \mathbf{X}, \mathbf{y}^t) \approx \frac{\prod_{i=1}^{n^*} p(y_i^{t*} = k|x_i^*, \mathbf{X}, \mathbf{y}^t)}{\sum_{m=1}^{M} \prod_{j=1}^{n^*} p(y_j^{t*} = m|x_j^*, \mathbf{X}, \mathbf{y}^t)}, \tag{14}$$

where $p(y_i^{t*} = k|x_i^*, \mathbf{X}, \mathbf{y}^t)$ are the task responsibilities computed individually for each test point. We will adopt this approximation in the experimental section for computational reasons; calculations using the full covariances in Equation (13) are unfeasible with more than 100 points (test or training). While this approximation may appear crude, we experimented extensively in medium-scale problems using a reduced rank approximation for $\boldsymbol{\Upsilon}^*$ (capturing up to 90% of the total variance), but this did not appear to yield significant empirical advantages justifying the substantial computational costs. Note though that although the i.i.d. approximation misses the correlations between the test samples, it still uses information from all test points to produce overall test task class posterior probabilities.

The pseudo-algorithm for the stage of Meta-generalisation for both types of predictions, P2PGat and Batch, is given in algorithm 3.3.1.

4. Experiments

This section aims at providing insights into the workings of our meta-generalising model through empirical evidence. Experiments are presented for both the fully observed and partially observed task scenarios described in Section 2, and in both cases we investigate both the P2P gating and the Batch mode of predictions on new tasks. The fully observed tasks case, considered in Section 4.1, investigates the situation where data generating distribution of the target task is actually the same as that of one of the source tasks. In this case all available tasks are used in the training phase, but in the testing phase the model has no information from which of the source task the target task comes from. The second set of experiments, described in Section 4.2, considers the case of the partially observed tasks. In this case the data generating distribution of the target task does not match the distribution of one of the source tasks, so that the set of source tasks is strictly a subset of the set of all tasks. Training is performed on the source tasks, and testing on the totally unseen target tasks. While both scenarios are plausible applications of meta-generalising, Section 4.2 gives more insight into the connections between the correlation structure of the tasks and the task prediction mechanism on totally unseen tasks.

Five different data sets are considered in the experiments. The first two data sets are artificially generated to demonstrate the strengths and the limitations of the method; the first one satisfies the assumptions of the model, and the second one, which is only considered in Section 4.1, is in conflict with them. The third data set is a character classification problem between commonly confused handwritten letters. The fourth data set is an automated diagnosis problem: annotated heartbeats from ECG recordings are used to discriminate normal from arrhythmic beats, and each patient is considered as a task. The last data set, which is considered only in the second set of experiments, is a landmine detection problem. More details are given in each section separately. We present results for different training set sizes, and for each training size experiments are repeated 25 times by randomly selecting the data points used for training from each task. Furthermore, in both scenarios three types of outputs are considered from the CMTMC model; the batch written as "BatchMCAppr", the P2P gating written as "P2PMCGat", and the "MAP" estimate which simply selects the output of the multi-task classifier that has the highest posterior, something that is usually considered in classifier fusion techniques (Kuncheva, 2002). As our method essentially relies on the covariance structure between tasks, two types of baseline comparisons are possible: in the worst case, results should not be worse than completely ignoring the task structure and pooling together all training data. We refer to this baseline as Pool. In the best case, our method should not be statistically better than a method which leverages the same covariance structure and has access to all the task label information, for example, a standard multi-task learning approach. We refer to this best-case scenario as MTL; we compare with this only in the fully observed task scenario, as in the partially observed case the meta-generalising results are generally quite far from this best case.

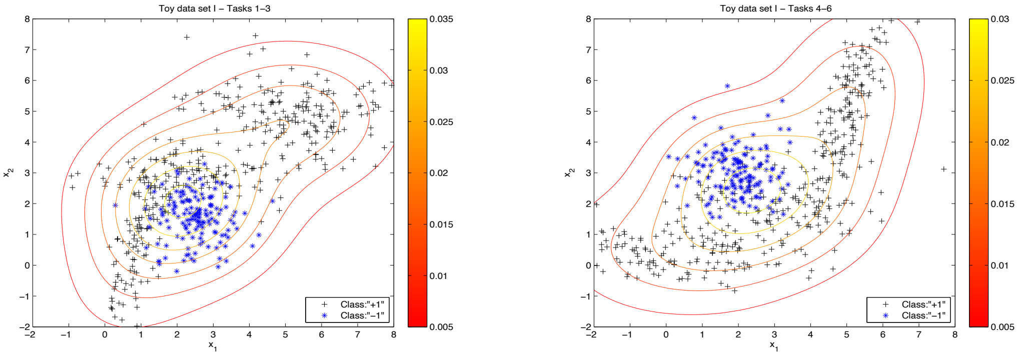
All methods are compared in terms of the area under the precision-recall curve, also known as the *Average Precision* (AP) (Davis and Goadrich, 2006). Simulation results were processed based on the work of Brodersen et al. (2010), that provides a smooth estimate of the precision-recall curve;[3] an equivalent performance measure that could have been used is the Area Under the Curve (AUC) (Hanley and Mcneil, 1982), which is also appropriate for imbalanced data sets. Note that simulation results follow the same pattern with both measures. In all experiments the task covariance matrix K^t

[3] Code downloaded from: `http://people.inf.ethz.ch/bkay/downloads`.

was parameterized as a correlation matrix (Rebonato and Jäckel, 2000), with unit diagonal, while the data covariance function K^x is set specifically for each data set depending the application.

4.1 Fully Observed Tasks

In this scenario, the data distribution of the target task is the same as that of (at least) one of the source tasks. This guarantees that the similarity of distribution assumption is met, however, as we'll see in the case of Toy data *II*, the low joint prediction error assumption is not automatically satisfied. Obviously, the actual input data will be different, due to the stochasticity of the data generating process. Intuitively, the success of the model depends strongly on whether the model will be able to infer correctly from which of the source tasks the target task actually comes from.



(a)

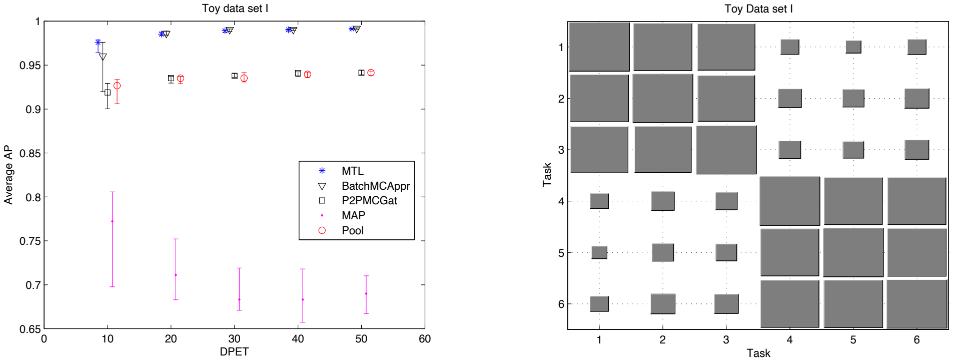
(b)

Figure 2: Toy data set I distribution; (a) scatter plot and density for the first cluster of tasks (1-3), (b) scatter plot and density for the second cluster of tasks (4-6).

4.1.1 TOY DATA SET *I*

The first toy data set is comprised of six binary classification tasks. This toy problem was previously used in Liu et al. (2009) in the context of semi-supervised multi-task learning. Data for the first three tasks are generated from a mixture of two partially overlapping Gaussian distributions, and similarly for the remaining three tasks. Hence, the six tasks cluster in two groups; for each task 600 data points were generated, which were equally divided between the two classes. The scatter plots of the two clusters are shown in Figures 2.a and 2.b.

This data set is ideal for demonstrating the concept of the meta-generalising for three reasons. First of all the assumptions of the model are satisfied. Secondly, the tasks group in two clusters. The third reason is that the densities of the clusters though similar are not exactly the same; this is illustrated in Figures 2.a and 2.b, which shows the contour plot of the densities of the two clusters. We use an Automatic Relevance Determination (ARD) data covariance function, which employs a different characteristic length scale for each feature, and is able to identify which features are more relevant for classification (Rasmussen and Williams, 2005).



(a)

(b)

Figure 3: Toy data set I classification Results; (a) Average AP over the 6 tasks, (b) Hinton Diagram of the task covariance matrix of the CMTMC model computed by averaging over the 25 repetitions with 50 data points per task.

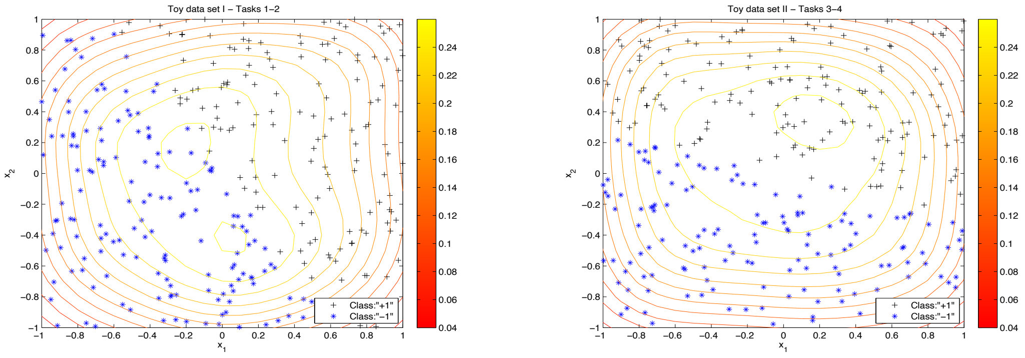
Classification results are presented in Figure 3.a; the Y axis is the AP, and the X axis is the number of data points from each task (DPET) used for training. The results show that, in this toy problem, the Batch mode performs similarly to the ideal MTL case, although it has a high variance for the case of 10 DPET. The P2PGat and Pooling method perform approximately 10% worse than the Batch, while the MAP estimate gives roughly 20% less than the Batch. Moreover, Figure 3.b shown the Hinton diagram[4] (Hinton, 1989) of the task covariance matrix of the CMTMC model which accurately recovers the structure of the tasks.

4.1.2 TOY DATA SET *II*

The second toy data set consists of four tasks which group into two clusters. The scatter plot as well as the density of the two clusters are shown in Figures 4.a and 4.b, for the first and second cluster respectively. The main feature of this data set, evident visually from Figure 4, is the similarity of the data generating distribution for the two tasks. While the densities are peaked in different locations, without class labels the tasks are almost identical, meaning that the multi-class classifier cannot learn to discriminate between the two tasks. As in the previous example, each task consisted of 600 data points equally divided between the two classes, and we used the ARD covariance function.

Figure 5.a shows the results the different methods produced. As expected, the Batch mode fails to correctly identify the task responsibilities; as a result, it gives a lower average AP than the MTL, a difference which does not decrease with the number of DPET, indicating statistical inconsistency. This is reinforced by the Hinton diagram of $\mathbf{K}^t$ in Figure 5.b, where it fails to identify the clusters of the tasks. Even though this difference is small it is significant for this easy problem, where the MTL algorithm performs close to 100%. Additionally, the P2PGat, the Pooling, and the MAP estimates perform better that the Batch, but they also fail to reach the performance of MTL.

[4] The Hinton diagram is a graphical representation of the values in a data matrix; here, it is used to display the correlations between the tasks.

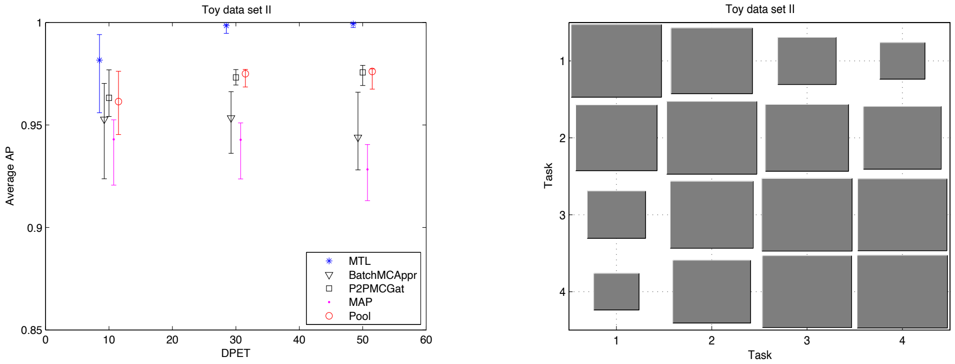


(a) (b)

Figure 4: Toy data set II distribution;(a) scatter plot and density for the first cluster of tasks(1-2), (b) scatter plot and density for the second cluster of task(3-4).

4.1.3 CHARACTER CLASSIFICATION

In this data set the task is to learn to classify between commonly confused handwritten letters, which is included in the "Transfer learning Toolkit" of Berkeley University available at `http://multitask.cs.berkeley.edu/`. This data set is comprised of eight binary classification tasks. The characters that are used and the number of samples are given in Table 1. Each sample is a 16×8 image, which results into a binary 128 feature vector. The covariance function that is employed for this data set is the *Radial Basis Function* (RBF).



(a) (b)

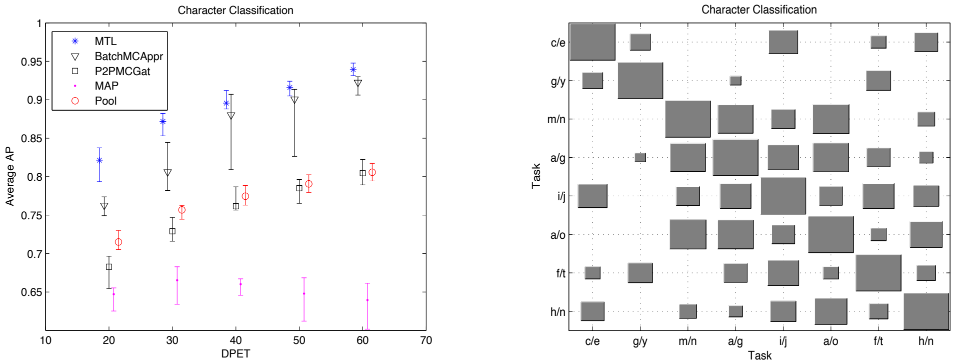
Figure 5: Toy data set II classification Results; (a) Average AP over the 4 tasks, (b) Hinton Diagram of the task covariance matrix.

The classification results for this data set are presented in Figure 6.a. The Batch method follows closely the ideal MTL performance, and outperforms the P2PGat, Pooling, and the MAP methods

Task	1	2	3	4	5	6	7	8
Letter	**c**	**g**	**m**	**a**	**i**	**a**	**f**	**h**
Number of data	2017	2460	1596	4016	4895	4016	918	858
Letter	**e**	**y**	**n**	**g**	**j**	**o**	**t**	**n**
Number of data	4928	1218	5004	2460	188	3880	2131	5004

Table 1: Description of the Character data set; each column is a task showing the two letters as well as the corresponding number of examples per character.

(although there is significant variability for small numbers of labeled data per task). Figure 6.b shows the Hinton diagram of the task covariance matrix, which indicates a more random structure between the tasks, but finds that some tasks are more correlated than others, for example 'a/g' with 'a/o', and 'i/j' with 'f/t'. It should be noted though, that in this data set the "low-error joint prediction" assumption is partially violated since there is label disagreement between tasks 'a/g' and 'g/y', where the 'g' letter belongs to class "+1" in task 'a/g' and to "-1" in task 'g/y'. This does not seem to have any adverse effect on the performance of the model, presumably as the difference between letters 'a' and 'y' is sufficient to unambiguously assign the target task to the correct source task.



(a)

(b)

Figure 6: Character Classification Results; (a) Average AP over the 8 tasks, (b) Hinton Diagram of the task covariance matrix.

4.1.4 ARRHYTHMIA CLASSIFICATION

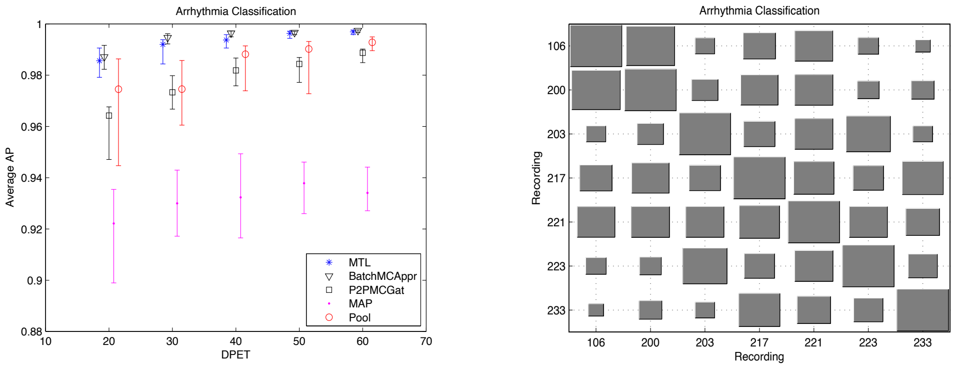
The arrhythmia data set consists of seven ECG recordings from different patients, which were acquired from the MIT-BIH Arrhythmia database (Goldberger et al., 2000). Each recording corresponds to a large number of heart beats, which is summarized in Table 2. Each patient is treated as a separate task, and the goal is to classify each heart beat into two classes, normal or premature ventricular contraction (PVC) arrhythmic beats. The same problem was considered in Skolidis et al.

(2008) using single task GP classifiers. Each recording was sampled at $360Hz$, and annotation provided by the database was used to separate the beats before any preprocessing. Each beat segment, consisting of 360 data points (one minute), was transformed into the frequency domain using a Fast Fourier Transform with a Hanning window. Only the first ten harmonics are used as features for classifying heart beats, as most of the information of the signal is contained in these harmonics.

Recording ID	106	200	203	217	221	223	233
Total number of data	2021	2567	2970	406	2349	2417	3053
Number of Normal heart beats	1503	1740	2526	244	1954	1955	2224
Number of PVC heart beats	518	827	444	162	395	462	829

Table 2: Description of the Arrhythmia data set.

Figure 7.a shows the average AP over the seven tasks. On average, the Batch method performs better than the P2PMCGat, the MAP, and the Pool, while it presents a small advantage compared to MTL. Interestingly, the MAP approach is consistently worse than other methods, a situation that will be reversed in the partially observed tasks scenario. As in the character classification problem the task covariance matrix $\mathbf{K}^t$, shown in Figure 7.b, demonstrates that there are correlations between the tasks but in more random way.



(a)

(b)

Figure 7: Arrhythmia Classification Results; (a) Average AP over the 7 tasks, (b) Hinton Diagram of the task covariance matrix.

4.1.5 OBSERVATIONS

This set of experiments has demonstrated the effectiveness of the CMTMC model in situations where the data distribution of the target task comes from one of the source tasks. Several observations are made:

1. In the fully observed tasks scenario, the space of tasks has been sampled sufficiently (by definition). In this case the Batch mode should theoretically be the best method, since all data

points are needed to produce an accurate estimate of the density of the target task. This is empirically confirmed in our investigation, as Batch closely approaches the MTL results in all cases.

2. If the "low-error joint prediction" assumption is violated, then meta-generalising becomes a very hard problem, possible unsolvable. The performance on the second toy example was not particularly bad, since all methods achieved higher that 90% in terms of AP, but none of methods reached the performance of the MTL algorithm, and the performance did not appreciably improve when more training data were provided, indicating statistical inconsistency. This effect could be dramatically increased if for example the classes between the clusters were anti-correlated, so that similar data generating distributions could be potentially associated with opposite predictions. Note though that if discriminative task descriptor features are available then this problem can be overcome, because augmenting the feature space would result in a different mapping of the latent function f.

3. If the model assumptions are met, the correlation structure of the tasks does not have a strong influence on the predictions, since the Batch mode outperformed the P2PGat gating and MAP estimate in all experiments. As we will see, this will be a crucial difference between the fully and partially observed tasks scenario.

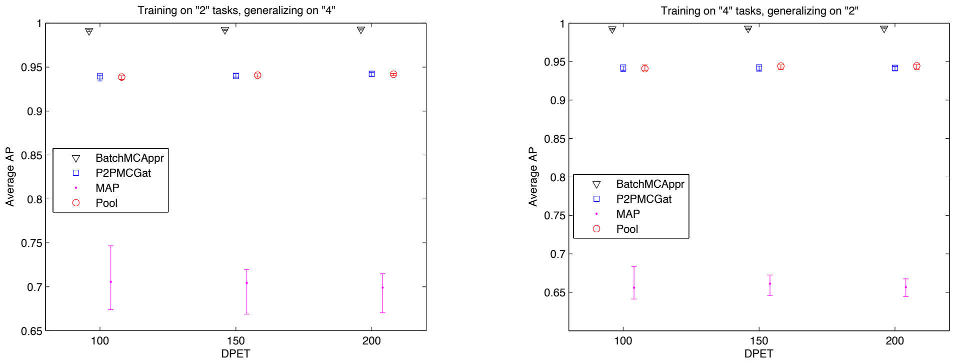
4.2 Partially Observed Tasks

We now consider the harder problem of making predictions on completely unseen tasks. In this case, *a priori* we have no guarantee that any of the underlying modelling assumptions (similarity of distribution and low-error joint prediction) may hold. However, in some situations it is not unrealistic to assume that inter-task correlations will be structured, for example by the presence of *clusters* of similar tasks. These clusters may be evident from the experimental design of the problem (as in the case of the landmine data set discussed below), or may become evident from the training phase on the source tasks, if the learned task covariance matrix exhibits a strong block structure.

We are not aware of other methods that has a distribution matching mechanism to perform predictions on totally unseen tasks. Therefore, in this section we will only compare the different inference mechanisms of the CMTMC model (Batch and P2PGat) with a GP model trained by pooling all data together and with the MAP combination of classifiers.

4.2.1 TOY DATA SET *I*

We consider the toy data set that was used in Section 4.1.1 consisting of two clusters of tasks; in this section, training tasks are selected by randomly selecting equal number of tasks from each cluster. The challenge for the model is to correctly classify the task, given the similarity of the task distributions between the two clusters (see Figure 2). While it could be argued that, as the tasks in each cluster have the same data generating distribution, this example is very close to the fully observed case scenario (and it certainly is if we consider the underlying tasks to be two rather than six), it is still a useful illustrative example as a limiting case where assumptions are perfectly met. Experimental results are presented for two and four training tasks in Figures 8.a and 8.b respectively. Naturally, as this data set is designed to match our modelling assumptions, the Batch method outperforms all other methods; it is interesting however that the method successfully detects from which cluster of tasks the unseen target task comes from even for relatively small training set



(a) (b)

Figure 8: Average AP on the unseen tasks of Toy data set *I*; (a) training on 2 tasks generalising on 4, (b) training on 4 tasks generalising on 2.

sizes. Comparing the performance of the Toy data set *I* in the fully and partially observed cases, in Figures 3 and 8 respectively, reveals that the same levels of AUC are achieved in both experimental setups, indicating that the task classification GP is highly confident of the correct result.

4.2.2 LANDMINE DETECTION

The landmine detection data set consists of images measured with airborne radar systems, and the goal is to predict landmines or clutter (Xue et al., 2007). Data are collected from 19 landmine fields, which are considered as subtasks, and each point is represented by a nine-dimensional feature vector. Tasks 1-10 correspond to regions that are relatively highly foliated while tasks 11-19 correspond to regions that are bare earth or desert. Figure 9 shows the number of data points from each task and each class, which indicates that this data set is highly imbalanced in favor of the Clutter ('-1') class. The experimental setup suggests the presence of two clusters of tasks corresponding to the

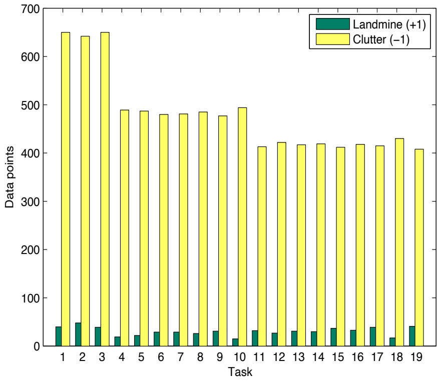
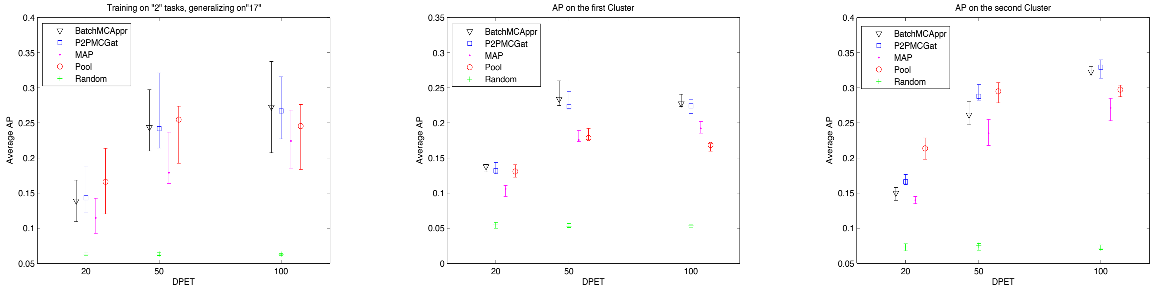


Figure 9: Landmine detection data distribution.

geomorphology of the region the observations come from; this is confirmed by our preliminary investigation (not shown), as well as from previously published results on this data set by Xue et al. (2007) and Liu et al. (2009). Thus, in this data set training tasks are set by randomly selecting equal number of tasks from the first cluster, tasks 1-10, and from the second cluster, tasks 11-19. Experiments are presented for two, four, and eight training tasks. The data covariance function that is used for this data set is the ARD.

Figures 10.a, 11.a, and 12.a shows the mean AP on the 17, 15, and 11 unseen target tasks for each partition respectively. Due to the high imbalance between the classes (Landmine-Clutter) the achieved AP of all methods is very low. Therefore, in this data set we also present the AP of a random predictor which clearly shows the improvement of each method considered. Note that in terms of AUC the results obtained in this work are consistent with previous studies in this data set (Xue et al., 2007; Liu et al., 2009), which are presented in Appendix C Figure 14 for completeness. Moreover, it is noticed that there are large overlapping error bars between all methods. Large error bars give evidence that there might be two levels of performance. Therefore, for each partition we provide the average AP for each cluster separately; subfigures (b) from Figures 10, 11, and 12 show the average AP for the first cluster, and subfigures (c) for the second cluster. Measuring the AP in each cluster separately gives significantly smaller error bars, and reveals interesting structures in the problem. Specifically, the performance on the second cluster is always better than on the first cluster by a considerable margin. Moreover, comparing the methods on each cluster separately we see that the Batch method outperformed the pooling and the P2PGat in most of the cases, particularly in the first cluster where the advantages become very significant as we increase the number of tasks/DPETs. The correlation structure within the second cluster is looser, immplying a weaker applicability of our modelling assumptions. However it should be pointed out that this is a substantially harder pattern recognition task compared to the toy data set considered above. For example, Liu et al. (2009) that investigated the application of semi-supervised MTL on this data set achieved a best performance of 78% AUC; the CMTMC (which relies on the more flexible GP framework for MTL) achieves an average AUC above 76% on totally unseen tasks having trained on *only* 8 source tasks with 100 DPET (see Figure 14).



(a) (b) (c)

Figure 10: AP on the 17 unseen tasks of Landmine data set; training on 2 tasks, generalising on 17; (a) AP over 17 tasks, (b) AP over 9 tasks of the first cluster, (c) AP over 8 tasks of the second cluster.

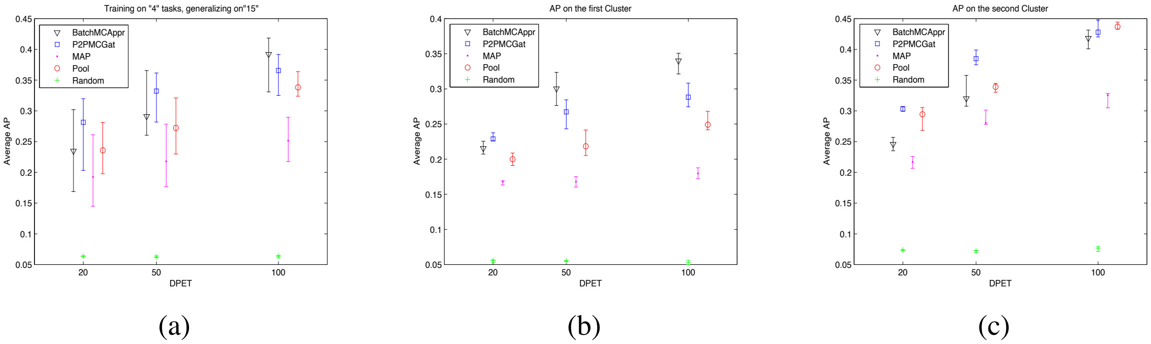


Figure 11: Average AP on the 15 unseen tasks of Landmine data set; training on 4 tasks, generalising on 15; (a) Overall AP over 15 tasks, (b) Average AP over 8 tasks of the first cluster, (c) Average AP over 7 tasks of the second cluster.

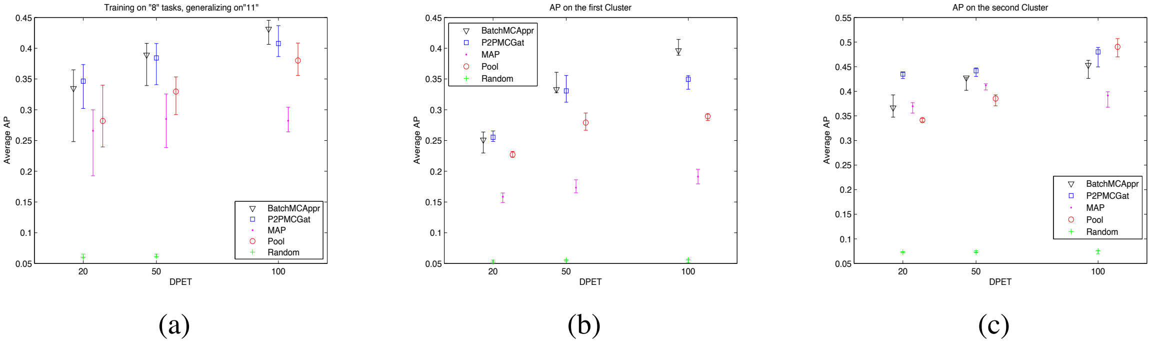
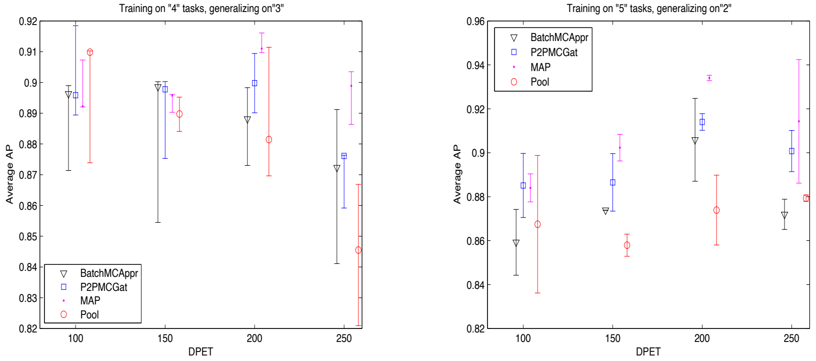


Figure 12: Average AP on the 11 unseen tasks of Landmine data set; training on 8 tasks, generalising on 11; (a) Overall AP over 11 tasks, (b) Average AP over 6 tasks of the first cluster, (c) Average AP over 5 tasks of the second cluster.

4.2.3 ARRHYTHMIA CLASSIFICATION

As a second real data set, we return to the arrhythmia classification problem introduced in Section 4.1.4. The results from the fully observed tasks scenario indicate an unclear pattern of correlations between the tasks, as summarised in the task covariance matrix Figure 7.b, which calls into question the validity of the similarity of distribution assumption. Fortunately, in this application the classes have a physical interpretation. For example normal heart beats between different patients, although not exactly the same, can be expected to be similar, and a PVC arrhythmic heart beat of one patient can not have the wave form of a normal heart beat from another patient. This allows us to assume that the classes between the tasks will not be anti-correlated, so that at least the low-error joint prediction assumption should approximately hold.

Since there are no obvious clusters among tasks, in this set of experiments the training tasks are chosen by randomly selecting some for training and keeping the rest as test tasks. Figure 13 presents the results on the unseen tasks that were obtained by training the CMCMT model with 4 and 5 tasks. First of all, we observe that the average AUC in the partially observed case is a lot lower than in the fully observed case, something perhaps to be expected since, contrary to the previous two examples, the model assumptions are not fully met in this data set. Surprisingly, the method that achieved the best performance was the MAP, and no principled justification can be given for that. Secondly, we observe that the performance in this set of experiments exhibits some interesting patterns as the number of training tasks increases. Specifically, for four training tasks the performance of all methods does not significantly improve as we increase the number of data points per task, and in some cases it even deteriorates, a phenomenon that was also observed for 2 and 3 training tasks but results are omitted for brevity. This indicates that if the space of tasks has not been sampled sufficiently, the model can not yield good generalisation performance to new tasks, even if the number of training data increases. In contrast, for five training tasks the MAP and P2PGat methods yield a significant improvement of performance as the number of data points increases (levelling off after 200 DPETs).



(a) (b)

Figure 13: Average AP on the unseen tasks of Arrhythmia data set on different number of training tasks; (a) training on 4 tasks, generalising on 3, (b) training on 5 tasks, generalising on 2.

Empirically, it would appear that the P2PGat method is preferable to the Batch method when the model assumptions are violated. Intuitively, one could argue that the Batch method is less flexible, as the relative contribution of the different single-class predictors is fixed across all points in the target task. Therefore, if the model assumptions are violated, leading to an incorrect task labelling, the propagated error could have a worse effect in Batch than in P2PGat. This is partly confirmed by the analysis of Toy data set *II* in Section 4.1.2, where the model assumptions were violated and P2PGat gave significantly higher AP than the Batch method.

4.2.4 CHARACTER CLASSIFICATION

For reasons of completeness, we present an analysis of the character classification problem in the partially observed tasks scenario. Here the validity of the model assumptions is dubious; nevertheless, we believe that interesting lessons can be learned from model failure. The fully observed tasks analysis of the character classification problem did not reveal any clusters of tasks. Furthermore, there is no reason to believe that the low-error joint prediction assumption may hold: some tasks might even be anticorrelated, as in tasks ‘a/g’ and ‘g/y’, where letter ‘g’ belongs to the negative class for task ‘a/g’, and to the positive class for task ‘g/y’. Therefore, the character classification problem is ill-suited for this type of experiments. This is borne out by experimental evidence: simulation results with 4, 5, and 6 training tasks, which are omitted for brevity, indicated that increasing the number of tasks and the number of training points per task does not improve the performance in any of the methods. Specifically, the results obtained were close to that of a random predictor indicating statistical inconsistency of the model assumptions with the data.

4.2.5 OBSERVATIONS

Meta-generalising in a partially observed tasks scenario is an extremely hard problem; nevertheless, we believe there are some interesting points that can be made from the previous experimental analysis. Below we summarise the most important observations for this scenario.

1. In situations where there are clusters of tasks, even though the model hasn’t seen all tasks, the Batch method can still make accurate predictions that reaches the performance of the fully observed tasks case. Pragmatically, one could consider whether the training phase of the model has revealed clusters of tasks when deciding which prediction method to apply.

2. In multi-task problems where the correlations between the tasks are less pronounced, but where the low-error joint prediction is satisfied and where a sufficient number of training tasks is available, the method that is most appropriate is the P2PGat, since it provides a more flexible task assignment mechanism than the Batch mode. The validity of the low-error joint prediction assumption can sometimes be assessed from the nature of the problem (as in the arrhythmia case).

3. Sufficient exploration of the task space is essential for the success of the method. While we have not tested our model for very large numbers of training tasks, the results suggest that often a significant improvement in performance can be achieved when the number of training tasks crosses a critical number, indicating a sufficient coverage of the task space. This phenomenon was observed in the Arrhythmia classification problem for 2 and 3 training tasks where the performance of the models remained the same as the number of training samples per task increased. In essence more training data lead to stronger biases for meta-generalisation in target tasks that are not correlated with any of the training tasks.

4. In most cases, when the assumptions of the model are only approximately met and when the exploration of the task space is insufficient, the generalisation performance on totally unseen tasks is still modest, and it may be that other approaches based on mixtures of GP experts (Tresp, 2000) achieve similar results. An extensive comparison with these approaches would be interesting, but outside the scope of the present work.

5. Conclusions

In this paper we presented an investigation on the use of Gaussian Processes for meta-generalisation, that is, predicting on unseen learning tasks by leveraging the information of several, related tasks. Our model attacks the meta-generalisation problem by coupling two GPs, a multi-class classifier that learns task responsibilities, and a multi-task classifier that learns prediction models on individual tasks as well as learning the global correlation structure between training tasks. While it should be emphasized that this is an initial attempt to address what is certainly a very ambitious problem, we believe the model will prove useful to understand meta-generalisation. First of all, it provides a constructive approach to meta-generalisation: most previous studies (Baxter, 2000) have been mainly theoretical investigations attempting to establish the necessary conditions for meta-generalisation to work, or have focused on the domain adaptation scenario (Ben-David et al., 2007, 2010). Our model is an attempt to translate these conditions into a model, and to investigate how well such a model may perform on real meta-generalisation problems.

It is important to remark that our method crucially relies on the ability to learn the covariance matrix of a GP: the fundamental ingredient in this work is the task correlation matrix which captures the correlations between source tasks. This not only has a significant impact on the prediction results, but can reveal the presence of clusters of tasks within the data, hence guiding the choice of the appropriate prediction method (Batch or P2PGat). Many multi-task learning approaches do not explicitly model the correlations, but transfer learning solely through some shared prior over parameters (Yu et al., 2005, e.g.). While this could have computational advantages, we would argue that the implicit modelling of task correlations would make them less suitable for meta-generalisation. A common problem, shared by many GP models, is the computational cost when samples become large, which would be the probable situation in many applications such as personalised medicine. Our approach also suffers from the cubic scaling of matrix inversions needed within GP inference; while sparsity inducing approaches could be helpful (Snelson and Ghahramani, 2006), it would be interesting to explore sparsity within the task space as well as within the data space.

While we believe that our results are encouraging and help clarify the importance of the various assumptions underlying meta-generalisation, it remains undeniable that in many practical situations it is impossible to assess the validity of these assumptions, making meta-generalisation an extremely challenging problem. Possible avenues to extend the applicability of the approach could be to consider task descriptor features, or to introduce a semi-supervised element in the model in the spirit of domain adaptation approaches.

Acknowledgments

We would like to acknowledge support for this project from the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/F009461/2. G. Sanguinetti acknowledges support from the Scottish government through the Scottish Informatics and Computer Science Alliance (SICSA). We wish to thank Eleni Vasilaki, Amos Storkey, Sethu Vijayakumar and Mark Girolami for the useful discussions and their support.

Appendix A. Approximate Inference

This appendix computes the approximate posteriors for $Q(\mathbf{g})$, $Q(\mathbf{f})$ and $Q(\mathbf{h}^x)$. The posterior of $Q(\mathbf{H}^t)$ can be found in Girolami and Rogers (2006) and therefore details are omitted.

A.1 $Q(\mathbf{g})$

The approximate posterior for $Q(\mathbf{g})$ is computed as Girolami and Rogers (2006)

$$\begin{aligned} Q(\mathbf{g}) &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{h}^t)}\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\log p(h^t_{ij}|g_{ij}) + \log p(\mathbf{g}_j|\mathbf{X})\right)\right\} \\ &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{h}^t)}\left(\sum_{j=1}^{M}\log \mathcal{N}_{\mathbf{h}^t_j}(\mathbf{g}_j,\mathbf{I}) + \log \mathcal{N}_{\mathbf{g}_j}(0,\mathbf{K}^x)\right)\right\} \\ &\propto \prod_{j=1}^{M}\mathcal{N}_{\tilde{\mathbf{h}}^t_j}(\mathbf{g}_j,\mathbf{I})\mathcal{N}_{\mathbf{g}_j}(0,\mathbf{K}^x), \end{aligned}$$

which gives that $Q(\mathbf{g}) = \prod_{j=1}^{M} Q(\mathbf{g}_j) = \prod_{j=1}^{M}\mathcal{N}_{\mathbf{g}_j}(\tilde{\mathbf{g}}_j,\mathbf{\Sigma}^g)$, where $\mathbf{\Sigma}^g = \left(\mathbf{I} + (\mathbf{K}^x)^{-1}\right)^{-1} = \mathbf{K}^x\left(\mathbf{I}+\mathbf{K}^x\right)^{-1}$, and $\tilde{\mathbf{g}}_j = \mathbf{\Sigma}^g\tilde{\mathbf{h}}^t_j$.

A.2 $Q(\mathbf{f})$

$$\begin{aligned} Q(\mathbf{f}) &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{h}^x)}\left\{\sum_{i=1}^{NM}\log p(h^x_i|f_i) + \log p(\mathbf{f}|\mathbf{X})\right\}\right\} \\ &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{h}^x)}\left\{-\frac{1}{2}\mathbf{h}^{xT}\mathbf{h}^x + \mathbf{f}^T\mathbf{h}^x - \frac{1}{2}\mathbf{f}^T\mathbf{f} - \frac{1}{2}\mathbf{f}^T\left(\mathbf{K}^t\otimes\mathbf{K}^x\right)^{-1}\mathbf{f} + const.\right\}\right\} \\ &\propto \exp\left\{-\frac{1}{2}\mathbf{f}^T(\mathbf{I} + \left(\mathbf{K}^t\otimes\mathbf{K}^x\right)^{-1})\mathbf{f} + \mathbf{f}^T\tilde{\mathbf{h}} + const.\right\}, \end{aligned}$$

which gives that $Q(\mathbf{f}) = \mathcal{N}_{\mathbf{f}}(\tilde{\mathbf{f}},\mathbf{\Sigma}^f)$ where $\tilde{\mathbf{f}} = \mathbf{\Sigma}^f\tilde{\mathbf{h}}^x$, and $\mathbf{\Sigma}^f = (\mathbf{I} + (\mathbf{K}^t\otimes\mathbf{K}^x)^{-1})^{-1} = \mathbf{K}^t\otimes\mathbf{K}^x(\mathbf{I} + \mathbf{K}^t\otimes\mathbf{K}^x)^{-1}$.

A.3 $Q(\mathbf{h}^x)$

$$\begin{aligned} Q(\mathbf{h}^x) &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{f})}\left\{\sum_{i=1}^{NM}\log p(y^x_i|h^x_i) + \log p(h^x_i|f_i)\right\}\right\} \\ &\propto \exp\left\{\log\left(\prod_{i=1}^{NM}p(y^x_i|h^x_i)\right) + \log\left(\prod_{i=1}^{NM}\mathcal{N}_{h^x_i}(\tilde{f}_i,1)\right)\right\} \\ &\propto \prod_{i=1}^{NM}\mathcal{N}_{h^x_i}(\tilde{f}_i,1)\delta(h^x_i) \end{aligned}$$

which gives that $Q(h^x_i) = \frac{1}{Z_i}\mathcal{N}_{h^x_i}(\tilde{f}_i,1)\delta(h^x_i)$, and we have that

$$Q(h^x_i) = \left\{\begin{array}{ll} \frac{1}{Z_i}\int_0^{+\infty} h^x_i\mathcal{N}_{h^x_i}(\tilde{f}_i,1) = \tilde{f}_i + \frac{\mathcal{N}_{\tilde{f}_i}(0,1)}{\Phi(\tilde{f}_i)} & \text{for } y^x_i = +1 \\ & \\ \frac{1}{Z_i}\int_{-\infty}^{0} h^x_i\mathcal{N}_{h^x_i}(\tilde{f}_i,1) = \tilde{f}_i - \frac{\mathcal{N}_{\tilde{f}_i}(0,1)}{\Phi(-\tilde{f}_i)}) & \text{for } y^x_i = -1 \end{array}\right\},$$

where $Z_i = \Phi(\pm\tilde{f}_i)$ for $y^x_i = \pm 1$. The approximate posterior of $Q(\mathbf{H}^t)$ can be computed in a similar manner, and we refer the interested reader to Girolami and Rogers (2006).

Appendix B. Lower Bound

This appendix presents the analytical form of the variational bound as well as the gradients of the bound with respect to the hyperparameters θ^x and θ^t.

B.1 Lower Bound on Log Marginal Likelihodd

The lower bound on the log marginal likelihood is computed by

$$\begin{aligned}\mathcal{L}(Q) = {} & \mathbb{E}_{Q(\Theta)}[\log p(\mathbf{y}^t, \mathbf{y}^x, \mathbf{g}, \mathbf{h}^t, \mathbf{f}, \mathbf{h}^x | X, \theta^t, \theta^x)] - \\ & \qquad \mathbb{E}_{Q(\Theta)}[\log Q(\mathbf{g})Q(\mathbf{h}^t)Q(\mathbf{f})Q(\mathbf{h}^x)] \\ = {} & -\frac{NM}{2}\log(2\pi) + \frac{N}{2}\log(2\pi) + \frac{NM}{2} - \frac{M}{2}\mathrm{trace}(\boldsymbol{\Sigma}^g) - \frac{1}{2}\sum_m \tilde{\mathbf{g}}_m^T \mathbf{K}^{x^{-1}} \tilde{\mathbf{g}}_m \\ & - \frac{M}{2}\mathrm{trace}\left(\mathbf{K}^{x^{-1}}\boldsymbol{\Sigma}^g\right) - \frac{M}{2}\log|\mathbf{K}^x| + \frac{M}{2}\log|\boldsymbol{\Sigma}^g| + \sum_n \log z_n^t \\ & + \sum_{n=1}^{N} \log z_n^x - \frac{1}{2}\log|\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x| - \frac{1}{2}\tilde{\mathbf{f}}^T (\mathbf{K}^t \otimes \mathbf{K}^x)^{-1}\tilde{\mathbf{f}}, \qquad (15)\end{aligned}$$

where $z_n^t = \mathbb{E}_{p(u)}\left\{\prod_{j \neq i} \Phi(u + \tilde{g}_{ni} - \tilde{g}_{nj})\right\}$, and $z_n^x = \Phi(y_n^x \tilde{f}_n)$.

Terms that depend on hyperparameters θ^x and θ^t are:

$$\begin{aligned}\mathcal{L}(Q)_{\theta^x,\theta^t} = {} & -\frac{M}{2}\mathrm{trace}(\boldsymbol{\Sigma}^g) - \frac{1}{2}\sum_m \tilde{\mathbf{g}}_m^T \mathbf{K}^{x^{-1}} \tilde{\mathbf{g}}_m - \frac{M}{2}\mathrm{trace}\left(\mathbf{K}^{x^{-1}}\boldsymbol{\Sigma}^g\right) \\ & - \frac{M}{2}\log|\mathbf{K}^x| + \frac{M}{2}\log|\boldsymbol{\Sigma}^g| - \frac{1}{2}\log|\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x| - \frac{1}{2}\tilde{\mathbf{f}}^T (\mathbf{K}^t \otimes \mathbf{K}^x)^{-1}\tilde{\mathbf{f}}.\end{aligned}$$

B.2 Gradients on Lower Bound

The gradients with respect to the parameters of the data covariance function $\mathbf{K}^x$ are computed from:

$$\begin{aligned}\frac{\partial}{\partial\theta^x}\mathcal{L}(q) = {} & -\frac{M}{2}\mathrm{trace}\left\{\boldsymbol{\Omega}(\mathbf{I} + \mathbf{K}^x)^{-1} - \mathbf{K}^x(\mathbf{I} + \mathbf{K}^x)^{-1}\boldsymbol{\Omega}(\mathbf{I} + \mathbf{K}^x)^{-1}\right\} + \frac{1}{2}\tilde{\mathbf{g}}_m^T \mathbf{K}^{x^{-1}}\boldsymbol{\Omega}\mathbf{K}^{x^{-1}}\tilde{\mathbf{g}}_m \\ & + \frac{M}{2}\mathrm{trace}\left\{(\mathbf{I} + \mathbf{K}^x)^{-1}\boldsymbol{\Omega}(\mathbf{I} + \mathbf{K}^x)^{-1}\right\} - \frac{M}{2}\mathrm{trace}\left\{\mathbf{K}^{x^{-1}}\boldsymbol{\Omega}\right\} \\ & + \frac{M}{2}\mathrm{trace}\left\{(\mathbf{I} + \mathbf{K}^{x^{-1}})^{-1}\mathbf{K}^{x^{-1}}\boldsymbol{\Omega}\mathbf{K}^{x^{-1}}\right\} + \frac{1}{2}\tilde{\mathbf{f}}^T\left(\mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\mathbf{K}^t \otimes \boldsymbol{\Omega}\left(\mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\tilde{\mathbf{f}} \\ & - \frac{1}{2}\mathrm{trace}\left(\left(\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\mathbf{K}^t \otimes \boldsymbol{\Omega}\right). \qquad (16)\end{aligned}$$

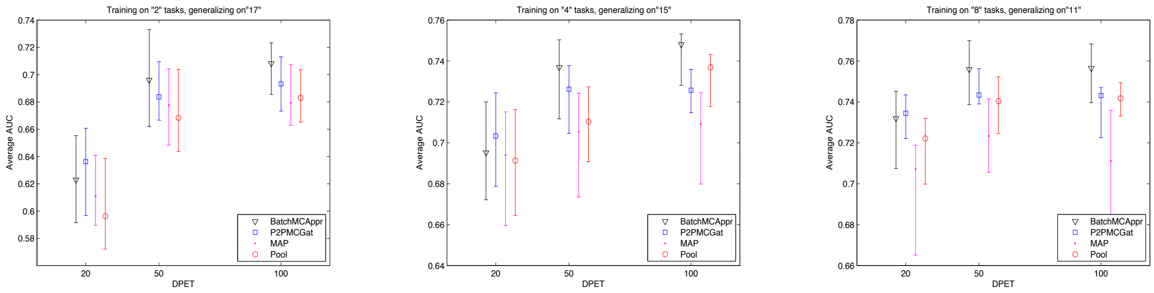
While the gradients with respect to the parameters of the task covariance matrix are computed from:

$$\frac{\partial}{\partial\theta^t}\mathcal{L}(q) = \frac{1}{2}\tilde{\mathbf{f}}^T\left(\mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\boldsymbol{\Xi} \otimes \mathbf{K}^x\left(\mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\tilde{\mathbf{f}} - \frac{1}{2}\mathrm{trace}\left(\left(\mathbf{I} + \mathbf{K}^t \otimes \mathbf{K}^x\right)^{-1}\boldsymbol{\Xi} \otimes \mathbf{K}^x\right),$$

where $\boldsymbol{\Omega} = \frac{\partial \mathbf{K}^x}{\partial\theta^x}$, and $\boldsymbol{\Xi} = \frac{\partial \mathbf{K}^t}{\partial\theta^t}$

Appendix C. Additional Results on the Landmine Detection Problem

This appendix provides additional results for the Landmine detection problem (section 4.2.2) from the Partially observed tasks scenario. In contrast to the results presented in section 4.2.2 where methods were compared in terms of AP, Figure 14 presents results in terms of AUC, similarly to previous studies in that data set (Xue et al., 2007; Liu et al., 2009).



(a) (b) (c)

Figure 14: AUC on the Landmine detection problem; (a) AUC over 17 tasks by training on 2 tasks, (b) AUC over 15 tasks by training on 4 tasks, (c) AUC over 11 tasks by training on 8 tasks.

References

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

R.K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

A. Arnold, R. Nallapati, and W.W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of the 7th IEEE International Conference on Data Mining Workshops*, pages 77–82, Omaha, Nebraska, USA, 2007.

B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, pages 137–145, Vancouver, Canada, 2007.

S. Ben-David, T. Luu, T. Lu, and D. Pál. Impossibility theorems for domain adaptation. In *Proceedings of the 13th International Workshop on Artificial Intelligence and Statistics*, volume 13, pages 129–136, Sardinia, Italy, 2010.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155, 2009.

E. Bonilla, K. M. Chai, and C.K.I. Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, pages 153–160, Vancouver, Canada, 2008.

K.H. Brodersen, C.S. Ong, K.E. Stephan, and J.M. Buhmann. The binormal assumption on precision-recall curves. In *Proceedings of the 2010 International Conference on Pattern Recognition*, pages 4263–4266, Istanbul, Turkey, 2010.

R. Caruana. Multi-task learning. *Machine Learning*, 28(1):41–75, 1997.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT Press, Cambridge, MA, 2006.

K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *The Journal of Machine Learning Research*, 9:1757–1774, 2008.

N. A.C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons. New York. US, 1993.

L. Csató, E. Fokoué, M. Opper, B. Schottky, and O. Winther. Efficient approaches to gaussian process classification. In *Advances in Neural Information Processing Systems 12*, pages 251–257, Denver, Colorado, 2000.

H. Daumé. Frustratingly easy domain adaptation. In *Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 256–263, 2007.

H. Daumé III. Bayesian multitask learning with latent hierarchies. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 135–142, Montreal, Canada, 2009.

H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, Pittsburgh, USA, 2006.

I. Deak. Three digit accurate multiple normal probabilities. *Numerische Mathematik*, 35(4):369–380, 1980.

H. I. Gassmann, I. Deak, and T. Szantai. Computing multivariate normal probabilities: A new look. *Journal of Computational and Graphical Statistics*, 11(4):920–949, 2002.

A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992.

M. Girolami and S. Rogers. Variational bayesian multinomial probit regression with gaussian process priors. *Neural Computation*, 18(8):1790–1817, 2006.

M. Girolami and M. Zhong. Data integration for classification problems employing Gaussian process priors. In *Advances in Neural Information Processing Systems 19*, pages 465–472, Vancouver, Canada, 2007.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23): 215–220", 2000.

A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall/CRC, 2000.

J. A. Hanley and B. J. Mcneil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.

G.E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234, 1989.

J. Huang, A. J. Smola, A. Gretton, K M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, pages 601–608, Vancouver, Canada, 2007.

R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

L.I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.

Q. Liu, X. Liao, H. Li, J. R. Stack, and L. Carin. Semisupervised multitask learning. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 31(6):1074–1086, 2009.

D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems 21*, pages 1041–1048, Vancouver, Canada, 2009.

T.P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, volume 17, pages 362–369, San Francisco, CA, USA, 2001.

M. Opper and O. Winther. Gaussian processes for classification: mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2009.

R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766, Corvallis, OR, USA, 2007.

C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. MIT press, 2005.

C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems 14*, pages 881–888, Vancouver, Canada, 2001.

R. Rebonato and P. Jäckel. The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Journal of Risk*, 2(2), 2000.

G. Skolidis and G. Sanguinetti. Bayesian multitask classification with gaussian process priors. *IEEE Transactions on Neural Networks*, 22(12):2011 –2021, Dec. 2011.

G. Skolidis, RH Clayton, and G. Sanguinetti. Automatic classification of arrhythmic beats using gaussian processes. In *IEEE Transactions on Computers in Cardiology, 2008*, pages 921–924, Bologna, Italy, 2008.

E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264, Vancouver, Canada, 2006.

A. J. Storkey and M. Sugiyama. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems 19*, pages 1337–1344, Vancouver, Canada, 2007.

M. Sugiyama, M. Krauledat, and K.R. Müller. Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005, 2007.

Volker Tresp. Mixtures of gaussian processes. In *Advances in Neural Information Processing Systems 13*, pages 654–660, Vancouver, Canada, 2000. MIT Press.

S.R. Waterhouse. *Classification and Regression Using Mixtures of Experts*. PhD thesis, Department of Engineering, Cambridge University, 1997.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63, 2007.

K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1012–1019, Bonn, Germany, 2005.

Journal of Machine Learning Research 13 (2012) 723-773 Submitted 4/08; Revised 11/11; Published 3/12

A Kernel Two-Sample Test

Arthur Gretton[*] ARTHUR.GRETTON@GMAIL.COM
MPI for Intelligent Systems
Spemannstrasse 38
72076 Tübingen, Germany

Karsten M. Borgwardt[†] KARSTEN.BORGWARDT@TUEBINGEN.MPG.DE
Machine Learning and Computational Biology Research Group
Max Planck Institutes Tübingen
Spemannstrasse 38
72076 Tübingen, Germany

Malte J. Rasch[‡] MALTE@MAIL.BNU.EDU.CN
19 XinJieKouWai St.
State Key Laboratory of Cognitive Neuroscience and Learning,
Beijing Normal University,
Beijing, 100875, P.R. China

Bernhard Schölkopf BERNHARD.SCHOELKOPF@TUEBINGEN.MPG.DE
MPI for Intelligent Systems
Spemannstrasse 38
72076, Tübingen, Germany

Alexander Smola[§] ALEX@SMOLA.ORG
Yahoo! Research
2821 Mission College Blvd
Santa Clara, CA 95054, USA

Editor: Nicolas Vayatis

Abstract

We propose a framework for analyzing and comparing distributions, which we use to construct statistical tests to determine if two samples are drawn from different distributions. Our test statistic is the largest difference in expectations over functions in the unit ball of a reproducing kernel Hilbert space (RKHS), and is called the *maximum mean discrepancy* (MMD). We present two distribution-free tests based on large deviation bounds for the MMD, and a third test based on the asymptotic distribution of this statistic. The MMD can be computed in quadratic time, although efficient linear time approximations are available. Our statistic is an instance of an integral probability metric, and various classical metrics on distributions are obtained when alternative function classes are used in place of an RKHS. We apply our two-sample tests to a variety of problems, including attribute matching for databases using the Hungarian marriage method, where they perform strongly. Excellent performance is also obtained when comparing distributions over graphs, for which these are the first such tests.

[*]. Also at Gatsby Computational Neuroscience Unit, CSML, 17 Queen Square, London WC1N 3AR, UK.

[†]. This work was carried out while K.M.B. was with the Ludwig-Maximilians-Universität München.

[‡]. This work was carried out while M.J.R. was with the Graz University of Technology.

[§]. Also at The Australian National University, Canberra, ACT 0200, Australia.

Keywords: kernel methods, two-sample test, uniform convergence bounds, schema matching, integral probability metric, hypothesis testing

1. Introduction

We address the problem of comparing samples from two probability distributions, by proposing statistical tests of the null hypothesis that these distributions are equal against the alternative hypothesis that these distributions are different (this is called the two-sample problem). Such tests have application in a variety of areas. In bioinformatics, it is of interest to compare microarray data from identical tissue types as measured by different laboratories, to detect whether the data may be analysed jointly, or whether differences in experimental procedure have caused systematic differences in the data distributions. Equally of interest are comparisons between microarray data from different tissue types, either to determine whether two subtypes of cancer may be treated as statistically indistinguishable from a diagnosis perspective, or to detect differences in healthy and cancerous tissue. In database attribute matching, it is desirable to merge databases containing multiple fields, where it is not known in advance which fields correspond: the fields are matched by maximising the similarity in the distributions of their entries.

We test whether distributions p and q are different on the basis of samples drawn from each of them, by finding a well behaved (e.g., smooth) function which is large on the points drawn from p, and small (as negative as possible) on the points from q. We use as our test statistic the difference between the mean function values on the two samples; when this is large, the samples are likely from different distributions. We call this test statistic the Maximum Mean Discrepancy (MMD).

Clearly the quality of the MMD as a statistic depends on the class $\mathcal{F}$ of smooth functions that define it. On one hand, $\mathcal{F}$ must be "rich enough" so that the population MMD vanishes if and only if $p = q$. On the other hand, for the test to be consistent in power, $\mathcal{F}$ needs to be "restrictive" enough for the empirical estimate of the MMD to converge quickly to its expectation as the sample size increases. We will use the unit balls in characteristic reproducing kernel Hilbert spaces (Fukumizu et al., 2008; Sriperumbudur et al., 2010b) as our function classes, since these will be shown to satisfy both of the foregoing properties. We also review classical metrics on distributions, namely the Kolmogorov-Smirnov and Earth-Mover's distances, which are based on different function classes; collectively these are known as integral probability metrics (Müller, 1997). On a more practical note, the MMD has a reasonable computational cost, when compared with other two-sample tests: given m points sampled from p and n from q, the cost is $O(m+n)^2$ time. We also propose a test statistic with a computational cost of $O(m+n)$: the associated test can achieve a given Type II error at a lower overall computational cost than the quadratic-cost test, by looking at a larger volume of data.

We define three nonparametric statistical tests based on the MMD. The first two tests are distribution-free, meaning they make no assumptions regarding p and q, albeit at the expense of being conservative in detecting differences between the distributions. The third test is based on the asymptotic distribution of the MMD, and is in practice more sensitive to differences in distribution at small sample sizes. The present work synthesizes and expands on results of Gretton et al. (2007a,b) and Smola et al. (2007),[1] who in turn build on the earlier work of Borgwardt et al. (2006). Note that

[1] In particular, most of the proofs here were not provided by Gretton et al. (2007a), but in an accompanying technical report (Gretton et al., 2008a), which this document replaces.

the latter addresses only the third kind of test, and that the approach of Gretton et al. (2007a,b) is rigorous in its treatment of the asymptotic distribution of the test statistic under the null hypothesis.

We begin our presentation in Section 2 with a formal definition of the MMD. We review the notion of a characteristic RKHS, and establish that when $\mathcal{F}$ is a unit ball in a characteristic RKHS, then the population MMD is zero if and only if $p = q$. We further show that universal RKHSs in the sense of Steinwart (2001) are characteristic. In Section 3, we give an overview of hypothesis testing as it applies to the two-sample problem, and review alternative test statistics, including the L_2 distance between kernel density estimates (Anderson et al., 1994), which is the prior approach closest to our work. We present our first two hypothesis tests in Section 4, based on two different bounds on the deviation between the population and empirical MMD. We take a different approach in Section 5, where we use the asymptotic distribution of the empirical MMD estimate as the basis for a third test. When large volumes of data are available, the cost of computing the MMD (quadratic in the sample size) may be excessive: we therefore propose in Section 6 a modified version of the MMD statistic that has a linear cost in the number of samples, and an associated asymptotic test. In Section 7, we provide an overview of methods related to the MMD in the statistics and machine learning literature. We also review alternative function classes for which the MMD defines a metric on probability distributions. Finally, in Section 8, we demonstrate the performance of MMD-based two-sample tests on problems from neuroscience, bioinformatics, and attribute matching using the Hungarian marriage method. Our approach performs well on high dimensional data with low sample size; in addition, we are able to successfully distinguish distributions on graph data, for which ours is the first proposed test.

A Matlab implementation of the tests is at `www.gatsby.ucl.ac.uk/` $\sim$ `gretton/mmd/mmd.htm`.

2. The Maximum Mean Discrepancy

In this section, we present the maximum mean discrepancy (MMD), and describe conditions under which it is a metric on the space of probability distributions. The MMD is defined in terms of particular function spaces that witness the difference in distributions: we therefore begin in Section 2.1 by introducing the MMD for an arbitrary function space. In Section 2.2, we compute both the population MMD and two empirical estimates when the associated function space is a reproducing kernel Hilbert space, and in Section 2.3 we derive the RKHS function that witnesses the MMD for a given pair of distributions.

2.1 Definition of the Maximum Mean Discrepancy

Our goal is to formulate a statistical test that answers the following question:

Problem 1 *Let x and y be random variables defined on a topological space $\mathcal{X}$, with respective Borel probability measures p and q . Given observations $X := \{x_1, \ldots, x_m\}$ and $Y := \{y_1, \ldots, y_n\}$, independently and identically distributed (i.i.d.) from p and q, respectively, can we decide whether $p \neq q$?*

Where there is no ambiguity, we use the shorthand notation $\mathbf{E}_x[f(x)] := \mathbf{E}_{x\sim p}[f(x)]$ and $\mathbf{E}_y[f(y)] := \mathbf{E}_{y\sim q}[f(y)]$ to denote expectations with respect to p and q, respectively, where $x \sim p$ indicates x has distribution p. To start with, we wish to determine a criterion that, in the population setting, takes on a unique and distinctive value only when $p = q$. It will be defined based on Lemma 9.3.2 of Dudley (2002).

Lemma 1 *Let $(\mathcal{X},d)$ be a metric space, and let p,q be two Borel probability measures defined on $\mathcal{X}$. Then $p=q$ if and only if $\mathbf{E}_x(f(x))=\mathbf{E}_y(f(y))$ for all $f\in C(\mathcal{X})$, where $C(\mathcal{X})$ is the space of bounded continuous functions on $\mathcal{X}$.*

Although $C(\mathcal{X})$ in principle allows us to identify $p=q$ uniquely, it is not practical to work with such a rich function class in the finite sample setting. We thus define a more general class of statistic, for as yet unspecified function classes $\mathcal{F}$, to measure the disparity between p and q (Fortet and Mourier, 1953; Müller, 1997).

Definition 2 *Let $\mathcal{F}$ be a class of functions $f:\mathcal{X}\to\mathbb{R}$ and let p,q,x,y,X,Y be defined as above. We define the maximum mean discrepancy (MMD) as*

$$\mathrm{MMD}\left[\mathcal{F},p,q\right]:=\sup_{f\in\mathcal{F}}\left(\mathbf{E}_x[f(x)]-\mathbf{E}_y[f(y)]\right). \tag{1}$$

In the statistics literature, this is known as an integral probability metric (Müller, 1997). A biased[2] *empirical estimate of the MMD is obtained by replacing the population expectations with empirical expectations computed on the samples X and Y,*

$$\mathrm{MMD}_b\left[\mathcal{F},X,Y\right]:=\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m}f(x_i)-\frac{1}{n}\sum_{i=1}^{n}f(y_i)\right). \tag{2}$$

We must therefore identify a function class that is rich enough to uniquely identify whether $p=q$, yet restrictive enough to provide useful finite sample estimates (the latter property will be established in subsequent sections).

2.2 The MMD in Reproducing Kernel Hilbert Spaces

In the present section, we propose as our MMD function class $\mathcal{F}$ the unit ball in a reproducing kernel Hilbert space $\mathcal{H}$. We will provide finite sample estimates of this quantity (both biased and unbiased), and establish conditions under which the MMD can be used to distinguish between probability measures. Other possible function classes $\mathcal{F}$ are discussed in Sections 7.1 and 7.2.

We first review some properties of $\mathcal{H}$ (Schölkopf and Smola, 2002). Since $\mathcal{H}$ is an RKHS, the operator of evaluation δ_x mapping $f\in\mathcal{H}$ to $f(x)\in\mathbb{R}$ is continuous. Thus, by the Riesz representation theorem (Reed and Simon, 1980, Theorem II.4), there is a feature mapping $\phi(x)$ from $\mathcal{X}$ to $\mathbb{R}$ such that $f(x)=\langle f,\phi(x)\rangle_{\mathcal{H}}$. This feature mapping takes the canonical form $\phi(x)=k(x,\cdot)$ (Steinwart and Christmann, 2008, Lemma 4.19), where $k(x_1,x_2):\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is positive definite, and the notation $k(x,\cdot)$ indicates the kernel has one argument fixed at x, and the second free. Note in particular that $\langle\phi(x),\phi(y)\rangle_{\mathcal{H}}=k(x,y)$. We will generally use the more concise notation $\phi(x)$ for the feature mapping, although in some cases it will be clearer to write $k(x,\cdot)$.

We next extend the notion of feature map to the embedding of a probability distribution: we will define an element $\mu_p\in\mathcal{H}$ such that $\mathbf{E}_xf=\langle f,\mu_p\rangle_{\mathcal{H}}$ for all $f\in\mathcal{H}$, which we call the *mean embedding* of p. Embeddings of probability measures into reproducing kernel Hilbert spaces are well established in the statistics literature: see Berlinet and Thomas-Agnan (2004, Chapter 4) for further detail and references. We begin by establishing conditions under which the mean embedding μ_p exists (Fukumizu et al., 2004, p. 93), (Sriperumbudur et al., 2010b, Theorem 1).

[2] The empirical MMD defined below has an upward bias—we will define an unbiased statistic in the following section.

Lemma 3 *If $k(\cdot,\cdot)$ is measurable and $\mathbf{E}_x\sqrt{k(x,x)} < \infty$ then $\mu_p \in \mathcal{H}$.*

Proof The linear operator $T_p f := \mathbf{E}_x f$ for all $f \in \mathcal{F}$ is bounded under the assumption, since

$$|T_p f| = |\mathbf{E}_x f| \leq \mathbf{E}_x |f| = \mathbf{E}_x |\langle f, \phi(x)\rangle_{\mathcal{H}}| \leq \mathbf{E}_x \left(\sqrt{k(x,x)}\, \|f\|_{\mathcal{H}}\right).$$

Hence by the Riesz representer theorem, there exists a $\mu_p \in \mathcal{H}$ such that $T_p f = \langle f, \mu_p\rangle_{\mathcal{H}}$. If we set $f = \phi(t) = k(t,\cdot)$, we obtain $\mu_p(t) = \langle \mu_p, k(t,\cdot)\rangle_{\mathcal{H}} = \mathbf{E}_x k(t,x)$: in other words, the mean embedding of the distribution p is the expectation under p of the canonical feature map. ■

We next show that the MMD may be expressed as the distance in $\mathcal{H}$ between mean embeddings (Borgwardt et al., 2006).

Lemma 4 *Assume the condition in Lemma 3 for the existence of the mean embeddings μ_p, μ_q is satisfied. Then*

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \left\|\mu_p - \mu_q\right\|^2_{\mathcal{H}}.$$

Proof

$$\begin{aligned}
\mathrm{MMD}^2[\mathcal{F}, p, q] &= \left[\sup_{\|f\|_{\mathcal{H}}\leq 1} \left(\mathbf{E}_x[f(x)] - \mathbf{E}_y[f(y)]\right)\right]^2 \\
&= \left[\sup_{\|f\|_{\mathcal{H}}\leq 1} \left\langle \mu_p - \mu_q, f\right\rangle_{\mathcal{H}}\right]^2 \\
&= \left\|\mu_p - \mu_q\right\|^2_{\mathcal{H}}.
\end{aligned}$$

■

We now establish a condition on the RKHS $\mathcal{H}$ under which the mean embedding μ_p is injective, which indicates that $\mathrm{MMD}[\mathcal{F}, p, q] = 0$ is a metric[3] on the Borel probability measures on $\mathcal{X}$. Evidently, this property will not hold for all $\mathcal{H}$: for instance, a polynomial RKHS of degree two cannot distinguish between distributions with the same mean and variance, but different kurtosis (Sriperumbudur et al., 2010b, Example 3). The MMD is a metric, however, when $\mathcal{H}$ is a *universal* RKHSs, defined on a compact metric space $\mathcal{X}$. Universality requires that $k(\cdot,\cdot)$ be continuous, and $\mathcal{H}$ be dense in $C(\mathcal{X})$ with respect to the L_∞ norm. Steinwart (2001) proves that the Gaussian and Laplace RKHSs are universal.

Theorem 5 *Let $\mathcal{F}$ be a unit ball in a universal RKHS $\mathcal{H}$, defined on the compact metric space $\mathcal{X}$, with associated continuous kernel $k(\cdot,\cdot)$. Then* $\mathrm{MMD}[\mathcal{F}, p, q] = 0$ *if and only if $p = q$.*

Proof The proof follows Cortes et al. (2008, Supplementary Appendix), whose approach is clearer than the original proof of Gretton et al. (2008a, p. 4).[4] First, it is clear that $p = q$ implies

[3] According to Dudley (2002, p. 26) a metric $d(x,y)$ satisfies the following four properties: symmetry, triangle inequality, $d(x,x) = 0$, and $d(x,y) = 0 \implies x = y$. A pseudo-metric only satisfies the first three properties.

[4] Note that the proof of Cortes et al. (2008) requires an application the of dominated convergence theorem, rather than using the Riesz representation theorem to show the existence of the mean embeddings μ_p and μ_q as we did in Lemma 3.

$\mathrm{MMD}\{\mathcal{F},p,q\}$ is zero. We now prove the converse. By the universality of $\mathcal{H}$, for any given $\varepsilon>0$ and $f\in C(\mathcal{X})$ there exists a $g\in\mathcal{H}$ such that

$$\|f-g\|_\infty\leq\varepsilon.$$

We next make the expansion

$$|\mathbf{E}_x f(x)-\mathbf{E}_y(f(y))|\leq|\mathbf{E}_x f(x)-\mathbf{E}_x g(x)|+|\mathbf{E}_x g(x)-\mathbf{E}_y g(y)|+|\mathbf{E}_y g(y)-\mathbf{E}_y f(y)|\,.$$

The first and third terms satisfy

$$|\mathbf{E}_x f(x)-\mathbf{E}_x g(x)|\leq\mathbf{E}_x\,|f(x)-g(x)|\leq\varepsilon.$$

Next, write

$$\mathbf{E}_x g(x)-\mathbf{E}_y g(y)=\left\langle g,\mu_p-\mu_q\right\rangle_{\mathcal{H}}=0,$$

since $\mathrm{MMD}\{\mathcal{F},p,q\}=0$ implies $\mu_p=\mu_q$. Hence

$$|\mathbf{E}_x f(x)-\mathbf{E}_y(f(y))|\leq2\varepsilon$$

for all $f\in C(\mathcal{X})$ and $\varepsilon>0$, which implies $p=q$ by Lemma 1. ■

While our result establishes the mapping μ_p is injective for universal kernels on compact domains, this result can also be shown in more general cases. Fukumizu et al. (2008) introduces the notion of *characteristic kernels*, these being kernels for which the mean map is injective. Fukumizu et al. establish that Gaussian and Laplace kernels are characteristic on $\mathbb{R}^d$, and thus that the associated MMD is a metric on distributions for this domain. Sriperumbudur et al. (2008, 2010b) and Sriperumbudur et al. (2011a) further explore the properties of characteristic kernels, providing a simple condition to determine whether translation invariant kernels are characteristic, and investigating the relation between universal and characteristic kernels on non-compact domains.

Given we are in an RKHS, we may easily obtain of the squared MMD, $\left\|\mu_p-\mu_q\right\|^2_{\mathcal{H}}$, in terms of kernel functions, and a corresponding unbiased finite sample estimate.

Lemma 6 *Given x and x' independent random variables with distribution p, and y and y' independent random variables with distribution q, the squared population* MMD *is*

$$\mathrm{MMD}^2\left[\mathcal{F},p,q\right]=\mathbf{E}_{x,x'}\left[k(x,x')\right]-2\mathbf{E}_{x,y}\left[k(x,y)\right]+\mathbf{E}_{y,y'}\left[k(y,y')\right],$$

where x' is an independent copy of x with the same distribution, and y' is an independent copy of y. An unbiased *empirical estimate is a sum of two U-statistics and a sample average,*

$$\begin{aligned}\mathrm{MMD}_u^2[\mathcal{F},X,Y] &= \frac{1}{m(m-1)}\sum_{i=1}^m\sum_{j\neq i}^m k(x_i,x_j)+\frac{1}{n(n-1)}\sum_{i=1}^n\sum_{j\neq i}^n k(y_i,y_j)\\ &\quad -\frac{2}{mn}\sum_{i=1}^m\sum_{j=1}^n k(x_i,y_j).\end{aligned}\tag{3}$$

When $m=n$, a slightly simpler empirical estimate may be used. Let $Z:=(z_1,\ldots,z_m)$ be m i.i.d. random variables, where $z:=(x,y)\sim p\times q$ (i.e., x and y are independent). An unbiased estimate of MMD^2 *is*

$$\mathrm{MMD}_u^2\left[\mathcal{F},X,Y\right]=\frac{1}{(m)(m-1)}\sum_{i\neq j}^m h(z_i,z_j),\tag{4}$$

which is a one-sample U-statistic with

$$h(z_i,z_j) := k(x_i,x_j)+k(y_i,y_j)-k(x_i,y_j)-k(x_j,y_i).$$

Proof Starting from the expression for $\mathrm{MMD}^2[\mathcal{F},p,q]$ in Lemma 4,

$$\begin{aligned}
\mathrm{MMD}^2[\mathcal{F},p,q] &= \left\|\mu_p-\mu_q\right\|^2_{\mathcal{H}} \\
&= \langle \mu_p,\mu_p\rangle_{\mathcal{H}} + \langle \mu_q,\mu_q\rangle_{\mathcal{H}} - 2\langle \mu_p,\mu_q\rangle_{\mathcal{H}} \\
&= \mathbf{E}_{x,x'}\left\langle \phi(x),\phi(x')\right\rangle_{\mathcal{H}} + \mathbf{E}_{y,y'}\left\langle \phi(y),\phi(y')\right\rangle_{\mathcal{H}} - 2\mathbf{E}_{x,y}\left\langle \phi(x),\phi(y)\right\rangle_{\mathcal{H}},
\end{aligned}$$

The proof is completed by applying $\langle \phi(x),\phi(x')\rangle_{\mathcal{H}} = k(x,x')$; the empirical estimates follow straightforwardly, by replacing the population expectations with their corresponding U-statistics and sample averages. This statistic is unbiased following Serfling (1980, Chapter 5). ■

Note that MMD_u^2 may be negative, since it is an unbiased estimator of $(\mathrm{MMD}[\mathcal{F},p,q])^2$. The only terms missing to ensure nonnegativity, however, are $h(z_i,z_i)$, which were removed to remove spurious correlations between observations. Consequently we have the bound

$$\mathrm{MMD}_u^2 + \frac{1}{m(m-1)}\sum_{i=1}^m k(x_i,x_i)+k(y_i,y_i)-2k(x_i,y_i) \geq 0.$$

Moreover, while the empirical statistic for $m=n$ is an unbiased estimate of MMD^2, it does not have minimum variance, since we ignore the cross-terms $k(x_i,y_i)$, of which there are $O(n)$. From (3), however, we see the minimum variance estimate is almost identical (Serfling, 1980, Section 5.1.4).

The biased statistic in (2) may also be easily computed following the above reasoning. Substituting the empirical estimates $\mu_X := \frac{1}{m}\sum_{i=1}^m \phi(x_i)$ and $\mu_Y := \frac{1}{n}\sum_{i=1}^n \phi(y_i)$ of the feature space means based on respective samples X and Y, we obtain

$$\mathrm{MMD}_b\left[\mathcal{F},X,Y\right] = \left[\frac{1}{m^2}\sum_{i,j=1}^m k(x_i,x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i,y_j) + \frac{1}{n^2}\sum_{i,j=1}^n k(y_i,y_j)\right]^{\frac{1}{2}}. \qquad (5)$$

Note that the U-statistics of (3) have been replaced by V-statistics. Intuitively we expect the empirical test statistic $\mathrm{MMD}[\mathcal{F},X,Y]$, whether biased or unbiased, to be small if $p=q$, and large if the distributions are far apart. It costs $O((m+n)^2)$ time to compute both statistics.

2.3 Witness Function of the MMD for RKHSs

We define the witness function f^* to be the RKHS function attaining the supremum in (1), and its empirical estimate $\hat{f}^*$ to be the function attaining the supremum in (2). From the reasoning in Lemma 4, it is clear that

$$\begin{aligned}
f^*(t) &\propto \left\langle \phi(t),\mu_p-\mu_q\right\rangle_{\mathcal{H}} &= \mathbf{E}_x\left[k(x,t)\right]-\mathbf{E}_y\left[k(y,t)\right], \\
\hat{f}^*(t) &\propto \left\langle \phi(t),\mu_X-\mu_Y\right\rangle_{\mathcal{H}} &= \tfrac{1}{m}\textstyle\sum_{i=1}^m k(x_i,t)-\tfrac{1}{n}\sum_{i=1}^n k(y_i,t).
\end{aligned}$$

where we have defined $\mu_X = m^{-1}\sum_{i=1}^m \phi(x_i)$, and μ_Y by analogy. The result follows since the unit vector v maximizing $\langle v,x\rangle_{\mathcal{H}}$ in a Hilbert space is $v = x/\left\|x\right\|_{\mathcal{H}}$.

We illustrate the behavior of MMD in Figure 1 using a one-dimensional example. The data X and Y were generated from distributions p and q with equal means and variances, with p Gaussian

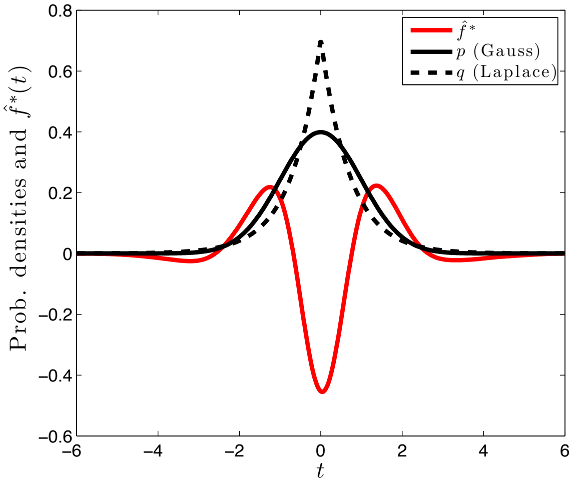


Figure 1: Illustration of the function maximizing the mean discrepancy in the case where a Gaussian is being compared with a Laplace distribution. Both distributions have zero mean and unit variance. The function $\hat{f}^*$ that witnesses the MMD has been scaled for plotting purposes, and was computed empirically on the basis of 2×10^4 samples, using a Gaussian kernel with $\sigma = 0.5$.

and q Laplacian. We chose $\mathcal{F}$ to be the unit ball in a Gaussian RKHS. The empirical estimate $\hat{f}^*$ of the function f^* that witnesses the MMD—in other words, the function maximizing the mean discrepancy in (1)—is smooth, negative where the Laplace density exceeds the Gaussian density (at the center and tails), and positive where the Gaussian density is larger. The magnitude of $\hat{f}^*$ is a direct reflection of the amount by which one density exceeds the other, insofar as the smoothness constraint permits it.

3. Background Material

We now present three background results. First, we introduce the terminology used in statistical hypothesis testing. Second, we demonstrate via an example that even for tests which have asymptotically no error, we cannot guarantee performance at any fixed sample size without making assumptions about the distributions. Third, we review some alternative statistics used in comparing distributions, and the associated two-sample tests (see also Section 7 for an overview of additional integral probability metrics).

3.1 Statistical Hypothesis Testing

Having described a metric on probability distributions (the MMD) based on distances between their Hilbert space embeddings, and empirical estimates (biased and unbiased) of this metric, we address the problem of determining whether the empirical MMD shows a *statistically significant* difference between distributions. To this end, we briefly describe the framework of statistical hypothesis testing as it applies in the present context, following Casella and Berger (2002, Chapter 8). Given i.i.d.

samples $X \sim p$ of size m and $Y \sim q$ of size n, the statistical test, $\mathcal{T}(X,Y) : \mathcal{X}^m \times \mathcal{X}^n \mapsto \{0,1\}$ is used to distinguish between the null hypothesis $\mathcal{H}_0 : p = q$ and the alternative hypothesis $\mathcal{H}_A : p \neq q$. This is achieved by comparing the test statistic[5] $\mathrm{MMD}[\mathcal{F},X,Y]$ with a particular threshold: if the threshold is exceeded, then the test rejects the null hypothesis (bearing in mind that a zero population MMD indicates $p = q$). The acceptance region of the test is thus defined as the set of real numbers below the threshold. Since the test is based on finite samples, it is possible that an incorrect answer will be returned. A Type I error is made when $p = q$ is rejected based on the observed samples, despite the null hypothesis having generated the data. Conversely, a Type II error occurs when $p = q$ is accepted despite the underlying distributions being different. The *level* α of a test is an upper bound on the probability of a Type I error: this is a design parameter of the test which must be set in advance, and is used to determine the threshold to which we compare the test statistic (finding the test threshold for a given α is the topic of Sections 4 and 5). The *power* of a test against a particular member of the alternative class $\mathcal{H}_A$ (i.e., a specific (p,q) such that $p \neq q$) is the probability of wrongly accepting $p = q$ in this instance. A consistent test achieves a level α, and a Type II error of zero, in the large sample limit. We will see that the tests proposed in this paper are consistent.

3.2 A Negative Result

Even if a test is consistent, it is not possible to distinguish distributions with high probability at a given, *fixed* sample size (i.e., to provide guarantees on the Type II error), without prior assumptions as to the nature of the difference between p and q. This is true regardless of the two-sample test used. There are several ways to illustrate this, which each give insight into the kinds of differences that might be undetectable for a given number of samples. The following example[6] is one such illustration.

Example 1 *Assume we have a distribution p from which we have drawn m i.i.d. observations. We construct a distribution q by drawing m^2 i.i.d. observations from p, and defining a discrete distribution over these m^2 instances with probability m^{-2} each. It is easy to check that if we now draw m observations from q, there is at least a $\binom{m^2}{m}\frac{m!}{m^{2m}} > 1 - e^{-1} > 0.63$ probability that we thereby obtain an m sample from p. Hence no test will be able to distinguish samples from p and q in this case. We could make the probability of detection arbitrarily small by increasing the size of the sample from which we construct q.*

3.3 Previous Work

We next give a brief overview of some earlier approaches to the two sample problem for multivariate data. Since our later experimental comparison is with respect to certain of these methods, we give abbreviated algorithm names in italics where appropriate: these should be used as a key to the tables in Section 8.

[5] This may be biased or unbiased.

[6] This is a variation of a construction for independence tests, which was suggested in a private communication by John Langford.

3.3.1 L_2 DISTANCE BETWEEN PARZEN WINDOW ESTIMATES

The prior work closest to the current approach is the Parzen window-based statistic of Anderson et al. (1994). We begin with a short overview of the Parzen window estimate and its properties (Silverman, 1986), before proceeding to a comparison with the RKHS approach. We assume a distribution p on $\mathbb{R}^d$, which has an associated density function f_p. The Parzen window estimate of this density from an i.i.d. sample X of size m is

$$\hat{f}_p(x) = \frac{1}{m}\sum_{i=1}^{m} \kappa\left(x_i - x\right), \text{ where } \kappa \text{ satisfies } \int_{\mathcal{X}} \kappa\left(x\right) dx = 1 \text{ and } \kappa\left(x\right) \geq 0.$$

We may rescale κ according to $\frac{1}{h_m^d}\kappa\left(\frac{x}{h_m}\right)$ for a bandwidth parameter h_m. To simplify the discussion, we use a single bandwidth h_{m+n} for both $\hat{f}_p$ and $\hat{f}_q$. Assuming m/n is bounded away from zero and infinity, consistency of the Parzen window estimates for f_p and f_q requires

$$\lim_{m,n\to\infty} h_{m+n}^d = 0 \quad \text{and} \quad \lim_{m,n\to\infty} (m+n)h_{m+n}^d = \infty. \tag{6}$$

We now show the L_2 distance between Parzen windows density estimates is a special case of the biased MMD in Equation (5). Denote by $D_r(p,q) := \left\|f_p - f_q\right\|_r$ the L_r distance between the densities f_p and f_q corresponding to the distributions p and q, respectively. For $r = 1$ the distance $D_r(p,q)$ is known as the Lévy distance (Feller, 1971), and for $r = 2$ we encounter a distance measure derived from the Renyi entropy (Gokcay and Principe, 2002). Assume that $\hat{f}_p$ and $\hat{f}_q$ are given as kernel density estimates with kernel $\kappa(x - x')$, that is, $\hat{f}_p(x) = m^{-1}\sum_{i=1}^{m}\kappa(x_i - x)$ and $\hat{f}_q(y)$ is defined by analogy. In this case

$$\begin{aligned} D_2(\hat{f}_p, \hat{f}_q)^2 &= \int \left[\frac{1}{m}\sum_{i=1}^{m}\kappa(x_i - z) - \frac{1}{n}\sum_{i=1}^{n}\kappa(y_i - z)\right]^2 dz \\ &= \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i - x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i - y_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i - y_j), \end{aligned}$$

where $k(x-y) = \int \kappa(x-z)\kappa(y-z)dz$. By its definition $k(x-y)$ is an RKHS kernel, as it is an inner product between $\kappa(x-z)$ and $\kappa(y-z)$ on the domain $\mathcal{X}$.

We now describe the asymptotic performance of a two-sample test using the statistic $D_2(\hat{f}_p, \hat{f}_q)^2$. We consider the power of the test under local departures from the null hypothesis. Anderson et al. (1994) define these to take the form

$$f_q = f_p + \delta g, \tag{7}$$

where $\delta \in \mathbb{R}$, and g is a fixed, bounded, integrable function chosen to ensure that f_q is a valid density for sufficiently small $|\delta|$. Anderson et al. consider two cases: the kernel bandwidth converging to zero with increasing sample size, ensuring consistency of the Parzen window estimates of f_p and f_q; and the case of a fixed bandwidth. In the former case, the minimum distance with which the test can discriminate f_p from f_q is[7] $\delta = (m+n)^{-1/2}h_{m+n}^{-d/2}$. In the latter case, this minimum distance is $\delta = (m+n)^{-1/2}$, under the assumption that the Fourier transform of the kernel κ does not vanish

[7] Formally, define s_α as a threshold for the statistic $D_2\left(\hat{f}_p, \hat{f}_q\right)^2$, chosen to ensure the test has level α, and let $\delta = (m+n)^{-1/2}h_{m+n}^{-d/2}c$ for some fixed $c \neq 0$. When $m,n \to \infty$ such that m/n is bounded away from 0 and ∞, and

on an interval (Anderson et al., 1994, Section 2.4), which implies the kernel k is characteristic (Sriperumbudur et al., 2010b). The power of the L_2 test against local alternatives is greater when the kernel is held fixed, since for *any* rate of decrease of h_{m+n} with increasing sample size, δ will decrease more slowly than for a fixed kernel.

An RKHS-based approach generalizes the L_2 statistic in a number of important respects. First, we may employ a much larger class of characteristic kernels that cannot be written as inner products between Parzen windows: several examples are given by Steinwart (2001, Section 3) and Micchelli et al. (2006, Section 3) (these kernels are universal, hence characteristic). We may further generalize to kernels on structured objects such as strings and graphs (Schölkopf et al., 2004), as done in our experiments (Section 8). Second, even when the kernel may be written as an inner product of Parzen windows on $\mathbb{R}^d$, the D_2^2 statistic with fixed bandwidth no longer converges to an L_2 distance between probability density functions, hence it is more natural to define the statistic as an integral probability metric for a particular RKHS, as in Definition 2. Indeed, in our experiments, we obtain good performance in experimental settings where the dimensionality greatly exceeds the sample size, and density estimates would perform very poorly[8] (for instance the Gaussian toy example in Figure 5B, for which performance actually improves when the dimensionality increases; and the microarray data sets in Table 1). This suggests it is not necessary to solve the more difficult problem of density estimation in high dimensions to do two-sample testing.

Finally, the kernel approach leads us to establish consistency against a larger class of local alternatives to the null hypothesis than that considered by Anderson et al. In Theorem 13, we prove consistency against a class of alternatives encoded in terms of the mean embeddings of p and q, which applies to any domain on which RKHS kernels may be defined, and not only densities on $\mathbb{R}^d$. This more general approach also has interesting consequences for distributions on $\mathbb{R}^d$: for instance, a local departure from $\mathcal{H}_0$ occurs when p and q differ at increasing frequencies in their respective characteristic functions. This class of local alternatives cannot be expressed in the form δg for fixed g, as in (7). We discuss this issue further in Section 5.

3.3.2 MMD FOR MULTINOMIALS

Assume a finite domain $\mathcal{X} := \{1, \ldots, d\}$, and define the random variables x and y on $\mathcal{X}$ such that $p_i := P(x = i)$ and $q_j := P(y = j)$. We embed x into an RKHS $\mathcal{H}$ via the feature mapping $\phi(x) := e_x$, where e_s is the unit vector in $\mathbb{R}^d$ taking value 1 in dimension s, and zero in the remaining entries. The kernel is the usual inner product on $\mathbb{R}^d$. In this case,

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \|p - q\|^2_{\mathbb{R}^d} = \sum_{i=1}^{d} (p_i - q_i)^2 . \tag{8}$$

Harchaoui et al. (2008, Section 1, long version) note that this L_2 statistic may not be the best choice for finite domains, citing a result of Lehmann and Romano (2005, Theorem 14.3.2) that Pearson's

assuming conditions (6), the limit

$$\pi(c) := \lim_{(m+n)\to\infty} \Pr_{\mathcal{H}_A}\left(D_2\left(\hat{f}_p, \hat{f}_q\right)^2 > s_\alpha\right)$$

is well-defined, and satisfies $\alpha < \pi(c) < 1$ for $0 < |c| < \infty$, and $\pi(c) \to 1$ as $c \to \infty$.

[8] The L_2 error of a kernel density estimate converges as $O(n^{-4/(4+d)})$ when the optimal bandwidth is used (Wasserman, 2006, Section 6.5).

Chi-squared statistic is optimal for the problem of goodness of fit testing for multinomials.[9] It would be of interest to establish whether an analogous result holds for two-sample testing in a wider class of RKHS feature spaces.

3.3.3 FURTHER MULTIVARIATE TWO-SAMPLE TESTS

Biau and Gyorfi (2005) *(Biau)* use as their test statistic the L_1 distance between discretized estimates of the probabilities, where the partitioning is refined as the sample size increases. This space partitioning approach becomes difficult or impossible for high dimensional problems, since there are too few points per bin. For this reason, we use this test only for low-dimensional problems in our experiments.

A generalisation of the Wald-Wolfowitz runs test to the multivariate domain was proposed and analysed by Friedman and Rafsky (1979) and Henze and Penrose (1999) *(FR Wolf)*, and involves counting the number of edges in the minimum spanning tree over the aggregated data that connect points in X to points in Y. The resulting test relies on the asymptotic normality of the test statistic, and is not distribution-free under the null hypothesis for finite samples (the test threshold depends on p, as with our asymptotic test in Section 5; by contrast, our tests in Section 4 are distribution-free). The computational cost of this method using Kruskal's algorithm is $O((m+n)^2 \log(m+n))$, although more modern methods improve on the $\log(m+n)$ term: see Chazelle (2000) for details. Friedman and Rafsky (1979) claim that calculating the matrix of distances, which costs $O((m+n)^2)$, dominates their computing time; we return to this point in our experiments (Section 8). Two possible generalisations of the Kolmogorov-Smirnov test to the multivariate case were studied by Bickel (1969) and Friedman and Rafsky (1979). The approach of Friedman and Rafsky *(FR Smirnov)* in this case again requires a minimal spanning tree, and has a similar cost to their multivariate runs test.

A more recent multivariate test was introduced by Rosenbaum (2005). This entails computing the minimum distance non-bipartite matching over the aggregate data, and using the number of pairs containing a sample from both X and Y as a test statistic. The resulting statistic is distribution-free under the null hypothesis at finite sample sizes, in which respect it is superior to the Friedman-Rafsky test; on the other hand, it costs $O((m+n)^3)$ to compute. Another distribution-free test *(Hall)* was proposed by Hall and Tajvidi (2002): for each point from p, it requires computing the closest points in the aggregated data, and counting how many of these are from q (the procedure is repeated for each point from q with respect to points from p). As we shall see in our experimental comparisons, the test statistic is costly to compute; Hall and Tajvidi consider only tens of points in their experiments.

4. Tests Based on Uniform Convergence Bounds

In this section, we introduce two tests for the two-sample problem that have exact performance guarantees at finite sample sizes, based on uniform convergence bounds. The first, in Section 4.1, uses the McDiarmid (1989) bound on the biased MMD statistic, and the second, in Section 4.2, uses a Hoeffding (1963) bound for the unbiased statistic.

[9] A goodness of fit test determines whether a sample from p is drawn from a *known* target multinomial q. Pearson's Chi-squared statistic weights each term in the sum (8) by its corresponding q_i^{-1}.

4.1 Bound on the Biased Statistic and Test

We establish two properties of the MMD, from which we derive a hypothesis test. First, we show that regardless of whether or not $p = q$, the empirical MMD converges in probability at rate $O((m+n)^{-\frac{1}{2}})$ to its population value. This shows the consistency of statistical tests based on the MMD. Second, we give probabilistic bounds for large deviations of the empirical MMD in the case $p = q$. These bounds lead directly to a threshold for our first hypothesis test. We begin by establishing the convergence of $\mathrm{MMD}_b[\mathcal{F},X,Y]$ to $\mathrm{MMD}[\mathcal{F},p,q]$. The following theorem is proved in A.2.

Theorem 7 *Let p,q,X,Y be defined as in Problem 1, and assume $0 \le k(x,y) \le K$. Then*

$$\Pr_{X,Y}\left\{|\mathrm{MMD}_b[\mathcal{F},X,Y] - \mathrm{MMD}[\mathcal{F},p,q]| > 2\left((K/m)^{\frac{1}{2}} + (K/n)^{\frac{1}{2}}\right) + \varepsilon\right\} \le 2\exp\left(\tfrac{-\varepsilon^2 mn}{2K(m+n)}\right),$$

where $\Pr_{X,Y}$ denotes the probability over the m-sample X and n-sample Y.

Our next goal is to refine this result in a way that allows us to define a test threshold under the null hypothesis $p = q$. Under this circumstance, the constants in the exponent are slightly improved. The following theorem is proved in Appendix A.3.

Theorem 8 *Under the conditions of Theorem 7 where additionally $p = q$ and $m = n$,*

$$\mathrm{MMD}_b[\mathcal{F},X,Y] \le \underbrace{m^{-\frac{1}{2}}\sqrt{2\mathbf{E}_{x,x'}\left[k(x,x) - k(x,x')\right]}}_{B_1(\mathcal{F},p)} + \varepsilon \le \underbrace{(2K/m)^{1/2}}_{B_2(\mathcal{F},p)} + \varepsilon,$$

both with probability at least $1 - \exp\left(-\frac{\varepsilon^2 m}{4K}\right)$.

In this theorem, we illustrate two possible bounds $B_1(\mathcal{F},p)$ and $B_2(\mathcal{F},p)$ on the bias in the empirical estimate (5). The first inequality is interesting inasmuch as it provides a link between the bias bound $B_1(\mathcal{F},p)$ and kernel size (for instance, if we were to use a Gaussian kernel with large σ, then $k(x,x)$ and $k(x,x')$ would likely be close, and the bias small). In the context of testing, however, we would need to provide an additional bound to show convergence of an empirical estimate of $B_1(\mathcal{F},p)$ to its population equivalent. Thus, in the following test for $p = q$ based on Theorem 8, we use $B_2(\mathcal{F},p)$ to bound the bias.[10]

Corollary 9 *A hypothesis test of level α for the null hypothesis $p = q$, that is, for $\mathrm{MMD}[\mathcal{F},p,q] = 0$, has the acceptance region $\mathrm{MMD}_b[\mathcal{F},X,Y] < \sqrt{2K/m}\left(1 + \sqrt{2\log\alpha^{-1}}\right)$.*

We emphasize that this test is distribution-free: the test threshold does not depend on the particular distribution that generated the sample. Theorem 7 guarantees the consistency of the test against fixed alternatives, and that the Type II error probability decreases to zero at rate $O\left(m^{-1/2}\right)$, assuming $m = n$. To put this convergence rate in perspective, consider a test of whether two normal distributions have equal means, given they have unknown but equal variance (Casella and Berger, 2002, Exercise 8.41). In this case, the test statistic has a Student-t distribution with $n+m-2$ degrees of freedom, and its Type II error probability converges at the same rate as our test.

It is worth noting that bounds may be obtained for the deviation between population mean embeddings μ_p and the empirical embeddings μ_X in a completely analogous fashion. The proof

[10] Note that we use a tighter bias bound than Gretton et al. (2007a).

requires symmetrization by means of a *ghost sample*, that is, a second set of observations drawn from the same distribution. While not the focus of the present paper, such bounds can be used to perform inference based on moment matching (Altun and Smola, 2006; Dudík and Schapire, 2006; Dudík et al., 2004).

4.2 Bound on the Unbiased Statistic and Test

The previous bounds are of interest since the proof strategy can be used for general function classes with well behaved Rademacher averages (see Sriperumbudur et al., 2010a). When $\mathcal{F}$ is the unit ball in an RKHS, however, we may very easily define a test via a convergence bound on the unbiased statistic MMD_u^2 in Lemma 4. We base our test on the following theorem, which is a straightforward application of the large deviation bound on U-statistics of Hoeffding (1963, p. 25).

Theorem 10 *Assume* $0 \leq k(x_i, x_j) \leq K$, *from which it follows* $-2K \leq h(z_i, z_j) \leq 2K$. *Then*

$$\mathrm{Pr}_{X,Y}\left\{\mathrm{MMD}_u^2(\mathcal{F}, X, Y) - \mathrm{MMD}^2(\mathcal{F}, p, q) > t\right\} \leq \exp\left(\frac{-t^2 m_2}{8K^2}\right)$$

where $m_2 := \lfloor m/2 \rfloor$ *(the same bound applies for deviations of* $-t$ *and below).*

A consistent statistical test for $p = q$ using MMD_u^2 is then obtained.

Corollary 11 *A hypothesis test of level* α *for the null hypothesis* $p = q$ *has the acceptance region* $\mathrm{MMD}_u^2 < (4K/\sqrt{m})\sqrt{\log(\alpha^{-1})}$.

This test is distribution-free. We now compare the thresholds of the above test with that in Corollary 9. We note first that the threshold for the biased statistic applies to an estimate of MMD, whereas that for the unbiased statistic is for an estimate of MMD^2. Squaring the former threshold to make the two quantities comparable, the squared threshold in Corollary 9 decreases as m^{-1}, whereas the threshold in Corollary 11 decreases as $m^{-1/2}$. Thus for sufficiently large[11] m, the McDiarmid-based threshold will be lower (and the associated test statistic is in any case biased upwards), and its Type II error will be better for a given Type I bound. This is confirmed in our Section 8 experiments. Note, however, that the rate of convergence of the squared, biased MMD estimate to its population value remains at $1/\sqrt{m}$ (bearing in mind we take the square of a biased estimate, where the bias term decays as $1/\sqrt{m}$).

Finally, we note that the bounds we obtained in this section and the last are rather conservative for a number of reasons: first, they do not take the actual distributions into account. In fact, they are finite sample size, distribution-free bounds that hold even in the worst case scenario. The bounds could be tightened using localization, moments of the distribution, etc.: see, for example, Bousquet et al. (2005) and de la Peña and Giné (1999). Any such improvements could be plugged straight into Theorem 19. Second, in computing bounds rather than trying to characterize the distribution of $\mathrm{MMD}(\mathcal{F}, X, Y)$ explicitly, we force our test to be conservative by design. In the following we aim for an exact characterization of the asymptotic distribution of $\mathrm{MMD}(\mathcal{F}, X, Y)$ instead of a bound. While this will not satisfy the uniform convergence requirements, it leads to superior tests in practice.

[11] In the case of $\alpha = 0.05$, this is $m \geq 12$.

5. Test Based on the Asymptotic Distribution of the Unbiased Statistic

We propose a third test, which is based on the asymptotic distribution of the unbiased estimate of MMD^2 in Lemma 6. This test uses the asymptotic distribution of MMD_u^2 under $\mathcal{H}_0$, which follows from results of Anderson et al. (1994, Appendix) and Serfling (1980, Section 5.5.2): see Appendix B.1 for the proof.

Theorem 12 *Let $\tilde{k}(x_i,x_j)$ be the kernel between feature space mappings from which the mean embedding of p has been subtracted,*

$$\begin{aligned} \tilde{k}(x_i,x_j) &:= \left\langle \phi(x_i)-\mu_p, \phi(x_j)-\mu_p \right\rangle_{\mathcal{H}} \\ &= k(x_i,x_j) - \mathbf{E}_x k(x_i,x) - \mathbf{E}_x k(x,x_j) + \mathbf{E}_{x,x'} k(x,x'), \end{aligned} \tag{9}$$

where x' is an independent copy of x drawn from p. Assume $\tilde{k} \in L_2(\mathcal{X}\times\mathcal{X}, p\times p)$ (i.e., the centred kernel is square integrable, which is true for all p when the kernel is bounded), and that for $t = m+n$, $\lim_{m,n\to\infty} m/t \to \rho_x$ and $\lim_{m,n\to\infty} n/t \to \rho_y := (1-\rho_x)$ for fixed $0<\rho_x<1$. Then under $\mathcal{H}_0$, MMD_u^2 converges in distribution according to

$$t\mathrm{MMD}_u^2[\mathcal{F},X,Y] \underset{D}{\to} \sum_{l=1}^{\infty} \lambda_l \left[(\rho_x^{-1/2} a_l - \rho_y^{-1/2} b_l)^2 - (\rho_x\rho_y)^{-1} \right], \tag{10}$$

where $a_l \sim \mathcal{N}(0,1)$ and $b_l \sim \mathcal{N}(0,1)$ are infinite sequences of independent Gaussian random variables, and the λ_i are eigenvalues of

$$\int_{\mathcal{X}} \tilde{k}(x,x')\psi_i(x)dp(x) = \lambda_i \psi_i(x').$$

We illustrate the MMD density under both the null and alternative hypotheses by approximating it empirically for $p=q$ and $p \neq q$. Results are plotted in Figure 2.

Our goal is to determine whether the empirical test statistic MMD_u^2 is so large as to be outside the $1-\alpha$ quantile of the null distribution in (10), which gives a level α test. Consistency of this test against local departures from the null hypothesis is provided by the following theorem, proved in Appendix B.2.

Theorem 13 *Define ρ_x, ρ_y, and t as in Theorem 12, and write $\mu_q = \mu_p + g_t$, where $g_t \in \mathcal{H}$ is chosen such that $\mu_p + g_t$ remains a valid mean embedding, and $\|g_t\|_{\mathcal{H}}$ is made to approach zero as $t\to\infty$ to describe local departures from the null hypothesis. Then $\|g_t\|_{\mathcal{H}} = ct^{-1/2}$ is the minimum distance between μ_p and μ_q distinguishable by the test.*

An example of a local departure from the null hypothesis is described earlier in the discussion of the L_2 distance between Parzen window estimates (Section 3.3.1). The class of local alternatives considered in Theorem 13 is more general, however: for instance, Sriperumbudur et al. (2010b, Section 4) and Harchaoui et al. (2008, Section 5, long version) give examples of classes of perturbations g_t with decreasing RKHS norm. These perturbations have the property that p differs from q at increasing frequencies, rather than simply with decreasing amplitude.

One way to estimate the $1-\alpha$ quantile of the null distribution is using the bootstrap on the aggregated data, following Arcones and Giné (1992). Alternatively, we may approximate the null

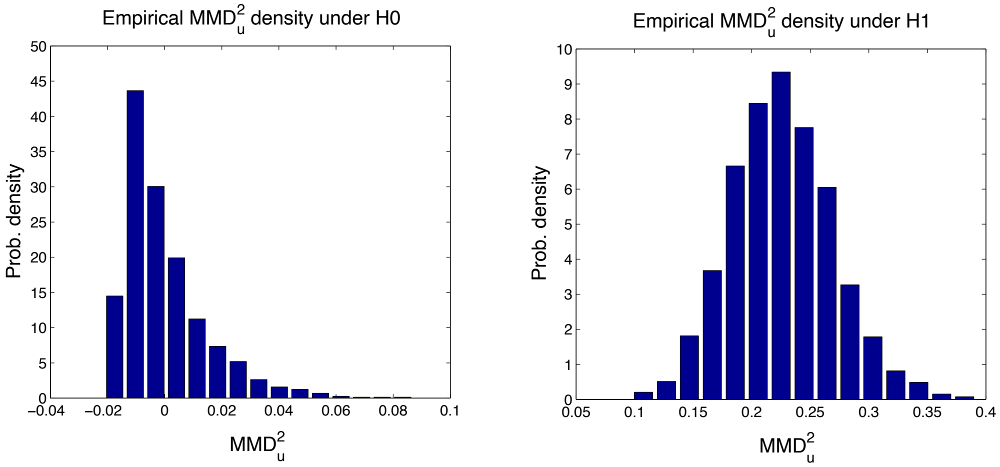


Figure 2: **Left:** Empirical distribution of the MMD under $\mathcal{H}_0$, with p and q both Gaussians with unit standard deviation, using 50 samples from each. **Right:** Empirical distribution of the MMD under $\mathcal{H}_A$, with p a Laplace distribution with unit standard deviation, and q a Laplace distribution with standard deviation $3\sqrt{2}$, using 100 samples from each. In both cases, the histograms were obtained by computing 2000 independent instances of the MMD.

distribution by fitting Pearson curves to its first four moments (Johnson et al., 1994, Section 18.8). Taking advantage of the degeneracy of the U-statistic, we obtain for $m = n$

$$\begin{aligned}
\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^2\right) &= \frac{2}{m(m-1)}\mathbf{E}_{z,z'}\left[h^2(z,z')\right] \text{ and} \\
\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^3\right) &= \frac{8(m-2)}{m^2(m-1)^2}\mathbf{E}_{z,z'}\left[h(z,z')\mathbf{E}_{z''}\left(h(z,z'')h(z',z'')\right)\right] + O(m^{-4}) \qquad (11)
\end{aligned}$$

(see Appendix B.3), where $h(z,z')$ is defined in Lemma 6, $z = (x,y) \sim p \times q$ where x and y are independent, and z', z'' are independent copies of z. The fourth moment $\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^4\right)$ is not computed, since it is both very small, $O(m^{-4})$, and expensive to calculate, $O(m^4)$. Instead, we replace the kurtosis[12] with a lower bound due to Wilkins (1944), $\mathrm{kurt}\left(\mathrm{MMD}_u^2\right) \geq \left(\mathrm{skew}\left(\mathrm{MMD}_u^2\right)\right)^2 + 1$. In Figure 3, we illustrate the Pearson curve fit to the null distribution: the fit is good in the upper quantiles of the distribution, where the test threshold is computed. Finally, we note that two alternative empirical estimates of the null distribution have more recently been proposed by Gretton et al. (2009): a consistent estimate, based on an empirical computation of the eigenvalues λ_l in (10); and an alternative Gamma approximation to the null distribution, which has a smaller computational cost but is generally less accurate. Further detail and experimental comparisons are given by Gretton et al.

[12] The kurtosis is defined in terms of the fourth and second moments as $\mathrm{kurt}\left(\mathrm{MMD}_u^2\right) = \frac{\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^4\right)}{\left[\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^2\right)\right]^2} - 3$.

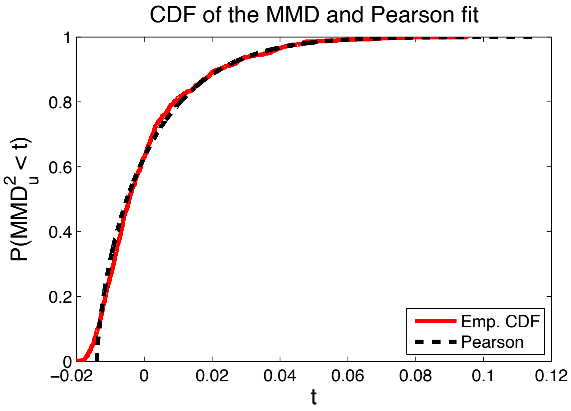


Figure 3: Illustration of the empirical CDF of the MMD and a Pearson curve fit. Both p and q were Gaussian with zero mean and unit variance, and 50 samples were drawn from each. The empirical CDF was computed on the basis of 1000 randomly generated MMD values. To ensure the quality of fit was determined only by the accuracy of the Pearson approximation, the moments used for the Pearson curves were also computed on the basis of these 1000 samples. The MMD used a Gaussian kernel with $\sigma = 0.5$.

6. A Linear Time Statistic and Test

The MMD-based tests are already more efficient than the $O(m^2 \log m)$ and $O(m^3)$ tests described in Section 3.3.3 (assuming $m = n$ for conciseness). It is still desirable, however, to obtain $O(m)$ tests which do not sacrifice too much statistical power. Moreover, we would like to obtain tests which have $O(1)$ storage requirements for computing the test statistic, in order to apply the test to data streams. We now describe how to achieve this by computing the test statistic using a subsampling of the terms in the sum. The empirical estimate in this case is obtained by drawing pairs from X and Y respectively *without* replacement.

Lemma 14 *Define $m_2 := \lfloor m/2 \rfloor$, assume $m = n$, and define $h(z_1, z_2)$ as in Lemma 6. The estimator*

$$\mathrm{MMD}_l^2[\mathcal{F}, X, Y] := \frac{1}{m_2} \sum_{i=1}^{m_2} h((x_{2i-1}, y_{2i-1}), (x_{2i}, y_{2i}))$$

can be computed in linear time, and is an unbiased estimate of $\mathrm{MMD}^2[\mathcal{F}, p, q]$.

While it is expected that MMD_l^2 has higher variance than MMD_u^2 (as we will see explicitly later), it is computationally much more appealing. In particular, the statistic can be used in stream computations with need for only $O(1)$ memory, whereas MMD_u^2 requires $O(m)$ storage and $O(m^2)$ time to compute the kernel h on all interacting pairs.

Since MMD_l^2 is just the average over a set of random variables, Hoeffding's bound and the central limit theorem readily allow us to provide both uniform convergence and asymptotic statements with little effort. The first follows directly from Hoeffding (1963, Theorem 2).

Theorem 15 *Assume* $0 \leq k(x_i, x_j) \leq K$. *Then*

$$\Pr_{X,Y}\left\{\mathrm{MMD}_l^2(\mathcal{F}, X, Y) - \mathrm{MMD}^2(\mathcal{F}, p, q) > t\right\} \leq \exp\left(\frac{-t^2 m_2}{8K^2}\right)$$

where $m_2 := \lfloor m/2 \rfloor$ *(the same bound applies for deviations of* $-t$ *and below).*

Note that the bound of Theorem 10 is identical to that of Theorem 15, which shows the former is rather loose. Next we invoke the central limit theorem (e.g., Serfling, 1980, Section 1.9).

Corollary 16 *Assume* $0 < \mathbf{E}\left(h^2\right) < \infty$. *Then* MMD_l^2 *converges in distribution to a Gaussian according to*

$$m^{\frac{1}{2}}\left(\mathrm{MMD}_l^2 - \mathrm{MMD}^2\left[\mathcal{F}, p, q\right]\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma_l^2\right),$$

where $\sigma_l^2 = 2\left[\mathbf{E}_{z,z'} h^2(z,z') - \left[\mathbf{E}_{z,z'} h(z,z')\right]^2\right]$, *where we use the shorthand* $\mathbf{E}_{z,z'} := \mathbf{E}_{z,z' \sim p \times q}$.

The factor of 2 arises since we are averaging over only $\lfloor m/2 \rfloor$ observations. It is instructive to compare this asymptotic distribution with that of the quadratic time statistic MMD_u^2 under $\mathcal{H}_A$, when $m = n$. In this case, MMD_u^2 converges in distribution to a Gaussian according to

$$m^{\frac{1}{2}}\left(\mathrm{MMD}_u^2 - \mathrm{MMD}^2\left[\mathcal{F}, p, q\right]\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma_u^2\right),$$

where $\sigma_u^2 = 4\left(\mathbf{E}_z\left[(\mathbf{E}_{z'} h(z,z'))^2\right] - \left[\mathbf{E}_{z,z'}(h(z,z'))\right]^2\right)$ (Serfling, 1980, Section 5.5). Thus for MMD_u^2, the asymptotic variance is (up to scaling) the variance of $\mathbf{E}_{z'}[h(z,z')]$, whereas for MMD_l^2 it is $\mathrm{Var}_{z,z'}[h(z,z')]$.

We end by noting another potential approach to reducing the cost of computing an empirical MMD estimate, by using a low rank approximation to the Gram matrix (Fine and Scheinberg, 2001; Williams and Seeger, 2001; Smola and Schölkopf, 2000). An incremental computation of the MMD based on such a low rank approximation would require $O(md)$ storage and $O(md)$ computation (where d is the rank of the approximate Gram matrix which is used to factorize *both* matrices) rather than $O(m)$ storage and $O(m^2)$ operations. That said, it remains to be determined what effect this approximation would have on the distribution of the test statistic under $\mathcal{H}_0$, and hence on the test threshold.

7. Related Metrics and Learning Problems

The present section discusses a number of topics related to the maximum mean discrepancy, including metrics on probability distributions using non-RKHS function classes (Sections 7.1 and 7.2), the relation with set kernels and kernels on probability measures (Section 7.3), an extension to kernel measures of independence (Section 7.4), a two-sample statistic using a distribution over witness functions (Section 7.5), and a connection to outlier detection (Section 7.6).

7.1 The MMD in Other Function Classes

The definition of the maximum mean discrepancy is by no means limited to RKHS. In fact, any function class $\mathcal{F}$ that comes with uniform convergence guarantees and is sufficiently rich will enjoy the above properties. Below, we consider the case where the scaled functions in $\mathcal{F}$ are dense in $C(\mathcal{X})$ (which is useful for instance when the functions in $\mathcal{F}$ are norm constrained).

Definition 17 *Let $\mathcal{F}$ be a subset of some vector space. The star $S[\mathcal{F}]$ of a set $\mathcal{F}$ is*

$$S[\mathcal{F}] := \{\alpha f | f \in \mathcal{F} \text{ and } \alpha \in [0, \infty)\}$$

Theorem 18 *Denote by $\mathcal{F}$ the subset of some vector space of functions from $\mathcal{X}$ to $\mathbb{R}$ for which $S[\mathcal{F}] \cap C(\mathcal{X})$ is dense in $C(\mathcal{X})$ with respect to the $L_\infty(\mathcal{X})$ norm. Then* $\mathrm{MMD}[\mathcal{F}, p, q] = 0$ *if and only if $p = q$, and* $\mathrm{MMD}[\mathcal{F}, p, q]$ *is a metric on the space of probability distributions. Whenever the star of $\mathcal{F}$ is* not *dense, the* MMD *defines a pseudo-metric space.*

Proof It is clear that $p = q$ implies $\mathrm{MMD}[\mathcal{F}, p, q] = 0$. The proof of the converse is very similar to that of Theorem 5. Define $\mathcal{H} := S(\mathcal{F}) \cap C(\mathcal{X})$. Since by assumption $\mathcal{H}$ is dense in $C(\mathcal{X})$, there exists an $h^* \in \mathcal{H}$ satisfying $\|h^* - f\|_\infty < \varepsilon$ for all $f \in C(\mathcal{X})$. Write $h^* := \alpha^* g^*$, where $g^* \in \mathcal{F}$. By assumption, $\mathbf{E}_x g^* - \mathbf{E}_y g^* = 0$. Thus we have the bound

$$\begin{aligned} |\mathbf{E}_x f(x) - \mathbf{E}_y(f(y))| &\leq |\mathbf{E}_x f(x) - \mathbf{E}_x h^*(x)| + \alpha^* |\mathbf{E}_x g^*(x) - \mathbf{E}_y g^*(y)| + |\mathbf{E}_y h^*(y) - \mathbf{E}_y f(y)| \\ &\leq 2\varepsilon \end{aligned}$$

for all $f \in C(\mathcal{X})$ and $\varepsilon > 0$, which implies $p = q$ by Lemma 1.

To show $\mathrm{MMD}[\mathcal{F}, p, q]$ is a metric, it remains to prove the triangle inequality. We have

$$\begin{aligned} \sup_{f \in \mathcal{F}} \left|E_p f - E_q f\right| + \sup_{g \in \mathcal{F}} \left|E_q g - E_r g\right| &\geq \sup_{f \in \mathcal{F}} \left[\left|E_p f - E_q f\right| + \left|E_q f - E_r\right|\right] \\ &\geq \sup_{f \in \mathcal{F}} |E_p f - E_r f|. \end{aligned}$$

■

Note that any uniform convergence statements in terms of $\mathcal{F}$ allow us immediately to characterize an estimator of $\mathrm{MMD}(\mathcal{F}, p, q)$ explicitly. The following result shows how (this reasoning is also the basis for the proofs in Section 4, although here we do not restrict ourselves to an RKHS).

Theorem 19 *Let $\delta \in (0, 1)$ be a confidence level and assume that for some $\varepsilon(\delta, m, \mathcal{F})$ the following holds for samples $\{x_1, \ldots, x_m\}$ drawn from p:*

$$\Pr_X \left\{ \sup_{f \in \mathcal{F}} \left| \mathbf{E}_x[f] - \frac{1}{m} \sum_{i=1}^m f(x_i) \right| > \varepsilon(\delta, m, \mathcal{F}) \right\} \leq \delta.$$

In this case we have that,

$$\Pr_{X,Y} \{|\mathrm{MMD}[\mathcal{F}, p, q] - \mathrm{MMD}_b[\mathcal{F}, X, Y]| > 2\varepsilon(\delta/2, m, \mathcal{F})\} \leq \delta,$$

where $\mathrm{MMD}_b[\mathcal{F}, X, Y]$ *is taken from Definition 2.*

Proof The proof works simply by using convexity and suprema as follows:

$$\begin{aligned} &|\mathrm{MMD}[\mathcal{F}, p, q] - \mathrm{MMD}_b[\mathcal{F}, X, Y]| \\ &= \left| \sup_{f \in \mathcal{F}} |\mathbf{E}_x[f] - \mathbf{E}_y[f]| - \sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m f(x_i) - \frac{1}{n} \sum_{i=1}^n f(y_i) \right| \right| \\ &\leq \sup_{f \in \mathcal{F}} \left| \mathbf{E}_x[f] - \mathbf{E}_y[f] - \frac{1}{m} \sum_{i=1}^m f(x_i) + \frac{1}{n} \sum_{i=1}^n f(y_i) \right| \\ &\leq \sup_{f \in \mathcal{F}} \left| \mathbf{E}_x[f] - \frac{1}{m} \sum_{i=1}^m f(x_i) \right| + \sup_{f \in \mathcal{F}} \left| \mathbf{E}_y[f] - \frac{1}{n} \sum_{i=1}^n f(y_i) \right|. \end{aligned}$$

Bounding each of the two terms via a uniform convergence bound proves the claim. ■

This shows that $\mathrm{MMD}_b[\mathcal{F},X,Y]$ can be used to estimate $\mathrm{MMD}[\mathcal{F},p,q]$, and that the quantity is asymptotically unbiased.

Remark 20 (Reduction to Binary Classification) *As noted by Friedman (2003), any classifier which maps a set of observations $\{z_i,l_i\}$ with $z_i \in \mathcal{X}$ on some domain $\mathcal{X}$ and labels $l_i \in \{\pm 1\}$, for which uniform convergence bounds exist on the convergence of the empirical loss to the expected loss, can be used to obtain a similarity measure on distributions—simply assign $l_i = 1$ if $z_i \in X$ and $l_i = -1$ for $z_i \in Y$ and find a classifier which is able to separate the two sets. In this case maximization of $\mathbf{E}_x[f] - \mathbf{E}_y[f]$ is achieved by ensuring that as many $z \sim p(z)$ as possible correspond to $f(z) = 1$, whereas for as many $z \sim q(z)$ as possible we have $f(z) = -1$. Consequently neural networks, decision trees, boosted classifiers and other objects for which uniform convergence bounds can be obtained can be used for the purpose of distribution comparison. Metrics and divergences on distributions can also be defined explicitly starting from classifiers. For instance, Sriperumbudur et al. (2009, Section 2) show the* MMD *minimizes the expected risk of a classifier with linear loss on the samples X and Y, and Ben-David et al. (2007, Section 4) use the error of a hyperplane classifier to approximate the $\mathcal{A}$-distance between distributions (Kifer et al., 2004). Reid and Williamson (2011) provide further discussion and examples.*

7.2 Examples of Non-RKHS Function Classes

Other function spaces $\mathcal{F}$ inspired by the statistics literature can also be considered in defining the MMD. Indeed, Lemma 1 defines an MMD with $\mathcal{F}$ the space of bounded continuous real-valued functions, which is a Banach space with the supremum norm (Dudley, 2002, p. 158). We now describe two further metrics on the space of probability distributions, namely the Kolmogorov-Smirnov and Earth Mover's distances, and their associated function classes.

7.2.1 KOLMOGOROV-SMIRNOV STATISTIC

The Kolmogorov-Smirnov (K-S) test is probably one of the most famous two-sample tests in statistics. It works for random variables $x \in \mathbb{R}$ (or any other set for which we can establish a total order). Denote by $F_p(x)$ the cumulative distribution function of p and let $F_X(x)$ be its empirical counterpart,

$$F_p(z) := \Pr\{x \leq z \text{ for } x \sim p\} \text{ and } F_X(z) := \frac{1}{|X|}\sum_{i=1}^{m} 1_{z \leq x_i}.$$

It is clear that F_p captures the properties of p. The Kolmogorov metric is simply the L_∞ distance $\|F_X - F_Y\|_\infty$ for two sets of observations X and Y. Smirnov (1939) showed that for $p = q$ the limiting distribution of the empirical cumulative distribution functions satisfies

$$\lim_{m,n\to\infty} \Pr_{X,Y}\left\{\left[\tfrac{mn}{m+n}\right]^{\frac{1}{2}} \|F_X - F_Y\|_\infty > x\right\} = 2\sum_{j=1}^{\infty}(-1)^{j-1}e^{-2j^2x^2} \text{ for } x \geq 0, \tag{12}$$

which is distribution independent. This allows for an efficient characterization of the distribution under the null hypothesis $\mathcal{H}_0$. Efficient numerical approximations to (12) can be found in numerical analysis handbooks (Press et al., 1994). The distribution under the alternative $p \neq q$, however, is unknown.

The Kolmogorov metric is, in fact, a special instance of $\mathrm{MMD}[\mathcal{F},p,q]$ for a certain Banach space (Müller, 1997, Theorem 5.2).

Proposition 21 *Let $\mathcal{F}$ be the class of functions $\mathcal{X} \to \mathbb{R}$ of bounded variation*[13] *1. Then* $\mathrm{MMD}[\mathcal{F},p,q] = \left\|F_p - F_q\right\|_\infty$.

7.2.2 EARTH-MOVER DISTANCES

Another class of distance measures on distributions that may be written as maximum mean discrepancies are the Earth-Mover distances. We assume $(\mathcal{X},\rho)$ is a separable metric space, and define $\mathcal{P}_1(\mathcal{X})$ to be the space of probability measures on $\mathcal{X}$ for which $\int \rho(x,z)dp(z) < \infty$ for all $p \in \mathcal{P}_1(\mathcal{X})$ and $x \in \mathcal{X}$ (these are the probability measures for which $\mathbf{E}_x |x| < \infty$ when $\mathcal{X} = \mathbb{R}$). We then have the following definition (Dudley, 2002, p. 420).

Definition 22 (Monge-Wasserstein metric) *Let $p \in \mathcal{P}_1(\mathcal{X})$ and $q \in \mathcal{P}_1(\mathcal{X})$. The Monge-Wasserstein distance is defined as*

$$W(p,q) := \inf_{\mu \in M(p,q)} \int \rho(x,y) d\mu(x,y),$$

where $M(p,q)$ is the set of joint distributions on $\mathcal{X} \times \mathcal{X}$ with marginals p and q.

We may interpret this as the cost (as represented by the metric $\rho(x,y)$) of transferring mass distributed according to p to a distribution in accordance with q, where μ is the movement schedule. In general, a large variety of costs of moving mass from x to y can be used, such as psycho-optical similarity measures in image retrieval (Rubner et al., 2000). The following theorem provides the link with the MMD (Dudley, 2002, Theorem 11.8.2).

Theorem 23 (Kantorovich-Rubinstein) *Let $p \in \mathcal{P}_1(\mathcal{X})$ and $q \in \mathcal{P}_1(\mathcal{X})$, where $\mathcal{X}$ is separable. Then a metric on $\mathcal{P}_1(S)$ is defined as*

$$W(p,q) = \|p-q\|_L^* = \sup_{\|f\|_L \leq 1} \left| \int f\, d(p-q) \right|,$$

where

$$\|f\|_L := \sup_{x \neq y \in \mathcal{X}} \frac{|f(x) - f(y)|}{\rho(x,y)}$$

is the Lipschitz seminorm[14] *for real valued f on $\mathcal{X}$.*

A simple example of this theorem is as follows (Dudley, 2002, Exercise 1, p. 425).

Example 2 *Let $\mathcal{X} = \mathbb{R}$ with associated $\rho(x,y) = |x-y|$. Then given f such that $\|f\|_L \leq 1$, we use integration by parts to obtain*

$$\left| \int f\, d(p-q) \right| = \left| \int (F_p - F_q)(x) f'(x) dx \right| \leq \int \left|(F_p - F_q)\right|(x) dx,$$

[13] A function f defined on $[a,b]$ is of bounded variation C if the total variation is bounded by C, that is, the supremum over all sums

$$\sum_{1 \leq i \leq n} |f(x_i) - f(x_{i-1})|,$$

where $a \leq x_0 \leq \ldots \leq x_n \leq b$ (Dudley, 2002, p. 184).

[14] A seminorm satisfies the requirements of a norm besides $\|x\| = 0$ only for $x = 0$ (Dudley, 2002, p. 156).

where the maximum is attained for the function g with derivative $g' = 2\,1_{F_p > F_q} - 1$ (and for which $\|g\|_L = 1$). We recover the L_1 distance between distribution functions,

$$W(P,Q) = \int \left|(F_p - F_q)\right|(x)dx.$$

One may further generalize Theorem 23 to the set of all laws $\mathcal{P}(\mathcal{X})$ on arbitrary metric spaces $\mathcal{X}$ (Dudley, 2002, Proposition 11.3.2).

Definition 24 (Bounded Lipschitz metric) *Let p and q be laws on a metric space $\mathcal{X}$. Then*

$$\beta(p,q) := \sup_{\|f\|_{BL} \leq 1} \left| \int f\, d(p-q) \right|$$

is a metric on $\mathcal{P}(\mathcal{X})$, where f belongs to the space of bounded Lipschitz functions with norm

$$\|f\|_{BL} := \|f\|_L + \|f\|_\infty .$$

Empirical estimates of the Monge-Wasserstein and Bounded Lipschitz metrics on $\mathbb{R}^d$ are provided by Sriperumbudur et al. (2010a).

7.3 Set Kernels and Kernels Between Probability Measures

Gärtner et al. (2002) propose kernels for Multi-Instance Classification (MIC) which deal with sets of observations. The purpose of MIC is to find estimators which are able to infer that if some elements in a set satisfy a certain property, then the set of observations also has this property. For instance, a dish of mushrooms is poisonous if it contains any poisonous mushrooms. Likewise a keyring will open a door if it contains a suitable key. One is only given the ensemble, however, rather than information about which instance of the set satisfies the property.

The solution proposed by Gärtner et al. (2002) is to map the ensembles $X_i := \{x_{i1}, \ldots, x_{im_i}\}$, where i is the ensemble index and m_i the number of elements in the ith ensemble, jointly into feature space via

$$\phi(X_i) := \frac{1}{m_i} \sum_{j=1}^{m_i} \phi(x_{ij}),$$

and to use the latter as the basis for a kernel method. This simple approach affords rather good performance. With the benefit of hindsight, it is now understandable why the kernel

$$k(X_i, X_j) = \frac{1}{m_i m_j} \sum_{u,v}^{m_i, m_j} k(x_{iu}, x_{jv})$$

produces useful results: it is simply the kernel between the empirical means in feature space $\left\langle \mu(X_i), \mu(X_j) \right\rangle$ (Hein et al., 2004, Equation 4). Jebara and Kondor (2003) later extended this setting by smoothing the empirical densities before computing inner products.

Note, however, that the empirical mean embedding μ_X may not be the best statistic to use for MIC: we are only interested in determining whether *some* instances in the domain have the desired property, rather than making a statement regarding the distribution over all instances. Taking this into account leads to an improved algorithm (Andrews et al., 2003).

7.4 Kernel Measures of Independence

We next demonstrate the application of MMD in determining whether two random variables x and y are independent. In other words, assume that pairs of random variables (x_i, y_i) are jointly drawn from some distribution $p := p_{xy}$. We wish to determine whether this distribution factorizes; that is, whether $q := p_x \times p_y$ is the same as p. One application of such an independence measure is in independent component analysis (Comon, 1994), where the goal is to find a linear mapping of the observations x_i to obtain mutually independent outputs. Kernel methods were employed to solve this problem by Bach and Jordan (2002), Gretton et al. (2005a,b), and Shen et al. (2009). In the following we re-derive one of the above kernel independence measures as a distance between mean embeddings (see also Smola et al., 2007).

We begin by defining

$$\begin{aligned} \mu\left[p_{xy}\right] &:= \mathbf{E}_{x,y}\left[v((x,y),\cdot)\right] \\ \text{and } \mu\left[p_x \times p_y\right] &:= \mathbf{E}_x \mathbf{E}_y\left[v((x,y),\cdot)\right]. \end{aligned}$$

Here we assume $\mathcal{V}$ is an RKHS over $\mathcal{X} \times \mathcal{Y}$ with kernel $v((x,y),(x',y'))$. If x and y are dependent, then $\mu[p_{xy}] \neq \mu[p_x \times p_y]$. Hence we may use $\Delta(\mathcal{V}, p_{xy}, p_x \times p_y) := \|\mu[p_{xy}] - \mu[p_x \times p_y]\|_{\mathcal{V}}$ as a measure of dependence.

Now assume that $v((x,y),(x',y')) = k(x,x')l(y,y')$, that is, the RKHS $\mathcal{V}$ is a direct product $\mathcal{H} \otimes \mathcal{G}$ of RKHSs on $\mathcal{X}$ and $\mathcal{Y}$. In this case it is easy to see that

$$\begin{aligned} \Delta^2\left(\mathcal{V}, p_{xy}, p_x \times p_y\right) &= \left\|\mathbf{E}_{xy}\left[k(x,\cdot)l(y,\cdot)\right] - \mathbf{E}_x\left[k(x,\cdot)\right]\mathbf{E}_y\left[l(y,\cdot)\right]\right\|_{\mathcal{V}}^2 \\ &= \mathbf{E}_{xy}\mathbf{E}_{x'y'}\left[k(x,x')l(y,y')\right] - 2\mathbf{E}_x\mathbf{E}_y\mathbf{E}_{x'y'}\left[k(x,x')l(y,y')\right] \\ &\quad + \mathbf{E}_x\mathbf{E}_y\mathbf{E}_{x'}\mathbf{E}_{y'}\left[k(x,x')l(y,y')\right]. \end{aligned}$$

The latter is also the squared Hilbert-Schmidt norm of the cross-covariance operator between RKHSs (Gretton et al., 2005a): for characteristic kernels, this is zero if and only if x and y are independent.

Theorem 25 *Denote by C_{xy} the covariance operator between random variables x and y, drawn jointly from p_{xy}, where the functions on $\mathcal{X}$ and $\mathcal{Y}$ are the reproducing kernel Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$ respectively. Then the Hilbert-Schmidt norm $\|C_{xy}\|_{\mathrm{HS}}$ equals $\Delta(\mathcal{V}, p_{xy}, p_x \times p_y)$.*

Empirical estimates of this quantity are as follows:

Theorem 26 *Denote by K and L the kernel matrices on X and Y respectively, and by $H = I - \mathbf{1}/m$ the projection matrix onto the subspace orthogonal to the vector with all entries set to 1 (where $\mathbf{1}$ is an $m \times m$ matrix of ones). Then $m^{-2}\operatorname{tr} HKHL$ is an estimate of Δ^2 with bias $O(m^{-1})$. The deviation from Δ^2 is $O_P(m^{-1/2})$.*

Gretton et al. (2005a) provide explicit constants. In certain circumstances, including in the case of RKHSs with Gaussian kernels, the empirical Δ^2 may also be interpreted in terms of a smoothed difference between the joint empirical characteristic function (ECF) and the product of the marginal ECFs (Feuerverger, 1993; Kankainen, 1995). This interpretation does not hold in all cases, however, for example, for kernels on strings, graphs, and other structured spaces. An illustration of the witness function $f^* \in \mathcal{V}$ from Section 2.3 is provided in Figure 4, for the case of dependence detection. This is a smooth function which has large magnitude where the joint density is most different from the product of the marginals.

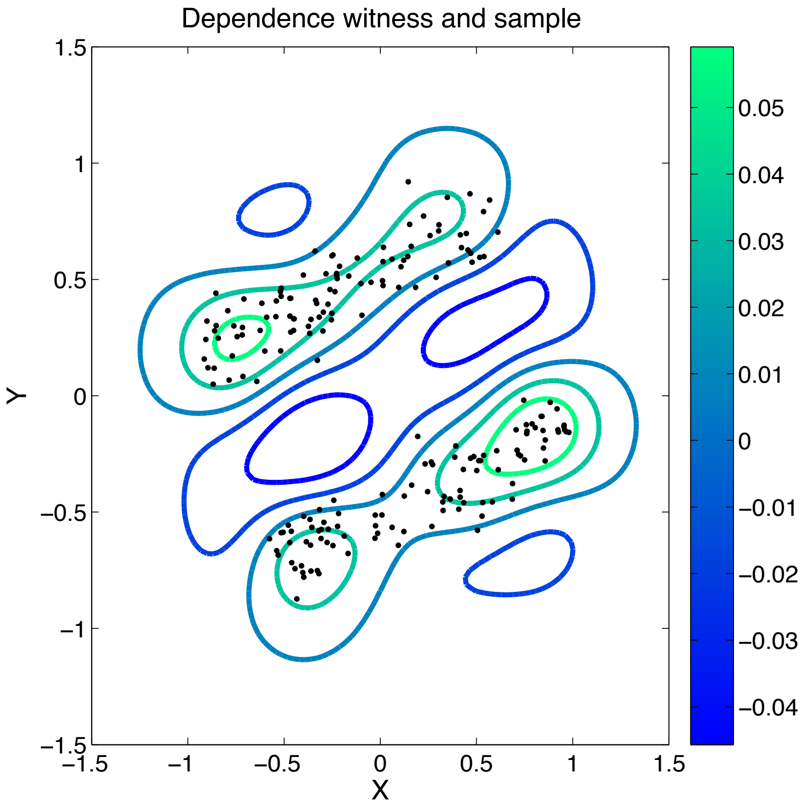


Figure 4: Illustration of the function maximizing the mean discrepancy when MMD is used as a measure of dependence. A sample from dependent random variables x and y is shown in black, and the associated function $\hat{f}^*$ that witnesses the MMD is plotted as a contour. The latter was computed empirically on the basis of 200 samples, using a Gaussian kernel with $\sigma = 0.2$.

We remark that a hypothesis test based on the above kernel statistic is more complicated than for the two-sample problem, since the product of the marginal distributions is in effect simulated by permuting the variables of the original sample. Further details are provided by Gretton et al. (2008b).

7.5 Kernel Statistics Using a Distribution over Witness Functions

Shawe-Taylor and Dolia (2007) define a distance between distributions as follows: let $\mathcal{H}$ be a set of functions on $\mathcal{X}$ and r be a probability distribution over $\mathcal{H}$. Then the distance between two distributions p and q is given by

$$D(p,q) := \mathbf{E}_{f\sim r(f)} \left|\mathbf{E}_x[f(x)] - \mathbf{E}_y[f(y)]\right|. \qquad (13)$$

That is, we compute the average distance between p and q with respect to a distribution over test functions. The following result shows the relation with the MMD, and is due to Song et al. (2008, Section 6).

Lemma 27 *Let $\mathcal{H}$ be a reproducing kernel Hilbert space, $f \in \mathcal{H}$, and assume $r(f) = r(\|f\|_{\mathcal{H}})$ with finite $\mathbf{E}_{f\sim r}[\|f\|_{\mathcal{H}}]$. Then $D(p,q) = C\left\|\mu_p - \mu_q\right\|_{\mathcal{H}}$ for some constant C which depends only on $\mathcal{H}$ and r.*

Proof By definition $\mathbf{E}_x[f(x)] = \langle \mu_p, f\rangle_{\mathcal{H}}$. Using linearity of the inner product, Equation (13) equals

$$\begin{aligned} &\int \left|\langle \mu_p - \mu_q, f\rangle_{\mathcal{H}}\right| \mathrm{d}r(f) \\ &= \left\|\mu_p - \mu_q\right\|_{\mathcal{H}} \int \left|\left\langle \frac{\mu_p - \mu_q}{\left\|\mu_p - \mu_q\right\|_{\mathcal{H}}}, f \right\rangle_{\mathcal{H}}\right| \mathrm{d}r(f), \end{aligned}$$

where the integral is independent of p,q. To see this, note that for any p,q, $\frac{\mu_p - \mu_q}{\left\|\mu_p - \mu_q\right\|_{\mathcal{H}}}$ is a unit vector which can be transformed into the first canonical basis vector (for instance) by a rotation which leaves the integral invariant, bearing in mind that r is rotation invariant. ■

7.6 Outlier Detection

An application related to the two sample problem is that of outlier detection: this is the question of whether a novel point is generated from the same distribution as a particular i.i.d. sample. In a way, this is a special case of a two sample test, where the second sample contains only one observation. Several methods essentially rely on the distance between a novel point to the sample mean in feature space to detect outliers.

For instance, Davy et al. (2002) use a related method to deal with nonstationary time series. Likewise Shawe-Taylor and Cristianini (2004, p. 117) discuss how to detect novel observations by using the following reasoning: the probability of being an outlier is bounded both as a function of the spread of the points in feature space and the uncertainty in the empirical feature space mean (as bounded using symmetrisation and McDiarmid's tail bound).

Instead of using the sample mean and variance, Tax and Duin (1999) estimate the center and radius of a minimal enclosing sphere for the data, the advantage being that such bounds can potentially lead to more reliable tests for single observations. Schölkopf et al. (2001) show that the minimal enclosing sphere problem is equivalent to novelty detection by means of finding a hyperplane separating the data from the origin, at least in the case of radial basis function kernels.

8. Experiments

We conducted distribution comparisons using our MMD-based tests on data sets from three real-world domains: database applications, bioinformatics, and neurobiology. We investigated both uniform convergence approaches (MMD$_b$ with the Corollary 9 threshold, and MMD$_u^2$ H with the Corollary 11 threshold); the asymptotic approaches with bootstrap (MMD$_u^2$ B) and moment matching to Pearson curves (MMD$_u^2$ M), both described in Section 5; and the asymptotic approach using the linear time statistic (MMD$_l^2$) from Section 6. We also compared against several alternatives from the literature (where applicable): the multivariate t-test, the Friedman-Rafsky Kolmogorov-Smirnov generalisation *(Smir)*, the Friedman-Rafsky Wald-Wolfowitz generalisation *(Wolf)*, the Biau-Györfi test *(Biau)* with a uniform space partitioning, and the Hall-Tajvidi test *(Hall)*. See Section 3.3 for details regarding these tests. Note that we do not apply the Biau-Györfi test to high-dimensional problems (since the required space partitioning is no longer possible), and that MMD is the only method applicable to structured data such as graphs.

An important issue in the practical application of the MMD-based tests is the selection of the kernel parameters. We illustrate this with a Gaussian RBF kernel, where we must choose the kernel width σ (we use this kernel for univariate and multivariate data, but not for graphs). The empirical MMD is zero both for kernel size $\sigma = 0$ (where the aggregate Gram matrix over X and Y is a unit matrix), and also approaches zero as $\sigma \to \infty$ (where the aggregate Gram matrix becomes uniformly constant). We set σ to be the median distance between points in the aggregate sample, as a compromise between these two extremes: this remains a heuristic, similar to those described in Takeuchi et al. (2006) and Schölkopf (1997), and the optimum choice of kernel size is an ongoing area of research. We further note that setting the kernel using the sample being tested may cause changes to the asymptotic distribution: in particular, the analysis in Sections 4 and 5 assumes the kernel not to be a function of the sample. An analysis of the convergence of MMD when the kernel is adapted on the basis of the sample is provided by Sriperumbudur et al. (2009), although the asymptotic distribution in this case remains a topic of research. As a practical matter, however, the median heuristic has not been observed to have much effect on the asymptotic distribution, and in experiments is indistinguishable from results obtained by computing the kernel on a small subset of the sample set aside for this purpose. See Appendix C for more detail.

8.1 Toy Example: Two Gaussians

In our first experiment, we investigated the scaling performance of the various tests as a function of the dimensionality d of the space $\mathcal{X} \subset \mathbb{R}^d$, when both p and q were Gaussian. We considered values of d up to 2500: the performance of the MMD-based tests cannot therefore be explained in the context of density estimation (as in Section 3.3.1), since the associated density estimates are necessarily meaningless here. The levels for all tests were set at $\alpha = 0.05, m = n = 250$ samples were used, and results were averaged over 100 repetitions. In the first case, the distributions had different means and unit variance. The percentage of times the null hypothesis was correctly rejected over a set of Euclidean distances between the distribution means (20 values logarithmically spaced from 0.05 to 50), was computed as a function of the dimensionality of the normal distributions. In case of the t-test, a ridge was added to the covariance estimate, to avoid singularity (the ratio of largest to smallest eigenvalue was ensured to be at most 2). In the second case, samples were drawn from distributions $\mathcal{N}(0, \mathbf{I})$ and $\mathcal{N}(0, \sigma^2\mathbf{I})$ with different variance. The percentage of null rejections was averaged over 20 σ values logarithmically spaced from $10^{0.01}$ to 10. The t-test was not compared in this case, since its output would have been irrelevant. Results are plotted in Figure 5.

In the case of Gaussians with differing means, we observe the t-test performs best in low dimensions, however its performance is severely weakened when the number of samples exceeds the number of dimensions. The performance of MMD_u^2 M is comparable to the t-test in low dimensions, and outperforms all other methods in high dimensions. The worst performance is obtained for MMD_u^2 H, though MMD_b also does relatively poorly: this is unsurprising given that these tests

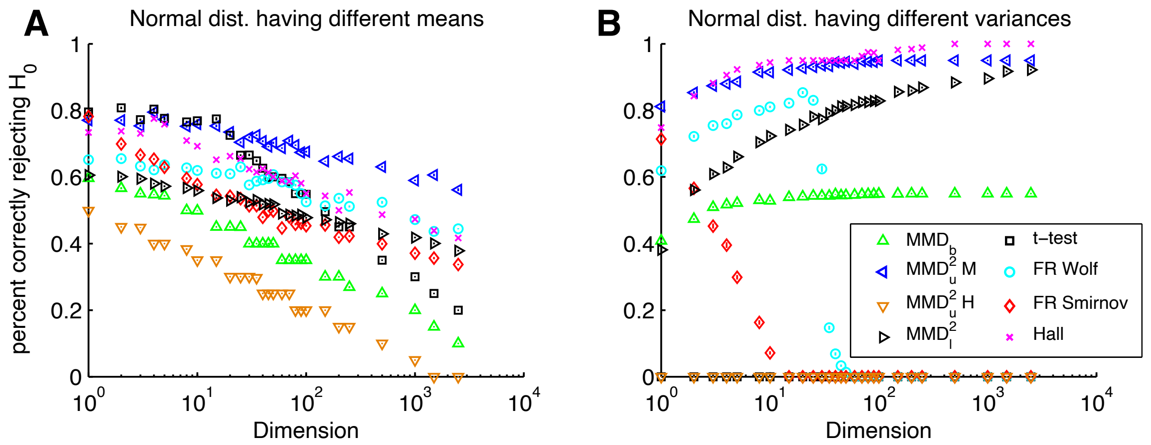


Figure 5: Type II performance of the various tests when separating two Gaussians, with test level $\alpha = 0.05$. **A** Gaussians having same variance and different means. **B** Gaussians having same mean and different variances.

derive from distribution-free large deviation bounds, and the sample size is relatively small. Remarkably, MMD_l^2 performs quite well compared with the Section 3.3.3 tests in high dimensions.

In the case of Gaussians of differing variance, the *Hall* test performs best, followed closely by MMD_u^2 M. *FR Wolf* and (to a much greater extent) *FR Smirnov* both have difficulties in high dimensions, failing completely once the dimensionality becomes too great. The linear-cost test MMD_l^2 again performs surprisingly well, almost matching the MMD_u^2 M performance at the highest dimensionality. Both MMD_u^2 H and MMD_b perform poorly, the former failing completely: this is one of several illustrations we will encounter of the much greater tightness of the Corollary 9 threshold over that in Corollary 11.

8.2 Data Integration

In our next application of MMD, we performed distribution testing for data integration: the objective being to aggregate two data sets into a single sample, with the understanding that both original samples were generated from the same distribution. Clearly, it is important to check this last condition before proceeding, or an analysis could detect patterns in the new data set that are caused by combining the two different source distributions. We chose several real-world settings for this task: we compared microarray data from normal and tumor tissues (Health status), microarray data from different subtypes of cancer (Subtype), and local field potential (LFP) electrode recordings from the Macaque primary visual cortex (V1) with and without spike events (Neural Data I and II, as described in more detail by Rasch et al., 2008). In all cases, the two data sets have different statistical properties, but the detection of these differences is made difficult by the high data dimensionality (indeed, for the microarray data, density estimation is impossible given the sample size and data dimensionality, and no successful test can rely on accurate density estimates as an intermediate step).

Data Set	Attr.	MMD_b	MMD_u^2 H	MMD_u^2 B	MMD_u^2 M	t-test	Wolf	Smir	Hall
Neural Data I	Same	100.0	100.0	96.5	96.5	100.0	97.0	95.0	96.0
	Different	38.0	100.0	**0.0**	**0.0**	42.0	**0.0**	10.0	49.0
Neural Data II	Same	100.0	100.0	94.6	95.2	100.0	95.0	94.5	96.0
	Different	99.7	100.0	3.3	3.4	100.0	**0.8**	31.8	5.9
Health status	Same	100.0	100.0	95.5	94.4	100.0	94.7	96.1	95.6
	Different	100.0	100.0	1.0	**0.8**	100.0	2.8	44.0	35.7
Subtype	Same	100.0	100.0	99.1	96.4	100.0	94.6	97.3	96.5
	Different	100.0	100.0	**0.0**	**0.0**	100.0	**0.0**	28.4	0.2

Table 1: Distribution testing for data integration on multivariate data. Numbers indicate the percentage of repetitions for which the null hypothesis (p=q) was accepted, given $\alpha = 0.05$. Sample size (dimension; repetitions of experiment): Neural I 4000 (63; 100) ; Neural II 1000 (100; 1200); Health Status 25 (12,600; 1000); Subtype 25 (2,118; 1000).

Data Set	Attr.	MMD_b	MMD_u^2 H	MMD_u^2 B	MMD_u^2 M	t-test	Wolf	Smir	Hall	Biau
BIO	Same	100.0	100.0	93.8	94.8	95.2	90.3	95.8	95.3	99.3
	Different	20.0	52.6	**17.2**	17.6	36.2	**17.2**	18.6	17.9	42.1
FOREST	Same	100.0	100.0	96.4	96.0	97.4	94.6	99.8	95.5	100.0
	Different	3.9	11.0	**0.0**	**0.0**	0.2	3.8	**0.0**	50.1	**0.0**
CNUM	Same	100.0	100.0	94.5	93.8	94.0	98.4	97.5	91.2	98.5
	Different	14.9	52.7	2.7	**2.5**	19.17	22.5	11.6	79.1	50.5
FOREST10D	Same	100.0	100.0	94.0	94.0	100.0	93.5	96.5	97.0	100.0
	Different	86.6	100.0	**0.0**	**0.0**	**0.0**	**0.0**	1.0	72.0	100.0

Table 2: Naive attribute matching on univariate (BIO, FOREST, CNUM) and multivariate (FOREST10D) data. Numbers indicate the percentage of times the null hypothesis $p = q$ was accepted with $\alpha = 0.05$, pooled over attributes. Sample size (dimension; attributes; repetitions of experiment): BIO 377 (1; 6; 100); FOREST 538 (1; 10; 100); CNUM 386 (1; 13; 100); FOREST10D 1000 (10; 2; 100).

We applied our tests to these data sets in the following fashion. Given two data sets A and B, we either chose one sample from A and the other from B *(attributes = different)*; or both samples from either A or B *(attributes = same)*. We then repeated this process up to 1200 times. Results are reported in Table 1. Our asymptotic tests perform better than all competitors besides *Wolf*: in the latter case, we have greater Type II error for one neural data set, lower Type II error on the Health Status data (which has very high dimension and low sample size), and identical (error-free) performance on the remaining examples. We note that the Type I error of the bootstrap test on the Subtype data set is far from its design value of 0.05, indicating that the Pearson curves provide a better threshold estimate for these low sample sizes. For the remaining data sets, the Type I errors of the Pearson and Bootstrap approximations are close. Thus, for larger data sets, the bootstrap is to be preferred, since it costs $O(m^2)$, compared with a cost of $O(m^3)$ for the Pearson curves (due to the cost of computing (11)). Finally, the uniform convergence-based tests are too conservative, with MMD_b finding differences in distribution only for the data with largest sample size, and MMD_u^2 H never finding differences.

8.3 Computational Cost

We next investigate the tradeoff between computational cost and performance of the various tests, with a particular focus on how the quadratic-cost MMD tests from Sections 4 and 5 compare with the linear time MMD-based asymptotic test from Section 6. We consider two 1-D data sets (CNUM and FOREST) and two higher-dimensional data sets (FOREST10D and NEUROII). Results are plotted in Figure 6. If cost is not a factor, then the MMD_u^2 B shows best overall performance as a function of sample size, with a Type II error dropping to zero as fast or faster than competing approaches in three of four cases, and narrowly trailing *FR Wolf* in the remaining case (FOREST10D). That said, for data sets CNUM, FOREST, and FOREST10D, the linear time MMD achieves a given Type II error at a far smaller computational cost than MMD_u^2 B, albeit by looking at a great deal more data. In the CNUM case, however, the linear test is not able to achieve zero error even for the largest data set size. For the NEUROII data, attaining zero Type II error has about the same cost for both approaches. The difference in cost of MMD_u^2 B and MMD_b is due to the bootstrapping required for the former, which produces a constant offset in cost between the two (here 150 resamplings were used).

The t-test also performs well in three of the four problems, and in fact represents the best cost-performance tradeoff in these three data sets (i.e., while it requires much more data than MMD_u^2 B for a given Type II error rate, it costs far less to compute). The t-test assumes that only the difference in means is important in distinguishing the distributions, and it requires an accurate estimate of the within-sample covariance; the test fails completely on the NEUROII data. We emphasise that the Kolmogorov-Smirnov results in 1-D were obtained using the classical statistic, and not the Friedman-Rafsky statistic, hence the low computational cost. The cost of both Friedman-Rafsky statistics is therefore given by the *FR Wolf* cost in this case. The latter scales similarly with sample size to the quadratic time MMD tests, confirming Friedman and Rafsky's observation that obtaining the pairwise distances between sample points is the dominant cost of their tests. We also remark on the unusual behaviour of the Type II error of the *FR Wolf* test in the FOREST data set, which worsens for increasing sample size.

We conclude that the approach to be recommended for two-sample testing will depend on the data available: for small amounts of data, the best results are obtained using every observation to maximum effect, and employing the quadratic time MMD_u^2 B test. When large volumes of data are available, a better option is to look at each point only once, which can yield lower Type II error for a given computational cost. It may also be worth doing a t-test first in this case, and only running more sophisticated nonparametric tests if the t-test accepts the null hypothesis, to verify the distributions are identical in more than just mean.

8.4 Attribute Matching

Our final series of experiments addresses automatic attribute matching. Given two databases, we want to detect corresponding attributes in the schemas of these databases, based on their data-content (as a simple example, two databases might have respective fields Wage and Salary, which are assumed to be observed via a subsampling of a particular population, and we wish to automatically determine that both Wage and Salary denote to the same underlying attribute). We use a two-sample test on pairs of attributes from two databases to find corresponding pairs.[15] This procedure

[15] Note that corresponding attributes may have different distributions in real-world databases. Hence, schema matching cannot solely rely on distribution testing.

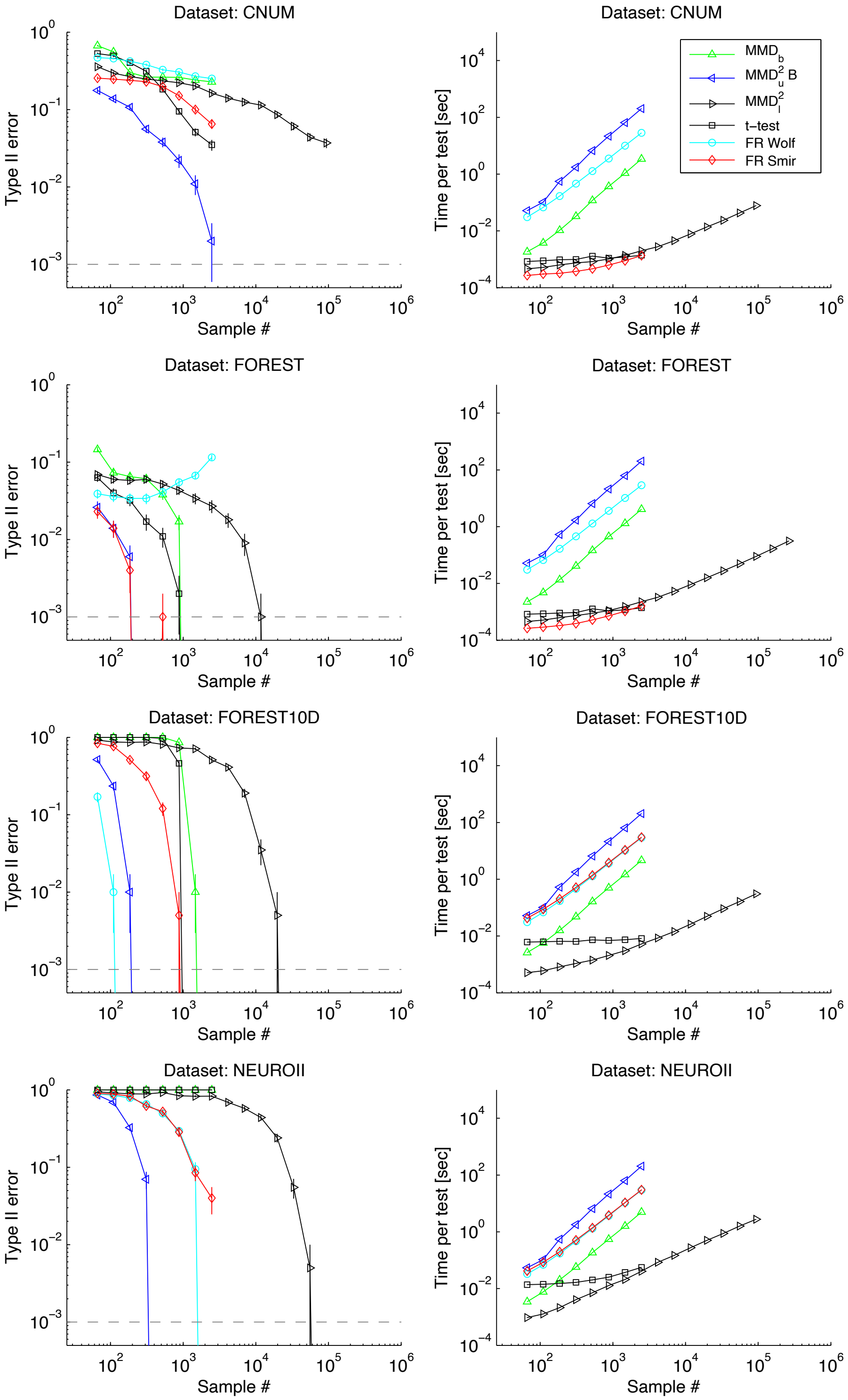


Figure 6: Linear-cost vs quadratic-cost MMD. The first column shows Type II performance, and the second shows runtime. The dashed grey horizontal line indicates zero Type II error (required due to log y-axis).

is also called *table matching* for tables from different databases. We performed attribute matching as follows: first, the data set D was split into two halves A and B. Each of the n attributes in A (and B, resp.) was then represented by its instances in A (resp. B). We then tested all pairs of attributes from A and from B against each other, to find the optimal assignment of attributes $A_1,\dots,A_n$ from A to attributes $B_1,\dots,B_n$ from B. We assumed that A and B contain the same number of attributes.

As a naive approach, we could assume that any possible pair of attributes might correspond, and thus that every attribute of A needs to be tested against all the attributes of B to find the optimal match. We report results for this naive approach, aggregated over all pairs of possible attribute matches, in Table 2. We used three data sets: the census income data set from the UCI KDD archive (CNUM), the protein homology data set from the 2004 KDD Cup (BIO) (Caruana and Joachims, 2004), and the forest data set from the UCI ML archive (Blake and Merz, 1998). For the final data set, we performed univariate matching of attributes (FOREST) and multivariate matching of tables (FOREST10D) from two different databases, where each table represents one type of forest. Both our asymptotic MMD_u^2-based tests perform as well as or better than the alternatives, notably for CNUM, where the advantage of MMD_u^2 is large. Unlike in Table 1, the next best alternatives are not consistently the same across all data: for example, in BIO they are *Wolf* or *Hall*, whereas in FOREST they are *Smir*, *Biau*, or the t-test. Thus, MMD_u^2 appears to perform more consistently across the multiple data sets. The Friedman-Rafsky tests do not always return a Type I error close to the design parameter: for instance, *Wolf* has a Type I error of 9.7% on the BIO data set (on these data, MMD_u^2 has the joint best Type II error without compromising the designed Type I performance). Finally, MMD_b performs much better than in Table 1, although surprisingly it fails to reliably detect differences in FOREST10D. The results of MMD_u^2 H are also improved, although it remains among the worst performing methods.

A more principled approach to attribute matching is also possible. Assume that $\phi(A)=(\phi_1(A_1),\phi_2(A_2),...,\phi_n(A_n))$: in other words, the kernel decomposes into kernels on the individual attributes of A (and also decomposes this way on the attributes of B). In this case, MMD^2 can be written $\sum_{i=1}^n \|\mu_i(A_i)-\mu_i(B_i)\|^2$, where we sum over the MMD terms on each of the attributes. Our goal of optimally assigning attributes from B to attributes of A via MMD is equivalent to finding the optimal permutation π of attributes of B that minimizes $\sum_{i=1}^n \|\mu_i(A_i)-\mu_i(B_{\pi(i)})\|^2$. If we define $C_{ij}=\|\mu_i(A_i)-\mu_i(B_j)\|^2$, then this is the same as minimizing the sum over $C_{i,\pi(i)}$. This is the linear assignment problem, which costs $O(n^3)$ time using the Hungarian method (Kuhn, 1955).

While this may appear to be a crude heuristic, it nonetheless defines a semi-metric on the sample spaces X and Y and the corresponding distributions p and q. This follows from the fact that matching distances are proper metrics if the matching cost functions are metrics. We formalize this as follows:

Theorem 28 *Let p,q be distributions on $\mathbb{R}^d$ and denote by p_i,q_i the marginal distributions on the i-th variable. Moreover, denote by Π the symmetric group on $\{1,\dots,d\}$. The following distance, obtained by optimal coordinate matching, is a semi-metric.*

$$\Delta[\mathcal{F},p,q]:=\min_{\pi\in\Pi}\sum_{i=1}^{d}\mathrm{MMD}[\mathcal{F},p_i,q_{\pi(i)}].$$

Proof Clearly $\Delta[\mathcal{F},p,q]$ is nonnegative, since it is a sum of nonnegative quantities. Next we show the triangle inequality. Denote by r a third distribution on $\mathbb{R}^d$ and let $\pi_{p,q},\pi_{q,r}$ and $\pi_{p,r}$ be the distance minimizing permutations over the associated pairs from $\{p,q,r\}$. It follows that

$$\begin{aligned}\Delta[\mathcal{F},p,q]+\Delta[\mathcal{F},q,r] &= \sum_{i=1}^{d}\mathrm{MMD}[\mathcal{F},p_i,q_{\pi_{p,q}(i)}]+\sum_{i=1}^{d}\mathrm{MMD}[\mathcal{F},q_i,r_{\pi_{q,r}(i)}]\\ &\geq \sum_{i=1}^{d}\mathrm{MMD}[\mathcal{F},p_i,r_{[\pi_{p,q}\circ\pi_{q,r}](i)}]\geq\Delta[\mathcal{F},p,r].\end{aligned}$$

The first inequality follows from the triangle inequality on MMD,

$$\mathrm{MMD}[\mathcal{F},p_i,q_{\pi_{p,q}(i)}]+\mathrm{MMD}[\mathcal{F},q_{\pi_{p,q}(i)},r_{[\pi_{p,q}\circ\pi_{q,r}](i)}]\geq\mathrm{MMD}[\mathcal{F},p_i,r_{[\pi_{p,q}\circ\pi_{q,r}](i)}].$$

The second inequality is a result of minimization over π. ■

We tested this 'Hungarian approach' to attribute matching via MMD_u^2 B on three univariate data sets (BIO, CNUM, FOREST) and for table matching on a fourth (FOREST10D). To study MMD_u^2 B on structured data, we used two data sets of protein graphs (PROTEINS and ENZYMES) and used the graph kernel for proteins from Borgwardt et al. (2005) for table matching via the Hungarian method (the other tests were not applicable to these graph data). The challenge here is to match tables representing one functional class of proteins (or enzymes) from data set A to the corresponding tables (functional classes) in B. Results are shown in Table 3. Besides on the BIO and CNUM data sets, MMD_u^2 B made no errors.

Data Set	Data type	No. attributes	Sample size	Repetitions	% correct
BIO	univariate	6	377	100	90.0
CNUM	univariate	13	386	100	99.8
FOREST	univariate	10	538	100	100.0
FOREST10D	multivariate	2	1000	100	100.0
ENZYME	structured	6	50	50	100.0
PROTEINS	structured	2	200	50	100.0

Table 3: Hungarian Method for attribute matching via MMD_u^2 B on univariate (BIO, CNUM, FOREST), multivariate (FOREST10D), and structured (ENZYMES, PROTEINS) data ($\alpha = 0.05$; "% correct" is the percentage of correct attribute matches over all repetitions).

9. Conclusion

We have established three simple multivariate tests for comparing two distributions p and q, based on samples of size m and n from these respective distributions. Our test statistic is the maximum mean discrepancy (MMD), defined as the maximum deviation in the expectation of a function evaluated on each of the random variables, taken over a sufficiently rich function class: in our case, a reproducing kernel Hilbert space (RKHS). Equivalently, the statistic can be written as the norm of the difference between distribution feature means in the RKHS. We do not require density estimates as an intermediate step. Two of our tests provide Type I error bounds that are exact and distribution-free for finite sample sizes. We also give a third test based on quantiles of the asymptotic distribution of the associated test statistic. All three tests can be computed in $O((m+n)^2)$ time, however when sufficient data are available, a linear time statistic can be used, which in our experiments was able to achieve a given Type II error at smaller computational cost, by looking at many more samples than the quadratic-cost tests.

We have seen in Section 7 that several classical metrics on probability distributions can be written as integral probability metrics with function classes that are not Hilbert spaces, but rather Banach or seminormed spaces (for instance the Kolmogorov-Smirnov and Earth Mover's distances). It is therefore of interest to establish under what conditions one could write these discrepancies in terms of norms of differences of mean embeddings. Sriperumbudur et al. (2011b) provide expressions for the maximum mean discrepancy in terms of mean embeddings in reproducing kernel Banach spaces. When the Banach space is not an RKBS, the question of establishing a mean embedding interpretation for the MMD remains open.

We also note (following Section 7.3) that the MMD for RKHSs is associated with a particular kernel between probability distributions. Hein et al. (2004) describe several further such kernels, which induce corresponding distances between feature space distribution mappings: these may in turn lead to new and powerful two-sample tests.

Two recent studies have shown that additional divergence measures between distributions can be obtained empirically through optimization in a reproducing kernel Hilbert space. Harchaoui et al. (2008) define a two-sample test statistic arising from the kernel Fisher discriminant, rather than the difference of RKHS means; and Nguyen et al. (2008) obtain a KL divergence estimate by approximating the ratio of densities (or its log) with a function in an RKHS. By design, both these kernel-based statistics prioritise different features of p and q when measuring the divergence between distributions, and the resulting effects on distinguishability of distributions are therefore of interest.

Acknowledgments

We would like to thank the anonymous referees, whose suggestions greatly improved the paper; Bharath Sriperumbudur, for thoroughly proofreading the final draft; Sivaraman Balakrishnan, Philipp Berens, Olivier Bousquet, Corinna Cortes, Omri Guttman, Peter Hall, Matthias Hein, John Langford, Mehryar Mohri, Novi Quadrianto, Le Song, and Vishy Vishwanathan, for constructive discussions; Patrick Warnat (DKFZ, Heidelberg), for providing the microarray data sets; and Nikos Logothetis, for providing the neural data sets. National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, and by the Austrian Science Fund (FWF), project # S9102-N04.

Appendix A. Large Deviation Bounds for Tests with Finite Sample Guarantees

This section contains proofs of the theorems of Section 4.1. We begin in Section A.1 with a review of McDiarmid's inequality and the Rademacher average of a function class. We prove Theorem 7 in Section A.2, and Theorem 8 in Section A.3.

A.1 Preliminary Definitions and Theorems

We need the following theorem, due to McDiarmid (1989).

Theorem 29 (McDiarmid's inequality) *Let $f : \mathcal{X}^m \to \mathbb{R}$ be a function such that for all $i \in \{1,\dots,m\}$, there exist $c_i < \infty$ for which*

$$\sup_{X \in \mathcal{X}^m, \tilde{x} \in \mathcal{X}} |f(x_1,\dots x_m) - f(x_1,\dots x_{i-1},\tilde{x},x_{i+1},\dots,x_m)| \leq c_i.$$

Then for all probability measures p and every $\varepsilon > 0$,

$$\Pr_X\left(f(X) - \mathbf{E}_X(f(X)) > t\right) < \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right),$$

where $\mathbf{E}_X$ denotes the expectation over the m random variables $x_i \sim p$, and $\Pr_X$ denotes the probability over these m variables.

We also define the Rademacher average of the function class $\mathcal{F}$ with respect to the m-sample X.

Definition 30 (Rademacher average of $\mathcal{F}$ on X) *Let $\mathcal{F}$ be the unit ball in an RKHS on the domain $\mathcal{X}$, with kernel bounded according to $0 \leq k(x,y) \leq K$. Let X be an i.i.d. sample of size m drawn according to a probability measure p on $\mathcal{X}$, and let σ_i be i.i.d and take values in $\{-1,1\}$ with equal probability. We define the Rademacher average*

$$\begin{aligned} R_m(\mathcal{F},X) &:= \mathbf{E}_\sigma \sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right| \\ &\leq (K/m)^{1/2}, \end{aligned}$$

where the upper bound is due to Bartlett and Mendelson (2002, Lemma 22), and $\mathbf{E}_\sigma$ denotes the expectation over all the σ_i. Similarly, we define

$$R_m(\mathcal{F},p) := \mathbf{E}_{x,\sigma} \sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i) \right|.$$

A.2 Bound when p and q May Differ

We want to show that the absolute difference between $\mathrm{MMD}(\mathcal{F},p,q)$ and $\mathrm{MMD}_b(\mathcal{F},X,Y)$ is close to its expected value, independent of the distributions p and q. To this end, we prove three intermediate results, which we then combine. The first result we need is an upper bound on the absolute difference between $\mathrm{MMD}(\mathcal{F},p,q)$ and $\mathrm{MMD}_b(\mathcal{F},X,Y)$. We have

$$\begin{aligned} & |\mathrm{MMD}(\mathcal{F},p,q) - \mathrm{MMD}_b(\mathcal{F},X,Y)| \\ = & \left| \sup_{f \in \mathcal{F}} \left(\mathbf{E}_x(f) - \mathbf{E}_y(f)\right) - \sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^m f(x_i) - \frac{1}{n} \sum_{i=1}^n f(y_i) \right) \right| \\ \leq & \underbrace{\sup_{f \in \mathcal{F}} \left| \mathbf{E}_x(f) - \mathbf{E}_y(f) - \frac{1}{m} \sum_{i=1}^m f(x_i) + \frac{1}{n} \sum_{i=1}^n f(y_i) \right|}_{\Delta(p,q,X,Y)}. \end{aligned} \tag{14}$$

Second, we provide an upper bound on the difference between $\Delta(p,q,X,Y)$ and its expectation. Changing either of x_i or y_i in $\Delta(p,q,X,Y)$ results in changes in magnitude of at most $2K^{1/2}/m$ or $2K^{1/2}/n$, respectively. We can then apply McDiarmid's theorem, given a denominator in the exponent of

$$m\left(2K^{1/2}/m\right)^2+n\left(2K^{1/2}/n\right)^2=4K\left(\frac{1}{m}+\frac{1}{n}\right)=4K\frac{m+n}{mn},$$

to obtain

$$\Pr_{X,Y}\left(\Delta(p,q,X,Y)-\mathbf{E}_{X,Y}\left[\Delta(p,q,X,Y)\right]>\varepsilon\right)\leq\exp\left(-\frac{\varepsilon^2 mn}{2K(m+n)}\right). \tag{15}$$

For our final result, we exploit symmetrisation, following, for example, van der Vaart and Wellner (1996, p. 108), to upper bound the expectation of $\Delta(p,q,X,Y)$. Denoting by X' an i.i.d sample of size m drawn independently of X (and likewise for Y'), we have

$$\begin{aligned}
& \mathbf{E}_{X,Y}\left[\Delta(p,q,X,Y)\right] \\
= \; & \mathbf{E}_{X,Y}\sup_{f\in\mathcal{F}}\left|\mathbf{E}_x(f)-\frac{1}{m}\sum_{i=1}^m f(x_i)-\mathbf{E}_y(f)+\frac{1}{n}\sum_{i=1}^n f(y_j)\right| \\
= \; & \mathbf{E}_{X,Y}\sup_{f\in\mathcal{F}}\left|\mathbf{E}_{X'}\left(\frac{1}{m}\sum_{i=1}^m f(x'_i)\right)-\frac{1}{m}\sum_{i=1}^m f(x_i)-\mathbf{E}_{Y'}\left(\frac{1}{n}\sum_{i=1}^n f(y'_j)\right)+\frac{1}{n}\sum_{i=1}^n f(y_j)\right| \\
\underset{(a)}{\leq} \; & \mathbf{E}_{X,Y,X',Y'}\sup_{f\in\mathcal{F}}\left|\frac{1}{m}\sum_{i=1}^m f(x'_i)-\frac{1}{m}\sum_{i=1}^m f(x_i)-\frac{1}{n}\sum_{i=1}^n f(y'_j)+\frac{1}{n}\sum_{i=1}^n f(y_j)\right| \\
= \; & \mathbf{E}_{X,Y,X',Y',\sigma,\sigma'}\sup_{f\in\mathcal{F}}\left|\frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(x'_i)-f(x_i)\right)+\frac{1}{n}\sum_{i=1}^n \sigma'_i\left(f(y'_j)-f(y_j)\right)\right| \\
\underset{(b)}{\leq} \; & \mathbf{E}_{X,X',\sigma}\sup_{f\in\mathcal{F}}\left|\frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(x'_i)-f(x_i)\right)\right|+\mathbf{E}_{Y,Y',\sigma}\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^n \sigma_i\left(f(y'_j)-f(y_j)\right)\right| \\
\underset{(c)}{\leq} \; & 2\left[R_m(\mathcal{F},p)+R_n(\mathcal{F},q)\right]. \\
\underset{(d)}{\leq} \; & 2\left[(K/m)^{1/2}+(K/n)^{1/2}\right],
\end{aligned} \tag{16}$$

where (a) uses Jensen's inequality, (b) uses the triangle inequality, (c) substitutes Definition 30 (the Rademacher average), and (d) bounds the Rademacher averages, also via Definition 30.

Having established our preliminary results, we proceed to the proof of Theorem 7.

Proof (Theorem 7) Combining Equations (15) and (16), gives

$$\Pr_{X,Y}\left(\Delta(p,q,X,Y)-2\left[(K/m)^{1/2}+(K/n)^{1/2}\right]>\varepsilon\right)\leq\exp\left(-\frac{\varepsilon^2 mn}{2K(m+n)}\right).$$

Substituting Equation (14) yields the result.

■

A.3 Bound when $p = q$ and $m = n$

In this section, we derive the Theorem 8 result, namely the large deviation bound on the MMD when $p = q$ and $m = n$. Note also that we consider only positive deviations of $\mathrm{MMD}_b(\mathcal{F}, X, Y)$ from $\mathrm{MMD}(\mathcal{F}, p, q)$, since negative deviations are irrelevant to our hypothesis test. The proof follows the same three steps as in the previous section. The first step in (14) becomes

$$\begin{aligned}\mathrm{MMD}_b(\mathcal{F},X,Y) - \mathrm{MMD}(\mathcal{F},p,q) &= \mathrm{MMD}_b(\mathcal{F},X,X') - 0 \\ &= \sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - f(x_i')\right)\right). \qquad (17)\end{aligned}$$

The McDiarmid bound on the difference between (17) and its expectation is now a function of $2m$ observations in (17), and has a denominator in the exponent of $2m\left(2K^{1/2}/m\right)^2 = 8K/m$. We use a different strategy in obtaining an upper bound on the expected (17), however: this is now

$$\begin{aligned}&\mathbf{E}_{X,X'}\left[\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - f(x_i')\right)\right] \\ =\ & \frac{1}{m}\mathbf{E}_{X,X'}\left\|\sum_{i=1}^{m}\left(\phi(x_i) - \phi(x_i')\right)\right\| \\ =\ & \frac{1}{m}\mathbf{E}_{X,X'}\left[\sum_{i=1}^{m}\sum_{j=1}^{m}\left(k(x_i,x_j) + k(x_i',x_j') - k(x_i,x_j') - k(x_i',x_j)\right)\right]^{\frac{1}{2}} \\ \leq\ & \frac{1}{m}\left[2m\mathbf{E}_x k(x,x) + 2m(m-1)\mathbf{E}_{x,x'}k(x,x') - 2m^2\mathbf{E}_{x,x'}k(x,x')\right]^{\frac{1}{2}} \\ =\ & \left[\frac{2}{m}\mathbf{E}_{x,x'}\left(k(x,x) - k(x,x')\right)\right]^{\frac{1}{2}} \qquad (18) \\ \leq\ & (2K/m)^{1/2}. \qquad (19)\end{aligned}$$

We remark that both (18) and (19) bound the amount by which our biased estimate of the population MMD exceeds zero under $\mathcal{H}_0$. Combining the three results, we find that under $\mathcal{H}_0$,

$$\begin{aligned}\Pr_{X,X'}\left(\mathrm{MMD}_b(\mathcal{F},X,X') - \left[\frac{2}{m}\mathbf{E}_{x,x'}\left(k(x,x) - k(x,x')\right)\right]^{\frac{1}{2}} > \varepsilon\right) &< \exp\left(\frac{-\varepsilon^2 m}{4K}\right) \quad \text{and} \\ \Pr_{X,X'}\left(\mathrm{MMD}_b(\mathcal{F},X,X') - (2K/m)^{1/2} > \varepsilon\right) &< \exp\left(\frac{-\varepsilon^2 m}{4K}\right).\end{aligned}$$

Appendix B. Proofs for Asymptotic Tests

We derive results needed in the asymptotic test of Section 5. Appendix B.1 describes the distribution of the empirical MMD under $\mathcal{H}_0$ (i.e., $p = q$). Appendix B.2 establishes consistency of the test under local departures from $\mathcal{H}_0$. Appendix B.3 contains derivations of the second and third moments of the empirical MMD, also under $\mathcal{H}_0$.

B.1 Convergence of the Empirical MMD under $\mathcal{H}_0$

In this appendix, we prove Theorem 12, which describes the distribution of the unbiased estimator $\mathrm{MMD}_u^2[\mathcal{F},X,Y]$ under the null hypothesis. Thus, throughout this section, the reader should bear in mind that y now has the same distribution as x, that is, $y \sim p$. We first recall from Lemma 6 in Section 2.2 the population expression,

$$\mathrm{MMD}^2[\mathcal{F},p,q] := \mathbf{E}_{x,x'}k(x,x') + \mathbf{E}_{y,y'}k(y,y') - 2\mathbf{E}_{x,y}k(x,y),$$

and its empirical counterpart,

$$\begin{aligned}\mathrm{MMD}_u^2[\mathcal{F},X,Y] &= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}k(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}k(y_i,y_j)\\ &\quad -\frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}k(x_i,y_j).\end{aligned} \tag{20}$$

We begin with the asymptotic analysis of $\mathrm{MMD}_u^2[\mathcal{F},X,Y]$ under the null hypothesis. This is based on the reasoning of Anderson et al. (1994, Appendix), bearing in mind the following changes:

- we do not need to deal with the bias terms S_{1j} in Anderson et al. (1994, Appendix) that vanish for large sample sizes, since our statistic is unbiased;
- we require greater generality, since our kernels are not necessarily inner products in L_2 between probability density functions (although this is a special case: see Section 3.3.1).

We first transform each term in the sum (20) by centering. Under $\mathcal{H}_0$, both x and y have the same mean embedding μ_p. Thus we replace each instance of $k(x_i,x_j)$ in the sum with a kernel $\tilde{k}(x_i,x_j)$ between feature space mappings from which the mean has been subtracted,

$$\begin{aligned}\tilde{k}(x_i,x_j) &:= \left\langle \phi(x_i)-\mu_p, \phi(x_j)-\mu_p \right\rangle_{\mathcal{H}}\\ &= k(x_i,x_j) - \mathbf{E}_x k(x_i,x) - \mathbf{E}_x k(x,x_j) + \mathbf{E}_{x,x'}k(x,x').\end{aligned}$$

The centering terms cancel across the three terms (the distance between the two points is unaffected by an identical global shift in both the points). This gives the equivalent form of the empirical MMD,

$$\begin{aligned}\mathrm{MMD}_u^2[\mathcal{F},X,Y] &= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\tilde{k}(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\tilde{k}(y_i,y_j)\\ &\quad -\frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{k}(x_i,y_j),\end{aligned} \tag{21}$$

where each of the three sums has expected value zero. Note in particular that the U-statistics in $\tilde{k}(x_i,x_j)$ are degenerate, meaning

$$\mathbf{E}_x\tilde{k}(x,v) = \mathbf{E}_x k(x,v) - \mathbf{E}_{x,x'}k(x,x') - \mathbf{E}_x k(x,v) + \mathbf{E}_{x,x'}k(x,x') = 0. \tag{22}$$

We define the operator $S_{\tilde{k}} : L_2(p) \to \mathcal{F}$ satisfying

$$S_{\tilde{k}}g(x) := \int_{\mathcal{X}} \tilde{k}(x,x')g(x')dp(x').$$

According to Reed and Simon (1980, Theorem VI.23), this operator is Hilbert-Schmidt, and hence compact, if and only if the kernel $\tilde{k}$ is square integrable under p,

$$\tilde{k} \in L_2\left(\mathcal{X} \times \mathcal{X}, p \times p\right). \tag{23}$$

We may write the kernel $\tilde{k}(x_i, x_j)$ in terms of eigenfunctions $\psi_l(x)$ with respect to the probability measure p,

$$\tilde{k}(x, x') = \sum_{l=1}^{\infty} \lambda_l \psi_l(x) \psi_l(x'), \tag{24}$$

where

$$\begin{aligned} \int_{\mathcal{X}} \tilde{k}(x, x') \psi_i(x) dp(x) &= \lambda_i \psi_i(x'), \\ \int_{\mathcal{X}} \psi_i(x) \psi_j(x) dp(x) &= \delta_{ij}, \end{aligned} \tag{25}$$

and the convergence is in $L_2\left(\mathcal{X} \times \mathcal{X}, p \times p\right)$. Since the operator is Hilbert-Schmidt, we have by Reed and Simon (1980, Theorem VI.22) that $\sum \lambda_i^2 < \infty$.

Using the degeneracy of the U-statistic in (22), then when $\lambda_i \neq 0$,

$$\begin{aligned} \lambda_i \mathbf{E}_{x'} \psi_i(x') &= \int_{\mathcal{X}} \mathbf{E}_{x'} \tilde{k}(x, x') \psi_i(x) dp(x) \\ &= 0, \end{aligned}$$

and hence

$$\mathbf{E}_x \psi_i(x) = 0. \tag{26}$$

In other words, the eigenfunctions $\psi_i(x)$ are zero mean and uncorrelated.

We now use these results to find the asymptotic distribution of (21). First,

$$\begin{aligned} \frac{1}{m} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \tilde{k}(x_i, x_j) &= \frac{1}{m} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \sum_{l=1}^{\infty} \lambda_l \psi_l(x_i) \psi_l(x_j) \\ &= \frac{1}{m} \sum_{l=1}^{\infty} \lambda_l \left(\left(\sum_i \psi_l(x_i) \right)^2 - \sum_i \psi_l^2(x_i) \right) \\ &\underset{D}{\rightarrow} \sum_{l=1}^{\infty} \lambda_l (a_l^2 - 1), \end{aligned} \tag{27}$$

where $a_l \sim \mathcal{N}(0,1)$ are i.i.d., and the final relation denotes convergence in distribution, which is proved by Serfling (1980, Section 5.5.2) using (25) and (26).[16] Given that the random variables a_l^2 are zero mean with finite variance, it can be shown either via Kolmogorov's inequality or by the Martingale convergence theorem that the above sum converges almost surely if $\sum_{l=1}^{\infty} \lambda_l^2 < \infty$ (Grimmet and Stirzaker, 2001, Chapter 7.11 Exercise 30). As we have seen, this is guaranteed under the assumption (23).

Likewise

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \tilde{k}(y_i, y_j) \underset{D}{\rightarrow} \sum_{l=1}^{\infty} \lambda_l (b_l^2 - 1),$$

[16] Simply replace $\tilde{h}_2(x_i, x_j)$ with $\tilde{k}(x_i, x_j)$ in Serfling (1980, top of p. 196).

where $b_l \sim \mathcal{N}(0,1)$ independent of the a_l, and

$$\frac{1}{\sqrt{mn}}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{k}(x_i,y_j) \underset{D}{\rightarrow} \sum_{l=1}^{\infty}\lambda_l a_l b_l, \tag{28}$$

both jointly in distribution with (27), where (28) is proved at the end of the section. We now combine these results. Define $t = m+n$, and assume $\lim_{m,n\to\infty} m/t \to \rho_x$ and $\lim_{m,n\to\infty} n/t \to \rho_y := (1-\rho_x)$ for fixed $0 < \rho_x < 1$. Then

$$\begin{aligned} t\mathrm{MMD}_u^2[\mathcal{F},X,Y] & \underset{D}{\rightarrow} \rho_x^{-1}\sum_{l=1}^{\infty}\lambda_l(a_l^2-1)+\rho_y^{-1}\sum_{l=1}^{\infty}\lambda_l(b_l^2-1)-\frac{2}{\sqrt{\rho_x\rho_y}}\sum_{l=1}^{\infty}\lambda_l a_l b_l \\ & = \sum_{l=1}^{\infty}\lambda_l\left[(\rho_x^{-1/2}a_l-\rho_y^{-1/2}b_l)^2-(\rho_x\rho_y)^{-1}\right]. \end{aligned}$$

Proof (Equation 28) The proof is a modification of the result for convergence of degenerate U-statistics of Serfling (1980, Section 5.5.2). We only provide those details that differ from the proof of Serfling, and otherwise refer to the steps in the original proof as needed. First, using (24) to expand out the centred kernel, we may write

$$T_{mn} := \frac{1}{\sqrt{mn}}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{k}(x_i,y_j) = \frac{1}{\sqrt{mn}}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{\infty}\lambda_l\psi_l(x_i)\psi_l(y_j).$$

We define a truncation of this sum,

$$T_{mnL} := \frac{1}{\sqrt{mn}}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{L}\lambda_l\psi_l(x_i)\psi_l(y_j).$$

The target distribution is written

$$V = \sum_{l=1}^{\infty}\lambda_l a_l b_l,$$

and its truncation is

$$V_L := \sum_{l=1}^{L}\lambda_l a_l b_l.$$

Our goal is to show

$$\left|\mathbf{E}_{X,Y}\left(e^{\imath sT_{mn}}\right)-\mathbf{E}_{a,b}\left(e^{\imath sV}\right)\right|$$

vanishes for all s as m and n increase, where the expectation $\mathbf{E}_{X,Y}$ is over all sample points, which implies $T_{mn} \underset{D}{\rightarrow} V$ (Dudley, 2002, Theorem 9.8.2). We achieve this via the upper bound

$$\begin{aligned} \left|\mathbf{E}_{X,Y}\left(e^{\imath sT_{mn}}\right)-\mathbf{E}_{a,b}\left(e^{\imath sV}\right)\right| & \leq \left|\mathbf{E}_{X,Y}\left(e^{\imath sT_{mn}}\right)-\mathbf{E}_{XY}\left(e^{\imath sT_{mnL}}\right)\right|+\left|\mathbf{E}_{XY}\left(e^{\imath sT_{mnL}}\right)-\mathbf{E}_{a,b}\left(e^{\imath sV_L}\right)\right| \\ & \quad +\left|\mathbf{E}_{a,b}\left(e^{\imath sV_L}\right)-\mathbf{E}_{a,b}\left(e^{\imath sV}\right)\right|, \end{aligned}$$

where we need to show that for large enough L, each of the three terms vanish.

First term: We first show that for large enough L, T_{mn} and T_{mnL} are close in distribution. From Serfling (1980, p. 197),

$$\left|\mathbf{E}_{X,Y}\left(e^{\imath sT_{mn}}\right)-\mathbf{E}_{X,Y}\left(e^{\imath sT_{mnL}}\right)\right| \leq |s|\left[\mathbf{E}_{X,Y}\left(T_{mn}-T_{mnL}\right)^2\right]^{1/2},$$

and we may write the difference between the full sum and its truncation as

$$T_{mn} - T_{mnL} = \frac{1}{\sqrt{mn}} \sum_{i=1}^{m} \sum_{j=1}^{n} \underbrace{\left(\tilde{k}(x_i, y_j) - \sum_{l=1}^{L} \lambda_l \psi_l(x_i) \psi_l(y_j) \right)}_{g_K(x_i, y_j)}.$$

Each of the properties (Serfling, 1980, Equations (6a)-(6c) p. 197) still holds for g_K, namely

$$\begin{aligned} \mathbf{E}_{x,x'} \left(g_K(x,x') \right) &= 0, \\ \mathbf{E}_{x,x'} \left(g_K^2(x,x') \right) &= \sum_{l=L+1}^{\infty} \lambda_l^2, \\ \mathbf{E}_{x} \left(g_K(x,x') \right) &= 0. \end{aligned}$$

Then

$$\begin{aligned} \mathbf{E}_{X,Y} \left(T_{mn} - T_{mnL} \right)^2 &= \frac{1}{mn} \sum_{i=1}^{m} \sum_{q=1}^{m} \sum_{j=1}^{n} \sum_{r=1}^{n} \mathbf{E}_{x_i,x_q,y_j,y_r} \left[g_K(x_i, y_j) g_K(x_q, y_r) \right] \\ &= \begin{cases} \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{E}_{x,x'} \left(g_K^2(x,x') \right) & i = q \text{ and } j = r, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

where we have used that $p = q$ under $\mathcal{H}_0$, which allows us to replace $\mathbf{E}_{x,y}$ with $\mathbf{E}_{x,x'}$ in the final line. It follows that for large enough L,

$$\begin{aligned} |s| \left[\mathbf{E}_{X,Y} \left(T_{mn} - T_{mnL} \right)^2 \right]^{1/2} &= |s| \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{E}_{x,x'} \left(g_K^2(x,x') \right) \right]^{1/2} \\ &= |s| \left[\sum_{l=L+1}^{\infty} \lambda_l^2 \right]^{1/2} \\ &< \varepsilon. \end{aligned}$$

Second term: We show that

$$T_{mnL} \xrightarrow[D]{} V_L \qquad (29)$$

as $m \to \infty$ and $n \to \infty$. We rewrite T_{mnL} as

$$T_{mnL} = \sum_{l=1}^{L} \lambda_l \left(\frac{1}{\sqrt{m}} \sum_{i=1}^{m} \psi_l(x_i) \right) \left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi_l(y_j) \right).$$

Define the length L vectors W_m and W'_n having lth entries

$$W_{ml} = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} \psi_l(x_i), \qquad W'_{nl} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi_l(y_j),$$

respectively. These have mean and covariance

$$\mathbf{E}_X(W_{ml}) = 0, \qquad \mathrm{Cov}_{X,Y}(W_{ml}, W_{ml'}) = \begin{cases} 1 & l = l', \\ 0 & l \neq l'. \end{cases}$$

Moreover, the vectors W_m and W'_n are independent. The result (29) then holds by the Lindberg-Lévy CLT (Serfling, 1980, Theorem 1.9.1A).

Third term: From Serfling (1980, p. 199), we have

$$\left|\mathbf{E}_{a,b}\left(e^{\imath s V_L}\right) - \mathbf{E}_{a,b}\left(e^{\imath s V}\right)\right| \leq |s| \left[\mathbf{E}_{a,b}\left(V - V_L\right)^2\right]^{1/2}.$$

We can bound the right hand term by

$$\begin{aligned}
\mathbf{E}_{a,b}\left(V - V_L\right)^2 &= \mathbf{E}_{a,b}\left(\sum_{l=L+1}^{\infty} \lambda_l a_l b_l\right)^2 \\
&= \sum_{l=L+1}^{\infty} \lambda_l^2 \mathbf{E}_y\left(a_l^2\right) \mathbf{E}_z\left(b_l^2\right) \\
&= \sum_{l=L+1}^{\infty} \lambda_l^2 \\
&\leq \varepsilon
\end{aligned}$$

for L sufficiently large. ■

B.2 Alternative Distribution: Consistency Against Local Alternatives

We prove Theorem 13, which gives the power against a local alternative hypothesis of a two-sample test based on MMD_u^2. The proof modifies a result of Anderson et al. (1994, Section 2.4), where we consider a more general class of local departures from the null hypothesis (rather than the class of perturbed densities described in Section 3.3.1).

First, we recall our test statistic,

$$\begin{aligned}
\mathrm{MMD}_u^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) \\
+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j).
\end{aligned}$$

We begin by transforming this statistic by centering the samples X and Y in feature space by μ_p and μ_q, respectively; unlike the $\mathcal{H}_0$ case, however, $\mu_p \neq \mu_q$, and the new statistic MMD_c^2 is *not* the same as MMD_u^2. The first term is centered as in (9). The second and third terms are respectively replaced by

$$\frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left\langle \phi(y_i) - \mu_q, \phi(y_j) - \mu_q \right\rangle_{\mathcal{H}}$$

and

$$\frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\langle \phi(x_i) - \mu_p, \phi(y_j) - \mu_q \right\rangle_{\mathcal{H}}.$$

The resulting centred statistic is

$$\begin{aligned}\mathrm{MMD}_c^2[\mathcal{F},X,Y] &= \frac{1}{m(m-1)}\sum_{i=1}^m\sum_{j\neq i}^m \left\langle \phi(x_i)-\mu_p,\phi(x_j)-\mu_p\right\rangle_{\mathcal{H}} \\ &+\frac{1}{n(n-1)}\sum_{i=1}^n\sum_{j\neq i}^n \left\langle \phi(y_i)-\mu_q,\phi(y_j)-\mu_q\right\rangle_{\mathcal{H}}-\frac{2}{mn}\sum_{i=1}^m\sum_{j=1}^n \left\langle \phi(x_i)-\mu_p,\phi(y_j)-\mu_q\right\rangle_{\mathcal{H}}.\end{aligned}$$

We write $\mu_q=\mu_p+g_t$, where $g_t\in\mathcal{H}$ is chosen such that μ_p+g_t remains a valid distribution embedding, and $\|g_t\|_{\mathcal{H}}$ can be made to approach zero to describe local departures from the null hypothesis. The difference between the original statistic and the centred statistic is then

$$\begin{aligned}&\mathrm{MMD}_u^2[\mathcal{F},X,Y]-\mathrm{MMD}_c^2[\mathcal{F},X,Y] \\ =&\frac{2}{m}\sum_{i=1}^m \langle\mu_p,\phi(x_i)\rangle_{\mathcal{H}}-\langle\mu_p,\mu_p\rangle_{\mathcal{H}}+\frac{2}{n}\sum_{i=1}^n\left\langle\mu_q,\phi(y_i)\right\rangle_{\mathcal{H}}-\left\langle\mu_q,\mu_q\right\rangle_{\mathcal{H}} \\ &\qquad -\frac{2}{m}\sum_{i=1}^m\left\langle\mu_q,\phi(x_i)\right\rangle_{\mathcal{H}}-\frac{2}{n}\sum_{i=1}^n\langle\mu_p,\phi(y_i)\rangle_{\mathcal{H}}+2\left\langle\mu_p,\mu_q\right\rangle_{\mathcal{H}} \\ =&\frac{2}{n}\sum_{i=1}^n\left\langle g_t,\phi(y_i)-\mu_q\right\rangle_{\mathcal{H}}-\frac{2}{m}\sum_{i=1}^m\langle g_t,\phi(x_i)-\mu_p\rangle_{\mathcal{H}}+\langle g_t,g_t\rangle_{\mathcal{H}}.\end{aligned}$$

We next show g_t can be used to encode a local departure from the null hypothesis. Define $t=m+n$, and assume $\lim_{m,n\to\infty}m/t\to\rho_x$ and $\lim_{m,n\to\infty}n/t\to\rho_y:=(1-\rho_x)$ where $0<\rho_x<1$. Consider the case where the departure from the null hypothesis satisfies $\|g_t\|_{\mathcal{H}}=ct^{-1/2}$. Then, as $t\to\infty$,

$$t\mathrm{MMD}_c^2[\mathcal{F},X,Y]\xrightarrow[D]{}\sum_{l=1}^\infty\lambda_l\left[(\rho_x^{-1/2}a_l+\rho_y^{-1/2}b_l)^2-(\rho_x\rho_y)^{-1}\right]=:S$$

as before, since the distance between μ_p and μ_q vanishes for large t (as $\|g_t\|_{\mathcal{H}}\to 0$). Next, the terms

$$\frac{1}{\sqrt{n}}\sum_{i=1}^n\left\langle\frac{g_t}{\|g_t\|_{\mathcal{H}}},\phi(y_i)-\mu_q\right\rangle_{\mathcal{H}}\quad\text{and}\quad\frac{1}{\sqrt{m}}\sum_{i=1}^m\left\langle\frac{g_t}{\|g_t\|_{\mathcal{H}}},\phi(x_i)-\mu_p\right\rangle_{\mathcal{H}}$$

in the difference between MMD_u^2 and MMD_c^2 are straightforward sums of independent zero mean random variables, and have Gaussian asymptotic distribution. Defining u_y to be the zero mean Gaussian random variable associated with the first term,

$$\begin{aligned}\frac{t}{n}\sum_{i=1}^n\left\langle g_t,\phi(y_i)-\mu_q\right\rangle_{\mathcal{H}} &= \frac{t}{n}\left(ct^{-1/2}\right)\sum_{i=1}^n\left\langle\frac{g_t}{\|g_t\|_{\mathcal{H}}},\phi(y_i)-\mu_q\right\rangle_{\mathcal{H}} \\ &\xrightarrow[D]{} c\rho_y^{-1/2}u_y.\end{aligned}$$

Likewise,

$$\frac{t}{m}\sum_{i=1}^m\langle g_t,\phi(x_i)-\mu_p\rangle_{\mathcal{H}}\xrightarrow[D]{}c\rho_x^{-1/2}u_x,$$

where u_x is a zero mean Gaussian random variable independent of u_y (note, however, that u_x and u_y are correlated with terms in S, and are defined on the same probability space as a_l and b_l in this sum). Finally,

$$t\,\langle g_t,g_t\rangle_{\mathcal{H}}=c^2.$$

This leads to our main result: given the threshold s_α, then

$$\Pr_{\mathcal{H}_A}\left(tMMD_u^2 > s_\alpha\right) \to \Pr\left(S + 2c\left(\rho_x^{-1/2}u_x - \rho_y^{-1/2}u_y\right) + c^2 > s_\alpha\right),$$

which is constant in t, and increases as $c \to \infty$. Thus, $\|g_t\|_{\mathcal{H}} = ct^{-1/2}$ is the minimum distance between μ_p and μ_q distinguishable by the asymptotic MMD-based test.

B.3 Moments of the Empirical MMD Under $\mathcal{H}_0$

In this section, we compute the moments of the U-statistic in Section 5 for $m = n$, under the null hypothesis conditions

$$\mathbf{E}_{z,z'} h(z,z') = 0, \tag{30}$$

and, importantly,

$$\mathbf{E}_{z'} h(z,z') = 0. \tag{31}$$

Note that the latter implies the former.

Variance/2nd moment: This was derived by Hoeffding (1948, p. 299), and is also described by Serfling (1980, Lemma A p. 183). Applying these results,

$$\begin{aligned}
&\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^2\right)\\
&= \left(\frac{2}{n(n-1)}\right)^2 \left[\frac{n(n-1)}{2}(n-2)(2)\mathbf{E}_z\left[(\mathbf{E}_{z'}h(z,z'))^2\right] + \frac{n(n-1)}{2}\mathbf{E}_{z,z'}\left[h^2(z,z')\right]\right]\\
&= \frac{2(n-2)}{n(n-1)}\mathbf{E}_z\left[(\mathbf{E}_{z'}h(z,z'))^2\right] + \frac{2}{n(n-1)}\mathbf{E}_{z,z'}\left[h^2(z,z')\right]\\
&= \frac{2}{n(n-1)}\mathbf{E}_{z,z'}\left[h^2(z,z')\right],
\end{aligned}$$

where the first term in the penultimate line is zero due to (31). Note that variance and 2nd moment are the same under the zero mean assumption.

3rd moment: We consider the terms that appear in the expansion of $\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^3\right)$. These are all of the form

$$\left(\frac{2}{n(n-1)}\right)^3 \mathbf{E}(h_{ab}h_{cd}h_{ef}),$$

where we shorten $h_{ab} = h(z_a, z_b)$, and we know z_a and z_b are always independent. Most of the terms vanish due to (30) and (31). The first terms that remain take the form

$$\left(\frac{2}{n(n-1)}\right)^3 \mathbf{E}(h_{ab}h_{bc}h_{ca}),$$

and there are

$$\frac{n(n-1)}{2}(n-2)(2)$$

of them, which gives us the expression

$$\begin{aligned}&\left(\frac{2}{n(n-1)}\right)^3 \frac{n(n-1)}{2}(n-2)(2)\mathbf{E}_{z,z'}\left[h(z,z')\mathbf{E}_{z''}\left(h(z,z'')h(z',z'')\right)\right] \\ &= \frac{8(n-2)}{n^2(n-1)^2}\mathbf{E}_{z,z'}\left[h(z,z')\mathbf{E}_{z''}\left(h(z,z'')h(z',z'')\right)\right]. \qquad (32)\end{aligned}$$

Note the scaling $\frac{8(n-2)}{n^2(n-1)^2} \sim \frac{1}{n^3}$. The remaining non-zero terms, for which $a = c = e$ and $b = d = f$, take the form

$$\left(\frac{2}{n(n-1)}\right)^3 \mathbf{E}_{z,z'}\left[h^3(z,z')\right],$$

and there are $\frac{n(n-1)}{2}$ of them, which gives

$$\left(\frac{2}{n(n-1)}\right)^2 \mathbf{E}_{z,z'}\left[h^3(z,z')\right].$$

However $\left(\frac{2}{n(n-1)}\right)^2 \sim n^{-4}$ so this term is negligible compared with (32). Thus, a reasonable approximation to the third moment is

$$\mathbf{E}\left(\left[\mathrm{MMD}_u^2\right]^3\right) \approx \frac{8(n-2)}{n^2(n-1)^2}\mathbf{E}_{z,z'}\left[h(z,z')\mathbf{E}_{z''}\left(h(z,z'')h(z',z'')\right)\right].$$

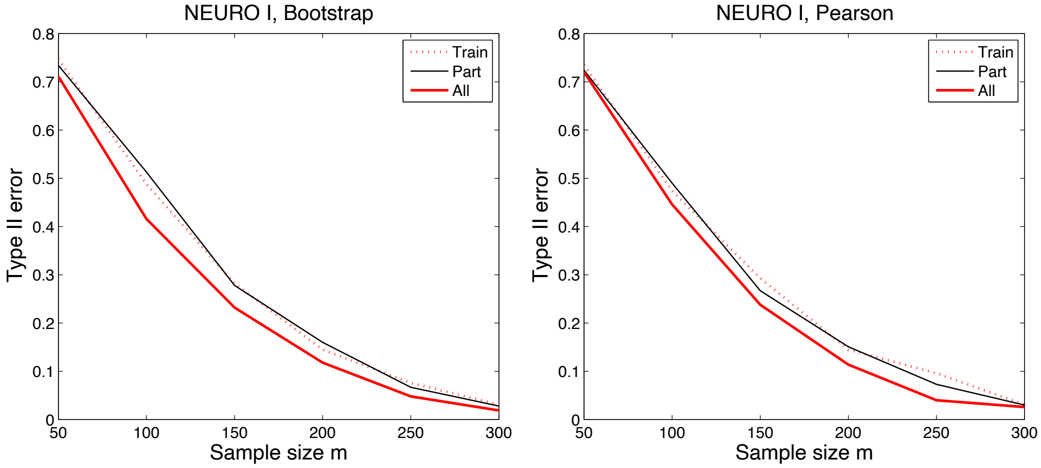
Appendix C. Empirical Evaluation of the Median Heuristic for Kernel Choice

In this appendix, we provide an empirical evaluation of the median heuristic for kernel choice, described at the start of Section 8: according to this heuristic, the kernel bandwidth is set at the median distance between points in the aggregate sample over p and q (in the case of a Gaussian kernel on $\mathbb{R}^d$). We investigated three kernel choice strategies: kernel selection on the entire sample from p and q; kernel selection on a hold-out set (10% of data), and testing on the remaining 90%; and kernel selection *and* testing on 90% of the available data. These strategies were evaluated on the Neural Data I data set described in Section 8.2, using a Gaussian kernel, and both the bootstrap and Pearson curve methods for selecting the test threshold. Results are plotted in Figure 7. We note that the Type II error of each approach follows the same trend. The Type II errors of the second and third approaches are indistinguishable, and the first approach has a slightly lower Type II error (as it is computed on slightly more data). In this instance, the null distribution with the kernel bandwidth set using the tested data is not substantially different to that obtained when a held-out set is used.

References

Y. Altun and A.J. Smola. Unifying divergence minimization and statistical inference via convex duality. In *Proc. Annual Conf. Computational Learning Theory*, LNCS, pages 139–153. Springer, 2006.

N. Anderson, P. Hall, and D. Titterington. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 50:41–54, 1994.


NEURO I, Bootstrap

NEURO I, Pearson

Figure 7: Type II error on the Neural Data I set, for kernel computed via the median heuristic on the full data set ("All"), kernel computed via the median heuristic on a 10% hold-out set ("Train"), and kernel computed via the median heuristic on 90% of the data ("Part"). Results are plotted over 1000 repetitions. **Left:** Bootstrap results. **Right:** Pearson curve results.

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.

M. Arcones and E. Giné. On the bootstrap of u and v statistics. *The Annals of Statistics*, 20(2): 655–674, 1992.

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

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, pages 137–144. MIT Press, 2007.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

G. Biau and L. Gyorfi. On the asymptotic properties of a nonparametric l_1-test statistic of homogeneity. *IEEE Transactions on Information Theory*, 51(11):3965–3973, 2005.

P. Bickel. A distribution free version of the Smirnov two sample test in the p-variate case. *The Annals of Mathematical Statistics*, 40(1):1–23, 1969.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

K. M. Borgwardt, C. S. Ong, S. Schonauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics (ISMB)*, 21(Suppl 1):i47–i56, Jun 2005.

K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics (ISMB)*, 22(14): e49–e57, 2006.

O. Bousquet, S. Boucheron, and G. Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:323– 375, 2005.

R. Caruana and T. Joachims. KDD cup. 2004. URL `http://kodiak.cs.cornell.edu/kddcup/index.html`.

G. Casella and R. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, 2nd edition, 2002.

B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47:1028–1047, 2000.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the International Conference on Algorithmic Learning Theory*, volume 5254 of *Lecture Notes in Computer Science*, pages 38–53. Springer, 2008.

M. Davy, A. Gretton, A. Doucet, and P. J. W. Rayner. Optimized support vector machines for nonstationary signal classification. *IEEE Signal Processing Letters*, 9(12):442–445, 2002.

V. de la Peña and E. Giné. *Decoupling: from Dependence to Independence*. Springer, New York, 1999.

M. Dudík and R. E. Schapire. Maximum entropy distribution estimation with generalized regularization. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 123–138. Springer Verlag, 2006.

M. Dudík, S. Phillips, and R.E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 472–486. Springer Verlag, 2004.

R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, Cambridge, UK, 2002.

W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley and Sons, New York, 2nd edition, 1971.

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

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

R. Fortet and E. Mourier. Convergence de la réparation empirique vers la réparation théorique. *Ann. Scient. École Norm. Sup.*, 70:266–285, 1953.

J. Friedman. On multivariate goodness-of-fit and two-sample testing. Technical Report SLAC-PUB-10325, University of Stanford Statistics Department, 2003.

J. Friedman and L. Rafsky. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *The Annals of Statistics*, 7(4):697–717, 1979.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, pages 489–496, Cambridge, MA, 2008. MIT Press.

T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann Publishers Inc., 2002.

E. Gokcay and J.C. Principe. Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–171, 2002.

A. Gretton, O. Bousquet, A.J. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 63–77. Springer-Verlag, 2005a.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005b.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 15*, pages 513–520, Cambridge, MA, 2007a. MIT Press.

A. Gretton, K. Borgwardt, M. Rasch, B. Schlkopf, and A. Smola. A kernel approach to comparing distributions. *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, pages 1637–1641, 2007b.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two sample problem. Technical Report 157, MPI for Biological Cybernetics, 2008a.

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

A. Gretton, K. Fukumizu, Z. Harchaoui, and B. Sriperumbudur. A fast, consistent kernel two-sample test. In *Advances in Neural Information Processing Systems 22*, Red Hook, NY, 2009. Curran Associates Inc.

G. R. Grimmet and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, third edition, 2001.

P. Hall and N. Tajvidi. Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, 89(2):359–374, 2002.

Z. Harchaoui, F. Bach, and E. Moulines. Testing for homogeneity with kernel Fisher discriminant analysis. In *Advances in Neural Information Processing Systems 20*, pages 609–616. MIT Press, Cambridge, MA, 2008.

M. Hein, T.N. Lal, and O. Bousquet. Hilbertian metrics on probability measures and their application in SVMs. In *Proceedings of the 26th DAGM Symposium*, pages 270–277, Berlin, 2004. Springer.

N. Henze and M. Penrose. On the multivariate runs test. *The Annals of Statistics*, 27(1):290–298, 1999.

W. Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In *Proceedings of the Annual Conference on Computational Learning Theory*, volume 2777 of *LNCS*, pages 57–71, Heidelberg, Germany, 2003. Springer-Verlag.

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions. Volume 1*. John Wiley and Sons, 2nd edition, 1994.

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

D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the International Conference on Very Large Data Bases*, pages 180–191. VLDB Endowment, 2004.

H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer, 3rd edition, 2005.

C. McDiarmid. On the method of bounded differences. In *Survey in Combinatorics*, pages 148–188. Cambridge University Press, 1989.

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7: 2651–2667, 2006.

A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

X.L. Nguyen, M. Wainwright, and M. Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems 20*, pages 1089–1096. MIT Press, Cambridge, MA, 2008.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C. The Art of Scientific Computation*. Cambridge University Press, Cambridge, UK, 1994.

M. Rasch, A. Gretton, Y. Murayama, W. Maass, and N. K. Logothetis. Predicting spiking activity from local field potentials. *Journal of Neurophysiology*, 99:1461–1476, 2008.

M. Reed and B. Simon. *Methods of modern mathematical physics. Vol. 1: Functional Analysis*. Academic Press, San Diego, 1980.

M. Reid and R. Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12:731–817, 2011.

P. Rosenbaum. An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society B*, 67(4):515–530, 2005.

Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997. Download: http://www.kernel-machines.org.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

R. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley, New York, 1980.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

J. Shawe-Taylor and A. Dolia. A framework for probability density estimation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 468–475, 2007.

H. Shen, S. Jegelka, and A. Gretton. Fast kernel-based independent component analysis. *IEEE Transactions on Signal Processing*, 57:3498 – 3511, 2009.

B. W. Silverman. *Density Estimation for Statistical and Data Analysis*. Monographs on statistics and applied probability. Chapman and Hall, London, 1986.

N.V. Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Moscow University Mathematics Bulletin*, 2:3–26, 1939. University of Moscow.

A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the International Conference on Machine Learning*, pages 911–918, San Francisco, 2000. Morgan Kaufmann Publishers.

A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Proceedings of the International Conference on Algorithmic Learning Theory*, volume 4754, pages 13–31. Springer, 2007.

L. Song, X. Zhang, A. Smola, A. Gretton, and B. Schölkopf. Tailoring density estimation via reproducing kernel moment matching. In *Proceedings of the International Conference on Machine Learning*, pages 992–999. ACM, 2008.

B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Injective Hilbert space embeddings of probability measures. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 111–122, 2008.

B. Sriperumbudur, K. Fukumizu, A. Gretton, G. Lanckriet, and B. Schoelkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems 22*, Red Hook, NY, 2009. Curran Associates Inc.

B. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. Lanckriet. Non-parametric estimation of integral probability metrics. In *International Symposium on Information Theory*, pages 1428 – 1432, 2010a.

B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010b.

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011a.

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Learning in Hilbert vs. Banach spaces: A measure embedding viewpoint. In *Advances in Neural Information Processing Systems 24*. Curran Associates Inc., Red Hook, NY, 2011b.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.

I. Takeuchi, Q. V. Le, T. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7, 2006.

D. M. J. Tax and R. P. W. Duin. Data domain description by support vectors. In *Proceedings ESANN*, pages 251–256, Brussels, 1999. D Facto.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

J. E. Wilkins. A note on skewness and kurtosis. *The Annals of Mathematical Statistics*, 15(3): 333–335, 1944.

C. K. I. Williams and M. Seeger. Using the Nystrom method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688, Cambridge, MA, 2001. MIT Press.

Journal of Machine Learning Research 13 (2012) 775-779 Submitted 9/11; Published 1/12

GPLP: A Local and Parallel Computation Toolbox for Gaussian Process Regression

Chiwoo Park CHIWOO.PARK@ENG.FSU.EDU
Department of Industrial and Manufacturing Engineering
Florida A&M - Florida State University College of Engineering
2525 Pottsdamer St
Tallahassee, FL 32310-6046, USA

Jianhua Z. Huang JIANHUA@STAT.TAMU.EDU
Department of Statistics
Texas A&M University
3143 TAMU
College Station, TX 77843-3143, USA

Yu Ding YUDING@IEMAIL.TAMU.EDU
Department of Industrial and Systems Engineering
Texas A&M University
3131 TAMU
College Station, TX 77843-3131, USA

Editor: Mikio Braun

Abstract

This paper presents the *Getting-started* style documentation for the local and parallel computation toolbox for Gaussian process regression (`GPLP`), an open source software package written in `Matlab` (but also compatible with `Octave`). The working environment and the usage of the software package will be presented in this paper.

Keywords: Gaussian process regression, domain decomposition method, partial independent conditional, bagging for Gaussian process, local probabilistic regression

1. Introduction

The Gaussian process regression (GP regression) has recently developed to be a useful tool in machine learning (Rasmussen and Williams, 2006). A GP regression provides the best unbiased linear estimator computable by a simple closed form expression and is a popular method for interpolation or extrapolation. A major limitation of GP regression is its computational complexity, scaled by $O(N^3)$, where N is the number of training observations.

Many fast computation methods have been introduced in the literature to relieve the computation burden: *matrix approximation* (Williams and Seeger, 2000; Smola and Bartlett, 2001), *likelihood approximation* (Seeger et al., 2003; Snelson and Ghahramani, 2006, 2007) and *localized regression* (Tresp, 2000; Schwaighofer et al., 2003; Urtasun and Darrell, 2008; Rasmussen and Ghahramani, 2002; Gramacy and Lee, 2008; Chen and Ren, 2009; Park et al., 2011).

Many of the computation methods have been implemented as software, which includes SOGP[1], GPML[2], SPGP[3], TGP[4] and GPStuff[5]. However, many of the methods are still not implemented, because of various complexities involved in the methods as well as in their implementation. In particular, most of the localized regression methods are not implemented in spite of their unique advantages such as adaptivity to non-stationary changes and easiness of being parallelized for faster computation.

The GPLP is the Octave and Matlab implementation of several localized regression methods: the domain decomposition method (Park et al., 2011, DDM), partial independent conditional (Snelson and Ghahramani, 2007, PIC), localized probabilistic regression (Urtasun and Darrell, 2008, LPR), and bagging for Gaussian process regression (Chen and Ren, 2009, BGP). Most of the localized regression methods can be applied for general machine learning problems although DDM is only applicable for spatial data sets. In addition, the GPLP provides two parallel computation versions of the domain decomposition method. The easiness of being parallelized is one of the advantages of the localized regression, and the two parallel implementations will provide a good guidance about how to materialize this advantage as software.

This manual is written in *Getting-started* style; it introduces the working environment of GPLP (in Section 2) and illustrates the usage with an simple example (in Section 3). If you need more detailed documentation, please refer to *User Manual* at ./doc directory.

2. Implementation

The GPLP is implemented in Matlab code such that it is executable and has been tested in Matlab Version 7.7 or later versions, and Octave Version 3.2.4 or later versions. It might be executable in any of Matlab Version 7.x and any of Octave Version 3.2.x, but it has not been tested on those versions. One exception is the implementation of LPR that only works in Matlab 7.12.0, in Matlab 7.7.0 or later versions with a compiler supporting mex-compile, or in Octave 3.2.4 or later versions. For information on the list of compilers to support the mex-compile in Matlab, please refer to the technical support webpage at http://www.mathworks.com/support/compilers/previous_releases.html.

The GPLP also includes the parallel computation version of DDM, which requires the open source message passing interface, MatMPI Version 1.2, to be pre-installed before executing the parallel version. All of the Matlab, Octave and MatMPI are working in many versions of Windows and Unix, so GPLP is virtually OS-independent.

The implementation consists of six different main modules for the six different methods implemented, but all of the main modules are structured in the common form having the similar input and output arguments. In addition, the implementation partially supports the separation of the main

[1] Implementation of SOGP (Smola and Bartlett, 2001) is available at http://cs.brown.edu/people/dang/code.shtml.

[2] Implementation of GPML (Williams and Seeger, 2000) is available at http://gaussianprocess.org/gpml/code/matlab/doc/index.html.

[3] Implementation of SPGP (Snelson and Ghahramani, 2006) is available at http://www.gatsby.ucl.ac.uk/~snelson.

[4] Implementation of TGP (Gramacy and Lee, 2008) is available at http://users.soe.ucsc.edu/~rbgramacy/tgp.html.

[5] Implementation of GPStuff (Snelson and Ghahramani, 2006, 2007; Schwaighofer et al., 2003) is available at http://www.lce.hut.fi/research/mm/gpstuff/.

logic from the specification of the covariance function and the mesh generation function (the specification of mesh generation function is only applicable for DDM and its two parallel computation versions; the explanation of the mesh generation function will be in the next section). With such separation, users can easily extend the function of GPLP by adding a new covariance function and adding a new mesh generation function without major modification of the main logic.

The code and documentation of GPLP are publicly available on the JMLR MOSS website at http://www.jmlr.org/mloss under GNU General Public License version 3.0 (GPL-3.0).

3. GPLP: A software Package for Localized and Parallel Computation of GP Regression

The GPLP provides an individual function for calling each one of the six localized regression methods (including two parallel implementations. The individual functions have a common structure of input and output arguments so that users can easily use all functions once they learn the common structure. In this section, we will explain the common structure by means of a simple example.

Consider a unknown random function $f : \mathcal{X} \to \mathbb{R}$. The GP regression predicts the realization of the random function at test locations xs, given a set of observations x from the realization. The localized GP regression partitions x into many smaller chunks, x_j's, and it does localized predictions at xs with each one of x_j's as the training data for every j. Finally, the localized GP regression combines the localized predictions to make a global prediction in many different ways. The key design parameters for the localized GP regression are (1) mean function and covariance function defining the GP, and (2) mesh generation function for partitioning x into x_j's.

```
 1  % define the structure of local regions
 2  param1.meshfunc = 'rectMesh';   % mesh generation function
 3  param1.mparam   = [14 21];      % mesh generation function parameters
 4  param1.p = 3; param1.q = 3;     % parameters defining the interaction
 5                                  %   between local regions for improving
 6                                  %   prediction accuracy
 7
 8  % set the prior GP by specifying a covariance function
 9  param2.covfunc  = {'covSum', {'covSEard','covNoise'}}; %covariance function
10  D               = size(x, 2);
11  logtheta0       = log(ones(D+2,1));
12  logtheta0(D+2)  = log(0.3);
13  param2.logtheta0 = logtheta0; % initial value of log hyperparameters
14  param2.frachyper = 0.5; % fraction of training data used for learning
15                          %                                  hyperparameters
16  param2.nIter     = 100; % maximum number of iterations in optimizing the
17                          %    log hyperparameters
18
19  % train the localized regression model for Gaussian process regression
20  [model, elpasedTrain] = ddmGP(x, y, param1, param2);
21
22  % predict at test inputs
23  [meanPred, varPred, elapsedPred] = ddm_pred(model, xs);
```

In line 2 and 3, we specify the mesh generation function as rectMesh with its input parameter (14, 21). The mesh generation function decomposes $\mathcal{X}$ (domain of f) into 14-by-21 rectangular meshes, $\{\mathcal{X}_j\}$, and it partitions x into x_j's such that x_j belongs to $\mathcal{X}_j$.

In line 4, there are two parameters that defines how many localized predictions are combined to produce a global prediction. In the domain decomposition method (DDM), a localized prediction is available for each mesh $\mathcal{X}_j$, which becomes the global prediction if the test input is in the interior of local domain $\mathcal{X}_j$. If the test input is over the common boundary of $\mathcal{X}_j$ and $\mathcal{X}_k$, the localized predictions are constrained by two factors: (1) the two local predictions for both of $\mathcal{X}_j$ and $\mathcal{X}_k$ should have limited degrees of freedom on the common boundary (called *flexibility of boundary prediction*); and (2) the two localized predictions should produce the same values on the boundary (called *consistency of boundary prediction*). The `param1.q` is the the number of control points on the boundary where the DDM checks the *consistency of boundary prediction*, and the `param1.p` is the number of degrees of freedom to constrain the *flexibility of boundary prediction*.

In line 9 through 13, we specify the `covSum` composite covariance function. The composite covariance function generates the covariance by summing two base covariance functions: the anisotropic version of squared exponential covariance function (`covSEard`) and the noise covariance function (`covNoise`). The `covSEard` is parameterized by $(D+1)$ hyperparameters as follows:

$$K(\mathbf{x},\mathbf{x}') = \theta_{D+1}^2 \exp\left\{ -\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d - x'_d}{\theta_d}\right)^2 \right\},$$

where D is the dimension of $\mathcal{X}$. The `covNoise` is parameterized by noise variance parameter σ^2 as $K(\mathbf{x},\mathbf{x}') = \sigma^2\delta(\mathbf{x},\mathbf{x}')$. In total, the composite covariance function is parameterized by $(D+2)$ parameter values, so the initial guess of hyperparameter, `logtheta0`, should be $(D+2)$-dimensional. In line 11 and 12, the first $(D+1)$ elements of `logtheta0` are initialized for the hyperparameter values of `covSEard`, and the last one element of `logtheta0` is initialized for the value of σ^2.

In line 14 and 15, `param2.frachyper` and `param2.nIter` are the process parameters used in maximizing the likelihood function with respect to the hyperparameters. The maximization is an iterative process that updates the log hyperparameter values, starting with the initial guess `logtheta0`. The `param2.nIter=100` implies that the number of the iterations allowed for the iterative maximization is at most one hundred. In each iteration, the likelihood function is evaluated. Since the evaluation is computationally expensive with big size of training data, people usually uses only a subset of the training data for the evaluation. The `param2.frachyper = 0.5` implies that only half of the training data `x` will be used for the evaluation of the likelihood function.

Last, in line 20, the function `ddmGP` trains the domain decomposition method for the localized GP regression with training data set `x` and the previously specified parameters, and `ddmGP` returns the trained model (`model`) and the elapsed time (`elpasedTrain`). The number of the parameters to be specified depends on the method used for the training. For more details, please refer to *User Manual* at `./doc` directory in this package. In line 22, the function `ddm_pred` produces the mean prediction `meanPred` and the variance prediction `varPred` at test locations `xs`, and also reports the time used for prediction (`elpasedPred`).

References

Tao Chen and Jianghong Ren. Bagging for Gaussian process regression. *Neurocomputing*, 72(7-9): 1605–1610, 2009.

Robert B. Gramacy and Herbert K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–

1130, 2008.

Chiwoo Park, Jianhua Z. Huang, and Yu Ding. Domain decomposition approach for fast gaussian process regression of large spatial data sets. *Journal of Machine Learning Research*, 12:1697–1728, 2011.

Carl E. Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2002.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Anton Schwaighofer, Marian Grigoras, Volker Tresp, and Clemens Hoffmann. Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems 16*, pages 977–984. MIT Press, 2003.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *International Workshop on Artificial Intelligence and Statistics 9*. Society for Artificial Intelligence and Statistics, 2003.

Alexander J. Smola and Peter L. Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems 13*, pages 619–625. MIT Press, 2001.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.

Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *International Conference on Artifical Intelligence and Statistics 11*, pages 524–531. Society for Artificial Intelligence and Statistics, 2007.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Raquel Urtasun and Trevor Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *IEEE Conference on Computer Vision and Pattern Recognition 2008*, pages 1–8, 2008.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 12*, pages 682–688. MIT Press, 2000.

Journal of Machine Learning Research 13 (2012) 781-794 Submitted 8/11; Revised 1/12; Published 3/12

Exact Covariance Thresholding into Connected Components for Large-Scale Graphical Lasso

Rahul Mazumder RAHULM@STANFORD.EDU
Trevor Hastie[*] HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305

Editor: Francis Bach

Abstract

We consider the sparse inverse covariance regularization problem or *graphical lasso* with regularization parameter λ. Suppose the sample *covariance graph* formed by thresholding the entries of the sample covariance matrix at λ is decomposed into connected components. We show that the *vertex-partition* induced by the connected components of the thresholded sample covariance graph (at λ) is *exactly* equal to that induced by the connected components of the estimated concentration graph, obtained by solving the graphical lasso problem for the *same* λ. This characterizes a very interesting property of a path of graphical lasso solutions. Furthermore, this simple rule, when used as a wrapper around existing algorithms for the graphical lasso, leads to enormous performance gains. For a range of values of λ, our proposal splits a large graphical lasso problem into smaller tractable problems, making it possible to solve an otherwise infeasible large-scale problem. We illustrate the graceful scalability of our proposal via synthetic and real-life microarray examples.

Keywords: sparse inverse covariance selection, sparsity, graphical lasso, Gaussian graphical models, graph connected components, concentration graph, large scale covariance estimation

1. Introduction

Consider a data matrix $\mathbf{X}_{n\times p}$ comprising of n sample realizations from a p dimensional Gaussian distribution with zero mean and positive definite covariance matrix $\Sigma_{p\times p}$ (unknown), that is, $\mathbf{x}_i \overset{\text{i.i.d}}{\sim} \text{MVN}(\mathbf{0},\Sigma)$, $i=1,\ldots,n$. The task is to estimate the unknown Σ based on the n samples. ℓ_1 regularized Sparse Inverse Covariance Selection also known as *graphical lasso* (Friedman et al., 2007; Banerjee et al., 2008; Yuan and Lin, 2007) estimates the covariance matrix Σ, under the assumption that the inverse covariance matrix, that is, Σ^{-1} is sparse. This is achieved by minimizing the regularized negative log-likelihood function:

$$\underset{\Theta\succeq\mathbf{0}}{\text{minimize}} \quad -\log\det(\Theta)+\text{tr}(\mathbf{S}\Theta)+\lambda\sum_{i,j}|\Theta_{ij}|, \tag{1}$$

where $\mathbf{S}$ is the sample covariance matrix. Problem (1) is a convex optimization problem in the variable Θ (Boyd and Vandenberghe, 2004). Let $\widehat{\Theta}^{(\lambda)}$ denote the solution to (1). We note that (1) can also be used in a more non-parametric fashion for any positive semi-definite input matrix $\mathbf{S}$, not necessarily a sample covariance matrix of a MVN sample as described above.

[*] Also in the Department of Health, Research and Policy

A related criterion to (1) is one where the diagonals are not penalized—by substituting $\mathbf{S} \leftarrow \mathbf{S} + \lambda I_{p\times p}$ in the "unpenalized" problem we get (1). In this paper we concentrate on problem (1).

Developing efficient large-scale algorithms for (1) is an active area of research across the fields of Convex Optimization, Machine Learning and Statistics. Many algorithms have been proposed for this task (Friedman et al., 2007; Banerjee et al., 2008; Lu, 2009, 2010; Scheinberg et al., 2010; Yuan, 2009, for example). However, it appears that certain special properties of the solution to (1) have been largely ignored. This paper is about one such (surprising) property—namely establishing an equivalence between the *vertex-partition* induced by the connected components of the non-zero pattern of $\widehat{\Theta}^{(\lambda)}$ and the thresholded sample covariance matrix $\mathbf{S}$. This paper is *not* about a specific algorithm for the problem (1)—it focuses on the aforementioned observation that leads to a novel thresholding/screening procedure based on $\mathbf{S}$. This provides interesting insight into the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$ obtained by solving (1), over a path of λ values. The behavior of the connected-components obtained from the non-zero patterns of $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$ can be completely understood by simple screening rules on $\mathbf{S}$. This can be done without *even attempting* to solve (1)—arguably a very challenging convex optimization problem. Furthermore, this thresholding rule can be used as a *wrapper* to enormously boost the performance of existing algorithms, as seen in our experiments. This strategy becomes extremely effective in solving large problems over a range of values of λ—sufficiently restricted to ensure sparsity and the separation into connected components. Of course, for sufficiently small values of λ there will be no separation into components, and hence no computational savings.

At this point we introduce some notation and terminology, which we will use throughout the paper.

1.1 Notations and Preliminaries

For a matrix $\mathbf{Z}$, its $(i, j)^{\text{th}}$ entry is denoted by $\mathbf{Z}_{ij}$.

We also introduce some graph theory notations and definitions (Bollobas, 1998) sufficient for this exposition. A finite undirected graph $\mathcal{G}$ on p vertices is given by the ordered tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the collection of (undirected) edges. The edge-set is equivalently represented via a (symmetric) 0-1 matrix[1] (also known as the *adjacency* matrix) with p rows/columns. We use the convention that a node is not connected to itself, so the diagonals of the adjacency matrix are all zeros. Let $|\mathcal{V}|$ and $|\mathcal{E}|$ denote the number of nodes and edges respectively.

We say two nodes $u, v \in \mathcal{V}$ are *connected* if there is a *path* between them. A maximal connected *subgraph*[2] is a *connected component* of the graph $\mathcal{G}$. *Connectedness* is an equivalence relation that decomposes a graph $\mathcal{G}$ into its connected components $\{(\mathcal{V}_\ell, \mathcal{E}_\ell)\}_{1\leq \ell\leq K}$—with $\mathcal{G} = \cup_{\ell=1}^{K}(\mathcal{V}_\ell, \mathcal{E}_\ell)$, where K denotes the number of connected components. This decomposition partitions the vertices $\mathcal{V}$ of $\mathcal{G}$ into $\{\mathcal{V}_\ell\}_{1\leq \ell\leq K}$. Note that the labeling of the components is unique upto permutations on $\{1, \ldots, K\}$. Throughout this paper we will often refer to this partition as the *vertex-partition* induced by the components of the graph $\mathcal{G}$. If the size of a component is one, that is, $|\mathcal{V}_\ell| = 1$, we say that the node is *isolated*. Suppose a graph $\widehat{\mathcal{G}}$ defined on the set of vertices $\mathcal{V}$ admits the following decomposition into connected components: $\widehat{\mathcal{G}} = \cup_{\ell=1}^{\widehat{K}}(\widehat{\mathcal{V}}_\ell, \widehat{\mathcal{E}}_\ell)$. We say the vertex-

[1] 0 denotes absence of an edge and 1 denotes its presence.

[2] $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a *subgraph* of $\mathcal{G}$ if $\mathcal{V}' \subset \mathcal{V}$ and $\mathcal{E}' \subset \mathcal{E}$.

partitions induced by the connected components of $\mathcal{G}$ and $\widehat{\mathcal{G}}$ are *equal* if $\widehat{K} = K$ and there is a permutation π on $\{1,\dots,K\}$ such that $\widehat{\mathcal{V}}_{\pi(\ell)} = \mathcal{V}_\ell$ for all $\ell \in \{1,\dots,K\}$.

The paper is organized as follows. Section 2 describes the covariance graph thresholding idea along with theoretical justification and related work, followed by complexity analysis of the algorithmic framework in Section 3. Numerical experiments appear in Section 4, concluding remarks in Section 5 and the proofs are gathered in the Appendix A.

2. Methodology: *Exact* Thresholding of the Covariance Graph

The sparsity pattern of the solution $\widehat{\Theta}^{(\lambda)}$ to (1) gives rise to the symmetric edge matrix/skeleton $\in \{0,1\}^{p\times p}$ defined by:

$$\mathcal{E}^{(\lambda)}_{ij} = \begin{cases} 1 & \text{if } \widehat{\Theta}^{(\lambda)}_{ij} \neq 0, i \neq j; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The above defines a symmetric graph $\mathcal{G}^{(\lambda)} = (\mathcal{V}, \mathcal{E}^{(\lambda)})$, namely the *estimated concentration graph* (Cox and Wermuth, 1996; Lauritzen, 1996) defined on the nodes $\mathcal{V} = \{1,\dots,p\}$ with edges $\mathcal{E}^{(\lambda)}$.

Suppose the graph $\mathcal{G}^{(\lambda)}$ admits a decomposition into $\kappa(\lambda)$ connected components:

$$\mathcal{G}^{(\lambda)} = \cup_{\ell=1}^{\kappa(\lambda)} \mathcal{G}^{(\lambda)}_\ell, \tag{3}$$

where $\mathcal{G}^{(\lambda)}_\ell = (\widehat{\mathcal{V}}^{(\lambda)}_\ell, \mathcal{E}^{(\lambda)}_\ell)$ are the components of the graph $\mathcal{G}^{(\lambda)}$. Note that $\kappa(\lambda) \in \{1,\dots,p\}$, with $\kappa(\lambda) = p$ (large λ) implying that all nodes are isolated and for small enough values of λ, there is only one component, that is, $\kappa(\lambda) = 1$.

We now describe the simple screening/thresholding rule. Given λ, we perform a thresholding on the entries of the sample covariance matrix $\mathbf{S}$ and obtain a graph edge skeleton $\mathrm{E}^{(\lambda)} \in \{0,1\}^{p\times p}$ defined by:

$$\mathrm{E}^{(\lambda)}_{ij} = \begin{cases} 1 & \text{if } |\mathbf{S}_{ij}| > \lambda, i \neq j; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The symmetric matrix $\mathrm{E}^{(\lambda)}$ defines a symmetric graph on the nodes $\mathcal{V} = \{1,\dots,p\}$ given by $\mathrm{G}^{(\lambda)} = (\mathcal{V}, \mathrm{E}^{(\lambda)})$. We refer to this as the *thresholded sample covariance graph*. Similar to the decomposition in (3), the graph $\mathrm{G}^{(\lambda)}$ also admits a decomposition into connected components:

$$\mathrm{G}^{(\lambda)} = \cup_{\ell=1}^{k(\lambda)} \mathrm{G}^{(\lambda)}_\ell, \tag{5}$$

where $\mathrm{G}^{(\lambda)}_\ell = (\mathcal{V}^{(\lambda)}_\ell, \mathrm{E}^{(\lambda)}_\ell)$ are the components of the graph $\mathrm{G}^{(\lambda)}$.

Note that the components of $\mathcal{G}^{(\lambda)}$ require knowledge of $\widehat{\Theta}^{(\lambda)}$—the solution to (1). Construction of $\mathrm{G}^{(\lambda)}$ and its components require operating on $\mathbf{S}$—an operation that can be performed completely independent of the optimization problem (1), which is arguably more expensive (See Section 3). The surprising message we describe in this paper is that the *vertex-partition* of the connected components of (5) is *exactly* equal to that of (3).

This observation has the following consequences:

1. We obtain a very interesting property of the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$—the behavior of the connected components of the estimated concentration graph can be completely understood by simple screening rules on $\mathbf{S}$.

2. The cost of computing the connected components of the thresholded sample covariance graph (5) is orders of magnitude smaller than the cost of fitting graphical models (1). Furthermore, the computations pertaining to the covariance graph can be done off-line and are amenable to parallel computation (See Section 3).

3. The optimization problem (1) completely separates into $k(\lambda)$ separate optimization sub-problems of the form (1). The sub-problems have size equal to the number of nodes in each component $p_i := |\mathcal{V}_i|, i = 1, \ldots, k(\lambda)$. Hence for certain values of λ, solving problem (1) becomes feasible although it may be impossible to operate on the $p \times p$ dimensional (global) variable Θ on a single machine.

4. Suppose that for λ_0, there are $k(\lambda_0)$ components and the graphical model computations are distributed.[3] Since the vertex-partitions induced via (3) and (5) are nested with increasing λ (see Theorem 2), it suffices to operate independently on these separate machines to obtain the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_\lambda$ for all $\lambda \geq \lambda_0$.

5. Consider a distributed computing architecture, where every machine allows operating on a graphical lasso problem (1) of maximal size $p_{\max}$. Then with relatively small effort we can find the smallest value of $\lambda = \lambda_{p_{\max}}$, such that there are no connected components of size larger than $p_{\max}$. Problem (1) thus 'splits up' independently into manageable problems across the different machines. When this structure is not exploited the global problem (1) remains intractable.

The following theorem establishes the main technical contribution of this paper—the equivalence of the vertex-partitions induced by the connected components of the thresholded sample covariance graph and the estimated concentration graph.

Theorem 1 *For any $\lambda > 0$, the components of the estimated concentration graph $\mathcal{G}^{(\lambda)}$, as defined in (2) and (3) induce* exactly *the same vertex-partition as that of the thresholded sample covariance graph $\mathrm{G}^{(\lambda)}$, defined in (4) and (5). That is $\kappa(\lambda) = k(\lambda)$ and there exists a permutation π on $\{1, \ldots, k(\lambda)\}$ such that:*

$$\widehat{\mathcal{V}}_i^{(\lambda)} = \mathcal{V}_{\pi(i)}^{(\lambda)}, \;\; \forall i = 1, \ldots, k(\lambda). \tag{6}$$

Proof *The proof of the theorem appears in Appendix A.1.* ■

Since the decomposition of a symmetric graph into its connected components depends upon the ordering/ labeling of the components, the permutation π appears in Theorem 1.

Remark 1 *Note that the edge-structures* within *each block need not be preserved. Under a matching reordering of the labels of the components of $\mathcal{G}^{(\lambda)}$ and $\mathrm{G}^{(\lambda)}$:*
for every fixed ℓ such that $\widehat{\mathcal{V}}_\ell^{(\lambda)} = \mathcal{V}_\ell^{(\lambda)}$ the edge-sets $\mathcal{E}_\ell^{(\lambda)}$ and $\mathrm{E}_\ell^{(\lambda)}$ are not *necessarily equal.*

[3] Distributing these operations depend upon the number of processors available, their capacities, communication lag, the number of components and the maximal size of the blocks across all machines. These of-course depend upon the computing environment. In the context of the present problem, it is often desirable to club smaller components into a single machine.

Theorem 1 leads to a special property of the path-of-solutions to (1), that is, the vertex-partition induced by the connected components of $\mathcal{G}^{(\lambda)}$ are nested with increasing λ. This is the content of the following theorem.

Theorem 2 *Consider two values of the regularization parameter such that* $\lambda > \lambda' > 0$*, with corresponding concentration graphs* $\mathcal{G}^{(\lambda)}$ *and* $\mathcal{G}^{(\lambda')}$ *as in (2) and connected components (3). Then the vertex-partition induced by the components of* $\mathcal{G}^{(\lambda)}$ *are* nested *within the partition induced by the components of* $\mathcal{G}^{(\lambda')}$*. Formally,* $\kappa(\lambda) \geq \kappa(\lambda')$ *and the vertex-partition* $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\leq \ell \leq \kappa(\lambda)}$ *forms a finer resolution of* $\{\widehat{\mathcal{V}}_\ell^{(\lambda')}\}_{1\leq \ell \leq \kappa(\lambda')}$*.*

Proof *The proof of this theorem appears in the Appendix A.2.* ■

Remark 2 *It is worth noting that Theorem 2 addresses the nesting of the edges* across *connected components and not within a component. In general, the edge-set* $\mathcal{E}^{(\lambda)}$ *of the estimated concentration graph need not be nested as a function of* λ*:*
for $\lambda > \lambda'$*, in general,* $\mathcal{E}^{(\lambda)} \not\subset \mathcal{E}^{(\lambda')}$*.*

See Friedman et al. (2007, Figure 3), for numerical examples demonstrating the non-monotonicity of the edge-set across λ, as described in Remark 2.

2.1 Node-Thresholding

A simple consequence of Theorem 1 is that of *node-thresholding*. If $\lambda \geq \max_{j\neq i} |\mathbf{S}_{ij}|$, then the i^{th} node will be isolated from the other nodes, the off-diagonal entries of the i^{th} row/column are all zero, that is, $\max_{j\neq i} |\widehat{\Theta}_{ij}^{(\lambda)}| = 0$. Furthermore, the i^{th} diagonal entries of the estimated covariance and precision matrices are given by $(\mathbf{S}_{ii} + \lambda)$ and $\frac{1}{\mathbf{S}_{ii}+\lambda}$, respectively. Hence, as soon as $\lambda \geq \max_{i=1,\ldots,p}\{\max_{j\neq i} |\mathbf{S}_{ij}|\}$, the estimated covariance and precision matrices obtained from (1) are both diagonal.

2.2 Related Work

Witten et al. (2011) independently discovered block screening as described in this paper. At the time of our writing, an earlier version of their paper was available (Witten and Friedman, 2011); it proposed a scheme to detect isolated nodes for problem (1) via a simple screening of the entries of $\mathbf{S}$, but no block screening. Earlier, Banerjee et al. (2008, Theorem 4) made the same observation about isolated nodes. The revised manuscript (Witten et al., 2011) that includes block screening became available shortly after our paper was submitted for publication.

Zhou et al. (2011) use a thresholding strategy followed by re-fitting for estimating Gaussian graphical models. Their approach is based on the node-wise lasso-regression procedure of Meinshausen and Bühlmann (2006). A hard thresholding is performed on the ℓ_1-penalized regression coefficient estimates at every node to obtain the graph structure. A restricted MLE for the concentration matrix is obtained for the graph. The proposal in our paper differs since we are interested in solving the GLASSO problem (1).

3. Computational Complexity

The overall complexity of our proposal depends upon (a) the graph partition stage and (b) solving (sub)problems of the form (1). In addition to these, there is an unavoidable complexity associated with handling and/or forming $\mathbf{S}$.

The cost of computing the connected components of the thresholded covariance graph is fairly negligible when compared to solving a similar sized graphical lasso problem (1)—see also our simulation studies in Section 4. In case we observe samples $\mathbf{x}_i \in \Re^p, i = 1, \ldots, n$ the cost for creating the sample covariance matrix $\mathbf{S}$ is $O(n \cdot p^2)$. Thresholding the sample covariance matrix costs $O(p^2)$. Obtaining the connected components of the thresholded covariance graph costs $O(|\mathrm{E}^{(\lambda)}| + p)$ (Tarjan, 1972). Since we are interested in a region where the thresholded covariance graph is sparse enough to be broken into smaller connected components—$|\mathrm{E}^{(\lambda)}| \ll p^2$. Note that all computations pertaining to the construction of the connected components and the task of computing $\mathbf{S}$ can be computed off-line. Furthermore the computations are parallelizable. Gazit (1991, for example) describes parallel algorithms for computing connected components of a graph—they have a time complexity $O(\log(p))$ and require $O((|\mathrm{E}^{(\lambda)}| + p)/\log(p))$ processors with space $O(p + |\mathrm{E}^{(\lambda)}|)$.

There are a wide variety of algorithms for the task of solving (1). While an exhaustive review of the computational complexities of the different algorithms is beyond the scope of this paper, we provide a brief summary for a few algorithms below.

Banerjee et al. (2008) proposed a smooth accelerated gradient based method (Nesterov, 2005) with complexity $O(\frac{p^{4.5}}{\varepsilon})$ to obtain an ε accurate solution—the per iteration cost being $O(p^3)$. They also proposed a block coordinate method which has a complexity of $O(p^4)$.

The complexity of the GLASSO algorithm (Friedman et al., 2007) which uses a row-by-row block coordinate method is roughly $O(p^3)$ for reasonably sparse-problems with p nodes. For denser problems the cost can be as large as $O(p^4)$.

The algorithm SMACS proposed in Lu (2010) has a per iteration complexity of $O(p^3)$ and an overall complexity of $O(\frac{p^4}{\sqrt{\varepsilon}})$ to obtain an $\varepsilon > 0$ accurate solution.

It appears that most existing algorithms for (1), have a complexity of at least $O(p^3)$ to $O(p^4)$ or possibly larger, depending upon the algorithm used and the desired accuracy of the solution—making computations for (1) almost impractical for values of p much larger than 2000.

It is quite clear that the role played by covariance thresholding is indeed crucial in this context. Assume that we use a solver of complexity $O(p^J)$ with $J \in \{3, 4\}$, along with our screening procedure. Suppose for a given λ, the thresholded sample covariance graph has $k(\lambda)$ components—the total cost of solving these smaller problems is then $\sum_{i=1}^{k(\lambda)} O(|\mathcal{V}_i^{(\lambda)}|^J) \ll O(p^J)$, with $J \in \{3, 4\}$. This difference in practice can be enormous—see Section 4 for numerical examples. This is what makes large scale graphical lasso problems solvable!

4. Numerical Examples

In this section we show via numerical experiments that the screening property helps in obtaining many fold speed-ups when compared to an algorithm that does not exploit it. Section 4.1 considers synthetic examples and Section 4.2 discusses real-life microarray data-examples.

4.1 Synthetic Examples

Experiments are performed with two publicly available algorithm implementations for the problem (1):

GLASSO: The algorithm of Friedman et al. (2007). We used the MATLAB wrapper available at `http://www-stat.stanford.edu/~tibs/glasso/index.html` to the Fortran code. The specific criterion for convergence (lack of progress of the diagonal entries) was set to 10^{-5} and the maximal number of iterations was set to 1000.

SMACS: denotes the algorithm of Lu (2010). We used the MATLAB implementation `smooth_covsel` available at `http://people.math.sfu.ca/~zhaosong/Codes/SMOOTH_COVSEL/`. The criterion for convergence (based on duality gap) was set to 10^{-5} and the maximal number of iterations was set to 1000.

We will like to note that the convergence criteria of the two algorithms GLASSO and SMACS are not the same. For obtaining the connected components of a symmetric adjacency matrix we used the MATLAB function `graphconncomp`. All of our computations are done in MATLAB 7.11.0 on a 3.3 GhZ Intel Xeon processor.

The simulation examples are created as follows. We generated a block diagonal matrix given by $\tilde{\mathbf{S}} = \text{blkdiag}(\tilde{\mathbf{S}}_1, \ldots, \tilde{\mathbf{S}}_K)$, where each block $\tilde{\mathbf{S}}_\ell = \mathbf{1}_{p_\ell \times p_\ell}$—a matrix of all ones and $\sum_\ell p_\ell = p$. In the examples we took all p_ℓs to be equal to p_1 (say). Noise of the form $\sigma \cdot UU'$ (U is a $p \times p$ matrix with i.i.d. standard Gaussian entries) is added to $\tilde{\mathbf{S}}$ such that 1.25 times the largest (in absolute value) off block-diagonal (as in the block structure of $\tilde{\mathbf{S}}$) entry of $\sigma \cdot UU'$ equals the smallest absolute non-zero entry in $\tilde{\mathbf{S}}$, that is, one. The sample covariance matrix is $\mathbf{S} = \tilde{\mathbf{S}} + \sigma \cdot UU'$.

We consider a number of examples for varying K and p_1 values, as shown in Table 1. Sizes were chosen such that it is at-least 'conceivable' to solve (1) on the full dimensional problem, without screening. In all the examples shown in Table 1, we set $\lambda_I := (\lambda_{\max} + \lambda_{\min})/2$, where for all values of λ in the interval $[\lambda_{\min}, \lambda_{\max}]$ the thresh-holded version of the sample covariance matrix has exactly K connected components. We also took a larger value of λ, that is, $\lambda_{II} := \lambda_{\max}$, which gave sparser estimates of the precision matrix but the number of connected components were the same.

The computations across different connected blocks could be distributed into as many machines. This would lead to almost a K fold improvement in timings, however in Table 1 we report the timings by operating serially across the blocks. The serial 'loop' across the different blocks are implemented in MATLAB.

Table 1 shows the rather remarkable improvements obtained by using our proposed covariance thresholding strategy as compared to operating on the whole matrix. Timing comparisons between GLASSO and SMACS are not fair, since GLASSO is written in Fortran and SMACS in MATLAB. However, we note that our experiments are meant to demonstrate how the thresholding helps in improving the overall computational time over the baseline method of not exploiting screening. Clearly our proposed strategy makes solving larger problems (1), not only feasible but with quite attractive computational time. The time taken by the graph-partitioning step in splitting the thresholded covariance graph into its connected components is negligible as compared to the timings for the optimization problem.

				Algorithm Timings (sec)		**Ratio**	**Time** (sec)
K	p_1 / p	λ	**Algorithm**	with	without	Speedup	graph
				screen	screen	factor	partition
2	200 / 400	λ_I	GLASSO	11.1	25.97	2.33	0.04
			SMACS	12.31	137.45	11.16	
		λ_{II}	GLASSO	1.687	4.783	2.83	0.066
			SMACS	10.01	42.08	4.20	
2	500 /1000	λ_I	GLASSO	305.24	735.39	2.40	0.247
			SMACS	175	2138*	12.21	
		λ_{II}	GLASSO	29.8	121.8	4.08	0.35
			SMACS	272.6	1247.1	4.57	
5	300 /1500	λ_I	GLASSO	210.86	1439	6.82	0.18
			SMACS	63.22	6062*	95.88	
		λ_{II}	GLASSO	10.47	293.63	28.04	0.123
			SMACS	219.72	6061.6	27.58	
5	500 /2500	λ_I	GLASSO	1386.9	-	-	0.71
			SMACS	493	-	-	
		λ_{II}	GLASSO	17.79	963.92	54.18	0.018
			SMACS	354.81	-	-	
8	300 /2400	λ_I	GLASSO	692.25	-	-	0.713
			SMACS	185.75	-	-	
		λ_{II}	GLASSO	9.07	842.7	92.91	0.023
			SMACS	153.55	-	-	

Table 1: Table showing (a) the times in seconds with screening, (b) without screening, that is, on the whole matrix and (c) the ratio (b)/(a)—'Speedup factor' for algorithms GLASSO and SMACS. Algorithms with screening are operated serially—the times reflect the total time summed across all blocks. The column 'graph partition' lists the time for computing the connected components of the thresholded sample covariance graph. Since $\lambda_{II} > \lambda_I$, the former gives sparser models. '*' denotes the algorithm did not converge within 1000 iterations. '-' refers to cases where the respective algorithms failed to converge within 2 hours.

4.2 Micro-array Data Examples

The graphical lasso is often used in learning connectivity networks in gene-microarray data (Friedman et al., 2007, see for example). Since in most real examples the number of genes p is around tens of thousands, obtaining an inverse covariance matrix by solving (1) is computationally impractical. The covariance thresholding method we propose easily applies to these problems—and as we see gracefully delivers solutions over a large range of the parameter λ. We study three different micro-array examples and observe that as one varies λ from large to small values, the thresholded covariance graph splits into a number of non-trivial connected components of varying sizes. We continue till a small/moderate value of λ when the maximal size of a connected component gets larger than a predefined machine-capacity or the 'computational budget' for a single graphical lasso problem. Note that in relevant micro-array applications, since $p \gg n$ (n, the number of samples is at most a few hundred) heavy regularization is required to control the variance of the covariance estimates—so it does seem reasonable to restrict to solutions of (1) for large values of λ.

Following are the data-sets we used for our experiments:

(A) This data-set appears in Alon et al. (1999) and has been analyzed by Rothman et al. (2008, for example). In this experiment, tissue samples were analyzed using an Affymetrix Oligonucleotide array. The data were processed, filtered and reduced to a subset of $p = 2000$ gene expression values. The number of Colon Adenocarcinoma tissue samples is $n = 62$.

(B) This is an early example of an expression array, obtained from the Patrick Brown Laboratory at Stanford University. There are $n = 385$ patient samples of tissue from various regions of the body (some from tumors, some not), with gene-expression measurements for $p = 4718$ genes.

(C) The third example is the by now famous NKI data set that produced the 70-gene prognostic signature for breast cancer (Van-De-Vijver et al., 2002). Here there are $n = 295$ samples and $p = 24481$ genes.

Among the above, both (B) and (C) have few missing values—which we imputed by the respective global means of the observed expression values. For each of the three data-sets, we took $\mathbf{S}$ to be the corresponding sample correlation matrix. The *exact thresholding* methodolgy could have also been applied to the sample covariance matrix. Since it is a common practice to standardize the "genes", we operate on the sample correlation matrix.

Figure 1 shows how the component sizes of the thresholded covariance graph change across λ. We describe the strategy we used to arrive at the figure. Note that the connected components change *only* at the absolute values of the entries of $\mathbf{S}$. From the sorted absolute values of the off-diagonal entries of $\mathbf{S}$, we obtained the smallest value of λ, say λ'_{min}, for which the size of the maximal connected component was 1500. For a grid of values of λ till λ'_{min}, we computed the connected components of the thresholded sample-covariance matrix and obtained the size-distribution of the various connected components. Figure 1 shows how these components change over a range of values of λ for the three examples (A), (B) and (C). The number of connected components of a particular size is denoted by a color-scheme, described by the color-bar in the figures. With increasing λ: the larger connected components gradually disappear as they decompose into smaller components; the sizes of the connected components decrease and the frequency of the smaller components increase. Since these are all correlation matrices, for $\lambda \geq 1$ all the nodes in the graph become isolated. The range of λ values for which the maximal size of the components is smaller than 1500 differ across the three examples. For (C) there is a greater variety in the sizes of the components as compared to (A) and (B). Note that by Theorem 1, the pattern of the components appearing in Figure 1 are exactly the same as the components appearing in the solution of (1) for that λ.

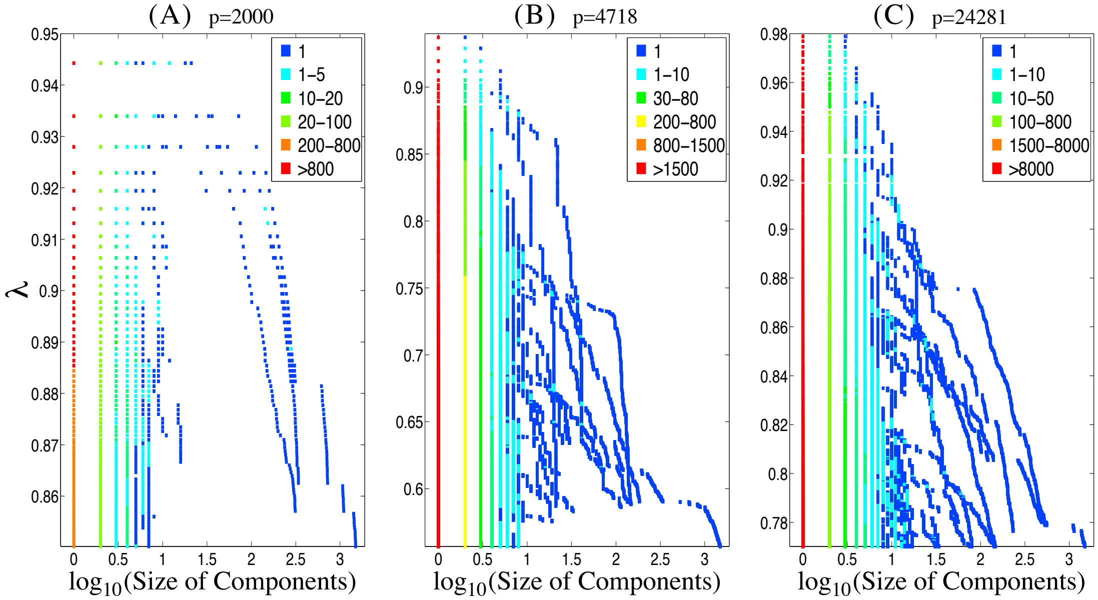


Figure 1: Figure showing the size distribution (in the log-scale) of connected components arising from the thresholded sample covariance graph for examples (A)-(C). For every value of λ (vertical axis), the horizontal slice denotes the sizes of the different components appearing in the thresholded covariance graph. The colors represent the number of components in the graph having that specific size. For every figure, the range of λ values is chosen such that the maximal size of the connected components do not exceed 1500.

For examples (B) and (C) we found that the full problem sizes are beyond the scope of GLASSO and SMACS—the screening rule is apparently the *only* way to obtain solutions for a reasonable range of λ-values as shown in Figure 1.

5. Conclusions

In this paper we present a novel property characterizing the family of solutions to the graphical lasso problem (1), as a function of the regularization parameter λ. The property is fairly surprising—the vertex partition induced by the connected components of the non-zero pattern of the estimated concentration matrix (at λ) and the thresholded sample covariance matrix **S** (at λ) are *exactly equal*. This property seems to have been unobserved in the literature. Our observation not only provides interesting insights into the properties of the graphical lasso solution-path but also opens the door to solving large-scale graphical lasso problems, which are otherwise intractable. This simple rule when used as a wrapper around existing algorithms leads to enormous performance boosts—on occasions by a factor of thousands!

Acknowledgments

This work is supported by NSF-DMS grant 1007719.

Appendix A. Proofs

Here we provide proofs of Theorems 1 and 2.

A.1 Proof of Theorem 1

Proof Suppose $\widehat{\Theta}$ (we suppress the superscript λ for notational convenience) solves problem (1), then standard KKT conditions of optimality (Boyd and Vandenberghe, 2004) give:

$$|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \leq \lambda \quad \forall\, \widehat{\Theta}_{ij} = 0; \quad \text{and} \tag{7}$$

$$\widehat{\mathbf{W}}_{ij} = \mathbf{S}_{ij} + \lambda \quad \forall\, \widehat{\Theta}_{ij} > 0; \quad \widehat{\mathbf{W}}_{ij} = \mathbf{S}_{ij} - \lambda\, \forall\, \widehat{\Theta}_{ij} < 0; \tag{8}$$

where $\widehat{\mathbf{W}} = (\widehat{\Theta})^{-1}$. The diagonal entries satisfy $\widehat{\mathbf{W}}_{ii} = \mathbf{S}_{ii} + \lambda$, for $i = 1, \ldots, p$.

Using (4) and (5), there exists an ordering of the vertices $\{1, \ldots, p\}$ of the graph such that $\mathrm{E}^{(\lambda)}$ is block-diagonal. For notational convenience, we will assume that the matrix is already in that order. Under this ordering of the vertices, the edge-matrix of the thresholded covariance graph is of the form:

$$\mathrm{E}^{(\lambda)} = \begin{pmatrix} \mathrm{E}_1^{(\lambda)} & 0 & \cdots & 0 \\ 0 & \mathrm{E}_2^{(\lambda)} & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathrm{E}_{k(\lambda)}^{(\lambda)} \end{pmatrix} \tag{9}$$

where the different components represent blocks of indices given by: $\mathcal{V}_\ell^{(\lambda)}, \ell = 1, \ldots, k(\lambda)$.

We will construct a matrix $\widehat{\mathbf{W}}$ having the same structure as (9) which is a solution to (1). Note that if $\widehat{\mathbf{W}}$ is block diagonal then so is its inverse. Let $\widehat{\mathbf{W}}$ and its inverse $\widehat{\Theta}$ be given by:

$$\widehat{\mathbf{W}} = \begin{pmatrix} \widehat{\mathbf{W}}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\mathbf{W}}_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \widehat{\mathbf{W}}_{k(\lambda)} \end{pmatrix}, \quad \widehat{\Theta} = \begin{pmatrix} \widehat{\Theta}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\Theta}_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \widehat{\Theta}_{k(\lambda)} \end{pmatrix}$$

Define the block diagonal matrices $\widehat{\mathbf{W}}_\ell$ or equivalently $\widehat{\Theta}_\ell$ via the following sub-problems

$$\widehat{\Theta}_\ell = \arg\min_{\Theta_\ell} \quad \{-\log\det(\Theta_\ell) + \mathrm{tr}(\mathbf{S}_\ell \Theta_\ell) + \lambda \sum_{ij} |\,(\Theta_\ell)_{ij}\,|\} \tag{10}$$

for $\ell = 1, \ldots, k(\lambda)$, where $\mathbf{S}_\ell$ is a sub-block of $\mathbf{S}$, with row/column indices from $\mathcal{V}_\ell^{(\lambda)} \times \mathcal{V}_\ell^{(\lambda)}$. The same notation is used for Θ_ℓ. Denote the inverses of the block-precision matrices by $\{\widehat{\Theta}_\ell\}^{-1} = \widehat{\mathbf{W}}_\ell$. We will show that the above $\widehat{\Theta}$ satisfies the KKT conditions—(7) and (8).

Note that by construction of the thresholded sample covariance graph,
if $i \in \mathcal{V}_\ell^{(\lambda)}$ and $j \in \mathcal{V}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$, then $|\mathbf{S}_{ij}| \leq \lambda$.

Hence, for $i \in \mathcal{V}_\ell^{(\lambda)}$ and $j \in \mathcal{V}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$; the choice $\widehat{\Theta}_{ij} = \widehat{\mathbf{W}}_{ij} = 0$ satisfies the KKT conditions (7)

$$|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \leq \lambda$$

for all the off-diagonal entries in the block-matrix (9).

By construction (10) it is easy to see that for every ℓ, the matrix $\widehat{\Theta}_\ell$ satisfies the KKT conditions (7) and (8) corresponding to the ℓ^{th} block of the $p \times p$ dimensional problem. Hence $\widehat{\Theta}$ solves problem (1).

The above argument shows that the connected components obtained from the estimated precision graph $\mathcal{G}^{(\lambda)}$ leads to a partition of the vertices $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\leq\ell\leq\kappa(\lambda)}$ such that for every $\ell \in \{1,\ldots,k(\lambda)\}$, there is a $\ell' \in \{1,\ldots,\kappa(\lambda)\}$ such that $\widehat{\mathcal{V}}_{\ell'}^{(\lambda)} \subset \mathcal{V}_\ell^{(\lambda)}$. In particular $k(\lambda) \leq \kappa(\lambda)$.

Conversely, if $\widehat{\Theta}$ admits the decomposition as in the statement of the theorem, then it follows from (7) that:
for $i \in \widehat{\mathcal{V}}_\ell^{(\lambda)}$ and $j \in \widehat{\mathcal{V}}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$; $|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \leq \lambda$. Since $\widehat{\mathbf{W}}_{ij} = 0$, we have $|\mathbf{S}_{ij}| \leq \lambda$. This proves that the connected components of $\mathrm{G}^{(\lambda)}$ leads to a partition of the vertices, which is finer than the vertex-partition induced by the components of $\mathcal{G}^{(\lambda)}$. In particular this implies that $k(\lambda) \geq \kappa(\lambda)$.

Combining the above two we conclude $k(\lambda) = \kappa(\lambda)$ and also the equality (6). The permutation π in the theorem appears since the labeling of the connected components is not unique. ■

A.2 Proof of Theorem 2

Proof This proof is a direct consequence of Theorem 1, which establishes that the vertex-partitions induced by the the connected components of the estimated precision graph and the thresholded sample covariance graph are equal.

Observe that, by construction, the connected components of the thresholded sample covariance graph, that is, $\mathrm{G}^{(\lambda)}$ are nested within the connected components of $\mathrm{G}^{(\lambda')}$. In particular, the vertex-partition induced by the components of the thresholded sample covariance graph at λ, is contained inside the vertex-partition induced by the components of the thresholded sample covariance graph at λ'. Now, using Theorem 1 we conclude that the vertex-partition induced by the components of the estimated precision graph at λ, given by $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\leq\ell\leq\kappa(\lambda)}$ is contained inside the vertex-partition induced by the components of the estimated precision graph at λ', given by $\{\widehat{\mathcal{V}}_\ell^{(\lambda')}\}_{1\leq\ell\leq\kappa(\lambda')}$. The proof is thus complete. ■

References

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, June 1999. ISSN 0027-8424. doi: 10.1073/pnas.96.12.6745. URL http://dx.doi.org/10.1073/pnas.96.12.6745.

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. Bollobas. *Modern Graph Theory*. Springer, New York, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

D.R Cox and N. Wermuth. *Multivariate Dependencies*. Chapman and Hall, London, 1996.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2007.

H. Gazit. An optimal randomized parallel algorithm for finding connected components in a graph. *SIAM Journal on Computing*, 20(6):1046–1067, 1991.

S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

Z. Lu. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19:1807–1827, February 2009. ISSN 1052-6234. doi: 10.1137/070695915. URL `http://portal.acm.org/citation.cfm?id=1654243.1654257`.

Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31:2000–2016, May 2010. ISSN 0895-4798. doi: http://dx.doi.org/10.1137/080742531. URL `http://dx.doi.org/10.1137/080742531`.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming, Series A*, 103:127–152, 2005.

A.J. Rothman, P.J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Neural Information Processing Systems*, pages 2101–2109, 2010.

R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2): 146160, 1972.

M. J. Van-De-Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, Dec 2002.

D. Witten and J. Friedman. A fast screening rule for the graphical lasso. *accepted for publication in Journal of Computational and Graphical Statistics*, 2011. Report dated 5-12-2011.

D. Witten, J. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011. (Reference draft dated 9/8/2011).

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

X. Yuan. Alternating direction methods for sparse covariance selection. *Methods*, (August):1–12, 2009. URL `http://www.optimization-online.org/DB-FILE/2009/09/2390.pdf`.

S. Zhou, P. Rütimann, M. Xu, and P. Bühlmann. High-dimensional covariance estimation based on gaussian graphical models. *Journal of Machine Learning Research*, 999888:2975–3026, November 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=2078183.2078201`.

Journal of Machine Learning Research 13 (2012) 795-828 Submitted 1/11; Revised 10/11; Published 3/12

Algorithms for Learning Kernels Based on Centered Alignment

Corinna Cortes CORTES@GOOGLE.COM
Google Research
76 Ninth Avenue
New York, NY 10011

Mehryar Mohri MOHRI@CIMS.NYU.EDU
Courant Institute and Google Research
251 Mercer Street
New York, NY 10012

Afshin Rostamizadeh* ROSTAMI@GOOGLE.COM
Google Research
76 Ninth Avenue
New York, NY 10011

Editor: Francis Bach

Abstract

This paper presents new and effective algorithms for learning kernels. In particular, as shown by our empirical results, these algorithms consistently outperform the so-called uniform combination solution that has proven to be difficult to improve upon in the past, as well as other algorithms for learning kernels based on convex combinations of base kernels in both classification and regression. Our algorithms are based on the notion of centered alignment which is used as a similarity measure between kernels or kernel matrices. We present a number of novel algorithmic, theoretical, and empirical results for learning kernels based on our notion of centered alignment. In particular, we describe efficient algorithms for learning a maximum alignment kernel by showing that the problem can be reduced to a simple QP and discuss a one-stage algorithm for learning both a kernel and a hypothesis based on that kernel using an alignment-based regularization. Our theoretical results include a novel concentration bound for centered alignment between kernel matrices, the proof of the existence of effective predictors for kernels with high alignment, both for classification and for regression, and the proof of stability-based generalization bounds for a broad family of algorithms for learning kernels based on centered alignment. We also report the results of experiments with our centered alignment-based algorithms in both classification and regression.

Keywords: kernel methods, learning kernels, feature selection

1. Introduction

One of the key steps in the design of learning algorithms is the choice of the features. This choice is typically left to the user and represents his prior knowledge, but it is critical: a poor choice makes learning challenging while a better choice makes it more likely to be successful. The general objective of this work is to define effective methods that partially relieve the user from the requirement of specifying the features.

*. A significant amount of the presented work was completed while AR was a graduate student at the Courant Institute of Mathematical Sciences and a postdoctoral scholar at the University of Califorinia at Berkeley.

For kernel-based algorithms the features are provided intrinsically via the choice of a positive-definite symmetric kernel function (Boser et al., 1992; Cortes and Vapnik, 1995; Vapnik, 1998). To limit the risk of a poor choice of kernel, in the last decade or so, a number of publications have investigated the idea of *learning the kernel* from data (Cristianini et al., 2001; Chapelle et al., 2002; Bousquet and Herrmann, 2002; Lanckriet et al., 2004; Jebara, 2004; Argyriou et al., 2005; Micchelli and Pontil, 2005; Lewis et al., 2006; Argyriou et al., 2006; Kim et al., 2006; Cortes et al., 2008; Sonnenburg et al., 2006; Srebro and Ben-David, 2006; Zien and Ong, 2007; Cortes et al., 2009a, 2010a,b). This reduces the requirement from the user to only specifying a family of kernels rather than a specific kernel. The task of selecting (or learning) a kernel out of that family is then reserved to the learning algorithm which, as for standard kernel-based methods, must also use the data to choose a hypothesis in the reproducing kernel Hilbert space (RKHS) associated to the kernel selected.

Different kernel families have been studied in the past, but the most widely used one has been that of convex combinations of a finite set of base kernels. However, while different learning kernel algorithms have been introduced in that case, including those of Lanckriet et al. (2004), to our knowledge, in the past, none has succeeded in consistently and significantly outperforming the *uniform combination* solution, in binary classification or regression tasks. The uniform solution consists of simply learning a hypothesis out of the RKHS associated to a uniform combination of the base kernels. This disappointing performance of learning kernel algorithms has been pointed out in different instances, including by many participants at different NIPS workshops organized on the theme in 2008 and 2009, as well as in a survey talk (Cortes, 2009) and tutorial (Cortes et al., 2011b). The empirical results we report further confirm this observation. Other kernel families have been considered in the literature, including hyperkernels (Ong et al., 2005), Gaussian kernel families (Micchelli and Pontil, 2005), or non-linear families (Bach, 2008; Cortes et al., 2009b; Varma and Babu, 2009). However, the performance reported for these other families does not seem to be consistently superior to that of the uniform combination either.

In contrast, on the theoretical side, favorable guarantees have been derived for learning kernels. For general kernel families, learning bounds based on covering numbers were given by Srebro and Ben-David (2006). Stronger margin-based generalization guarantees based on an analysis of the Rademacher complexity, with only a square-root logarithmic dependency on the number of base kernels were given by Cortes et al. (2010b) for convex combinations of kernels with an L_1 constraint. The dependency of theses bounds, as well as others given for L_q constraints, were shown to be optimal with respect to the number of kernels. These L_1 bounds generalize those presented in Koltchinskii and Yuan (2008) in the context of ensembles of kernel machines. The learning guarantees suggest that learning kernel algorithms even with a relatively large number of base kernels could achieve a good performance.

This paper presents new algorithms for learning kernels whose performance is more consistent with expectations based on these theoretical guarantees. In particular, as can be seen by our experimental results, several of the algorithms we describe consistently outperform the uniform combination solution. They also surpass in performance the algorithm of Lanckriet et al. (2004) in classification and improve upon that of Cortes et al. (2009a) in regression. Thus, this can be viewed as the first series of algorithmic solutions for learning kernels in classification and regression with consistent performance improvements.

Our learning kernel algorithms are based on the notion of *centered alignment* which is a similarity measure between kernels or kernel matrices. This can be used to measure the similarity of each base kernel with the target kernel K_Y derived from the output labels. Our definition of centered alignment is close to the uncentered kernel alignment originally introduced by Cristianini et al. (2001). This closeness is only superficial however: as we shall see both from the analysis of several cases and from experimental results, in contrast with our notion of alignment, the uncentered kernel alignment of Cristianini et al. (2001) does not correlate well with performance and thus, in general, cannot be used effectively for learning kernels. We note that other kernel optimization criteria similar to centered alignment, but without the key normalization have been used by some authors (Kim et al., 2006; Gretton et al., 2005). Both the centering and the normalization are critical components of our definition.

We present a number of novel algorithmic, theoretical, and empirical results for learning kernels based on our notion of centered alignment. In Section 2, we introduce and analyze the properties of centered alignment between kernel functions and kernel matrices, and discuss its benefits. In particular, the importance of the centering is justified theoretically and validated empirically. We then describe several algorithms based on the notion of centered alignment in Section 3.

We present two algorithms that each work in two subsequent stages (Sections 3.1 and 3.2): the first stage consists of *learning* a kernel K that is a non-negative linear combination of p base kernels; the second stage combines this kernel with a standard kernel-based learning algorithm such as support vector machines (SVMs) (Cortes and Vapnik, 1995) for classification, or kernel ridge regression (KRR) for regression (Saunders et al., 1998), to select a prediction hypothesis. These two algorithms differ in the way centered alignment is used to learn K. The simplest and most straightforward to implement algorithm selects the weight of each base kernel matrix independently, only from the centered alignment of that matrix with the target kernel matrix. The other more accurate algorithm instead determines these weights jointly by measuring the centered alignment of a convex combination of base kernel matrices with the target one. We show that this more accurate algorithm is very efficient by proving that the base kernel weights can be obtained by solving a simple quadratic program (QP). We also give a closed-form expression for the weights in the case of a linear, but not necessarily convex, combination. Note that an alternative two-stage technique consists of first learning a prediction hypothesis using each base kernel and then learning the best linear combination of these hypotheses. But, as pointed out in Section 3.3, in general, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms. In addition, we present and analyze an algorithm that uses centered alignment to both select a convex combination kernel and a hypothesis based on that kernel, these two tasks being performed in a single stage by solving a single optimization problem (Section 3.4).

We also present an extensive theoretical analysis of the notion of centered alignment and algorithms based on that notion. We prove a concentration bound for the notion of centered alignment showing that the centered alignment of two kernel matrices is sharply concentrated around the centered alignment of the corresponding kernel functions, the difference being bounded by a term in $O(1/\sqrt{m})$ for samples of size m (Section 4.1). Our result is simpler and directly bounds the difference between these two relevant quantities, unlike previous work by Cristianini et al. (2001) (for uncentered alignments). We also show the existence of good predictors for kernels with high centered alignment, both for classification and for regression (Section 4.2). This result justifies the search for good learning kernel algorithms based on the notion of centered alignment. We note that the proofs given for similar results in classification for uncentered alignments by Cristianini et al. (2001, 2002) are erroneous. We also present stability-based generalization bounds for two-stage learning kernel algorithms based on centered alignment when the second stage is kernel ridge regression (Section 4.3). We further study the application of these bounds in the case of our alignment maximization algorithm and initiate a detailed analysis of the stability of this algorithm (Appendix B).

Finally, in Section 5, we report the results of experiments with our centered alignment-based algorithms both in classification and regression, and compare our results with L_1- and L_2-regularized learning kernel algorithms (Lanckriet et al., 2004; Cortes et al., 2009a), as well as with the uniform kernel combination method. The results show an improvement both over the uniform combination and over the one-stage kernel learning algorithms. They also demonstrate a strong correlation between the centered alignment achieved and the performance of the algorithm.[1]

2. Alignment Definitions

The notion of kernel alignment was first introduced by Cristianini et al. (2001). Our definition of kernel alignment is different and is based on the notion of centering in the feature space. Thus, we start with the definition of centering and the analysis of its relevant properties.

2.1 Centered Kernel Functions

Let D be the distribution according to which training and test points are drawn. A feature mapping $\Phi\colon \mathcal{X} \to H$ is centered by subtracting from it its expectation, that is forming it by $\Phi - \mathrm{E}_x[\Phi]$, where E_x denotes the expected value of Φ when x is drawn according to the distribution D. Centering a positive definite symmetric (PDS) kernel function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ consists of centering any feature mapping Φ associated to K. Thus, the centered kernel K_c associated to K is defined for all $x, x' \in \mathcal{X}$ by

$$\begin{aligned} K_c(x,x') &= (\Phi(x) - \mathop{\mathrm{E}}_{x}[\Phi])^\top(\Phi(x') - \mathop{\mathrm{E}}_{x'}[\Phi]) \\ &= K(x,x') - \mathop{\mathrm{E}}_{x}[K(x,x')] - \mathop{\mathrm{E}}_{x'}[K(x,x')] + \mathop{\mathrm{E}}_{x,x'}[K(x,x')]. \end{aligned}$$

This also shows that the definition does not depend on the choice of the feature mapping associated to K. Since $K_c(x,x')$ is defined as an inner product, K_c is also a PDS kernel.[2] Note also that for a centered kernel K_c, $\mathrm{E}_{x,x'}[K_c(x,x')] = 0$, that is, centering the feature mapping implies centering the kernel function.

2.2 Centered Kernel Matrices

Similar definitions can be given for a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D: a feature vector $\Phi(x_i)$ with $i \in [1, m]$ is centered by subtracting from it its empirical expectation, that is forming it with $\Phi(x_i) - \overline{\Phi}$, where $\overline{\Phi} = \frac{1}{m}\sum_{i=1}^m \Phi(x_i)$. The kernel matrix $\mathbf{K}$ associated to K and the sample

[1] This is an extended version of Cortes et al. (2010a) with much additional material, including additional empirical evidence supporting the importance of centered alignment, the description and discussion of a single-stage algorithm for learning kernels based on centered alignment, an analysis of unnormalized centered alignment and the proof of the existence of good predictors for large values of centered alignment, generalization bounds for two-stage learning kernel algorithms based on centered alignment, and an experimental investigation of the single-stage algorithm.

[2] For convenience, we use a matrix notation for feature vectors and use $\mathbf{\Phi}(x)^\top \mathbf{\Phi}(x')$ to denote the inner product between two feature vectors and similarly $\mathbf{\Phi}(x)\mathbf{\Phi}(x')^\top$ for the outer product, including in the case where the dimension of the feature space is infinite, in which case we are using infinite matrices.

S is centered by replacing it with $\mathbf{K}_c$ defined for all $i, j \in [1, m]$ by

$$[\mathbf{K}_c]_{ij} = \mathbf{K}_{ij} - \frac{1}{m}\sum_{i=1}^{m} \mathbf{K}_{ij} - \frac{1}{m}\sum_{j=1}^{m} \mathbf{K}_{ij} + \frac{1}{m^2}\sum_{i,j=1}^{m} \mathbf{K}_{ij}. \tag{1}$$

Let $\mathbf{\Phi} = [\Phi(x_1), \ldots, \Phi(x_m)]^\top$ and $\overline{\mathbf{\Phi}} = [\overline{\Phi}, \ldots, \overline{\Phi}]^\top$. Then, it is not hard to verify that $\mathbf{K}_c = (\mathbf{\Phi} - \overline{\mathbf{\Phi}})(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^\top$, which shows that $\mathbf{K}_c$ is a positive semi-definite (PSD) matrix. Also, as with the kernel function, $\frac{1}{m^2}\sum_{i,j=1}^{m}[\mathbf{K}_c]_{ij} = 0$. Let $\langle \cdot, \cdot \rangle_F$ denote the Frobenius product and $\| \cdot \|_F$ the Frobenius norm defined by

$$\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}, \langle \mathbf{A}, \mathbf{B} \rangle_F = \operatorname{Tr}[\mathbf{A}^\top \mathbf{B}] \text{ and } \|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F}.$$

Then, the following basic properties hold for centering kernel matrices.

Lemma 1 *Let* $\mathbf{1} \in \mathbb{R}^{m \times 1}$ *denote the vector with all entries equal to one, and* $\mathbf{I}$ *the identity matrix.*

1. *For any kernel matrix* $\mathbf{K} \in \mathbb{R}^{m \times m}$*, the centered kernel matrix* $\mathbf{K}_c$ *can be expressed as follows*

$$\mathbf{K}_c = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right] \mathbf{K} \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right].$$

2. *For any two kernel matrices* $\mathbf{K}$ *and* $\mathbf{K}'$*,*

$$\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F.$$

Proof The first statement can be shown straightforwardly from the definition of $\mathbf{K}_c$ (Equation (1)). The second statement follows from

$$\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \operatorname{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right] \mathbf{K} \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right] \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right] \mathbf{K}' \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\right],$$

the fact that $[\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]^2 = [\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]$, and the trace property $\operatorname{Tr}[\mathbf{AB}] = \operatorname{Tr}[\mathbf{BA}]$, valid for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$. ■

We shall use these properties in the proofs of the results presented in Section 4.

2.3 Centered Kernel Alignment

In the following sections, in the absence of ambiguity, to abbreviate the notation, we often omit the variables over which an expectation is taken. We define the alignment of two kernel functions as follows.

Definition 2 (Kernel function alignment) *Let* K *and* K' *be two kernel functions defined over* $\mathcal{X} \times \mathcal{X}$ *such that* $0 < \mathrm{E}[K_c^2] < +\infty$ *and* $0 < \mathrm{E}[K_c'^2] < +\infty$*. Then, the* alignment *between* K *and* K' *is defined by*

$$\rho(K, K') = \frac{\mathrm{E}[K_c K'_c]}{\sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[K_c'^2]}}\ .$$

Since $|\,\mathrm{E}[K_c K'_c]| \leq \sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[K_c'^2]}$ by the Cauchy-Schwarz inequality, we have $\rho(K, K') \in [-1, 1]$. The following lemma shows more precisely that $\rho(K, K') \in [0, 1]$ when K and K' are PDS kernels.

Lemma 3 *For any two PDS kernels K and K',* $\mathrm{E}[KK'] \geq 0$.

Proof Let Φ be a feature mapping associated to K and Φ' a feature mapping associated to K'. By definition of Φ and Φ', and using the properties of the trace, we can write:

$$\begin{aligned}\mathop{\mathrm{E}}_{x,x'}[K(x,x')K'(x,x')] &= \mathop{\mathrm{E}}_{x,x'}[\Phi(x)^\top \Phi(x')\Phi'(x')^\top \Phi'(x)] \\ &= \mathop{\mathrm{E}}_{x,x'}\left[\mathrm{Tr}[\Phi(x)^\top \Phi(x')\Phi'(x')^\top \Phi'(x)]\right] \\ &= \langle \mathop{\mathrm{E}}_{x}[\Phi(x)\Phi'(x)^\top], \mathop{\mathrm{E}}_{x'}[\Phi(x')\Phi'(x')^\top]\rangle_F = \|\mathbf{U}\|_F^2 \geq 0,\end{aligned}$$

where $\mathbf{U} = \mathrm{E}_x[\Phi(x)\Phi'(x)^\top]$. ■

The lemma applies in particular to any two centered kernels K_c and K'_c which, as previously shown, are PDS kernels if K and K' are PDS. Thus, for any two PDS kernels K and K', the following holds:

$$\mathrm{E}[K_c K'_c] \geq 0.$$

We can define similarly the alignment between two kernel matrices $\mathbf{K}$ and $\mathbf{K}'$ based on a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D.

Definition 4 (Kernel matrix alignment) *Let $\mathbf{K} \in \mathbb{R}^{m\times m}$ and $\mathbf{K}' \in \mathbb{R}^{m\times m}$ be two kernel matrices such that $\|\mathbf{K}_c\|_F \neq 0$ and $\|\mathbf{K}'_c\|_F \neq 0$. Then, the* alignment *between $\mathbf{K}$ and $\mathbf{K}'$ is defined by*

$$\widehat{\rho}(\mathbf{K}, \mathbf{K}') = \frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{\|\mathbf{K}_c\|_F \|\mathbf{K}'_c\|_F}\ .$$

Here too, by the Cauchy-Schwarz inequality, $\widehat{\rho}(\mathbf{K}, \mathbf{K}') \in [-1, 1]$ and in fact $\widehat{\rho}(\mathbf{K}, \mathbf{K}') \geq 0$ since the Frobenius product of any two positive semi-definite matrices $\mathbf{K}$ and $\mathbf{K}'$ is non-negative. Indeed, for such matrices, there exist matrices $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{K} = \mathbf{U}\mathbf{U}^\top$ and $\mathbf{K}' = \mathbf{V}\mathbf{V}^\top$. The statement follows from

$$\langle \mathbf{K}, \mathbf{K}'\rangle_F = \mathrm{Tr}(\mathbf{U}\mathbf{U}^\top\mathbf{V}\mathbf{V}^\top) = \mathrm{Tr}\left((\mathbf{U}^\top\mathbf{V})^\top(\mathbf{U}^\top\mathbf{V})\right) = \|\mathbf{U}^\top\mathbf{V}\|_F^2 \geq 0. \tag{2}$$

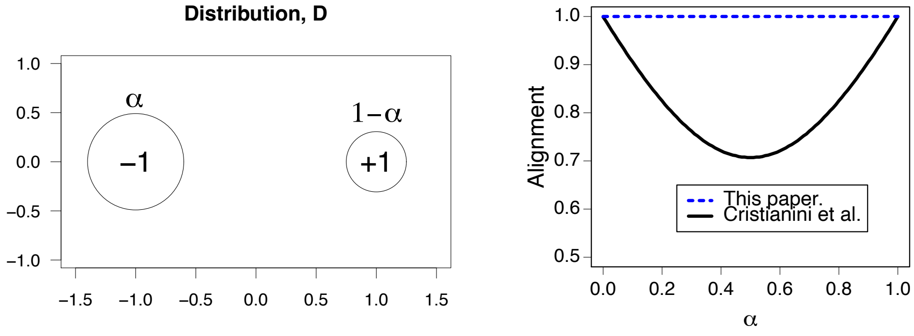
This applies in particular to the kernel matrices of the PDS kernels K_c and K'_c:

$$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F \geq 0.$$

Our definitions of alignment between kernel functions or between kernel matrices differ from those originally given by Cristianini et al. (2001, 2002):

$$A = \frac{\mathrm{E}[KK']}{\sqrt{\mathrm{E}[K^2]\,\mathrm{E}[K'^2]}}, \quad \widehat{A} = \frac{\langle \mathbf{K}, \mathbf{K}'\rangle_F}{\|\mathbf{K}\|_F \|\mathbf{K}'\|_F},$$

which are thus in terms of K and K' instead of K_c and K'_c and similarly for matrices. This may appear to be a technicality, but it is in fact a critical difference. Without that centering, the definition of alignment does not correlate well with performance. To see this, consider the standard case where K' is the target label kernel, that is $K'(x, x') = yy'$, with y the label of x and y' the label of x', and examine the following simple example in dimension two ($\mathcal{X} = \mathbb{R}^2$), where $K(x, x') = x \cdot x' + 1$ and



(a) (b)

Figure 1: (a) Representation of the distribution D. In this simple two-dimensional example, a fraction α of the points are at $(-1,0)$ and have the label -1. The remaining points are at $(1,0)$ and have the label $+1$. (b) Alignment values computed for two different definitions of alignment. The solid line in black plots the definition of alignment computed according to Cristianini et al. (2001) $A = (\alpha^2 + (1-\alpha)^2)^{1/2}$, while our definition of centered alignment results in the straight dotted blue line $\rho = 1$.

	KINEMATICS	IONOSPHERE	GERMAN	SPAMBASE	SPLICE
	(REGR.)	(REGR.)	(CLASS.)	(CLASS.)	(CLASS.)
$\widehat{\rho}$	0.9624	0.9979	0.9439	0.9918	0.9515
$\widehat{A}$	0.8627	0.9841	0.9390	0.9889	-0.4484

Table 1: The correlations of the alignment values and error-rates of various kernels. The top row reports the correlation of the accuracy of the base kernels used in Section 5 with the centered alignments $\widehat{\rho}$, the bottom row the correlation with the non-centered alignment $\widehat{A}$.

where the distribution D is defined by a fraction $\alpha \in [0,1]$ of all points being at $(-1,0)$ and labeled with -1, and the remaining points at $(1,0)$ with label $+1$, as shown in Figure 1.

Clearly, for any value of $\alpha \in [0,1]$, the problem is separable, for example with the simple vertical line going through the origin, and one would expect the alignment to be 1. However, the alignment A calculated using the definition of the distribution D admits a different expression. Using

$$\begin{aligned}
&\mathrm{E}[K'^2] = 1,\\
&\mathrm{E}[K^2] = \alpha^2 \cdot 4 + (1-\alpha)^2 \cdot 4 + 2\alpha(1-\alpha) \cdot 0 = 4\big(\alpha^2 + (1-\alpha)^2\big),\\
&\mathrm{E}[KK'] = \alpha^2 \cdot 2 + (1-\alpha)^2 \cdot 2 + 2\alpha(1-\alpha) \cdot 0 = 2\big(\alpha^2 + (1-\alpha)^2\big),
\end{aligned}$$

gives $A = (\alpha^2 + (1-\alpha)^2)^{1/2}$. Thus, A is never equal to one except for $\alpha = 0$ or $\alpha = 1$ and for the balanced case where $\alpha = 1/2$, its value is $A = 1/\sqrt{2} \approx .707 < 1$. In contrast, with our definition, $\rho(K,K') = 1$ for all $\alpha \in [0,1]$ (see Figure 1).

This mismatch between A (or $\widehat{A}$) and the performance values can also be seen in real world data sets. Instances of this problem have been noticed by Meila (2003) and Pothin and Richard

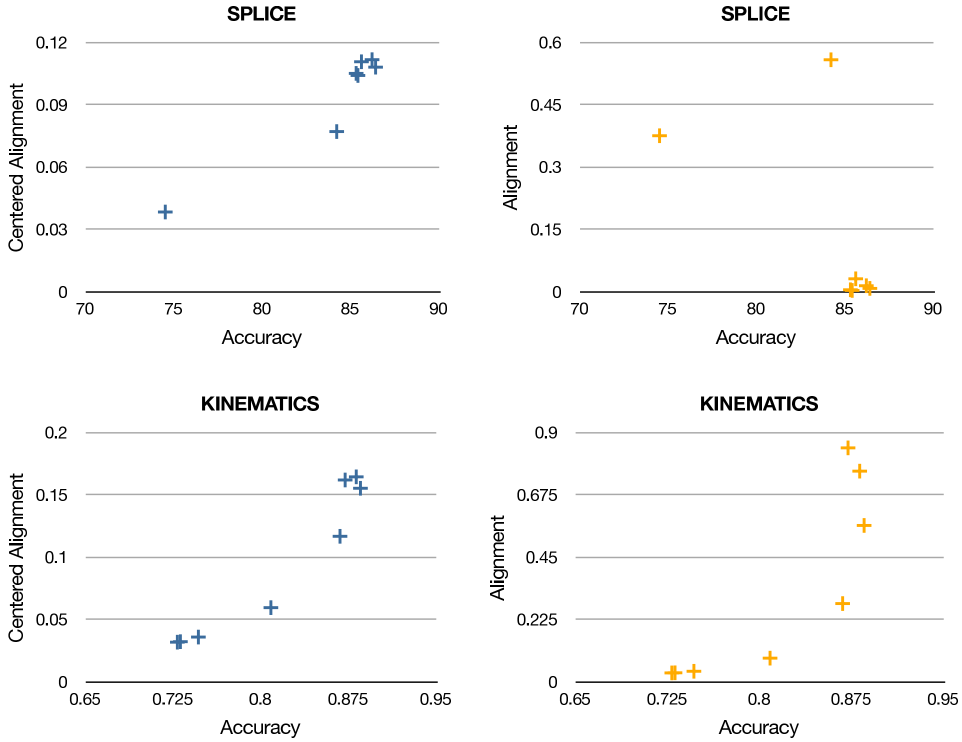


Figure 2: Detailed view of the splice and kinematics experiments presented in Table 1. Both the centered (plots in blue on left) and non-centered alignment (plots in orange on right) are plotted as a function of the accuracy (for the regression problem in the kinematics task "accuracy" is 1 - RMSE). It is apparent from these plots that the non-centered alignment can be misleading when evaluating the quality of a kernel.

(2008) who have suggested various (input) data translation methods, and by Cristianini et al. (2002) who observed an issue for unbalanced data sets. Table 1, as well as Figure 2, give a series of empirical results in several classification and regression tasks based on data sets taken from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`) and Delve data sets (`http://www.cs.toronto.edu/~delve/data/datasets.html`). The table and the figure illustrate the fact that the quantity $\widehat{A}$ measured with respect to several different kernels does not always correlate well with the performance achieved by each kernel. In fact, for the splice classification task, the non-centered alignment is negatively correlated with the accuracy, while a large positive correlation is expected of a good quality measure. The centered notion of alignment $\widehat{\rho}$ however, shows good correlation along all data sets and is always better correlated than $\widehat{A}$.

The notion of alignment seeks to capture the correlation between the random variables $K(x,x')$ and $K'(x,x')$ and one could think it natural, as for the standard correlation coefficients, to consider the following definition:

$$\rho'(K,K') = \frac{\mathrm{E}[(K-\mathrm{E}[K])(K'-\mathrm{E}[K'])]}{\sqrt{\mathrm{E}[(K-\mathrm{E}[K])^2]\,\mathrm{E}[(K'-\mathrm{E}[K'])^2]}} \ .$$

However, centering the kernel values, as opposed to centering the feature values, is not directly relevant to linear predictions in feature space, while our definition of alignment ρ is precisely related to that. Also, as already shown in Section 2.1, centering in the feature space implies the centering of the kernel values, since $\mathrm{E}[K_c] = 0$ and $\frac{1}{m^2}\sum_{i,j=1}^{m}[\mathbf{K}_c]_{ij} = 0$ for any kernel K and kernel matrix $\mathbf{K}$. Conversely, however, centering the kernel does not imply centering in feature space. For example, consider any kernel where all the row marginals are not all equal.

3. Algorithms

This section discusses several learning kernel algorithms based on the notion of centered alignment. In all cases, the family of kernels considered is that of non-negative combinations of p base kernels K_k, $k \in [1, p]$. Thus, the final hypothesis learned belongs to the reproducing kernel Hilbert space (RKHS) $\mathbb{H}_{K_{\boldsymbol{\mu}}}$ associated to a kernel of the form $K_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k K_k$, with $\boldsymbol{\mu} \geq 0$, which guarantees that $K_{\boldsymbol{\mu}}$ is PDS, and $\|\boldsymbol{\mu}\| = \Lambda \geq 0$, for some regularization parameter Λ.

We first describe and analyze two algorithms that both work in two stages: in the first stage, these algorithms determine the mixture weights $\boldsymbol{\mu}$. In the second stage, they train a standard kernel-based algorithm, for example, SVMs for classification, or KRR for regression, in combination with the kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$ associated to $K_{\boldsymbol{\mu}}$, to learn a hypothesis $h \in \mathbb{H}_{K_{\boldsymbol{\mu}}}$. Thus, these *two-stage algorithms* differ only by their first stage, which determines $K_{\boldsymbol{\mu}}$. We describe first in Section 3.1 a simple algorithm that determines each mixture weight μ_k independently, (`align`), then, in Section 3.2, an algorithm that determines the weights μ_ks jointly (`alignf`) by selecting $\boldsymbol{\mu}$ to maximize the alignment with the target kernel. We briefly discuss in Section 3.3 the relationship of such two-stage learning algorithms with algorithms based on ensemble techniques, which also consist of two stages. Finally, we introduce and analyze a *single-stage alignment-based algorithm* which learns $\boldsymbol{\mu}$ and the hypothesis $h \in \mathbb{H}_{K_{\boldsymbol{\mu}}}$ simultaneously in Section 3.4.

3.1 Independent Alignment-based Algorithm (`align`)

This is a simple but efficient method which consists of using the training sample to independently compute the alignment between each kernel matrix $\mathbf{K}_k$ and the target kernel matrix $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$, based on the labels $\mathbf{y}$, and to choose each mixture weight μ_k proportional to that alignment. Thus, the resulting kernel matrix is defined by:

$$\mathbf{K}_{\boldsymbol{\mu}} \propto \sum_{k=1}^{p} \widehat{\rho}(\mathbf{K}_k, \mathbf{K}_Y)\mathbf{K}_k = \frac{1}{\|\mathbf{K}_Y\|_F} \sum_{k=1}^{p} \frac{\langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F}{\|\mathbf{K}_k\|_F} \mathbf{K}_k. \tag{3}$$

When the base kernel matrices $\mathbf{K}_k$ have been normalized with respect to the Frobenius norm, the independent alignment-based algorithm can also be viewed as the solution of a joint maximization of the unnormalized alignment defined as follows, with a L_2-norm constraint on the norm of $\boldsymbol{\mu}$.

Definition 5 (Unnormalized alignment) *Let K and K' be two PDS kernels defined over $\mathcal{X} \times \mathcal{X}$ and $\mathbf{K}$ and $\mathbf{K}'$ their kernel matrices for a sample of size m. Then, the* unnormalized alignment $\rho_{\mathrm{u}}(K, K')$ *between K and K' and the* unnormalized alignment $\widehat{\rho_{\mathrm{u}}}(\mathbf{K}, \mathbf{K}')$ *between $\mathbf{K}$ and $\mathbf{K}'$ are defined by*

$$\rho_{\mathrm{u}}(K, K') = \mathop{\mathrm{E}}_{x,x'}[K_c(x, x')K'_c(x, x')] \quad \textit{and} \quad \widehat{\rho_{\mathrm{u}}}(\mathbf{K}, \mathbf{K}') = \frac{1}{m^2}\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F.$$

Since they are not normalized, the alignment values a and $\widehat{a}$ are no longer guaranteed to be in the interval $[0,1]$. However, assuming the kernel function K and labels are bounded, the unnormalized alignment between K and K_Y are bounded as well.

Lemma 6 *Let K be a PDS kernel. Assume that for all $x \in \mathcal{X}$, $K_c(x,x) \leq R^2$ and for all output label y, $|y| \leq M$. Then, the following bounds hold:*

$$0 \leq \rho_{\mathrm{u}}(K, K_Y) \leq MR^2 \quad \textit{and} \quad 0 \leq \widehat{\rho}_{\mathrm{u}}(\mathbf{K}, \mathbf{K}_Y) \leq MR^2.$$

Proof The lower bounds hold by Lemma 3 and Inequality (2). The upper bounds can be obtained straightforwardly via the application of the Cauchy-Schwarz inequality:

$$\rho_{\mathrm{u}}^2(K, K_Y) = \mathop{\mathrm{E}}_{(x,y),(x',y')}[K_c(x,x')yy']^2 \leq \mathop{\mathrm{E}}_{x,x'}[K_c^2(x,x')] \mathop{\mathrm{E}}_{y,y'}[yy']^2 \leq R^4M^2$$

$$\widehat{\rho}_{\mathrm{u}}(\mathbf{K}, \mathbf{K}') = \frac{1}{m^2}\langle \mathbf{K}_c, \mathbf{K}_Y \rangle_F \leq \frac{1}{m^2}\|\mathbf{K}_c\|_F \|\mathbf{K}_y\|_F \leq \frac{mR^2mM}{m^2} = R^2M,$$

where we used the identity $\langle \mathbf{K}_c, \mathbf{K}_{Yc} \rangle_F = \langle \mathbf{K}_c, \mathbf{K}_Y \rangle_F$ from Lemma 1. ■

We will consider more generally the corresponding optimization with an L_q-norm constraint on $\boldsymbol{\mu}$ with $q > 1$:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \; & \widehat{\rho}_{\mathrm{u}}\Big(\sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y\Big) = \Big\langle \sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y \Big\rangle_F \\ \text{subject to: } & \sum_{k=1}^{p} \mu_k^q \leq \Lambda. \end{aligned} \tag{4}$$

An explicit constraint enforcing $\boldsymbol{\mu} \geq \mathbf{0}$ is not necessary since, as we shall see, the optimal solution found always satisfies this constraint.

Proposition 7 *Let $\boldsymbol{\mu}^*$ be the solution of the optimization problem* (4)*, then $\mu_k^* \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F^{\frac{1}{q-1}}$.*

Proof The Lagrangian corresponding to the optimization (4) is defined as follows,

$$L(\boldsymbol{\mu}, \beta) = -\sum_{k=1}^{p} \mu_k \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F + \beta\Big(\sum_{k=1}^{p} \mu_k^q - \Lambda\Big),$$

where the dual variable β is non-negative. Differentiating with respect to μ_k and setting the result to zero gives

$$\frac{\partial L}{\partial \mu_k} = -\langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F + q\beta\mu_k^{q-1} = 0 \implies \mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F^{\frac{1}{q-1}},$$

which concludes the proof. ■

Thus, for $q = 2, \mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F$ is exactly the solution given by Equation (3) modulo normalization by the Frobenius norm of the base matrix. Note that for $q = 1$, the optimization becomes trivial and can be solved by simply placing all the weight on μ_k with the largest coefficient, that is the μ_k whose corresponding kernel matrix $\mathbf{K}_k$ has the largest alignment with the target kernel.

3.2 Alignment Maximization Algorithm

The independent alignment-based method ignores the correlation between the base kernel matrices. The alignment maximization method takes these correlations into account. It determines the mixture weights μ_k jointly by seeking to maximize the alignment between the convex combination kernel $\mathbf{K}_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k \mathbf{K}_k$ and the target kernel $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$.

This was also suggested in the case of uncentered alignment by Cristianini et al. (2001); Kandola et al. (2002a) and later studied by Lanckriet et al. (2004) who showed that the problem can be solved as a QCQP (however, as already discussed in Section 2.1, the uncentered alignment is not well correlated with performance). In what follows, we present even more efficient algorithms for computing the weights μ_k by showing that the problem can be reduced to a simple QP. We start by examining the case of a non-convex linear combination where the components of $\boldsymbol{\mu}$ can be negative, and show that the problem admits a closed-form solution in that case. We then partially use that solution to obtain the solution of the convex combination.

3.2.1 LINEAR COMBINATION

We can assume without loss of generality that the centered base kernel matrices $\mathbf{K}_{kc}$ are independent, that is, no linear combination is equal to the zero matrix, otherwise we can select an independent subset. This condition ensures that $\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F > 0$ for arbitrary $\boldsymbol{\mu}$ and that $\widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top)$ is well defined (Definition 4). By Lemma 1, $\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_{Yc} \rangle_F = \langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_Y \rangle_F$. Thus, since $\|\mathbf{K}_{Yc}\|_F$ does not depend on $\boldsymbol{\mu}$, the alignment maximization problem $\max_{\boldsymbol{\mu} \in \mathcal{M}} \widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top)$ can be equivalently written as the following optimization problem:

$$\max_{\boldsymbol{\mu} \in \mathcal{M}} \frac{\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{y}\mathbf{y}^\top \rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F}, \tag{5}$$

where $\mathcal{M} = \{\boldsymbol{\mu} \colon \|\boldsymbol{\mu}\|_2 = 1\}$. A similar set can be defined via the L_1-norm instead of L_2. As we shall see, however, the direction of the solution $\boldsymbol{\mu}^\star$ does not change with respect to the choice of norm. Thus, the problem can be solved in the same way in both cases and subsequently scaled appropriately. Note that, by Lemma 1, $\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \mathbf{K}_{\boldsymbol{\mu}} \mathbf{U}_m$ with $\mathbf{U}_m = \mathbf{I} - \mathbf{1}\mathbf{1}^\top/m$, thus,

$$\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \Big(\sum_{k=1}^{p} \mu_k \mathbf{K}_k \Big) \mathbf{U}_m = \sum_{k=1}^{p} \mu_k \mathbf{U}_m \mathbf{K}_k \mathbf{U}_m = \sum_{k=1}^{p} \mu_k \mathbf{K}_{kc}.$$

Let

$$\mathbf{a} = (\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \ldots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top \rangle_F)^\top,$$

and let $\mathbf{M}$ denote the matrix defined by

$$\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F,$$

for $k, l \in [1, p]$. Note that, in view of the non-negativity of the Frobenius product of symmetric PSD matrices shown in Section 2.3, the entries of $\mathbf{a}$ and $\mathbf{M}$ are all non-negative. Observe also that $\mathbf{M}$ is a symmetric PSD matrix since for any vector $\mathbf{X} = (x_1, \ldots, x_p)^\top \in \mathbb{R}^p$,

$$\begin{aligned}
\mathbf{X}^\top \mathbf{M} \mathbf{X} &= \sum_{k,l=1}^{p} x_k x_l \mathbf{M}_{kl} \\
&= \mathrm{Tr}\Big[\sum_{k,l=1}^{p} x_k x_l \mathbf{K}_{kc} \mathbf{K}_{lc}\Big] \\
&= \mathrm{Tr}\Big[(\sum_{k=1}^{p} x_k \mathbf{K}_{kc})(\sum_{l=1}^{p} x_l \mathbf{K}_{lc})\Big] = \| \sum_{k=1}^{p} x_k \mathbf{K}_{kc} \|_F^2 > 0.
\end{aligned}$$

The strict inequality follows from the fact that the base kernels are linearly independent. Since this inequality holds for any non-zero $\mathbf{X}$, it also shows that $\mathbf{M}$ is invertible.

Proposition 8 *The solution $\boldsymbol{\mu}^\star$ of the optimization problem (5) is given by $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$.*

Proof With the notation introduced, problem (5) can be rewritten as $\boldsymbol{\mu}^\star = \mathrm{argmax}_{\|\boldsymbol{\mu}\|_2=1} \frac{\boldsymbol{\mu}^\top \mathbf{a}}{\sqrt{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}}$. Thus, clearly, the solution must verify $\boldsymbol{\mu}^{\star\top}\mathbf{a} \geq 0$. We will square the objective and yet not enforce this condition since, as we shall see, it will be verified by the solution we find. Therefore, we consider the problem

$$\boldsymbol{\mu}^\star = \underset{\|\boldsymbol{\mu}\|_2=1}{\mathrm{argmax}} \frac{(\boldsymbol{\mu}^\top \mathbf{a})^2}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}} = \underset{\|\boldsymbol{\mu}\|_2=1}{\mathrm{argmax}} \frac{\boldsymbol{\mu}^\top \mathbf{a}\mathbf{a}^\top \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}.$$

In the final equality, we recognize the general Rayleigh quotient. Let $\boldsymbol{\nu} = \mathbf{M}^{1/2}\boldsymbol{\mu}$ and $\boldsymbol{\nu}^\star = \mathbf{M}^{1/2}\boldsymbol{\mu}^\star$, then

$$\boldsymbol{\nu}^\star = \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\mathrm{argmax}} \frac{\boldsymbol{\nu}^\top \big[\mathbf{M}^{-1/2}\mathbf{a}\mathbf{a}^\top \mathbf{M}^{-1/2}\big]\boldsymbol{\nu}}{\boldsymbol{\nu}^\top \boldsymbol{\nu}}.$$

Hence, the solution is

$$\boldsymbol{\nu}^\star = \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\mathrm{argmax}} \frac{\big[\boldsymbol{\nu}^\top \mathbf{M}^{-1/2}\mathbf{a}\big]^2}{\|\boldsymbol{\nu}\|_2^2} = \underset{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1}{\mathrm{argmax}} \bigg[\Big[\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|}\Big]^\top \mathbf{M}^{-1/2}\mathbf{a}\bigg]^2.$$

Thus, $\boldsymbol{\nu}^\star \in \mathrm{Vec}(\mathbf{M}^{-1/2}\mathbf{a})$ with $\|\mathbf{M}^{-1/2}\boldsymbol{\nu}^\star\|_2 = 1$. This yields immediately $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$, which verifies $\boldsymbol{\mu}^{\star\top}\mathbf{a} = \mathbf{a}^\top \mathbf{M}^{-1}\mathbf{a}/\|\mathbf{M}^{-1}\mathbf{a}\| \geq 0$ since $\mathbf{M}$ and $\mathbf{M}^{-1}$ are PSD. ■

3.2.2 CONVEX COMBINATION (alignf)

In view of the proof of Proposition 8, the alignment maximization problem with the set $\mathcal{M}' = \{\|\boldsymbol{\mu}\|_2 = 1 \wedge \boldsymbol{\mu} \geq \mathbf{0}\}$ can be written as

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu} \in \mathcal{M}'}{\mathrm{argmax}} \frac{\boldsymbol{\mu}^\top \mathbf{a}\mathbf{a}^\top \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}. \tag{6}$$

The following proposition shows that the problem can be reduced to solving a simple QP.

Proposition 9 *Let $\mathbf{v}^\star$ be the solution of the following QP:*

$$\min_{\mathbf{v}\geq\mathbf{0}} \mathbf{v}^\top\mathbf{M}\mathbf{v} - 2\mathbf{v}^\top\mathbf{a}. \tag{7}$$

Then, the solution $\boldsymbol{\mu}^*$ of the alignment maximization problem (6) is given by $\boldsymbol{\mu}^\star = \mathbf{v}^\star/\|\mathbf{v}^\star\|$.

Proof Note that problem (7) is equivalent to the following one defined over $\boldsymbol{\mu}$ and b

$$\min_{\substack{\boldsymbol{\mu}\geq\mathbf{0},\|\boldsymbol{\mu}\|_2=1\\ b>0}} b^2\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu} - 2b\boldsymbol{\mu}^\top\mathbf{a}, \tag{8}$$

where the relation $\mathbf{v} = b\boldsymbol{\mu}$ can be used to retrieve $\mathbf{v}$. The optimal choice of b as a function of $\boldsymbol{\mu}$ can be found by setting the gradient of the objective function with respect to b to zero, giving the closed-form solution $b^* = \frac{\boldsymbol{\mu}^\top\mathbf{a}}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}$. Plugging this back into (8) results in the following optimization after straightforward simplifications:

$$\min_{\boldsymbol{\mu}\geq\mathbf{0},\|\boldsymbol{\mu}\|_2=1} -\frac{(\boldsymbol{\mu}^\top\mathbf{a})^2}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}},$$

which is equivalent to (6). This shows that $\mathbf{v}^\star = b^*\boldsymbol{\mu}^*$ where $\boldsymbol{\mu}^*$ is the solution of (6) and concludes the proof. ■

It is not hard to see that this problem is equivalent to solving a hard margin SVM problem, thus, any SVM solver can also be used to solve it. A similar problem with the non-centered definition of alignment is treated by Kandola et al. (2002b), but their optimization solution differs from ours and requires cross-validation.

Also, note that solving this QP problem does not require a matrix inversion of $\mathbf{M}$. In fact, the assumption about the invertibility of matrix $\mathbf{M}$ is not necessary and a maximal alignment solution can be computed using the same optimization as that of Proposition 9 in the non-invertible case. The optimization problem is then not strictly convex however and the alignment solution $\boldsymbol{\mu}$ not unique.

We now further analyze the properties of the solution $\mathbf{v}$ of problem (7). Let $\widehat{\rho}_0(\boldsymbol{\mu})$ denote the partially normalized alignment maximized by (5):

$$\widehat{\rho}_0(\boldsymbol{\mu}) = \|\mathbf{y}\mathbf{y}^\top\|_F^2\,\widehat{\rho}(\boldsymbol{\mu}) = \frac{\langle\mathbf{K}_{\boldsymbol{\mu}_c},\mathbf{y}\mathbf{y}^\top\rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F} = \frac{\boldsymbol{\mu}^\top\mathbf{a}}{\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}} = \frac{\langle\boldsymbol{\mu},\mathbf{M}^{-1}\mathbf{a}\rangle_\mathbf{M}}{\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}} = \frac{\langle\boldsymbol{\mu},\mathbf{M}^{-1}\mathbf{a}\rangle_\mathbf{M}}{\|\boldsymbol{\mu}\|_\mathbf{M}}.$$

The following proposition gives a simple expression for $\widehat{\rho}_0(\boldsymbol{\mu})$.

Proposition 10 *For $\boldsymbol{\mu} = \mathbf{v}/\|\mathbf{v}\|$, with $\mathbf{v}\neq 0$ solution of the alignment maximization problem* (7)*, the following identity holds:*

$$\widehat{\rho}_0(\boldsymbol{\mu}) = \|\mathbf{v}\|_\mathbf{M}.$$

Proof Since $\|\mathbf{v}\|_\mathbf{M}^2 - 2\mathbf{v}^\top\mathbf{a} = \|\mathbf{v}\|_\mathbf{M}^2 - 2\langle\mathbf{v},\mathbf{M}^{-1}\mathbf{a}\rangle_\mathbf{M} = \|\mathbf{v}-\mathbf{M}^{-1}\mathbf{a}\|_\mathbf{M}^2 - \|\mathbf{M}^{-1}\mathbf{a}\|_\mathbf{M}^2$ the optimization problem (7) can be equivalently written as

$$\min_{\mathbf{v}\geq 0} \|\mathbf{v}-\mathbf{M}^{-1}\mathbf{a}\|_\mathbf{M}^2.$$

This implies that the solution $\mathbf{v}$ is the $\mathbf{M}$-orthogonal projection of $\mathbf{M}^{-1}\mathbf{a}$ over the convex set $\{\mathbf{v}\colon \mathbf{v} \geq 0\}$. Therefore, $\mathbf{v} - \mathbf{M}^{-1}\mathbf{a}$ is $\mathbf{M}$-orthogonal to $\mathbf{v}$:

$$\langle \mathbf{v}, \mathbf{v} - \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}} = 0 \implies \|\mathbf{v}\|_{\mathbf{M}}^2 = \langle \mathbf{v}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}.$$

Thus,

$$\|\mathbf{v}\|_{\mathbf{M}} = \frac{\langle \mathbf{v}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}}{\|\mathbf{v}\|_{\mathbf{M}}} = \frac{\langle \boldsymbol{\mu}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}}{\|\boldsymbol{\mu}\|_{\mathbf{M}}} = \rho(\boldsymbol{\mu}),$$

which concludes the proof. ■

Thus, the proposition gives a straightforward way of computing $\rho_0(\boldsymbol{\mu})$, thereby also $\rho(\boldsymbol{\mu})$, from the $\mathbf{M}$-norm of the solution vector $\mathbf{v}$ that $\boldsymbol{\mu}$ is derived from.

3.3 Relationship with Ensemble Techniques

An alternative two-stage technique for learning with multiple kernels consists of first learning a prediction hypothesis h_k using each kernel K_k, $k \in [1, p]$, and then of learning the best linear combination of these hypotheses: $h = \sum_{k=1}^{p} \mu_k h_k$. But, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms such as that of Lanckriet et al. (2004). For ensemble techniques, each hypothesis h_k, $k \in [1, p]$, is of the form $h_k = \sum_{i=1}^{m} \alpha_{ik} K_k(x_i, \cdot)$ for some $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{mk})^\top \in \mathbb{R}^m$ with different constraints $\|\boldsymbol{\alpha}_k\| \leq \Lambda_k$, $\Lambda_k \geq 0$, and the final hypothesis is of the form

$$\sum_{k=1}^{p} \mu_k h_k = \sum_{k=1}^{p} \mu_k \sum_{j=1}^{m} \alpha_{ik} K_k(x_i, \cdot) = \sum_{i=1}^{m} \sum_{k=1}^{p} \mu_k \alpha_{ik} K_k(x_i, \cdot).$$

In contrast, the general form of the hypothesis learned using kernel learning algorithms is

$$\sum_{i=1}^{m} \alpha_i K_{\boldsymbol{\mu}}(x_i, \cdot) = \sum_{i=1}^{m} \alpha_i \sum_{k=1}^{p} \mu_k K_k(x_i, \cdot) = \sum_{k=1}^{p} \sum_{i=1}^{m} \mu_k \alpha_i K_k(x_i, \cdot),$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^m$ with $\|\boldsymbol{\alpha}\| \leq \Lambda$, $\Lambda \geq 0$. When the coefficients α_{ik} can be decoupled, that is $\alpha_{ik} = \alpha_i \beta_k$ for some β_ks, the two solutions seem to have the same form but they are in fact different since in general the coefficients must obey different constraints (different Λ_ks). Furthermore, the combination weights μ_i are not required to be positive in the ensemble case. We present a more detailed theoretical and empirical comparison of the ensemble and learning kernel techniques elsewhere (Cortes et al., 2011a).

3.4 Single-stage Alignment-based Algorithm

This section analyzes an optimization based on the notion of centered alignment, which can be viewed as the single-stage counterpart of the two-stage algorithm discussed in Sections 3.1 - 3.2.

As in Sections 3.1 and 3.2, we denote by $\mathbf{a}$ the vector $(\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \ldots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top \rangle_F)^\top$ and let $\mathbf{M} \in \mathbb{R}^{p \times p}$ be the matrix defined by $\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F$. The optimization is then defined by augmenting standard single-stage learning kernel optimizations with an alignment maximization constraint. Thus, the domain $\mathcal{M}$ of the kernel combination vector $\boldsymbol{\mu}$ is defined by:

$$\mathcal{M} = \{\boldsymbol{\mu}\colon \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \rho(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top) \geq \Omega\},$$

for non-negative parameters Λ and Ω. The alignment constraint $\rho(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top) \geq \Omega$ can be rewritten as $\Omega\sqrt{\boldsymbol{\mu}^\top \mathbf{M}\boldsymbol{\mu}} - \boldsymbol{\mu}^\top \mathbf{a} \leq 0$, which defines a convex region. Thus, $\mathcal{M}$ is a convex subset of $\mathbb{R}^p$.

For a fixed $\boldsymbol{\mu} \in \mathcal{M}$ and corresponding kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$, let $F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ denote the objective function of the dual optimization problem minimize$_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ solved by an algorithm such as SVM, KRR, or more generally any other algorithm for which $\mathcal{A}$ is a convex set and $F(\boldsymbol{\mu}, \cdot)$ a concave function for all $\boldsymbol{\mu} \in \mathcal{M}$, and $F(\cdot, \boldsymbol{\alpha})$ convex for all $\boldsymbol{\alpha} \in \mathcal{A}$. Then, the general form of a single-stage alignment-based learning kernel optimization is

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}).$$

Note that, by the convex-concave properties of F and the convexity of $\mathcal{M}$ and $\mathcal{A}$, von Neumann's minimax theorem applies:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}).$$

We now further examine this optimization problem in the specific case of the kernel ridge regression algorithm. In the case of KRR, $F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -\boldsymbol{\alpha}^\top(\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}$. Thus, the max-min problem can be rewritten as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} -\boldsymbol{\alpha}^\top(\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}.$$

Let $\mathbf{b}_{\boldsymbol{\alpha}}$ denote the vector $(\boldsymbol{\alpha}^\top \mathbf{K}_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top \mathbf{K}_p \boldsymbol{\alpha})^\top$, then the problem can be rewritten as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} - \max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \mathbf{b}_{\boldsymbol{\alpha}},$$

where $\lambda = \lambda_0 m$ in the notation of Equation (10). We first focus on analyzing only the term $-\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top \mathbf{b}_{\boldsymbol{\alpha}}$. Since the last constraint in $\mathcal{M}$ is convex, standard Lagrange multiplier theory guarantees that for any Ω there exists a $\gamma \geq 0$ such that the following optimization is equivalent to the original maximization over $\boldsymbol{\mu}$.

$$\begin{aligned} \min_{\boldsymbol{\mu}} \; & -\boldsymbol{\mu}^\top \mathbf{b}_{\boldsymbol{\alpha}} + \gamma(\Omega\sqrt{\boldsymbol{\mu}^\top \mathbf{M}\boldsymbol{\mu}} - \boldsymbol{\mu}^\top \mathbf{a}) \\ \text{subject to } & \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \gamma \geq 0. \end{aligned}$$

Note that γ is not a variable, but rather a parameter that will be hand-tuned. Now, again applying standard Lagrange multiplier theory we have that for any $(\gamma\Omega) \geq 0$ there exists an Ω' such that the following optimization is equivalent:

$$\begin{aligned} \min \; & -\boldsymbol{\mu}^\top(\gamma\mathbf{a} + \mathbf{b}_{\boldsymbol{\alpha}}) \\ \text{subject to } & \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \gamma \geq \mathbf{0} \wedge \boldsymbol{\mu}^\top \mathbf{M}\boldsymbol{\mu} \leq \Omega'^2. \end{aligned}$$

Applying the Lagrange technique a final time (for any Λ there exists a $\gamma' \geq 0$ and for any Ω'^2 there exists a $\gamma'' \geq 0$) leads to

$$\begin{aligned} \min \; & -\boldsymbol{\mu}^\top(\gamma\mathbf{a} + \mathbf{b}_{\boldsymbol{\alpha}}) + \gamma'\boldsymbol{\mu}^\top\boldsymbol{\mu} + \gamma''\boldsymbol{\mu}^\top \mathbf{M}\boldsymbol{\mu} \\ \text{subject to } & \boldsymbol{\mu} \geq \mathbf{0} \wedge \gamma, \gamma', \gamma'' \geq 0. \end{aligned}$$

This is a simple QP problem. Note that the overall problem can now be written as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A},\boldsymbol{\mu}\geq\mathbf{0}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}+2\boldsymbol{\alpha}^\top\mathbf{y}+\boldsymbol{\mu}^\top(\gamma\mathbf{a}+\mathbf{b}_{\boldsymbol{\alpha}})-\gamma'\boldsymbol{\mu}^\top\boldsymbol{\mu}-\gamma''\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}.$$

This last problem is not convex in $(\boldsymbol{\alpha},\boldsymbol{\mu})$, but the problem is convex in each variable. In the case of kernel ridge regression, the maximization in $\boldsymbol{\alpha}$ admits a closed form solution. Plugging in that solution yields the following convex optimization problem in $\boldsymbol{\mu}$:

$$\min_{\boldsymbol{\mu}\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\lambda\mathbf{I})^{-1}\mathbf{y}-\gamma\boldsymbol{\mu}^\top\mathbf{a}+\boldsymbol{\mu}^\top(\gamma''\mathbf{M}+\gamma'\mathbf{I})\boldsymbol{\mu}.$$

Note that multiplying the objective by λ using the substitution $\boldsymbol{\mu}'=\frac{1}{\lambda}\boldsymbol{\mu}$ results in the following equivalent problem,

$$\min_{\boldsymbol{\mu}'\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}'}+\mathbf{I})^{-1}\mathbf{y}-\lambda^2\gamma\boldsymbol{\mu}'^\top\mathbf{a}+\boldsymbol{\mu}'^\top(\lambda^3\gamma''\mathbf{M}+\lambda^3\gamma'\mathbf{I})\boldsymbol{\mu}',$$

which makes clear that the trade-off parameter λ can be subsumed by the γ,γ' and γ'' parameters. This leads to the following simpler problem with a reduced number of trade-off parameters,

$$\min_{\boldsymbol{\mu}\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\mathbf{I})^{-1}\mathbf{y}-\gamma\boldsymbol{\mu}^\top\mathbf{a}+\boldsymbol{\mu}^\top(\gamma''\mathbf{M}+\gamma'\mathbf{I})\boldsymbol{\mu}. \tag{9}$$

This is a convex optimization problem. In particular, $\boldsymbol{\mu}\mapsto\mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\mathbf{I})^{-1}\mathbf{y}$ is a convex funtion by convexity of $f\colon\mathbf{M}\mapsto\mathbf{y}^\top\mathbf{M}^{-1}\mathbf{y}$ over the set of positive definite symmetric matrices. The convexity of f can be seen from that of its epigraph, which, by the property of the Schur complement, can be written as follows (Boyd and Vandenberghe, 2004):

$$\operatorname{epi} f=\{(\mathbf{M},t)\colon \mathbf{M}\succ\mathbf{0},\mathbf{y}^\top\mathbf{M}^{-1}\mathbf{y}\leq t\}=\{(\mathbf{M},t)\colon \begin{pmatrix}\mathbf{M} & \mathbf{y}\\ \mathbf{y}^\top & t\end{pmatrix}\succeq\mathbf{0},\mathbf{M}\succ\mathbf{0}\}.$$

This defines a linear matrix inequality in $(\mathbf{M},t)$ and thus a convex set. The convex optimization (9) can be solved efficiently using a simple iterative algorithm as in Cortes et al. (2009a). In practice, the algorithm converges within 10-50 iterations.

4. Theoretical Results

This section presents a series of theoretical guarantees related to the notion of kernel alignment. Section 4.1 proves a concentration bound of the form $|\rho-\widehat{\rho}|\leq O(1/\sqrt{m})$, which relates the centered alignment ρ to its empirical estimate $\widehat{\rho}$. In Section 4.2, we prove the existence of accurate predictors in both classification and regression in the presence of a kernel K with good alignment with respect to the target kernel. Section 4.3 presents stability-based generalization bounds for the two-stage alignment maximization algorithm whose first stage was described in Section 3.2.2.

4.1 Concentration Bounds for Centered Alignment

Our concentration bound differs from that of Cristianini et al. (2001) both because our definition of alignment is different and because we give a bound directly on the quantity of interest $|\rho-\widehat{\rho}|$. Instead, Cristianini et al. (2001) give a bound on $|A'-\widehat{A}|$, where $A'\neq A$ can be related to A by replacing each Frobenius product with its expectation over samples of size m.

The following proposition gives a bound on the essential quantities appearing in the definition of the alignments.

Proposition 11 *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *drawn according to* D*. Assume that for any* $x \in \mathcal{X}$*,* $K(x,x) \leq R^2$ *and* $K'(x,x) \leq R'^2$*. Then, for any* $\delta > 0$*, with probability at least* $1-\delta$*, the following inequality holds:*

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \frac{18R^2R'^2}{m} + 24R^2R'^2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Note that in the case $K'(x_i,x_j) = y_i y_j$, we then have $R'^2 \leq \max_i y_i^2$.

Proof The proof relies on a series of lemmas given in the Appendix. By the triangle inequality and in view of Lemma 19, the following holds:

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right]\right| + \frac{18R^2R'^2}{m}.$$

Now, in view of Lemma 18, the application of McDiarmid's inequality (McDiarmid, 1989) to $\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}$ gives for any $\varepsilon > 0$:

$$\Pr\left[\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right]\right| > \varepsilon\right] \leq 2\exp[-2m\varepsilon^2/(24R^2R'^2)^2].$$

Setting δ to be equal to the right-hand side yields the statement of the proposition. ■

Theorem 12 *Under the assumptions of Proposition 11, and further assuming that the conditions of the Definitions 2-4 are satisfied for* $\rho(K,K')$ *and* $\widehat{\rho}(\mathbf{K},\mathbf{K}')$*, for any* $\delta > 0$*, with probability at least* $1-\delta$*, the following inequality holds:*

$$|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| \leq 18\beta\left[\frac{3}{m} + 8\sqrt{\frac{\log\frac{6}{\delta}}{2m}}\right],$$

with $\beta = \max(R^2R'^2/\mathrm{E}[K_c^2], R^2R'^2/\mathrm{E}[K_c'^2])$.

Proof To shorten the presentation, we first simplify the notation for the alignments as follows:

$$\rho(K,K') = \frac{b}{\sqrt{aa'}} \qquad \widehat{\rho}(\mathbf{K},\mathbf{K}') = \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}},$$

with $b = \mathrm{E}[K_c K'_c]$, $a = \mathrm{E}[K_c^2]$, $a' = \mathrm{E}[K_c'^2]$ and similarly, $\widehat{b} = (1/m^2)\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F$, $\widehat{a} = (1/m^2)\|\mathbf{K}_c\|^2$, and $\widehat{a}' = (1/m^2)\|\mathbf{K}'_c\|^2$. By Proposition 11 and the union bound, for any $\delta > 0$, with probability at least $1-\delta$, all three differences $a - \widehat{a}$, $a' - \widehat{a}'$, and $b - \widehat{b}$ are bounded by $\alpha = \frac{18R^2R'^2}{m} + 24R^2R'^2\sqrt{\frac{\log\frac{6}{\delta}}{2m}}$. Using the definitions of ρ and $\widehat{\rho}$, we can write:

$$\begin{aligned}
|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &= \left|\frac{b}{\sqrt{aa'}} - \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}}\right| = \left|\frac{b\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}\sqrt{aa'}}{\sqrt{aa'\widehat{a}\widehat{a}'}}\right| \\
&= \left|\frac{(b-\widehat{b})\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}(\sqrt{aa'} - \sqrt{\widehat{a}\widehat{a}'})}{\sqrt{aa'\widehat{a}\widehat{a}'}}\right| \\
&= \left|\frac{(b-\widehat{b})}{\sqrt{aa'}} - \widehat{\rho}(\mathbf{K},\mathbf{K}')\frac{aa' - \widehat{a}\widehat{a}'}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})}\right|.
\end{aligned}$$

Since $\widehat{\rho}(\mathbf{K},\mathbf{K}') \in [0,1]$, it follows that

$$|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| \leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|aa' - \widehat{a}\widehat{a}'|}{\sqrt{aa'}(\sqrt{aa'}+\sqrt{\widehat{a}\widehat{a}'})}.$$

Assume first that $\widehat{a} \leq \widehat{a}'$. Rewriting the right-hand side to make the differences $a - \widehat{a}$ and $a' - \widehat{a}'$ appear, we obtain:

$$\begin{aligned}
|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &\leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|(a-\widehat{a})a' + \widehat{a}(a'-\widehat{a}')|}{\sqrt{aa'}(\sqrt{aa'}+\sqrt{\widehat{a}\widehat{a}'})} \\
&\leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a'+\widehat{a}}{\sqrt{aa'}+\sqrt{\widehat{a}\widehat{a}'}}\right] \leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a'}{\sqrt{aa'}} + \frac{\widehat{a}}{\sqrt{\widehat{a}\widehat{a}'}}\right] \\
&\leq \frac{\alpha}{\sqrt{aa'}}\left[2 + \sqrt{\frac{a'}{a}}\right] = \left[\frac{2}{\sqrt{aa'}} + \frac{1}{a}\right]\alpha.
\end{aligned}$$

We can similarly obtain $\left[\frac{2}{\sqrt{aa'}} + \frac{1}{a'}\right]\alpha$ when $\widehat{a}' \leq \widehat{a}$. Both bounds are less than or equal to $3\max(\frac{\alpha}{a}, \frac{\alpha}{a'})$. ■

Equivalently, one can set the right hand side of the high probability statement presented in Theorem 12 equal to ϵ and solve for δ, which shows that $\Pr\left[|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| > \epsilon\right] \leq O(e^{-m\epsilon^2})$.

4.2 Existence of Good Alignment-based Predictors

For classification and regression tasks, the target kernel is based on the labels and defined by $K_Y(x,x') = yy'$, where we denote by y the label of point x and y' that of x'. This section shows the existence of predictors with high accuracy both for classification and regression when the alignment $\rho(K,K_Y)$ between the kernel K and K_Y is high.

In the regression setting, we shall assume that the labels have been normalized such that $\mathrm{E}[y^2] = 1$. In classification, $y = \pm 1$ and thus $\mathrm{E}[y^2] = 1$ without any normalization. Denote by h^* the hypothesis defined for all $x \in \mathcal{X}$ by

$$h^*(x) = \frac{\mathrm{E}_{x'}[y' K_c(x,x')]}{\sqrt{\mathrm{E}[K_c^2]}}.$$

Observe that by definition of h^*, $\mathrm{E}_x[yh^*(x)] = \rho(K,K_Y)$. For any $x \in \mathcal{X}$, define $\gamma(x) = \sqrt{\frac{\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}_{x,x'}[K_c^2(x,x')]}}$ and $\Gamma = \max_x \gamma(x)$. The following result shows that the hypothesis h^* has high accuracy when the kernel alignment is high and Γ not too large.[3]

Theorem 13 (classification) *Let* $R(h^*) = \Pr[yh^*(x) < 0]$ *denote the error of* h^* *in binary classification. For any kernel* K *such that* $0 < \mathrm{E}[K_c^2] < +\infty$*, the following holds:*

$$R(h^*) \leq 1 - \rho(K,K_Y)/\Gamma.$$

[3] A version of this result was presented by Cristianini, Shawe-Taylor, Elisseeff, and Kandola (2001) and Cristianini, Kandola, Elisseeff, and Shawe-Taylor (2002) for the so-called Parzen window solution and non-centered kernels. However, both proofs are incorrect since they rely implicitly on the fact that $\max_x \left[\frac{\mathrm{E}_{x'}[K^2(x,x')]}{\mathrm{E}_{x,x'}[K^2(x,x')]}\right]^{\frac{1}{2}} = 1$, which can only hold in the trivial case where the kernel function K^2 is a constant: by definition of the maximum and expectation operators, $\max_x \left[\mathrm{E}_{x'}[K^2(x,x')]\right] \geq \mathrm{E}_x\left[\mathrm{E}_{x'}[K^2(x,x')]\right]$, with equality only in the constant case.

Proof Note that for all $x \in \mathcal{X}$,

$$|yh^*(x)| = \frac{|y\,\mathrm{E}_{x'}[y'K_c(x,x')]|}{\sqrt{\mathrm{E}[K_c^2]}} \le \frac{\sqrt{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\mathrm{E}[K_c^2]}} = \frac{\sqrt{\mathrm{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\mathrm{E}[K_c^2]}} \le \Gamma.$$

In view of this inequality, and the fact that $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$, we can write:

$$\begin{aligned} 1 - R(h^*) &= \Pr[yh^*(x) \ge 0] \\ &= \mathrm{E}[\mathbf{1}_{\{yh^*(x)\ge 0\}}] \\ &\ge \mathrm{E}\left[\frac{yh^*(x)}{\Gamma}\mathbf{1}_{\{yh^*(x)\ge 0\}}\right] \\ &\ge \mathrm{E}\left[\frac{yh^*(x)}{\Gamma}\right] = \rho(K, K_Y)/\Gamma, \end{aligned}$$

where $\mathbf{1}_\omega$ is the indicator function of the event ω. ■

A probabilistic version of the theorem can be straightforwardly derived by noting that by Markov's inequality, for any $\delta > 0$, with probability at least $1-\delta$, $|\gamma(x)| \le 1/\sqrt{\delta}$.

Theorem 14 (regression) *Let $R(h^*) = \mathrm{E}_x[(y - h^*(x))^2]$ denote the error of h^* in regression. For any kernel K such that $0 < \mathrm{E}[K_c^2] < +\infty$, the following holds:*

$$R(h^*) \le 2(1 - \rho(K, K_Y)).$$

Proof By the Cauchy-Schwarz inequality, it follows that:

$$\begin{aligned} \mathop{\mathrm{E}}_x[h^{*2}(x)] &= \mathop{\mathrm{E}}_x\left[\frac{\mathrm{E}_{x'}[y'K_c(x,x')]^2}{\mathrm{E}[K_c^2]}\right] \\ &\le \mathop{\mathrm{E}}_x\left[\frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]}\right] \\ &= \frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x,x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]} = \mathop{\mathrm{E}}_{x'}[y'^2] = 1. \end{aligned}$$

Using again the fact that $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$, the error of h^* can be bounded as follows:

$$\mathrm{E}[(y - h^*(x))^2] = \mathop{\mathrm{E}}_x[h^*(x)^2] + \mathop{\mathrm{E}}_x[y^2] - 2\mathop{\mathrm{E}}_x[yh^*(x)] \le 1 + 1 - 2\rho(K, K_Y),$$

which concludes the proof. ■

The hypothesis h^* is closely related to the hypothesis h_S^* derived as follows from a finite sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$:

$$h_S(x) = \frac{\frac{1}{m}\sum_{i=1}^m y_i K_c(x, x_i)}{\sqrt{\frac{1}{m^2}\sum_{i,j=1}^m K_c(x_i, x_j)^2}\sqrt{\frac{1}{m^2}\sum_{i,j=1}^m (y_i y_j)^2}}.$$

Note in particular that $\widehat{\mathrm{E}}_x[yh_S(x)] = \widehat{\rho}(\mathbf{K}, \mathbf{K_Y})$, where we denote by $\widehat{\mathrm{E}}$ the expectation based on the empirical distribution. Using this and other results of this section, it is not hard to show that with high probability $|R(h^*) - R(h_S^*)| \le O(1/\sqrt{m})$ both in the classification and regression settings.

For classification, the existence of a good predictor g^* based on the unnormalized alignment ρ_u (see Definition 5) can also be shown. The corresponding guarantees are simpler and do not depend on a term such as Γ. However, unlike the normalized case, the loss of the predictor g^*_S derived from a finite sample may not always be close to that of g^*. Note that in classification, for any label y, $|y| = 1$, thus, by Lemma 6, the following holds: $0 \leq \rho_{\mathrm{u}}(K, K_Y)| \leq R^2$. Let g^* be the hypothesis defined by:

$$g^*(x) = \mathop{\mathrm{E}}_{x'}[y' K_c(x, x')].$$

Since $0 \leq \rho_{\mathrm{u}}(K, K_Y)| \leq R^2$, the following theorem provides strong guarantees for g^* when the unnormalized alignment a is sufficiently large, that is close to R^2.

Theorem 15 (classification) *Let $R(g^*) = \Pr[yg^*(x) < 0]$ denote the error of g^* in binary classification. For any kernel K such that $\sup_{x \in \mathcal{X}} K_c(x, x) \leq R^2$, we have:*

$$R(g^*) \leq 1 - \rho_{\mathrm{u}}(K, K_Y)/R^2.$$

Proof Note that for all $x \in \mathcal{X}$,

$$|yg^*(x)| = |g^*(x)| = |\mathop{\mathrm{E}}_{x'}[y' K_c(x, x')]| \leq R^2.$$

Using this inequality, and the fact that $\mathrm{E}_x[yg^*(x)] = \rho_{\mathrm{u}}(K, K_Y)$, we can write:

$$\begin{aligned} 1 - R(g^*) = \Pr[yg^*(x) \geq 0] &= \mathrm{E}[\mathbf{1}_{\{yg^*(x) \geq 0\}}] \\ &\geq \mathrm{E}\left[\frac{yg^*(x)}{R^2} \mathbf{1}_{\{yh^*(x) \geq 0\}}\right] \\ &\geq \mathrm{E}\left[\frac{yg^*(x)}{R^2}\right] = \rho_{\mathrm{u}}(K, K_Y)/R^2, \end{aligned}$$

which concludes the proof. ■

4.3 Generalization Bounds for Two-stage Learning Kernel Algorithms

This section presents stability-based generalization bounds for two-stage learning kernel algorithms. The proof of a stability-based learning bound hinges on showing that the learning algorithm is *stable*, that is the pointwise loss of a learned hypothesis does not change drastically if the training sample changes only slightly. We refer the reader to Bousquet and Elisseeff (2000) for a full introduction.

We present learning bounds for the case where the second stage of the algorithm is kernel ridge regression (KRR). Similar results can be given in classification using algorithms such as SVMs in the second stage. Thus, in the first stage, the algorithms we examine select a combination weight parameter $\boldsymbol{\mu} \in \mathcal{M}_q = \{\boldsymbol{\mu} \colon \boldsymbol{\mu} \geq \mathbf{0}, \|\boldsymbol{\mu}\|_q^q = \Lambda_q\}$ which defines a kernel $K_{\boldsymbol{\mu}}$, and in the second stage use KRR to select a hypothesis from the RKHS associated to $K_{\boldsymbol{\mu}}$. While several of our results hold in general, we will be more specifically interested in the alignment maximization algorithm presented in Section 3.2.2.

Recall that for a fixed kernel function $K_{\boldsymbol{\mu}}$ with associated RKHS $\mathbb{H}_{K_{\boldsymbol{\mu}}}$ and training set $S = ((x_1, y_1), \ldots, (x_m, y_m))$, the KRR optimization problem is defined by the following constraint optimization problem:

$$\min_{h \in \mathbb{H}_{K_{\boldsymbol{\mu}}}} G(h) = \lambda_0 \|h\|^2_{K_{\boldsymbol{\mu}}} + \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)^2. \tag{10}$$

We first analyze the stability of two-stage algorithms and then use that to derive a stability-based generalization bound (Bousquet and Elisseeff, 2000). More precisely, we examine the pointwise difference in hypothesis values obtained on any point x when the algorithm has been trained on two data sets S and S' of size m that differ in exactly one point.

In what follows, we denote by $\|\mathbf{K}\|_{s,t} = (\sum_{k=1}^{p} \|\mathbf{K}_k\|_s^t)^{1/t}$ the (s,t)-norm of a collection of matrices and by $\Delta\boldsymbol{\mu}$ the difference $\boldsymbol{\mu}' - \boldsymbol{\mu}$ of the combination vector $\boldsymbol{\mu}'$ and $\boldsymbol{\mu}$ returned by the first stage of the algorithm by training on S, respectively S'.

Theorem 16 (Stability of two-stage learning kernel algorithm) *Let S and S' be two samples of size m that differ in exactly one point and let h and h' be the associated hypotheses generated by a two-stage KRR learning kernel algorithm with the constraint $\boldsymbol{\mu} \in \mathcal{M}_1$. Then, for any $s, t \geq 1$ with $\frac{1}{s} + \frac{1}{r} = 1$ and any $x \in \mathcal{X}$:*

$$|h'(x) - h(x)| \leq \frac{2\Lambda_1 R^2 M}{\lambda_0 m} \Big[1 + \frac{\|\Delta\boldsymbol{\mu}\|_s \|\mathbf{K}_c\|_{2,t}}{2\lambda_0} \Big],$$

where M is an upper bound on the target labels and $R^2 = \sup_{\substack{k \in [1,p] \\ x \in \mathcal{X}}} K_k(x,x)$.

Proof The KRR algorithm returns the hypothesis $h(x) = \sum_{i=1}^{m} \alpha_i K_{\boldsymbol{\mu}}(x_i, x)$, where $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}} + m\lambda_0 \mathbf{I})^{-1}\mathbf{y}$. Thus, this hypothesis is parametrized by the kernel weight vector $\boldsymbol{\mu}$, which defines the kernel function, and the sample S, which is used to populate the kernel matrix, and will be explicitly denoted $h_{\boldsymbol{\mu},S}$. To estimate the stability of the overall two-stage algorithm, $\Delta h_{\boldsymbol{\mu},S} = h_{\boldsymbol{\mu}',S'} - h_{\boldsymbol{\mu},S}$, we use the decomposition

$$\Delta h_{\boldsymbol{\mu},S} = (h_{\boldsymbol{\mu}',S'} - h_{\boldsymbol{\mu}',S}) + (h_{\boldsymbol{\mu}',S} - h_{\boldsymbol{\mu},S})$$

and bound each parenthesized term separately. The first parenthesized term measures the pointwise stability of KRR due to a change of a single training point with a fixed kernel. This can be bounded using Theorem 2 of Cortes et al. (2009a). Since, for all $x \in \mathcal{X}$, $K_{\boldsymbol{\mu}}(x,x) = \sum_{k=1}^{p} \mu_k K_k(x,x) \leq R^2 \sum_{k=1}^{p} \mu_k \leq \Lambda_1 R^2$, using that theorem yields the following bound:

$$\forall x \in \mathcal{X}, \quad |h_{\boldsymbol{\mu},S'}(x) - h_{\boldsymbol{\mu},S}(x)| \leq \frac{2\Lambda_1 R^2 M}{\lambda_0 m}.$$

The second parenthesized term measures the pointwise difference of the hypotheses due to the change of kernel from $\mathbf{K}_{\boldsymbol{\mu}'}$ to $\mathbf{K}_{\boldsymbol{\mu}}$ for a fixed training sample when using KRR. By Proposition 1 of Cortes et al. (2010c), this term can be bounded as follows:

$$\forall x \in \mathcal{X}, |h_{\boldsymbol{\mu}',S}(x) - h_{\boldsymbol{\mu},S}(x)| \leq \frac{\Lambda_1 R^2 M}{\lambda_0^2 m} \|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\|.$$

The term $\|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\|$ can be bounded using Hölder's inequality as follows:

$$\|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\| = \| \sum_{k=1}^{p} (\Delta\mu_k) \mathbf{K}_k \| \leq \sum_{k=1}^{p} |\Delta\mu_k| \, \|\mathbf{K}_k\| \leq \|\Delta\boldsymbol{\mu}\|_s \|\mathbf{K}\|_{2,t},$$

which completes the proof. ■

The pointwise stability result just presented can be used directly to derive a generalization bound for two-stage learning kernel algorithms as in Bousquet and Elisseeff (2000).

For a hypothesis h, we denote by $R(h)$ its generalization error and by $\widehat{R}(h)$ its empirical error on a $S = ((x_1, y_1), \ldots, (x_m, y_m))$:

$$R(h) = \operatorname*{E}_{x,y}[(h_S(x) - y)^2] \quad \widehat{R}(h) = \frac{1}{m}\sum_{i=1}^{m}(h_S(x_i) - y_i)^2.$$

Theorem 17 (Stability-based generalization bound) *Let h_S denote the hypothesis returned by a two-stage KRR kernel learning algorithm with the constraint $\boldsymbol{\mu} \in \mathcal{M}_1$ when trained on sample S. For any $s, t \geq 1$ with $\frac{1}{s} + \frac{1}{r} = 1$, with probability at least $1 - \delta$ over samples S of size m, the following bound holds:*

$$R(h_S) \leq \widehat{R}(h_S) + \frac{2M_1M_2}{m} + \Big(1 + \frac{16M_2}{M_1}\Big)\frac{M_1M_2}{4}\sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

with $M_1 = 2\Big[1 + \sqrt{\frac{\Lambda_1 R^2}{\lambda_0}}\Big]M$ and $M_2 = \frac{2\Lambda_1 R^2}{\lambda_0}\Big[1 + \frac{\|\Delta\boldsymbol{\mu}\|_s \|\mathbf{K}_c\|_{2,t}}{2\lambda_0}\Big]M$.

Proof Since h_S is the minimizer of the objective (10) and since $\mathbf{0}$ belongs to the hypothesis space,

$$G(h_S) \leq G(\mathbf{0}) = \frac{1}{m}\sum_{i=1}^{m}(0 - y_i)^2 \leq M^2.$$

Furthermore, since the mean squared loss is non-negative, we can write: $\lambda_0 \|h_S\|^2_{K_\mu} \leq G(h_S)$. Therefore, $\|h_S\|^2_{K_\mu} \leq \frac{M^2}{\lambda_0}$. By the reproducing property, for any $x \in \mathcal{X}$,

$$\begin{aligned} |h_S(x)| = |\langle h_S, K_{\boldsymbol{\mu}}(x, \cdot)\rangle_{K_{\boldsymbol{\mu}}}| &\leq \|h_S\|_{K_{\boldsymbol{\mu}}}\sqrt{K_{\boldsymbol{\mu}}(x,x)} \\ &= \sqrt{\frac{M}{\lambda_0}}\sqrt{\sum_{k=1}^{p}\mu_k K_k(x,x)} \\ &\leq \sqrt{\frac{M}{\lambda_0}}\sqrt{\|\boldsymbol{\mu}\|_1 R^2} \leq RM\sqrt{\frac{\Lambda_1}{\lambda_0}}. \end{aligned}$$

Thus, for all $x \in \mathcal{X}$ and $y \in [-M, M]$, the squared loss can be bounded as follows

$$|h_S(x) - y| \leq \Big(M + RM\sqrt{\frac{\Lambda_1}{\lambda_0}}\Big) = \frac{M_1}{2}.$$

This implies that the squared loss is M_1-Lipschitz and by Theorem 16 that the algorithm is stable with a uniform stability parameter $\beta \leq \frac{M_1M_2}{m}$ bounded as follows:

$$|(h_{S'}(x) - y)^2 - (h_S(x) - y)^2| \leq M_1|h_{S'}(x) - h_S(x)| \leq \frac{M_1M_2}{m}.$$

The application of Theorem 12 of Bousquet and Elisseeff (2000) with the bound on the loss $\frac{M_1}{2}$ and the uniform stability parameter β directly yields the statement. ■

The inequality just presented holds for all two-stage learning kernel algorithms. To determine its convergence rate, the term $\|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}_c\|_{2,t}$ must be bounded. Let $s=1$ and $t=\infty$, and assume that the base kernels $\mathbf{K}_k$, $k\in[1,p]$, are trace-normalized as in our experiments (Section 3), then a straightforward bound can be given for this term:

$$\|\Delta\boldsymbol{\mu}\|_1\|\mathbf{K}_c\|_{2,\infty}\le(\|\boldsymbol{\mu}'\|_1+\|\boldsymbol{\mu}\|_1)\max_{k\in[1,k]}\|\mathbf{K}_{kc}\|_2\le\max_{k\in[1,k]}2\Lambda_1\operatorname{Tr}[\mathbf{K}_{kc}]\le 2\Lambda_1.$$

Thus, in the statement of Theorem 17, M_2 can be replaced with $\frac{2\Lambda_1R^2}{\lambda_0}\left[1+\frac{\Lambda_1}{\lambda_0}\right]M$ and, for Λ_1 and λ_0 constant, the learning bound converges in $O(1/\sqrt{m})$.

The straightforward upper bound on $\|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}_c\|_{2,t}$ applies to all such two-stage learning kernel algorithms. For a specific algorithm, finer or more favorable bounds could be derived. We have initiated this study in the specific case of the alignment maximization algorithm. The result given in Proposition 21 (Appendix B) can be used to bound $\|\Delta\boldsymbol{\mu}\|_2$ and thus $\|\Delta\boldsymbol{\mu}\|_2\|\mathbf{K}_c\|_{2,2}$.

Note that in the specific case of the alignment maximization algorithm, if $\boldsymbol{\mu}^*$ is the solution obtained for the constraint $\boldsymbol{\mu}\in\mathcal{M}_2$, then it is also the alignment maximizing solution found in the set $\boldsymbol{\mu}\in\mathcal{M}_1$ with $\Lambda_1=\|\boldsymbol{\mu}^*\|_1\le\sqrt{p}\|\boldsymbol{\mu}\|_2\le\sqrt{p}\Lambda_2$. This makes the dependence on p explicit in the case of the constraint $\boldsymbol{\mu}\in\mathcal{M}_2$.

5. Experiments

This section compares the performance of several learning kernel algorithms for classification and regression. We compare the alignment-based two-stage learning kernel algorithms `align` and `alignf`, as well as the single-stage algorithm presented in Section 3 with the following algorithms:

Uniform combination (`unif`*)*: this is the most straightforward method, which consists of choosing equal mixture weights, thus the kernel matrix used is,

$$\mathbf{K}_{\boldsymbol{\mu}}=\frac{\Lambda}{p}\sum_{k=1}^{p}\mathbf{K}_k.$$

Nevertheless, improving upon the performance of this method has been surprisingly difficult for standard (one-stage) learning kernel algorithms (Cortes, 2009; Cortes et al., 2011b).

Norm-1 regularized combination (`l1-svm`): this algorithm optimizes the SVM objective

$$\begin{gathered}\min_{\boldsymbol{\mu}}\max_{\boldsymbol{\alpha}}\ 2\boldsymbol{\alpha}^\top\mathbf{1}-\boldsymbol{\alpha}^\top\mathbf{Y}^\top\mathbf{K}_{\boldsymbol{\mu}}\mathbf{Y}\boldsymbol{\alpha}\\ \text{subject to: }\boldsymbol{\mu}\ge\mathbf{0},\operatorname{Tr}[\mathbf{K}_{\boldsymbol{\mu}}]\le\Lambda,\boldsymbol{\alpha}^\top\mathbf{y}=0,\mathbf{0}\le\boldsymbol{\alpha}\le\mathbf{C},\end{gathered}$$

as described by Lanckriet et al. (2004). Here, $\mathbf{Y}$ is the diagonal matrix constructed from the labels $\mathbf{y}$ and $\mathbf{C}$ is the regularization parameter of the SVM.

Norm-2 regularized combination (`l2-krr`): this algorithm optimizes the kernel ridge regression objective

$$\begin{gathered}\min_{\boldsymbol{\mu}}\max_{\boldsymbol{\alpha}}-\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}-\boldsymbol{\alpha}^\top\mathbf{K}_{\boldsymbol{\mu}}\boldsymbol{\alpha}+2\boldsymbol{\alpha}^\top\mathbf{y}\\ \text{subject to: }\boldsymbol{\mu}\ge\mathbf{0},\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_2\le\Lambda.\end{gathered}$$

	KINEMATICS	IONOSPHERE	GERMAN	SPAMBASE	SPLICE
SIZE	1000	351	1000	1000	1000
γ	-3, 3	-3, 3	-4, 3	-12, -7	-9, -3
`unif`	.138 ± .005 .158 ± .013	.479 ± .033 .246 ± .033	.259 ± .018 .089 ± .008	.187 ± .028 .138 ± .031	.152 ± .022 .122 ± .011
`1-stage`	.137 ± .005 .155 ± .012	.470 ± .032 .251 ± .035	.260 ± .026 .082 ± .003	.209 ± .028 .099 ± .024	.153 ± .025 .105 ± .006
`align`	.125 ± .004 .173 ± .016	.456 ± .036 .261 ± .040	.255 ± .015 .089 ± .008	.186 ± .026 .140 ± .031	.151 ± .024 .123 ± .011
`alignf`	.115 ± .004 .176 ± .017	.444 ± .034 .278 ± .057	.242 ± .015 .093 ± .009	.180 ± .024 .146 ± .028	.139 ± .013 .124 ± .011

REGRESSION CLASSIFICATION

Table 2: Error measures (top) and alignment values (bottom) for `unif`, `1-stage` (`l2-krr` or `l1-svm`), `align` and `alignf` with kernels built from linear combinations of Gaussian base kernels. The choice of γ_0, γ_1 is listed in the row labeled γ and the total size of the data set used is listed under SIZE. The results are shown with ±1 standard deviation measured by 5-fold cross-validation. Further measures of significance are shown in Appendix C, Table 4.

The L_2 regularized method is used for regression since it is shown in Cortes et al. (2009a) to outperform the alternative L_1 regularized method in similar settings. Here, λ is the regularization parameter of KRR and $\boldsymbol{\mu}_0$ is an additional regularization parameter for the kernel selection.

In all experiments, the error measures reported are for 5-fold cross validation, where, in each trial, three folds are used for training, one used for validation, and one for testing. For the two-stage methods, the same training and validation data is used for both stages of the learning. The regularization parameter Λ is chosen via a grid search based on the performance on the validation set, while the regularization parameters $\mathbf{C}$ for `l1-svm` and λ for `l2-krr` are fixed since only the ratios $\mathbf{C}/\Lambda$ and λ/Λ are important. More explicitly, for the KRR algorithm, scaling the vector $\boldsymbol{\mu}$ by Λ results in a scaled dual solution: $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}}\Lambda + \lambda\mathbf{I})^{-1}\mathbf{y} = \Lambda^{-1}(\mathbf{K}_{\boldsymbol{\mu}} + \frac{\lambda}{\Lambda}\mathbf{I})^{-1}\mathbf{y}$. In turn, we see that the primal solution $h(x) = \sum_{i=1}^{m} \Lambda^{-1}\alpha_i \Lambda K_{\boldsymbol{\mu}}(x, x_i) = \sum_{i=1}^{m} \alpha_i K_{\boldsymbol{\mu}}(x, x_i)$ is equivalent to the solution of the KRR algorithm that uses a regularization parameter equal to λ/Λ without scaling $\boldsymbol{\mu}$ and, thus, it suffices to vary only one regularization parameter. In the case of SVMs, the scale of the hypothesis does not change its sign (or the binary prediction) and thus the same property can be shown to hold. The $\boldsymbol{\mu}_0$ parameter is set to zero in our experiments.

5.1 General Kernel Combinations

In the first set of experiments, we consider combinations of Gaussian kernels of the form

$$\mathbf{K}_\gamma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

with varying bandwidth parameter $\gamma \in \{2^{\gamma_0}, 2^{\gamma_0+1}, \ldots, 2^{1-\gamma_1}, 2^{\gamma_1}\}$. The values γ_0 and γ_1 are chosen such that the base kernels are sufficiently different in alignment and performance. Each base kernel is centered and normalized to have trace one. We test the algorithms on several data sets taken from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`) and Delve (`http://www.cs.toronto.edu/~delve/data/datasets.html`).

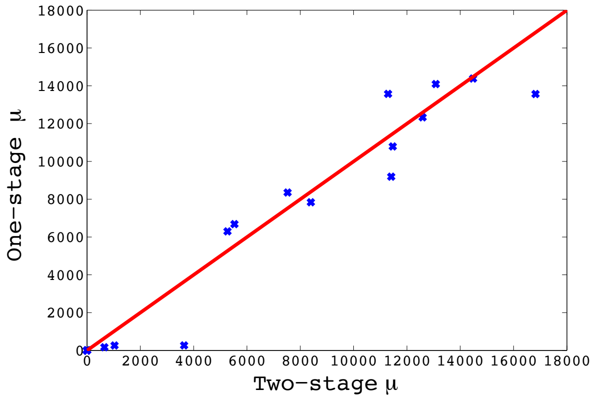


Figure 3: A scatter plot comparison of the different kernel combination weight values obtained by optimally tuned one-stage and two-stage algorithms on the kinematics data set.

Table 2 summarizes our results. For regression, we compare against the `l2-krr` method and report RMSE. For classification, we compare against the `l1-svm` method and report the misclassification percentage. In general, we see that performance and alignment are well correlated. In all data sets, we see improvement over the uniform combination as well as the one-stage kernel learning algorithms. Note that although the `align` method often increases the alignment of the final kernel, as compared to the uniform combination, the `alignf` method gives the best alignment since it directly maximizes this quantity. Nonetheless, `align` provides an inexpensive heuristic that increases the alignment and performance of the final combination kernel.

In our experiments with the one-stage KRR algorithm presented in Section 3.4, there was no significant improvement found over the two-stage `alignf` algorithm with respect to the kinematics and ionosphere data sets. In fact, for optimally cross-validated parameters γ, γ' and γ'' the solution combination weights were found to closely coincide with the `alignf` solution (see Figure 3). This would suggest the use of the two-stage algorithm over the one-stage, since there are fewer parameters to tune and the problem can be solved as a standard QP.

To the best of our knowledge, these are the first kernel combination experiments for alignment with general base kernels. Previous experiments seem to have dealt exclusively with rank-one base kernels built from the eigenvectors of a single kernel matrix (Cristianini et al., 2001). In the next section, we also examine rank-one kernels, although not generated from a spectral decomposition.

5.2 Rank-one Kernel Combinations

In this set of experiments we use the sentiment analysis data set version 1 from Blitzer et al. (2007): *books*, *dvd*, *electronics* and *kitchen*. Each domain has 2,000 examples. In the regression setting, the goal is to predict a rating between 1 and 5, while for classification the goal is to discriminate positive (ratings ≥ 4) from negative reviews (ratings ≤ 2). We use rank-one kernels based on the 4,000 most frequent bigrams. The kth base kernel, $\mathbf{K}_k$, corresponds to the kth bigram count $\mathbf{v}_k$, $\mathbf{K}_k = \mathbf{v}_k\mathbf{v}_k^\top$. Each base kernel is normalized to have trace one and the labels are centered.

The `alignf` method returns a sparse weight vector due to the constraint $\boldsymbol{\mu} \geq \mathbf{0}$. As is demonstrated by the performance of the `l1-svm` method, Table 3, and also previously observed by Cortes

	BOOKS	DVD	ELEC	KITCHEN
`unif`	1.442 ± .015 0.029 ± .005	1.438 ± .033 0.029 ± .005	1.342 ± .030 0.038 ± .002	1.356 ± .016 0.039 ± .006
`l2-krr`	1.410 ± .024 0.036 ± .008	1.423 ± .034 0.036 ± .009	1.318 ± .033 0.050 ± .004	1.333 ± .015 0.056 ± .005
`align`	1.401 ± .035 0.046 ± .006	1.414 ± .017 0.047 ± .005	1.308 ± .033 0.065 ± .004	1.312 ± .012 0.076 ± .008

REGRESSION

	BOOKS	DVD	ELEC	KITCHEN
`unif`	0.258 ± .017 0.030 ± .004	0.243 ± .015 0.030 ± .005	0.188 ± .014 0.040 ± .002	0.201 ± .020 0.039 ± .007
`l1-svm`	0.286 ± .016 0.030 ± .011	0.292 ± .025 0.033 ± .014	0.238 ± .019 0.051 ± .004	0.236 ± .024 0.058 ± .007
`align`	0.243 ± .020 0.043 ± .003	0.214 ± .020 0.045 ± .005	0.166 ± .016 0.063 ± .004	0.172 ± .022 0.070 ± .010

CLASSIFICATION

Table 3: The error measures (top) and alignment values (bottom) on four sentiment analysis domains using kernels learned as combinations of rank-one base kernels corresponding to individual features. The results are shown with ±1 standard deviation as measured by 5-fold cross-validation. Further measures of significance are shown in Appendix C, Table 5.

et al. (2009a), a sparse weight vector $\boldsymbol{\mu}$ does not generally offer an improvement over the uniform combination in the rank-one setting. Thus, we focus on the performance of `align` and compare it to `unif` and one-stage learning methods. Table 3 shows that `align` significantly improves both the alignment and the error percentage over `unif` and also improves somewhat over the one-stage `l2-krr` algorithm. Evidence of statistical significance is provided in Appendix C, Table 5. Note that, although the sparse weighting provided by `l1-svm` improves the alignment in certain cases, it does not improve performance.

6. Conclusion

We presented a series of novel algorithmic, theoretical, and empirical results for learning kernels based on the notion of centered alignment. Our experiments show a consistent improvement of the performance of alignment-based algorithms over previous learning kernel techniques, as well as the straightforward uniform kernel combination, which has been difficult to surpass in the past, in both classification and regression. The algorithms we described are efficient and easy to implement. All the algorithms presented in this paper are available in the open-source C++ library available at `www.openkernel.org`. They can be used in a variety of applications to improve performance. We also gave an extensive theoretical analysis which provides a number of guarantees for centered alignment-based algorithms and methods. Several of the algorithmic and theoretical results presented can be extended to other learning settings. In particular, methods based on similar ideas could be used to design learning kernel algorithms for dimensionality reduction.

The notion of centered alignment served as a key similarity measure to achieve these results. Note that we are not proving that good alignment is necessarily needed for a good classifier, but both our theory and empirical results do suggest the existence of accurate predictors with a good centered alignment. Different methods based on possibly different efficiently computable similarity measures could be used to design effective learning kernel algorithms. In particular, the notion of similarity suggested by Balcan and Blum (2006), if it could be computed from finite samples, could be used in a equivalent way.

Acknowledgments

The work of author MM was partly supported by a Google Research Award.

Appendix A. Lemmas Supporting Proof of Proposition 11

For a function f of the sample S, we denote by $\Delta(f)$ the difference $f(S') - f(S)$, where S' is a sample differing from S by just one point, say the m-th point is x_m in S and x'_m in S'. The following perturbation bound will be needed in order to apply McDiarmid's inequality.

Lemma 18 *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *according to the distribution* D. *Assume that for any* $x \in \mathcal{X}$, $K(x,x) \leq R^2$ *and* $K'(x,x) \leq R'^2$. *Then, the following perturbation inequality holds when changing one point of the sample:*

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F)| \leq \frac{24 R^2 R'^2}{m}.$$

Proof By Lemma 1, we can write:

$$\begin{aligned}\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F &= \mathrm{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}'\right] \\ &= \mathrm{Tr}\left[\mathbf{K}\mathbf{K}' - \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\mathbf{K}' - \mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}' + \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}'\right] \\ &= \langle \mathbf{K}, \mathbf{K}' \rangle_F - \frac{\mathbf{1}^\top(\mathbf{K}\mathbf{K}' + \mathbf{K}'\mathbf{K})\mathbf{1}}{m} + \frac{(\mathbf{1}^\top\mathbf{K}\mathbf{1})(\mathbf{1}^\top\mathbf{K}'\mathbf{1})}{m^2}.\end{aligned}$$

The perturbation of the first term is given by

$$\Delta(\langle \mathbf{K}, \mathbf{K}' \rangle_F) = \sum_{i=1}^{m} \Delta(\mathbf{K}_{im}\mathbf{K}'_{im}) + \sum_{i \neq m} \Delta(\mathbf{K}_{mi}\mathbf{K}'_{mi}).$$

By the Cauchy-Schwarz inequality, for any $i, j \in [1, m]$,

$$|\mathbf{K}_{ij}| = |K(x_i, x_j)| \leq \sqrt{K(x_i, x_i) K(x_j, x_j)} \leq R^2$$

and the product can be bound as $|\mathbf{K}_{i,j}\mathbf{K}'_{i,j}| \leq |\mathbf{K}_{i,j}||\mathbf{K}'_i j| \leq R^2R'^2$. The difference of products is then bound as $|\Delta(\mathbf{K}_{i,j}\mathbf{K}'_{i,j})| \leq 2R^2R'^2$. Thus,

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K}, \mathbf{K}' \rangle_F)| \leq \frac{2m-1}{m^2}(2R^2R'^2) \leq \frac{4R^2R'^2}{m}.$$

Similarly, for the first part of the second term, we obtain

$$\begin{aligned}
\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top \mathbf{K}\mathbf{K}'\mathbf{1}}{m}\right)\right| &= \left|\Delta\left(\sum_{i,j,k=1}^{m} \frac{\mathbf{K}_{ik}\mathbf{K}'_{kj}}{m^3}\right)\right| \\
&= \left|\Delta\left(\frac{\sum_{i,k=1}^{m} \mathbf{K}_{ik}\mathbf{K}'_{km} + \sum_{i,j\neq m} \mathbf{K}_{im}\mathbf{K}'_{mj}}{m^3} + \frac{\sum_{k\neq m, j\neq m} \mathbf{K}_{mk}\mathbf{K}'_{kj}}{m^3}\right)\right| \\
&\leq \frac{m^2 + m(m-1) + (m-1)^2}{m^3}(2R^2R'^2) \leq \frac{3m^2 - 3m + 1}{m^3}(2R^2R'^2) \\
&\leq \frac{6R^2R'^2}{m}.
\end{aligned}$$

Similarly, we have:

$$\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top \mathbf{K}'\mathbf{K}\mathbf{1}}{m}\right)\right| \leq \frac{6R^2R'^2}{m}.$$

The final term is bounded as follows,

$$\begin{aligned}
\frac{1}{m^2}\left|\Delta\left(\frac{(\mathbf{1}^\top \mathbf{K}\mathbf{1})(\mathbf{1}^\top \mathbf{K}'\mathbf{1})}{m^2}\right)\right| &\leq \left|\Delta\left(\frac{\sum_{i,j,k} \mathbf{K}_{ij}\mathbf{K}'_{km} + \sum_{i,j,k\neq m} \mathbf{K}_{ij}\mathbf{K}'_{mk}}{m^4} + \right.\right. \\
&\qquad \left.\left.\frac{\sum_{i,j\neq m,k\neq m} \mathbf{K}_{im}\mathbf{K}'_{jk} + \sum_{i\neq m, j\neq m, k\neq m} \mathbf{K}_{mi}\mathbf{K}'_{jk}}{m^4}\right)\right| \\
&\leq \frac{m^3 + m^2(m-1) + m(m-1)^2 + (m-1)^3}{m^4}(2R^2R'^2) \\
&\leq \frac{8R^2R'^2}{m}.
\end{aligned}$$

Combining these last four inequalities leads directly to the statement of the lemma. ■

Because of the diagonal terms of the matrices, $\frac{1}{m^2}\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F$ is not an unbiased estimate of $\mathrm{E}[K_c K'_c]$. However, as shown by the following lemma, the estimation bias decreases at the rate $O(1/m)$.

Lemma 19 *Under the same assumptions as Lemma 18, the following bound on the difference of expectations holds:*

$$\left|\mathop{\mathrm{E}}_{x,x'}[K_c(x,x')K'_c(x,x')] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right]\right| \leq \frac{18R^2R'^2}{m}.$$

Proof To simplify the notation, unless otherwise specified, the expectation is taken over x, x' drawn according to the distribution D. The key observation used in this proof is that

$$\mathop{\mathrm{E}}_{S}[\mathbf{K}_{ij}\mathbf{K}'_{ij}] = \mathop{\mathrm{E}}_{S}[K(x_i,x_j)K'(x_i,x_j)] = \mathrm{E}[KK'], \tag{11}$$

for i, j distinct. For expressions such as $\mathrm{E}_S[\mathbf{K}_{ik}\mathbf{K}'_{kj}]$ with i, j, k distinct, we obtain the following:

$$\operatorname*{E}_S[\mathbf{K}_{ik}\mathbf{K}'_{kj}] = \operatorname*{E}_S[K(x_i,x_k)K'(x_k,x_j)] = \operatorname*{E}_{x'}[\operatorname*{E}_x[K]\operatorname*{E}_x[K']]. \tag{12}$$

Let us start with the expression of $\mathrm{E}[K_cK'_c]$:

$$\mathrm{E}[K_cK'_c] = \mathrm{E}\left[\left(K - \operatorname*{E}_{x'}[K] - \operatorname*{E}_x[K] + \mathrm{E}[K]\right)\left(K' - \operatorname*{E}_{x'}[K'] - \operatorname*{E}_x[K'] + \mathrm{E}[K']\right)\right]. \tag{13}$$

After expanding this expression, applying the expectation to each of the terms, and simplifying, we obtain:

$$\mathrm{E}[K_cK'_c] = \mathrm{E}[KK'] - 2\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right] + \mathrm{E}[K]\,\mathrm{E}[K'].$$

$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F$ can be expanded and written more explicitly as follows:

$$\begin{aligned}
\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F &= \langle \mathbf{K}, \mathbf{K}'\rangle_F - \frac{\mathbf{1}^\top\mathbf{K}\mathbf{K}'\mathbf{1}}{m} - \frac{\mathbf{1}^\top\mathbf{K}'\mathbf{K}\mathbf{1}}{m} + \frac{\mathbf{1}^\top\mathbf{K}'\mathbf{1}\mathbf{1}^\top\mathbf{K}\mathbf{1}}{m^2} \\
&= \sum_{i,j=1}^m \mathbf{K}_{ij}\mathbf{K}'_{ij} - \frac{1}{m}\sum_{i,j,k=1}^m (\mathbf{K}_{ik}\mathbf{K}'_{kj} + \mathbf{K}'_{ik}\mathbf{K}_{kj}) + \frac{1}{m^2}(\sum_{i,j=1}^m \mathbf{K}_{ij})(\sum_{i,j=1}^m \mathbf{K}'_{ij}).
\end{aligned}$$

To take the expectation of this expression, we use the observations (11) and (12) and similar identities. Counting terms of each kind, leads to the following expression of the expectation:

$$\begin{aligned}
\operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right] = {} & \left[\frac{m(m-1)}{m^2} - \frac{2m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[KK'] \\
&+ \left[\frac{-2m(m-1)(m-2)}{m^3} + \frac{2m(m-1)(m-2)}{m^4}\right]\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right] \\
&+ \left[\frac{m(m-1)(m-2)(m-3)}{m^4}\right]\mathrm{E}[K]\,\mathrm{E}[K'] \\
&+ \left[\frac{m}{m^2} - \frac{2m}{m^3} + \frac{m}{m^4}\right]\operatorname*{E}_x[K(x,x)K'(x,x)] \\
&+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x)K'(x,x')] \\
&+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x')K'(x,x)] \\
&+ \left[\frac{m(m-1)}{m^4}\right]\operatorname*{E}_x[K(x,x)]\operatorname*{E}_x[K'(x,x)] \\
&+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\operatorname*{E}_x[K(x,x)]\,\mathrm{E}[K'] \\
&+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\mathrm{E}[K]\operatorname*{E}_x[K'(x,x)].
\end{aligned}$$

Taking the difference with the expression of $\mathrm{E}[K_cK'_c]$ (Equation 13), using the fact that terms of form $\mathrm{E}_x[K(x,x)K'(x,x)]$ and other similar ones are all bounded by $R^2R'^2$ and collecting the terms gives

$$\begin{aligned}
\left|\mathrm{E}[K_cK'_c] - \operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| \leq {} & \frac{3m^2-4m+2}{m^3}\mathrm{E}[KK'] - 2\frac{4m^2-5m+2}{m^3}\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right] \\
&+ \frac{6m^2-11m+6}{m^3}\mathrm{E}[K]\,\mathrm{E}[K'] + \gamma,
\end{aligned}$$

with $|\gamma| \leq \frac{m-1}{m^2} R^2 R'^2$. Using again the fact that the expectations are bounded by $R^2 R'^2$ yields

$$\left| \mathrm{E}[K_c K'_c] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right] \right| \leq \left[\frac{3}{m} + \frac{8}{m} + \frac{6}{m} + \frac{1}{m}\right] R^2 R'^2 \leq \frac{18}{m} R^2 R'^2,$$

and concludes the proof. ■

Appendix B. Stability Bounds for Alignment Maximization Algorithm

Lemma 20 *Let $\boldsymbol{\mu} = \mathbf{v}/\|\mathbf{v}\|$ and $\boldsymbol{\mu}' = \mathbf{v}'/\|\mathbf{v}'\|$. Then, the following identity holds for $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}' - \boldsymbol{\mu}$:*

$$\Delta\boldsymbol{\mu} = \left[\frac{\Delta\mathbf{v}}{\|\mathbf{v}'\|} - \frac{(\Delta\mathbf{v})^\top (\mathbf{v} + \mathbf{v}')\mathbf{v}}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)}\right].$$

Proof By definition of $\Delta\boldsymbol{\mu}$, we can write

$$\Delta\boldsymbol{\mu} = \Delta\left(\frac{\mathbf{v}}{\|\mathbf{v}\|}\right) = \left[\frac{\mathbf{v}' - \mathbf{v}}{\|\mathbf{v}'\|} - \frac{\mathbf{v}\|\mathbf{v}'\| - \mathbf{v}\|\mathbf{v}\|}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right] = \left[\frac{\Delta\mathbf{v}}{\|\mathbf{v}'\|} - \frac{\mathbf{v}\Delta(\|\mathbf{v}\|)}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right]. \tag{14}$$

Observe that:

$$\Delta(\|\mathbf{v}\|) = \frac{\Delta(\|\mathbf{v}\|^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\Delta(\sum_{i=1}^p v_i^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\sum_{i=1}^p \Delta(v_i)(v_i + v'_i)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{(\Delta\mathbf{v})^\top (\mathbf{v} + \mathbf{v}')}{\|\mathbf{v}\| + \|\mathbf{v}'\|}.$$

Plugging in this expression in (14) yields the statement of the lemma. ■

Consider the minimization (7) shown by Proposition 9 to provide the solution of the alignment maximization problem for a convex combination. The matrix $\mathbf{M}$ and vector $\mathbf{a}$ are functions of the training sample S. To emphasize this dependency, we rewrite that optimization for a sample S as

$$\min_{\mathbf{v} \geq \mathbf{0}} F(S, \mathbf{v}), \tag{15}$$

where $F(S, \mathbf{v}) = \mathbf{v}^\top \mathbf{M}\mathbf{v} - 2\mathbf{v}^\top \mathbf{a} = \|\mathbf{v}\|_{\mathbf{M}}^2 - 2\mathbf{v}^\top \mathbf{a}$. The following lemma provides a stability result for this optimization problem.

Proposition 21 *Let S and S' denote two samples of size m differing by only one point. Let $\mathbf{v}$ and $\mathbf{v}'$ be the solution of (15), respectively, for sample S and S'. Then, the following inequality holds for $\Delta\mathbf{v} = \mathbf{v}' - \mathbf{v}$:*

$$\|\Delta\mathbf{v}\|_{\mathbf{M}}^2 \leq \left[\Delta\mathbf{a} - (\Delta\mathbf{M})\mathbf{v}'\right]^\top \Delta\mathbf{v}.$$

Proof Since $C = \{\mathbf{v} \colon \mathbf{v} \geq 0\}$ is convex, for any $s \in [0, 1]$, $\mathbf{v} + s\Delta\mathbf{v}$ and $\mathbf{v}' - s\Delta\mathbf{v}$ are in C. Thus, by definition of $\mathbf{v}'$ and $\mathbf{v}$,

$$F(S, \mathbf{v}) \leq F(S, \mathbf{v} + s\Delta\mathbf{v}) \quad \text{and} \quad F(S', \mathbf{v}') \leq F(S', \mathbf{v}' - s\Delta\mathbf{v}).$$

Summing up these inequalities, we obtain

$$\begin{aligned}
\|\mathbf{v}\|_{\mathbf{M}}^2 - \|\mathbf{v} + s\Delta\mathbf{v}\|_{\mathbf{M}}^2 + \|\mathbf{v}'\|_{\mathbf{M}'}^2 - \|\mathbf{v}' - s\Delta\mathbf{v}\|_{\mathbf{M}'}^2 & \\
&\leq 2\mathbf{v}^\top \mathbf{a} - 2(\mathbf{v} + s\Delta\mathbf{v})^\top \mathbf{a} + 2\mathbf{v}'^\top \mathbf{a}' - 2(\mathbf{v}' + s\Delta\mathbf{v})^\top \mathbf{a}' \\
&= -2[s\mathbf{a}^\top \Delta\mathbf{v} - s\mathbf{a}'^\top \Delta\mathbf{v}] = 2s(\Delta\mathbf{a})^\top \Delta\mathbf{v}.
\end{aligned}$$

KINEMATICS

	unif	l2-krr	align	alignf
unif	–	1	1	1
l2-krr	0	–	1	1
align	0	0	–	1
alignf	0	0	0	–

IONOSPHERE

	unif	l2-krr	align	alignf
unif	–	1	1	1
l2-krr	0	–	1	1
align	0	0	–	1
alignf	0	0	0	–

GERMAN

	unif	l1-svm	align	alignf
unif	–	0	1	1
l1-svm	0	–	0	1
align	0	0	–	1
alignf	0	0	0	–

SPAMBASE

	unif	l1-svm	align	alignf
unif	–	0	0	0
l1-svm	1	–	1	1
align	0	0	–	0
alignf	0	0	0	–

SPLICE

	unif	l1-svm	align	alignf
unif	–	0	0	1
l1-svm	0	–	0	0
align	0	0	–	0
alignf	0	0	0	–

Table 4: Significance tests for general kernel combination results presented in Table 2. An entry of 1 indicates that the algorithm listed in the column has a significantly better accuracy than the algorithm listed in the row.

The left-hand side of this inequality can be rewritten as follows after expansion and using the identity $\|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\mathbf{M}'} - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\mathbf{M}} = \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}$:

$$\begin{aligned} -\|s\Delta\mathbf{v}\|^2_{\mathbf{M}} - 2s\mathbf{v}^\top\mathbf{M}\Delta\mathbf{v} + \|\mathbf{v}'\|^2_{\mathbf{M}'} - \|\mathbf{v}'\|^2_{\mathbf{M}} - \|s\Delta\mathbf{v}\|^2_{\mathbf{M}} + 2s\mathbf{v}'^\top\mathbf{M}(\Delta\mathbf{v}) - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}} \\ = 2s(1-s)\|\Delta\mathbf{v}\|^2_{\mathbf{M}} + \|\mathbf{v}'\|^2_{\Delta\mathbf{M}} - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}. \end{aligned}$$

Then, expanding $\|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}$ results in the final inequality

$$2s(1-s)\|\Delta\mathbf{v}\|^2_{\mathbf{M}} - s^2\|\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}} + 2s\mathbf{v}'^\top(\Delta\mathbf{M})(\Delta\mathbf{v}) \leq 2s(\Delta\mathbf{a})^\top\Delta\mathbf{v}.$$

Dividing by s and setting $s=0$ yields

$$\|\Delta\mathbf{v}\|^2_{\mathbf{M}} + \mathbf{v}'^\top(\Delta\mathbf{M})(\Delta\mathbf{v}) \leq (\Delta\mathbf{a})^\top\Delta\mathbf{v},$$

which concludes the proof of the lemma. ■

Appendix C. Significance Tests for Empirical Results

Tables 4 and 5 show the results of paired-sample one-sided T-tests for all pairs of algorithms compared across all data sets presented in Section 5 for both regression and classification. Each entry of the tables indicates whether the mean error of the algorithm listed in the column is significantly less than the mean error of the algorithm listed in the row at significance level $p = 0.1$. An entry

Books

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

Dvd

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	0
align	0	0	–

Elec

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

Kitchen

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

Regression

Books

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

Dvd

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

Elec

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

Kitchen

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

Classification

Table 5: Significance tests for rank-one kernel combination results presented in Table 3. An entry of 1 indicates that the algorithm listed in the column has a significantly better accuracy then the algorithm listed in the row.

of 1 indicates a significant difference, while an entry of 0 indicates that the null hypothesis (that the errors are not significantly different) cannot be rejected.

Table 4 indicates that the `alignf` method offers significant improvement over `unif` in all data sets with the exception of spambase and significantly improves over the compared one-stage method in all data sets apart from splice. Table 5 indicates that the `align` method significantly improves over both the uniform and one-stage combination in all data sets apart from dvd in the regression setting, where improvement over `l2-krr` is not deemed significant.

References

Andreas Argyriou, Charles Micchelli, and Massimiliano Pontil. Learning convex combinations of continuously parameterized basic kernels. In *COLT*, 2005.

Andreas Argyriou, Raphael Hauser, Charles Micchelli, and Massimiliano Pontil. A DC-programming algorithm for kernel selection. In *ICML*, 2006.

Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, 2008.

Maria-Florina Balcan and Avrim Blum. On a theory of learning with similarity functions. In *ICML*, 2006.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*, 2007.

Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, volume 5, 1992.

Olivier Bousquet and André Elisseeff. Algorithmic stability and generalization performance. In *NIPS*, 2000.

Olivier Bousquet and Daniel J. L. Herrmann. On the complexity of learning the kernel matrix. In *NIPS*, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3), 2002.

Corinna Cortes. Invited talk: Can learning kernels help performance? In *ICML*, 2009.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3), 1995.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning sequence kernels. In *MLSP*, 2008.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L_2-regularization for learning kernels. In *UAI*, 2009a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning non-linear combinations of kernels. In *NIPS*, 2009b.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel methods. In *ICML*, 2010a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *ICML*, 2010b.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the Impact of Kernel Approximation on Learning Accuracy. In *AISTATS*, 2010c.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Ensembles of kernel predictors. In *UAI*, 2011a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Tutorial: Learning kernels. In *ICML*, 2011b.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. On kernel-target alignment. In *NIPS*, 2001.

Nello Cristianini, Jaz S. Kandola, André Elisseeff, and John Shawe-Taylor. On kernel target alignment. http://www.support-vector.net/papers/alignment_JMLR.ps, unpublished, 2002.

Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic learning theory*, 2005.

Tony Jebara. Multi-task feature and kernel selection for SVMs. In *ICML*, 2004.

Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. On the extensions of kernel alignment. technical report 120, Department of Computer Science, Univ. of London, UK, 2002a.

Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. Optimizing kernel alignment over combinations of kernels. technical report 121, Dept. of CS, Univ. of London, UK, 2002b.

Seung-Jean Kim, Alessandro Magnani, and Stephen Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *ICML*, 2006.

Vladimir Koltchinskii and Ming Yuan. Sparse recovery in large ensembles of kernel machines. In *COLT*, 2008.

Gert Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5, 2004.

Darrin P. Lewis, Tony Jebara, and William Stafford Noble. Nonstationary kernel combination. In *ICML*, 2006.

Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141, 1989.

Marina Meila. Data centering in feature space. In *AISTATS*, 2003.

Charles Micchelli and Massimiliano Pontil. Learning the kernel function via regularization. *JMLR*, 6, 2005.

Cheng Soon Ong, Alexander Smola, and Robert Williamson. Learning the kernel with hyperkernels. *JMLR*, 6, 2005.

Jean-Baptiste Pothin and Cédric Richard. Optimizing kernel alignment by data translation in feature space. In *ICASSP*, 2008.

Craig Saunders, A. Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

Nathan Srebro and Shai Ben-David. Learning bounds for support vector machines with learned kernels. In *COLT*, 2006.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *ICML*, 2009.

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *ICML*, 2007.

Journal of Machine Learning Research 13 (2012) 829-848 Submitted 3/11; Revised 6/11; Published 3/12

Causal Bounds and Observable Constraints for Non-deterministic Models

Roland R. Ramsahai RAMSAHAI@STATSLAB.CAM.AC.UK
Statistical Laboratory
Centre for Mathematical Sciences
University of Cambridge
Wilberforce Road
Cambridge CB3 0WB, UK

Editor: Peter Spirtes

Abstract

Conditional independence relations involving latent variables do not necessarily imply observable independences. They may imply inequality constraints on observable parameters and causal bounds, which can be used for falsification and identification. The literature on computing such constraints often involve a deterministic underlying data generating process in a counterfactual framework. If an analyst is ignorant of the nature of the underlying mechanisms then they may wish to use a model which allows the underlying mechanisms to be probabilistic. A method of computation for a weaker model without any determinism is given here and demonstrated for the instrumental variable model, though applicable to other models. The approach is based on the analysis of mappings with convex polytopes in a decision theoretic framework and can be implemented in readily available polyhedral computation software. Well known constraints and bounds are replicated in a probabilistic model and novel ones are computed for instrumental variable models without non-deterministic versions of the randomization, exclusion restriction and monotonicity assumptions respectively.

Keywords: instrumental variables, instrumental inequality, causal bounds, convex polytope, latent variables, directed acyclic graph

1. Introduction

Conditional independence relations represent equality constraints on the parameters of a joint probability distribution. Such relations cannot be empirically validated if they involve latent variables. Collections of latent conditional independencies may imply inequality constraints on parameters of the observable distribution. The classical motivation is the *instrumental variable* (IV) model (Durbin, 1954; Angrist et al., 1996). It includes the IV, A, and inference is required about the effect of a variable, B, on another, C, in the presence of latent confounders, U. The IV model is defined by $A \not\perp\!\!\!\perp B, C \perp\!\!\!\perp A \,|\, (B,U)$ and $U \perp\!\!\!\perp A$. The latter two involve the latent variable U so it was traditionally thought that the model could not be empirically verified (Imbens and Angrist, 1994). However Pearl (1995) derived the 'instrumental inequality'

$$\max_B \sum_C \left\{ \max_A \mathrm{P}(C,B\,|\,A) \right\} \leq 1, \tag{1}$$

a set of constraints which are implied by and can be used to falsify the discrete IV model. To compute the constraints, Pearl (1995) defines the IV model as a deterministic counterfactual model (Rubin, 1974). Such models involve latent deterministic relations, which marginalise to produce observed probabilistic relationships, and are technically equivalent to structural equation models (Strotz and Wold, 1960) and other functional models (Heckerman and Shachter, 1995).

Without intervention data and further assumptions, the causal effect of B on C cannot be point identified (Durbin, 1954; Angrist et al., 1996), but, using the deterministic counterfactual model, it can be bounded with the joint distribution of A, B and C (Pearl, 1995; Robins, 1989; Manski, 1990). Thus making it possible to acquire non-trivial information about the effect of the intervention when intervention studies cannot be conducted; because of ethical, financial or other reasons. Using the deterministic counterfactual approach and linear programming software developed by Balke (1995), the constraints on the causal effect of B on C were improved by Balke and Pearl (1997) and extended to other models by Kaufman et al. (2009). This linear programming approach within a deterministic counterfactual model has become the standard tool for computing such constraints, with some exceptions (Geiger and Meek, 1998; Kang and Tian, 2006).

As a technical construct for computations, deterministic counterfactual models are widely accepted as valuable. Applications of deterministic counterfactual models assume there are underlying deterministic relations (Angrist et al., 1996) and pose no issues if the determinism can be practically justified. For certain applications though, for example, mutations that cause cancer (Aalen and Frigessi, 2007), subject matter knowledge suggests that assumptions about the existence of deterministic mechanisms are unrealistic and spawns controversy (Dawid, 2000). Even if an analyst is unaware of the type of mechanisms involved in their study, it would be desirable to avoid deterministic counterfactuals if alternative computations are no more difficult. The method in §2 provides such an alternative, which is agnostic to whether the underlying mechanisms are probabilistic or deterministic, to deriving falsifiable constraints and causal bounds of the type previously described. The method described does not use counterfactuals, which has certain advantages (Dawid, 2000), but more importantly demonstrates that the determinism in the models is unnecessary.

In this discussion, causal inference is formalized within standard decision theory (Spirtes et al., 1993; Pearl, 1993), with conditional independence assumptions (Lauritzen, 2001; Dawid, 2002). The model has been successfully applied in defining direct effects (Geneletti, 2007) and dynamic treatment strategies (Dawid and Didelez, 2010), to name a few. The approach in §2 and throughout uses this framework and it is compared to the counterfactual framework in §3. The method is based on the analysis of convex polytopes and can be implemented in standard polytope representation software such as Polymake (Gawrilow and Joswig, 2000) or PORTA (Christof and Loebel, 1998). Known constraints, which have been previously derived using deterministic counterfactuals, are derived in §5. Graphical models for representing causal assumptions are described in §4. Non-trivial modifications of the computation technique are considered in §6, §7 and §8 to derive novel constraints and causal bounds when various assumptions in the IV model are weakened.

Example 1 *Consider an IV model of partial compliance, where $A \in \{1,2\}$ is treatment assigned, $B \in \{0,1\}$ is treatment taken and C is an outcome of interest. The counterfactual IV model involves counterfactual variables (B_1, B_2) which represent a unit's deterministic compliance behaviour when A is set to* 1 *(no treatment) or* 2 *(treatment) respectively. Analyses of this model often make the monotonicity assumption $B_2 \geq B_1$, meaning that a unit which does not take treatment if assigned it, will never take it.*

The deterministic counterfactual framework only allows the compliance behaviour in Example 1 to be modelled as deterministic. Monotonicity assumptions, used to compute bounds in IV models, in the literature (Pearl, 1995; Balke and Pearl, 1997) are imposed in models which are stronger than necessary. It is shown in §8 that the known bounds and novel bounds can be derived for a weaker model. In the context of Example 1, monotonicity in the weaker model is equivalent to assuming that units are more likely to take treatment if assigned it than if not assigned it.

The counterfactual IV model in Example 1 uses the exclusion restriction assumption (Imbens and Angrist, 1994). This assumption restricts C to be a deterministic function of compliance behaviour and treatment taken only. Stochastic exclusion restrictions are considered within the deterministic counterfactual framework in Hirano et al. (2000). In a weaker fully probabilistic model, the exclusion restriction assumption $C \perp\!\!\!\perp A \,|\, (B,U)$ is used in §2 to replicate results which were derived under the stronger model (Balke and Pearl, 1993; Pearl, 1995; Balke and Pearl, 1997). Whilst varying the strength of the exclusion restriction, novel constraints are derived in §7 with the probabilistic approach. This allows a sensitivity analysis to the non-deterministic exclusion restriction, which is important when assumptions involve unobservable variables (Shepherd et al., 2006).

Another assumption in the IV model in Example 1 is that treatment assignment is independent of compliance behaviour. In the probabilistic framework, novel constraints are computed for a weaker IV model with $U \not\perp\!\!\!\perp A$, as described in §6. Applications to data are given in §9. The IV model provides motivation for this discussion but the approach extends to other models. The notation used throughout is listed in Appendix A.

2. Computation of Constraints in the Instrumental Variable Model

Consider a model involving the random variables A, B, C and U, where the state space of A is $\{1,2\}$, B is $\{0,1\}$ and C is $\{0,1\}$. U is unobservable by definition so no assumption is made about it. Let $\vec{v}^* = (\zeta^*_{00.1}, \zeta^*_{01.1}, \ldots, \zeta^*_{11.2})$ be a random vector with components $\zeta^*_{cb.a}$, which are random variables that are functions of U, where $\zeta^*_{cb.a} = \mathrm{P}(C=c, B=b \,|\, A=a, U)$. Similarly, let $\vec{v} = (\zeta_{00.1}, \zeta_{01.1}, \ldots, \zeta_{11.2})$ be a fixed vector of probabilities that are not functions of U, where $\zeta_{cb.a} = \mathrm{P}(C=c, B=b \,|\, A=a)$. Let $\vec{\tau}^* = (\eta^*_0, \eta^*_1, \delta^*_1, \delta^*_2)$, where

$$\eta^*_b = \mathrm{P}(C=1 \,|\, B=b, U), \quad \delta^*_a = \mathrm{P}(B=1 \,|\, A=a, U). \tag{2}$$

Since $\vec{\tau}^*$ is a vector of probabilities then $\vec{\tau}^* \in \mathcal{T}$ since the components of $\vec{\tau}^*$ satisfy the axioms of probability, where $\mathcal{T} = [0,1]^4$. To derive falsifiable constraints on $\vec{v}$ for the IV model, it is necessary to determine the set of $\vec{v}$ which does not satisfy the assumptions in the IV model. Under the IV model, $C \perp\!\!\!\perp A \,|\, (B,U)$, which implies that

$$\mathrm{P}(C,B \,|\, A,U) = \mathrm{P}(C \,|\, B,U)\mathrm{P}(B \,|\, A,U), \tag{3}$$

and $\vec{v}^*$ can be parameterised by $\vec{\tau}^*$. The relation in Equation (3) together with the codes in (2) define a mapping $\Xi : \vec{\tau}^* \in \mathcal{T} \to \vec{v}^* \in \mathcal{V}$, where $\vec{\tau}^*$ is unrestricted by the IV model and $\mathcal{V} = \Xi(\mathcal{T})$ contains all $\vec{v}^*$ which obey the IV model. Since the components of each $\vec{v}^*$ obey the axioms of probability then $\mathcal{V} \subseteq \mathcal{Z}$, where $\mathcal{Z} \subset [0,1]^8$ is the intersection of the hyperplanes defined by $\sum_{c,b} \zeta^*_{cb.a} = 1$ for $a \in \{1,2\}$.

Under the IV model, $U \perp\!\!\!\perp A$, which implies that $\zeta_{cb.a} = \mathrm{E}_U(\zeta^*_{cb.a})$ and thus all $\vec{v}$ that obey the IV model lie in $\mathcal{H}$, where $\mathcal{H}$ is the set of all possible convex combinations of all $\vec{v}^* \in \mathcal{V}$ or the convex hull of $\mathcal{V}$. Let $\hat{\mathcal{V}} = \Xi(\hat{\mathcal{T}})$ and $\hat{\mathcal{H}}$ be the convex hull of $\hat{\mathcal{V}}$, where $\hat{\mathcal{T}}$ is the collection of extreme vertices of $\mathcal{T}$. The vertices $\hat{\mathcal{T}}$ and $\hat{\mathcal{V}}$ are partially listed in Figure 1 (top) and the transformation $\Xi(\cdot)$ is represented in Figure 1 (bottom).

$$\begin{array}{cccccccccccccc}
\eta_0^* & \eta_1^* & \delta_1^* & \delta_2^* & & \zeta_{00.1}^* & \zeta_{01.1}^* & \zeta_{10.1}^* & \zeta_{11.1}^* & \zeta_{00.2}^* & \zeta_{01.2}^* & \zeta_{10.2}^* & \zeta_{11.2}^* \\
0 & 0 & 0 & 0 & & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \rightarrow & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}$$

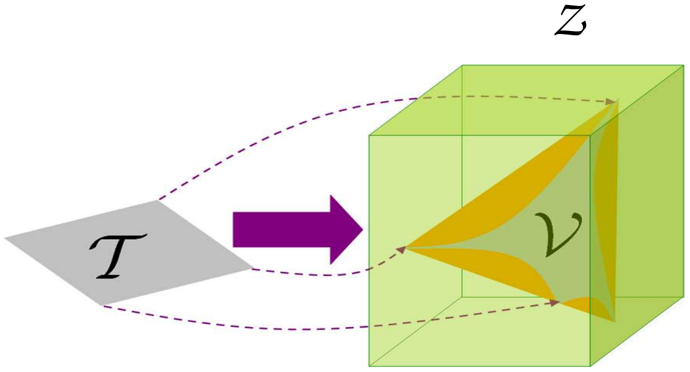


Figure 1: Transformation of extreme vertices (top) of polytope (bottom).

Since $\mathcal{H} = \hat{\mathcal{H}}$, from Theorem 1 in Appendix B, then all $\vec{v}$ that obey the IV model lie in $\hat{\mathcal{H}}$. The proof of Theorem 1 does not use the specific form of $\Xi(\cdot)$, only its monotonicity in each coordinate. A program such as Polymake (Gawrilow and Joswig, 2000) or PORTA (Christof and Loebel, 1998) can be used to transform the representation of $\hat{\mathcal{H}}$ in terms of its extreme vertices to a representation in terms of its facets or inequalities. The inequalities are constraints which are satisfied by $\vec{v}$ if $\vec{v}$ obeys the IV model. This specific computation produces the falsifiable 'instrumental inequality' constraints in (1) and is exactly the approach of Dawid (2003).

It is possible for the randomization or exclusion restriction assumption to fail without violation of any of the constraints in (1). This is because there are distributions $\mathrm{P}(C,B,A,U)$ which either violate the assumption $U \perp\!\!\!\perp A$ or Equation (3) but give rise to margins $\mathrm{P}(C,B,A)$ that obey the inequalities in (1). For example, if all $\vec{v}^*$ lie in $\mathcal{H} \backslash \mathcal{V}$ and randomization holds then the exclusion restriction in Equation (3) is not satisfied but all $\vec{v} \in \mathcal{H}$, which means that the inequalities in (1) are satisfied. I conjecture that the condition that U has a certain small state space is sufficient to imply that it is possible for the IV model to fail without violation of any of the constraints in (1).

3. Geometry of Counterfactuals and Latent Variables

The binary IV model can be re-parameterised by replacing U in $\mathrm{P}(C,B\,|\,A,U)$ with $\vec{\tau}^*$ and considering $\mathrm{P}(C,B\,|\,A,\vec{\tau}^*)$. The polytope $\mathcal{H}$ represents the model for $\mathrm{P}(C,B\,|\,A)$ and, since $\mathcal{H} = \hat{\mathcal{H}}$, a computationally and empirically indistinguishable model is formed by restricting $\vec{\tau}^*$ to $\hat{\mathcal{T}}$. In this minimal representation of the model, where $\vec{\tau}^* \in \hat{\mathcal{T}}$, the parameters $\eta_b^* \in \{0,1\}$ and $\delta_a^* \in \{0,1\}$. Therefore η_b^* is a deterministic function of B and the latent variable U and can be interpreted as a counterfactual variable which is the value of C when $B = b$ for a given $\vec{\tau}^*$, and similarly for δ_a^*. Similar comments are given in Lauritzen (2004).

If U is interpreted as the collection of variables which define a unit then $\vec{\tau}^*$ is the vector of potential responses for a unit and each vertex of the polytope corresponds to a certain type of unit. In the partial compliance model of Example 1, the vertex $\vec{\tau}^* = (0,0,0,0)$ corresponds to a unit which is classified as *never recover* (response is 0 regardless of treatment taken) and a *never taker* (treatment taken is 0 regardless of treatment assigned).

The probabilistic model is parameterised by $\vec{\tau}^*$ over the entire polytope $\mathcal{T}$ whereas the counterfactual model is parameterised by $\vec{\tau}^*$ only at the extreme vertices of the polytope $\mathcal{T}$. In special cases where latent determinism is realistic then such a parameterisation is meaningful and assumptions about the non-existence of certain vertices of the polytope or $\vec{\tau}^* \in \hat{\mathcal{T}}$ can potentially be justified. If latent determinism is known to be unrealistic (Aalen and Frigessi, 2007) and the reparameterisation is a technical construct then it may be wise to steer clear of any interpretation beyond simply saying that they are the vertices of the polytope defining the model.

The concepts are demonstrated in the reformulation of the monotonicity assumption in §8. The deterministic counterfactual approach assumes latent determinism and interprets the vertices as having real meaning. Under the deterministic interpretation, the monotonicity assumption implies that certain vertices are not valid for the model. The probabilistic approach defines monotonicity as a constraint on the latent conditional distributions to lie in a particular half-space, still allowing probabilistic behaviour.

4. Causal Graphical Models

The IV model considered so far, that is, without causal assumptions, is relatively simple. However extensions of it will be considered later and it will be useful, though not vital, to use graphical models to represent the assumptions involved. Graphs that are useful for representing conditional independence and causal assumptions are described in §4.1 and §4.2 respectively.

4.1 Directed Acyclic Graph

A purely probabilistic directed acyclic graph (DAG) (Lauritzen, 1996) consists of a set of *vertices* or *nodes*, $\mathcal{N}$, and a set of *directed edges*, $\mathcal{E}$. If $\lambda_1, \lambda_2 \in \mathcal{N}$ and $(\lambda_1, \lambda_2) \in \mathcal{E}$ then $(\lambda_2, \lambda_1) \notin \mathcal{E}$. It is said that there is a directed edge from λ_1 to λ_2, this is written as $\lambda_1 \to \lambda_2$ and λ_1 is called a *parent* of λ_2. In a DAG which represents the probability distribution of a set of random variables, $\mathcal{X}$, every $\lambda \in \mathcal{N}$ corresponds to a random variable $\mathcal{X}_\lambda \in \mathcal{X}$. The probability distribution function has the form

$$\mathrm{P}(\mathcal{X}) = \textstyle\prod_{\lambda \in \mathcal{N}} \mathrm{P}\{\mathcal{X}_\lambda \,|\, \mathcal{X}_{\mathrm{pa}(\lambda)}\}, \tag{4}$$

where 'pa$(\cdot)$' is the set of 'parents' of a node. This factorisation property is equivalent to a collection of conditional independence relations, which can be derived from the DAG using the concepts of 'd-separation' (Verma and Pearl, 1988) and a 'moral graph' (Lauritzen et al., 1990). The observational assumptions of the IV model can be represented by the DAG in Figure 2 (left).

4.2 Augmented Directed Acyclic Graph

The notation '$||$' (Lauritzen, 2001) is used for intervention conditioning and is equivalent to the 'do$(\cdot)$' notation (Goldszmidt and Pearl, 1992) and the '$\mathrm{P}_{\mathrm{man}(\cdot)}$' notation (Spirtes et al., 1993). Using

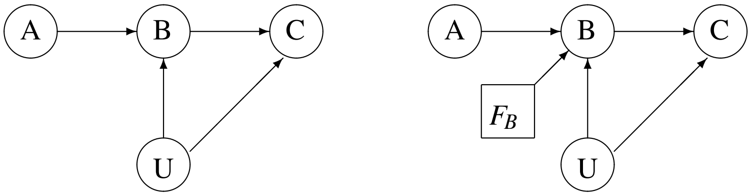


Figure 2: DAG which represents observational assumptions of the instrumental variable model (left) and augmented DAG for IV model, which includes causal assumptions (right).

the notation, $\mathrm{P}(C \,||\, B = b)$ is the probability of C given that B is actively forced to take the value b, and not passively observed to take the value b, as in $\mathrm{P}(C \,|\, B = b)$.

To derive intervention constraints, the assumptions represented by the augmented DAG (Spirtes et al., 1993; Pearl, 1993; Lauritzen, 2001; Dawid, 2002) in Figure 2 (right) are considered, where $\mathrm{ACE}(B \to C) = \alpha = P(C = 1 \,||\, B = 1) - P(C = 1 \,||\, B = 0)$ is the causal effect of interest. The intervention node F_B is a *regime indicator* decision variable which represents the way in which the value of B arises. Conditional independence relations can be derived in the same way as for the purely probabilistic DAGs since the probability distribution, conditional on F_B, still factorises according to Equation (4). The node F_B takes the values '*idle*', 0 or 1. If $F_B = idle$ then B takes a random value given by $\mathrm{P}\{B \,|\, \mathrm{pa}(B)\}$, but if F_B is either 0 or 1 then $B = F_B$. Using previous notation $\mathrm{P}(C \,||\, B = b) = \mathrm{P}(C \,|\, F_B = b)$. The relation $C \perp\!\!\!\perp B \,|\, (F_B = b, U)$ holds from the definition of F_B but is not represented in Figure 2 (right). Square nodes are decision nodes which represent fixed strategies, whereas circle nodes are random nodes which represent random variables.

The augmented DAGs which represent the IV model without randomization and the exclusion restriction are given in Figure 3.

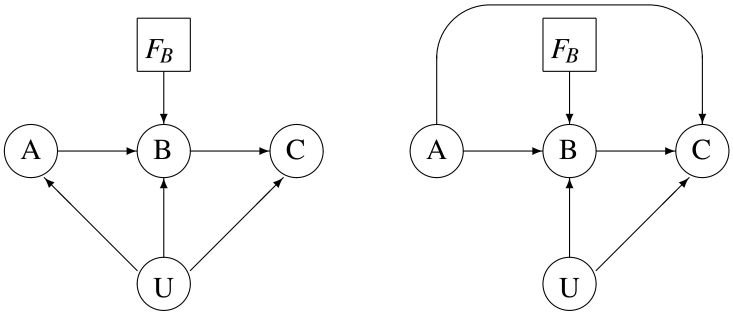


Figure 3: Augmented DAGs which represent the causal IV model without randomization (left) and without exclusion restriction (right).

The assumptions represented by the augmented DAGs in Figure 3 will be used in §6 and §7 respectively to derive constraints. The augmented DAG in Figure 2 (right) still applies under monotonicity since no extra conditional independences are assumed.

5. Applications of Computation

Many results in the literature are recovered by specific applications of the general method described in §2. It is based on parameterising with the factors of Equation (4) and transforming them according to a mapping. By defining the appropriate mapping, the various constraints are obtainable. Key requirements are the monotonicity of the mapping and that the space of valid parameters is the convex hull of the transformed polytope. Constraints on other quantities, such as $\mathrm{P}(C\,|\,A)$, $\mathrm{P}(B\,|\,A)$, $\mathrm{P}(C\,|\,B)$ etc., can be derived but some interesting examples are given in §5.1 and §5.2.

5.1 Falsifiable Constraints

Some applications, such as studies with partial compliance, require constraints involving the distribution $\mathrm{P}(C,B\,|\,A)$, whereas others can only identify the pairwise conditional distributions $\mathrm{P}(C\,|\,A)$ and $\mathrm{P}(B\,|\,A)$. For example, Mendelian randomization in genetic epidemiology involves the use of a genotype (A) as an instrument for the effect of a phenotype (B) on a disease (C). However only genotype-phenotype and genotype-disease data is usually available (Didelez and Sheehan, 2007) and thus constraints involving $\mathrm{P}(C\,|\,A)$ and $\mathrm{P}(B\,|\,A)$ are needed.

To derive the constraints, consider the monotone mapping $\vec{\tau}^* \longmapsto (\vec{\gamma}^*, \vec{\theta}^*)$ for the IV model of Figure 2 (left), where $\gamma^*_{ca} = \mathrm{P}(C = c\,|\,A = a, U)$ and $\theta^*_{ba} = \mathrm{P}(B = b\,|\,A = a, U)$. Since $U \perp\!\!\!\perp A$ then $(\vec{\gamma}, \vec{\theta})$ lies in the convex hull of the set of $(\vec{\gamma}^*, \vec{\theta}^*)$ which satisfy the IV model, where $\gamma_{ca} = \mathrm{P}(C = c\,|\,A = a)$ and $\theta_{ba} = \mathrm{P}(B = b\,|\,A = a)$. Similarly to the approach in §2, the constraints

$$\begin{aligned}\theta_{01} + \theta_{02} &\geq \gamma_{01} - \gamma_{02},\\ \theta_{01} + \theta_{02} &\geq \gamma_{02} - \gamma_{01},\\ \theta_{11} + \theta_{12} &\geq \gamma_{01} - \gamma_{02},\\ \theta_{11} + \theta_{12} &\geq \gamma_{02} - \gamma_{01},\end{aligned}$$

are obtained, which are the same as in Ramsahai (2007).

5.2 Bounds on Fixed Interventions

From the motivating Mendelian randomization example in §5.1, it may be necessary to obtain causal bounds in terms of the pairwise conditional distributions $\mathrm{P}(C\,|\,A)$ and $\mathrm{P}(B\,|\,A)$. Consider the model in Figure 2 (right). Since $C \perp\!\!\!\perp F_B\,|\,(B,U)$ and $U \perp\!\!\!\perp F_B$ then $\mathrm{P}(C\,||\,B) = \sum_U \mathrm{P}(C\,|\,B,U)\mathrm{P}(U)$. This implies that $(\vec{\gamma}, \vec{\theta}, \alpha)$ lies in the convex hull of $(\vec{\gamma}^*, \vec{\theta}^*, \alpha^*)$, where $\alpha^* = \mathrm{P}(C = 1\,|\,B = 1, U) - \mathrm{P}(C = 1\,|\,B = 0, U)$ and $\alpha = \mathrm{E}_U(\alpha^*)$. Therefore the monotone mapping $\vec{\tau}^* \longmapsto (\vec{\gamma}^*, \vec{\theta}^*, \alpha^*)$ can be used to compute constraints on $(\vec{\gamma}, \vec{\theta}, \alpha)$. The results of the computation are given in Appendix C and are the same as those derived in Ramsahai (2007).

Similarly, constraints and causal bounds in terms of the identifiable $\zeta_{cb.a}$ parameters can be obtained by considering the mapping $\vec{\tau}^* \longmapsto (\vec{v}^*, \alpha^*)$. The constraints involving the identifiable $\zeta_{cb.a}$ parameters only are the same as those obtained in §2, which are given in (1), and the rest constrain α. The bounds on α are given in Appendix C and are the same as those of Dawid (2003), which are derived by Balke and Pearl (1997) in a deterministic model.

6. Relaxing the Randomization Assumption in the Instrumental Variable Model

It is possible for treatment assignment in a partial compliance study, which is suitable for an IV model, to have invalid randomization, for example, if the doctor involved is aware of the health status of the patients. To analyze such a study, an analyst may opt for a model without randomization or at least assess the effect of the assumption on the inference. Both require constraints to be derived for a model without $U \perp\!\!\!\perp A$, as in Figure 3 (left). The decision framework is used here for computations without any assumptions of determinism. It is irrelevant to the computation whether U causes A, A causes U or both have a common cause. This is because the model in Figure 3 (left) only makes assumptions about distributions in the observational regime and the regime with intervention on B, since it includes the regime indicator F_B. No F_A or F_U regime indicators are included so no assumptions are made about interventions on A or U.

If there is data on $\mathrm{P}(C\,|\,A)$ and $\mathrm{P}(B\,|\,A)$ but not $\mathrm{P}(C\,|\,B,A)$ then constraints and bounds involving $\vec{\gamma}$ and $\vec{\theta}$ are useful. Without the randomization assumption, $U \perp\!\!\!\perp A$, $(\vec{\gamma},\vec{\theta},\alpha)$ does not necessarily lie in the convex hull of $(\vec{\gamma}^*,\vec{\theta}^*,\alpha^*)$ and similarly for the other applications in §5. Assuming the exclusion restriction in Equation (3) still holds, $\vec{\tau}^*$ still fully parameterises $\mathrm{P}(C,B\,|\,A,U)$. Consider the monotone mapping $\Xi_i(\cdot)$: $\vec{\tau}^* \longmapsto \vec{v}_i^*$, where $\vec{v}_i^* = (\gamma_{0i}^*,\gamma_{1i}^*,\theta_{0i}^*,\theta_{1i}^*,\alpha^*)$ for $i=1,2$, which can be expressed as

$$\begin{aligned}\alpha^* = \eta_1^* - \eta_0^*, \qquad \gamma_{0i}^* &= (1-\eta_0^*)(1-\delta_i^*) + (1-\eta_1^*)\delta_i^*, \qquad & \theta_{0i}^* &= 1-\delta_i^* \\ \gamma_{1i}^* &= \eta_0^*(1-\delta_i^*) + \eta_1^*\delta_i^*, & \theta_{1i}^* &= \delta_i^*.\end{aligned}$$

The transformation of $\hat{\mathcal{T}}$ by $\Xi_i(\cdot)$ is given in Figure 4. Since the relations $C \perp\!\!\!\perp F_B\,|\,(B,U), U \perp\!\!\!\perp F_B$ and $C \perp\!\!\!\perp A\,|\,(B,U)$ follow from Figure 3 and $C \perp\!\!\!\perp B\,|\,(U,F_B=B)$,

η_0^*	η_1^*	δ_i^*		γ_{0i}^*	γ_{1i}^*	θ_{0i}^*	θ_{1i}^*	α^*
0	0	0		1	0	1	0	0
0	0	1		1	0	0	1	0
0	1	0		1	0	1	0	1
0	1	1	→	0	1	0	1	1
1	0	0		0	1	1	0	-1
1	0	1		1	0	0	1	-1
1	1	0		0	1	1	0	0
1	1	1		0	1	0	1	0

Figure 4: Transformation of $\hat{\mathcal{T}}$ by $\Xi_i(\cdot)$ to the polytope which represents the IV model without randomization, in terms of the pairwise conditional distributions $\mathrm{P}(C\,|\,A)$ and $\mathrm{P}(B\,|\,A)$.

$$\mathrm{P}(C\,||\,B) = \textstyle\sum_A \sum_U \mathrm{P}(C\,|\,B,U)\mathrm{P}(U\,|\,A)\mathrm{P}(A) = \mathrm{E}_A(\alpha'_A), \tag{5}$$

where $\alpha'_A = \sum_U \mathrm{P}(C\,|\,B,U)\mathrm{P}(U\,|\,A)$. Since $\mathrm{P}(C\,|\,A) = \mathrm{E}_{U|A}\{\mathrm{P}(C\,|\,A,U)\}$ and $\mathrm{P}(B\,|\,A) = \mathrm{E}_{U|A}\{\mathrm{P}(B\,|\,A,U)\}$ then $\vec{w}_i$ lies in the convex hull of the set of $\vec{v}_i^*$, where $\vec{w}_i = (\gamma_{0i},\gamma_{1i},\theta_{0i},\theta_{1i},\alpha'_i)$, and the method of §2 computes the tight constraints

$$\begin{aligned}0 &\le \gamma_{0i} + 2\gamma_{1i} - \theta_{0i} + \alpha'_i,\\ 0 &\le \gamma_{0i} + \theta_{0i} + \alpha'_i,\\ 0 &\le \gamma_{1i} + \theta_{0i} - \alpha'_i,\\ 0 &\le 2\gamma_{0i} + \gamma_{1i} - \theta_{0i} - \alpha'_i,\end{aligned}$$

or

$$\max\{\ \gamma_{0i}+\theta_{0i}-2, -\gamma_{0i}-\theta_{0i}\ \} \le \alpha'_i \le \min\{\ -\gamma_{0i}+\theta_{0i}+1, \gamma_{0i}-\theta_{0i}+1\ \},$$

for all i. The constraints are tight since the vertices of the convex hull are a subset of the vertices of the transformed polytope and any vertex is achievable if the value of U, corresponding to the vertex, occurs with probability one. Since $\alpha = \mathrm{E}_A(\alpha'_A)$ from Equation (5) then

$$\min_i \left[\max \left\{ \begin{array}{c} \gamma_{0i} + \theta_{0i} - 2 \\ -\gamma_{0i} - \theta_{0i} \end{array} \right\} \right] \leq \alpha \leq \max_i \left[\min \left\{ \begin{array}{c} -\gamma_{0i} + \theta_{0i} + 1 \\ \gamma_{0i} - \theta_{0i} + 1 \end{array} \right\} \right].$$

These bounds always span zero and are tight since the bounds on α'_i are achievable by α if $\mathrm{P}(A = i) = 1$. If marginal A data are available, the bounds can be improved to

$$\mathrm{ACE}(B \to C) \geq \textstyle\sum_i \left[\max \left\{ \begin{array}{c} \gamma_{0i} + \theta_{0i} - 2 \\ -\gamma_{0i} - \theta_{0i} \end{array} \right\} \mathrm{P}(A = i) \right],$$

$$\mathrm{ACE}(B \to C) \leq \textstyle\sum_i \left[\min \left\{ \begin{array}{c} -\gamma_{0i} + \theta_{0i} + 1 \\ \gamma_{0i} - \theta_{0i} + 1 \end{array} \right\} \mathrm{P}(A = i) \right],$$

or

$$-1 + \mathrm{E}_A(|\gamma_{1A} - \theta_{0A}|) \leq \mathrm{ACE}(B \to C) \leq 1 - \mathrm{E}_A(|\gamma_{0A} - \theta_{0A}|). \tag{6}$$

Although the expression in (6) bounds the unobservable causal effect, there are no falsifiable constraints to invalidate the model. The bounds in (6) always span zero.

If a sample from $\mathrm{P}(C, B \,|\, A)$ is available, the mapping $\vec{\tau}^* \longmapsto \vec{v}_i^*$ can be used to compute observable constraints and causal bounds. The computation is possible since $\mathrm{P}(C, B \,|\, A) = \mathrm{E}_{U|A}\{\mathrm{P}(C, B \,|\, A, U)\}$, which implies that $\vec{w}_i$ lies in the convex hull of the set of $\vec{v}_i^*$, where $\vec{v}_i^* = (\zeta^*_{00.i}, \zeta^*_{01.i}, \zeta^*_{10.i}, \zeta^*_{11.i}, \alpha^*)$ and $\vec{w}_i = (\zeta_{00.i}, \zeta_{01.i}, \zeta_{10.i}, \zeta_{11.i}, \alpha'_i)$. The bounds $-\zeta_{01} - \zeta_{10} \leq \mathrm{ACE}(B \to C) \leq \zeta_{00} + \zeta_{11}$ are obtained, where $\zeta_{cb} = \mathrm{P}(C = c, B = b)$. All of the results in this section still hold if the state space of A is extended to $\{1, 2, \ldots, l\}$ but the state space of (B, C) kept binary.

The bounds on $\mathrm{ACE}(B \to C)$ by the ζ_{cb} parameters are derived by Manski (1990) in a model involving (B, C, C_0, C_1) under the assumptions that the potential outcomes (C_0, C_1) for a unit are the same regardless of how treatment is assigned, that is, whether by intervention or observation, and thus C is a deterministic function of (B, C_0, C_1) for a unit. The derivation, of the bounds on $\mathrm{ACE}(B \to C)$ by the ζ_{cb} parameters, given here only requires the analogous assumptions $U \perp\!\!\!\perp F_B$ and $C \perp\!\!\!\perp F_B \,|\, (B, U)$. The additional variable A used here, which satisfies the condition $C \perp\!\!\!\perp A \,|\, (B, U)$, trivially exists by constructing a variable $A = B$. Also, the conditional independence assumption $A \perp\!\!\!\perp F_B$ represented in Figure 3 (left) is unnecessary since it is not used in the derivation.

7. Relaxing the Exclusion Restriction in the Instrumental Variable Model

The exclusion restriction assumption may often be inapplicable, for example, if patients in a study with partial compliance become aware of their treatment assignment and this affects their outcome. There could be a direct relation between treatment assignment A and the outcome C, for which the model in Figure 3 (right) would be appropriate. The probabilistic nature of the exclusion restriction within the decision framework allows the strength of the direct relation to be varied and the sensitivity of inference to this assumption to be assessed.

A weaker alternative to the exclusion restriction assumption, $C \perp\!\!\!\perp A \,|\, (B, U)$, in the binary IV model is $0 \leq |\eta^*_{b1} - \eta^*_{b2}| \leq \varepsilon$ for $b = 0, 1$, where $\eta^*_{ba} = \mathrm{P}(C = 1 \,|\, B = b, A = a, U)$ and $0 \leq \varepsilon \leq 1$.

η^*_{01}	η^*_{02}	η^*_{11}	η^*_{12}	δ^*_1	δ^*_2
0	0	0	0	0	0
0	0	0	0.5	0	1
0	0	0.5	0	1	0
0	0	0.5	1	1	1
0	0	1	0.5	0	0
0	0	1	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮

↓

$\zeta^*_{00.1}$	$\zeta^*_{01.1}$	$\zeta^*_{10.1}$	$\zeta^*_{11.1}$	$\zeta^*_{00.2}$	$\zeta^*_{01.2}$	$\zeta^*_{10.2}$	$\zeta^*_{11.2}$
1	0	0	0	1	0	0	0
1	0	0	0	0	0.5	0	0.5
0	0.5	0	0.5	1	0	0	0
0	0.5	0	0.5	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 5: Transformation to the extreme vertices corresponding to the polytope which represents the IV model with the weaker exclusion restriction, for $\varepsilon = 0.5$, in terms of the distribution $\mathrm{P}(C,B\,|\,A)$.

The condition $\varepsilon = 0$ is equivalent to the exclusion restriction. For $\varepsilon = 1$, there are no constraints on $(\eta^*_{b1}, \eta^*_{b2})$ other than the axioms of probability and there are no falsifiable constraints or causal bounds for the IV model without the exclusion restriction. The augmented DAG in Figure 3 (right) does not represent any assumptions about ε but assumptions about ε are required to obtain non-trivial constraints and bounds. The application of the technique is considered for $\varepsilon = 0.5$. Consider the mapping of $\vec{\tau}^* = (\eta^*_{01}, \eta^*_{02}, \eta^*_{11}, \eta^*_{12}, \delta^*_1, \delta^*_2)$ to $\vec{v}^* = (\zeta^*_{00.1}, \zeta^*_{01.1}, \ldots, \zeta^*_{11.2})$ for a model with $A \in \{1,2\}$ and $B, C \in \{0,1\}$. The transformation of some of the extreme vertices are given in Figure 5. Use of the technique produces the causal bounds in Appendix D and the constraints

$$\begin{aligned}
\zeta_{00.1} + \zeta_{10.2} - \zeta_{10.1} - \zeta_{00.2} &\leq 1,\\
\zeta_{10.1} + \zeta_{00.2} - \zeta_{00.1} - \zeta_{10.2} &\leq 1,\\
\zeta_{11.1} + \zeta_{01.2} - \zeta_{01.1} - \zeta_{11.2} &\leq 1,\\
\zeta_{01.1} + \zeta_{11.2} - \zeta_{11.1} - \zeta_{01.2} &\leq 1,
\end{aligned}$$

which is a weaker version of the 'instrumental inequality' of Equation (1) and can be violated if the IV model with the weak exclusion restriction, $\varepsilon = 0.5$, is invalid. By adding the component $\mathrm{P}(C\,|\,B,A,U)$ to $\vec{v}^*$, causal bounds on $\mathrm{P}(C\,|\,A, F_B = B) = \sum_U \mathrm{P}(C\,|\,B,A,U)\mathrm{P}(U)$ can be derived for each A and used to compute bounds on $\mathrm{ACE}(B \to C)$ since $\mathrm{P}(C\,|\,F_B = B) = \sum_A \mathrm{P}(C\,|\,A, F_B = B)\mathrm{P}(A)$. Similarly to the bounds in §6, these bounds are tight. Although ε is currently defined as a constant, similar computations can be done if ε is allowed to be a function of b, that is, $|\eta^*_{b1} - \eta^*_{b2}|$ has a different range for each b.

7.1 Bounds on Direct Effects

Without assuming $C \perp\!\!\!\perp A \,|\, (B,U)$, if intervention on A is possible then the direct effect of A on C can be bounded with parameters of the distribution under no intervention. Consider extending the sample space of F_B to include the random regime d_A, which represents the regime in which $\mathrm{P}(B \,|\, A,U,F_B = d_{a^*}) = \mathrm{P}(B \,|\, A = a^*,U)$. Consider the controlled direct effect (CDE) (Didelez et al., 2006) and the random regime direct effect (RRDE)

$$\begin{aligned}
\mathrm{CDE}(B) &= \mathrm{E}(C \,|\, F_B = B, F_A = 2) - \mathrm{E}(C \,|\, F_B = B, F_A = 1) \\
&= \mathrm{E}_U\{\mathrm{E}(C \,|\, B, A = 2, U) - \mathrm{E}(C \,|\, B, A = 1, U)\} \\
&= \mathrm{E}_U(\eta^*_{B2} - \eta^*_{B1}),
\end{aligned}$$

$$\begin{aligned}
\mathrm{RRDE}(a^*) &= \mathrm{E}(C \,|\, F_B = d_{a^*}, F_A = 2) - \mathrm{E}(C \,|\, F_B = d_{a^*}, F_A = 1) \\
&= \mathrm{E}_U[\mathrm{E}_B\{\mathrm{E}(C \,|\, B, A = 2, U) - \mathrm{E}(C \,|\, B, A = 1, U) \,|\, A = a^*, U\}] \\
&= \mathrm{E}_U\{(\eta^*_{12} - \eta^*_{11})\delta^*_{a^*} + (\eta^*_{12} - \eta^*_{11})(1 - \delta^*_{a^*})\}.
\end{aligned}$$

The RRDE is called the NDE in Didelez et al. (2006) but Robins and Richardson (2010) argue that the parameter being referred to as NDE in Didelez et al. (2006) is not the same as the NDE in Pearl (2001). Thus a separate name is given here to RRDE. By considering the mapping of $\vec{\tau}^*$ to the vector with $\vec{v}^*$ and the extra component $\eta^*_{B2} - \eta^*_{B1}$, the bounds on $\mathrm{CDE}(B)$ of Cai et al. (2008) can be replicated. Similarly by mapping $\vec{\tau}^*$ to a vector with $\vec{v}^*$ and $\mathrm{E}_B(\eta^*_{B2} - \eta^*_{B1} \,|\, A = a^*, U)$, bounds on $\mathrm{RRDE}(a^*)$ are obtained, which are identical to the bounds on $\mathrm{NDE}(a^*)$ in Sjölander (2009). Unlike here, both references use counterfactuals and use the definition of CDE and NDE, sometimes called pure direct effect (Robins and Greenland, 1992), given in Pearl (2001).

8. Monotonicity Assumption in the Instrumental Variable Model

The monotonicity assumption in the literature (Imbens and Angrist, 1994; Angrist et al., 1996) is formulated in a deterministic model. In a partial compliance study, a patient may be more likely to take treatment under assignment but it may not be reasonable to assume that their behaviour is deterministically related to treatment assignment. A monotonicity assumption in a weaker probabilistic model is considered here and can be expressed mathematically for the binary IV model by $\delta^*_2 \geq \delta^*_1$, from Equation (2). It restricts the space of the vector of probabilities so the constraints are at least as strong as without it.

The IV model considered in this section includes the exclusion restriction and the randomization assumption, as in the augmented DAG in Figure 2 (right). As an illustrative example, consider the computation of falsifiable constraints and causal bounds on φ given $\vec{\gamma}$, without monotonicity, where $\varphi = \mathrm{ACE}(A \to B) = \theta_{01} - \theta_{02}$. This computation produces only trivial results. Under monotonicity, $\mathcal{T}$ must be redefined to omit all $\vec{\tau}^*$ which do not satisfy it. Therefore all of the vertices with $\delta^*_2 < \delta^*_1$ or $\varphi^* \geq 0$ should be removed to redefine $\hat{\mathcal{T}}$, where $\varphi^* = \mathrm{P}(B = 1 \,|\, A = 2, U) - \mathrm{P}(B = 1 \,|\, A = 1, U)$. The required mapping is $\vec{\tau}^* \longmapsto (\vec{\gamma}^*, \varphi^*)$ over the domain of the restricted $\mathcal{T}$. The transformation is given in Figure 6 and the non-trivial constraints obtained are

$$\max \left\{ \begin{array}{c} \gamma_{01} - \gamma_{02} \\ -\gamma_{01} + \gamma_{02} \end{array} \right\} \leq \varphi \leq 1 \quad \Longleftrightarrow \quad |\gamma_{01} - \gamma_{02}| \leq \varphi \leq 1, \tag{7}$$

or $\mathrm{ACE}(A \to B) \geq |\gamma_{01} - \gamma_{02}| = |\mathrm{ACE}(A \to C)|$. This makes sense since it is assumed that $\varphi^* \geq 0$ and B lies on the causal pathway from A to C. The transformed polytope in Figure 6 is 3 dimensional

η_0^*	η_1^*	δ_1^*	δ_2^*		γ_{01}^*	γ_{11}^*	γ_{02}^*	γ_{12}^*	φ^*
0	0	0	0		1	0	1	0	0
0	0	0	1		1	0	1	0	1
0	0	1	0		–	–	–	–	–
0	0	1	1		1	0	1	0	0
0	1	0	0		1	0	1	0	0
0	1	0	1		1	0	0	1	1
0	1	1	0		–	–	–	–	–
0	1	1	1	$\rightarrow$	0	1	0	1	0
1	0	0	0		0	1	0	1	0
1	0	0	1		0	1	1	0	1
1	0	1	0		–	–	–	–	–
1	0	1	1		1	0	1	0	0
1	1	0	0		0	1	0	1	0
1	1	0	1		0	1	0	1	1
1	1	1	0		–	–	–	–	–
1	1	1	1		0	1	0	1	0

Figure 6: Transformation to the extreme vertices corresponding to the polytope which represents the IV model, with the exclusion restriction, randomization and monotonicity assumptions, in terms of the distribution $\mathrm{P}(C\,|\,A)$. The dashes correspond to points ruled out by monotonicity, which are represented by $\circ$'s in Figure 7.

since $\gamma_{11}^* = 1 - \gamma_{01}^*$ and $\gamma_{12}^* = 1 - \gamma_{02}^*$; its projection in $(\gamma_{01}^*, \gamma_{02}^*, \varphi^*)$ space is given in Figure 7. The 6 $\bullet$'s are the vertices which are not removed after assuming monotonicity and the 6 $\circ$'s, which correspond to dashes in Figure 6, are the vertices which are removed. Figure 7 clearly demonstrates that the constraints without monotonicity are trivial whereas those with it are not.

To determine the effect of the monotonicity assumption on the constraints and bounds with $(\vec{\gamma}, \vec{\theta})$ in §5, the same mapping is used as in the derivation of the bivariate bounds on α but applied to the restricted $\mathcal{T}$ and $\hat{\mathcal{T}}$ formed by removing the appropriate vertices. The falsifiable constraints obtained are $\theta_{01} - \theta_{02} \geq |\gamma_{01} - \gamma_{02}|$ (equivalent to Equation (7)) and the causal bounds

$$\mathrm{ACE}(B \rightarrow C) \geq \max \left\{ \begin{array}{c} 2\gamma_{01} - \gamma_{02} + \theta_{01} - 2 \\ \gamma_{01} - 2\gamma_{02} - \theta_{02} \\ \gamma_{01} + \theta_{01} - 2 \\ -\gamma_{02} - \theta_{02} \\ \gamma_{01} - \gamma_{02} + \theta_{01} - \theta_{02} - 1 \end{array} \right\},$$

$$\mathrm{ACE}(B \rightarrow C) \leq \min \left\{ \begin{array}{c} 2\gamma_{01} - \gamma_{02} - \theta_{01} + 1 \\ \gamma_{01} - 2\gamma_{02} + \theta_{02} + 1 \\ \gamma_{01} - \theta_{01} + 1 \\ -\gamma_{02} + \theta_{02} + 1 \\ \gamma_{01} - \gamma_{02} - \theta_{01} + \theta_{02} + 1 \end{array} \right\}.$$

By considering the analogous mapping for $\vec{v}^*$, as defined in §2, the falsifiable constraints of Balke and Pearl (1997) and the causal bounds of Balke and Pearl (1993) for the IV model with mono-

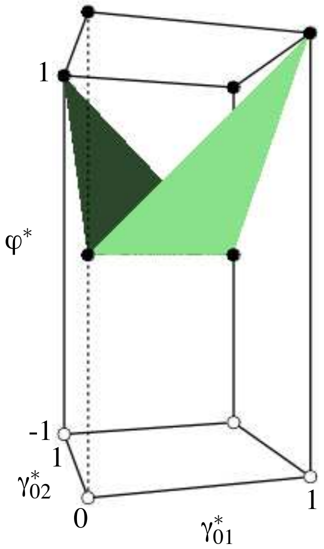


Figure 7: The extreme vertices which satisfy monotonicity are the 6 •'s, the 6 ○'s are those which do not. The convex hull of the transformed polytope for the IV model with the monotonicity assumption is the region above the shaded surface and without the monotonicity assumption is the entire cuboid.

tonicity are recovered. The bounds correspond to results in Robins (1989) and Manski (1990). Thus the IV model with monotonicity, introduced in this section, is empirically and computationally indistinguishable from the IV model with 'no defiers' considered in Example 1 and Angrist et al. (1996).

9. Data Analysis

The relative frequencies for two data sets are given in Table 1 and described below.

9.1 Lipid Research Clinics Coronary Data

Consider the Lipid Research Coronary Primary Prevention Trial (Lipid Research Clinic Program, 1984), which was analysed by Efron and Feldman (1991) and Balke and Pearl (1997). Subjects were randomized into two groups, 172 men were given the placebo and 165 were given the treatment, and the subjects' cholesterol levels were measured. There was partial compliance with the treatment assigned.

9.2 Vitamin A Supplementation

Another example of partial compliance is the study of Vitamin A supplementation in northern Sumatra, described by Sommer and Zeger (1991). The study consisted of children in 450 villages, 11588 children (221 villages) were assigned to the control group and 12094 (229 villages) to the treatment group.

Data Set	a	$\hat{\zeta}_{00.a}$	$\hat{\zeta}_{01.a}$	$\hat{\zeta}_{10.a}$	$\hat{\zeta}_{11.a}$
Lipid Research Clinic Program	1	0.919	0	0.081	0
	2	0.315	0.139	0.073	0.473
Vitamin A Supplementation	1	0.0064	0	0.9936	0
	2	0.0028	0.0010	0.1972	0.7990

Table 1: Relative frequencies derived from the data sets in Lipid Research Clinic Program (1984) and Sommer and Zeger (1991).

In these trials, the relative frequencies are the maximum likelihood estimates of the parameters $\vec{v}$. Bounds on ACE$(B \to C)$ are computed under various assumptions from the data in Table 1 and are given in Table 2. Sampling uncertainty is ignored here but can be properly considered using techniques described in Ramsahai and Lauritzen (2011).

Study	Assumptions	Lower bound	Upper bound
Lipid Research Clinic Program	IV model	0.392	0.780
	IV, no randomization	-0.145	0.855
	IV, partial exclusion restriction ($\varepsilon = 0.5$)	0.050	0.855
	IV, monotonicity	0.392	0.780
Vitamin A Supplement.	IV model	-0.1946	0.0054
	IV, no randomization	-0.587	0.413
	IV, partial exclusion restriction ($\varepsilon = 0.5$)	-0.392	0.212
	IV, monotonicity	-0.1946	0.0054

Table 2: Causal bounds on ACE$(B \to C)$ computed from Lipid Research Clinic Program (1984) and Vitamin A Supplementation Study under various assumptions.

From Table 2, the imposition of the monotonicity assumption has no effect and is unnecessary for these data sets. However the randomized treatment assignment is important since the bounds computed without randomization are very wide and not much can be inferred about the causal effect. Even though the bounds are much wider for the Lipid Research Clinic Program (1984) data, under the partial exclusion restriction with $\varepsilon = 0.5$, it can still be deduced that there is a positive causal effect.

10. Discussion

The methods given here are applied while relaxing various assumptions that are often used in the deterministic counterfactual IV model. By removing the assumption that there are latent deterministic mechanisms, it is shown that the same bounds and constraints are obtained and that the models are empirically equivalent §3. The results for models which relax the randomization and exclusion restriction assumptions are valuable for sensitivity analyses. They are also useful for applications in which some of the assumptions in the IV model are known to be false.

In §7, the constraints and bounds were computed for the IV model with a partial exclusion restriction for $\varepsilon = 0.5$. It is not obvious how the methods described in this paper can be extended to compute bounds and constraints as a function of ε but that would be worthy of future investigation. Such results would show how the bounds vary with ε and whether the data places any restrictions on the possible values of ε.

The ideas discussed can be extended to other models involving conditional independence since it is the factorization of the probability distribution which determines the algebraic structure of the polytope representing the model. The model must satisfy the condition that the observable distributions lie in the convex hull of the latent distribution. The vector of parameters $\mathrm{P}(\mathcal{X})$ always lies in the convex hull of $\mathrm{P}(\mathcal{X}\,|\,U)$ but there is no guarantee that the factorisation of $\mathrm{P}(\mathcal{X}\,|\,U)$ produces any non-trivial constraints, where $\mathcal{X}$ and U are collections of observed and unobserved variables respectively. However there may be non-trivial constraints on conditional probabilities derived from $\mathrm{P}(\mathcal{X})$.

Acknowledgments

Steffen Lauritzen has been very helpful with this research. Detailed comments from anonymous referees have helped improve the paper.

Appendix A. Notation

The symbols used throughout are listed below.

$$\begin{aligned}
&\zeta^*_{cb.a} = \mathrm{P}(C=c, B=b\,|\,A=a, U),\\
&\eta^*_b = \mathrm{P}(C=1\,|\,B=b, U),\\
&\delta^*_a = \mathrm{P}(B=1\,|\,A=a, U),\\
&\alpha = \mathrm{P}(C=1\,||\,B=1) - \mathrm{P}(C=1\,||\,B=0),\\
&\alpha^* = \mathrm{P}(C=1\,|\,B=1, U) - \mathrm{P}(C=1\,|\,B=0, U),\\
&\gamma^*_{ca} = \mathrm{P}(C=c\,|\,A=a, U),\\
&\theta^*_{ba} = \mathrm{P}(B=b\,|\,A=a, U).
\end{aligned}$$

The symbols with * are functions of U and the corresponding symbols without * are the marginals over U.

Appendix B. Equivalence of Convex Hulls

Theorem 1 $\mathcal{H} = \hat{\mathcal{H}}$.

Proof Following Dawid (2003), since $\hat{\mathcal{V}} \subseteq \mathcal{V}$ and $\hat{\mathcal{H}}$ is the minimal convex set containing $\hat{\mathcal{V}}$ then $\hat{\mathcal{H}} \subseteq \mathcal{H}$.

Let $m(\vec{v}^*)$ be an affine function, that is a linear function plus a constant, of $\vec{v}^*$, which returns a scalar, for $\vec{v}^* \in \mathcal{V}$. A closed half space in $[0,1]^8$ that contains $\vec{v}$ is defined by an affine function inequality $m(\vec{v}^*) \geq 0$ or $m\{\Xi(\vec{\tau}^*)\} \geq 0$ for $\vec{\tau}^* \in \mathcal{T}$. From Equations (3) and (2), $m\{\Xi(\vec{\tau}^*)\}$ is a monotonic function of any single component of $\vec{\tau}^*$ when the other three are fixed. Therefore the minimum of $m\{\Xi(\vec{\tau}^*)\}$ over $\mathcal{T}$ is attained for some $\vec{\tau}^* \in \hat{\mathcal{T}}$. Therefore

$$m\{\Xi(\vec{\tau}^*)\} \geq 0 \text{ for all } \vec{\tau}^* \in \hat{\mathcal{T}} \Rightarrow\ m\{\Xi(\vec{\tau}^*)\} \geq 0 \text{ for all } \vec{\tau}^* \in \mathcal{T}.$$

This means that any half space containing $\hat{\mathcal{V}}$ also contains $\mathcal{V}$. Since $\hat{\mathcal{H}}$ is the intersection of all half spaces containing $\hat{\mathcal{V}}$ then $\mathcal{V} \subseteq \hat{\mathcal{H}}$. Since $\hat{\mathcal{H}}$ is convex and $\mathcal{H}$ is the minimal convex set containing $\mathcal{V}$ then $\mathcal{H} \subseteq \hat{\mathcal{H}}$. ■

Appendix C. Causal Bounds for Binary Instrumental Variable Model

For the binary IV model of Figure 2 (right), bounds on α in terms of $\zeta_{cb.a}$ are

$$\alpha \geq \max \left\{ \begin{array}{c} \zeta_{00.1} + \zeta_{11.2} - 1 \\ \zeta_{11.1} + \zeta_{00.2} - 1 \\ -\zeta_{01.1} - \zeta_{10.1} + \zeta_{11.1} - \zeta_{10.2} - \zeta_{11.2} \\ -\zeta_{10.1} - \zeta_{11.1} - \zeta_{01.2} - \zeta_{10.2} + \zeta_{11.2} \\ -\zeta_{01.1} - \zeta_{10.1} \\ -\zeta_{01.2} - \zeta_{10.2} \\ -\zeta_{00.1} - \zeta_{01.1} + \zeta_{00.2} - \zeta_{01.2} - \zeta_{10.2} \\ \zeta_{00.1} - \zeta_{01.1} - \zeta_{10.1} - \zeta_{00.2} - \zeta_{01.2} \end{array} \right\},$$

and

$$\alpha \leq \min \left\{ \begin{array}{c} 1 - \zeta_{10.1} - \zeta_{01.2} \\ 1 - \zeta_{01.1} - \zeta_{10.2} \\ \zeta_{00.1} - \zeta_{01.1} + \zeta_{11.1} + \zeta_{00.2} + \zeta_{01.2} \\ \zeta_{00.1} + \zeta_{01.1} - \zeta_{01.2} + \zeta_{00.2} + \zeta_{11.2} \\ \zeta_{00.1} + \zeta_{11.1} \\ \zeta_{00.2} + \zeta_{11.2} \\ \zeta_{10.1} + \zeta_{11.1} + \zeta_{00.2} + \zeta_{11.2} - \zeta_{10.2} \\ \zeta_{00.1} - \zeta_{10.1} + \zeta_{11.1} + \zeta_{10.2} + \zeta_{11.2} \end{array} \right\}.$$

For the binary IV model of Figure 2 (right), bounds on α in terms of γ_{ca} and θ_{ba} are

$$\alpha \geq \max \left\{ \begin{array}{c} 2\gamma_{01} - \gamma_{02} + 2\theta_{01} - 3 \\ \gamma_{01} + \theta_{01} - 2 \\ \gamma_{02} + \theta_{02} - 2 \\ -\gamma_{01} + 2\gamma_{02} + 2\theta_{02} - 3 \\ -\gamma_{01} + \gamma_{02} - \theta_{01} + \theta_{02} - 1 \\ -\gamma_{01} - \theta_{01} \\ -\gamma_{02} - \theta_{02} \\ \gamma_{01} - 2\gamma_{02} - 2\theta_{02} \\ -2\gamma_{01} + \gamma_{02} - 2\theta_{01} \\ \gamma_{01} - \gamma_{02} + \theta_{01} - \theta_{02} - 1 \end{array} \right\},$$

and

$$\alpha \leq \min \left\{ \begin{array}{c} -2\gamma_{01} + \gamma_{02} + 2\theta_{01} + 1 \\ \gamma_{01} - 2\gamma_{02} + 2\theta_{02} + 1 \\ 2\gamma_{01} - \gamma_{02} - 2\theta_{01} + 2 \\ -\gamma_{01} + 2\gamma_{02} - 2\theta_{02} + 2 \\ \gamma_{01} - \gamma_{02} - \theta_{01} + \theta_{02} + 1 \\ -\gamma_{02} + \theta_{02} + 1 \\ \gamma_{01} - \theta_{01} + 1 \\ \gamma_{02} - \theta_{02} + 1 \\ -\gamma_{01} + \theta_{01} + 1 \\ -\gamma_{01} + \gamma_{02} + \theta_{01} - \theta_{02} + 1 \end{array} \right\}.$$

Appendix D. Causal Bounds for Instrumental Variable Model Without Exclusion Restriction

For the binary IV model without the exclusion restriction in §7, for $\varepsilon = 0.5$, the following bounds are obtained for $a = 1, 2$

$$\begin{array}{c} 2\{\mathrm{E}(C\,|\,A = a, F_B = 1) \\ -\mathrm{E}(C\,|\,A = a, F_B = 0)\} \end{array} \geq \max \left\{ \begin{array}{l} -2\zeta_{01.a} - 2\zeta_{10.a} \\ -\zeta_{01.a} - 2\zeta_{10.a} + \zeta_{11.a} + \zeta_{00.a'} - \zeta_{10.a'} - 1 \\ 2\zeta_{00.a} - 2 - \zeta_{01.a'} + \zeta_{11.a'} \\ -3\zeta_{01.a} - 2\zeta_{10.a} - \zeta_{11.a} - \zeta_{01.a'} + \zeta_{11.a'} \\ -3\zeta_{01.a} - 2\zeta_{10.a} + \zeta_{11.a} - 2\zeta_{10.a'} - 2\zeta_{11.a'} \\ -\zeta_{01.a} - 2\zeta_{10.a} - 3\zeta_{11.a} - 3\zeta_{01.a'} - 2\zeta_{10.a'} + \zeta_{11.a'} \\ 2\zeta_{11.a} + \zeta_{00.a'} - \zeta_{10.a'} - 2 \\ -2\zeta_{01.a'} - 2\zeta_{10.a'} - 1 \\ -3\zeta_{01.a} - 4\zeta_{10.a} - \zeta_{11.a} + 2\zeta_{10.a'} + 2\zeta_{11.a'} - 1 \\ \zeta_{01.a} + 2\zeta_{10.a} + 3\zeta_{11.a} - 3\zeta_{01.a'} - 4\zeta_{10.a'} - \zeta_{11.a'} - 2 \end{array} \right\},$$

and

$$\begin{array}{c} 2\{\mathrm{E}(C\,|\,A = a, F_B = 1) \\ -\mathrm{E}(C\,|\,A = a, F_B = 0)\} \end{array} \leq \min \left\{ \begin{array}{l} 2\zeta_{00.a} + 2\zeta_{11.a} \\ 2 - 2\zeta_{10.a} - \zeta_{01.a'} + \zeta_{11.a'} \\ 2 - \zeta_{01.a} - 2\zeta_{10.a} + \zeta_{11.a} - \zeta_{01.a'} + \zeta_{11.a'} \\ 1 + 2\zeta_{00.a'} + 2\zeta_{11.a'} \\ 2 - 2\zeta_{01.a} + \zeta_{00.a'} - \zeta_{10.a'} \\ 3\zeta_{00.a} - \zeta_{10.a} + 2\zeta_{11.a} + 2\zeta_{10.a'} + 2\zeta_{11.a'} \\ 3 - 3\zeta_{01.a} - 2\zeta_{10.a} - \zeta_{11.a} + \zeta_{00.a'} - \zeta_{10.a'} \\ 4 - 3\zeta_{01.a} - 2\zeta_{10.a} + \zeta_{11.a} - 2\zeta_{10.a'} - 2\zeta_{11.a'} \\ 4 + \zeta_{01.a} - 2\zeta_{10.a} - \zeta_{11.a} - 3\zeta_{01.a'} - 2\zeta_{10.a'} + \zeta_{11.a'} \\ 4 - \zeta_{01.a} + 2\zeta_{10.a} + \zeta_{11.a} - 3\zeta_{01.a'} - 4\zeta_{10.a'} - \zeta_{11.a'} \end{array} \right\}.$$

where $a' = 2$ if $a = 1$ and $a' = 1$ if $a = 2$.

References

O. O. Aalen and A. Frigessi. What can statistics contribute to a causal understanding? *Scandinavian Journal of Statistics*, 34(1):155–168, 2007.

J. D. Angrist, G. W. Imbens and D. B. Rubin. Identification of causal effects using instrumental variables. *J. Am. Statist. Assoc.*, 91(434):444–455, 1996.

A. Balke. Probabilistic counterfactuals: semantics, computation and applications. PhD Dissertation, University of California, Los Angeles, 1995.

A. Balke and J. Pearl. Non-parametric bounds on causal effects from partial compliance data. Technical Report R-199, Computer Science Department, University of California, Los Angeles, 1993.

A. Balke and J. Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1171–1176, 1997.

Z. Cai, M. Kuroki, J. Pearl and J. Tian. Bounds on direct effects in the presence of confounded intermediate variables. *Biometrics*, 64(3):695–701, 2008.

T. Christof and A. Loebel. PORTA. Available online at URL: http://www.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/, 1998.

A. P. Dawid. Conditional independence in statistical theory (with discussion). *Journal of the Royal Statisitical Society*, Ser. B, 41(1):1–31, 1979.

A. P. Dawid. Causal inference without counterfactuals (with discussion). *Journal of the American Statistical Association*, 95(450):407–448, 2000.

A. P. Dawid. Influence diagrams for causal modelling and inference. *International Statistical Review*, 70(2):161–189, 2002.

A. P. Dawid. Causal inference using influence diagrams: the problem of partial compliance. In P. J. Green, N. L. Hjort and S. Richardson, editors, *Highly Structured Stochastic Systems*, Oxford University Press, New York, 2003.

A. P. Dawid and V. Didelez. Identifying the consequences of dynamic treatment strategies: a decision theoretic overview. *Statistics Surveys*, 4:184–231, 2010.

V. Didelez, A. P. Dawid and S. Geneletti. Direct and indirect effects of sequential treatments. In *Proceedings of the Twenty Second Annual Conference on Uncertainty in Artificial Intelligence*, pages 138–146, AUAI Press, Arlington, Virginia, 2006.

V. Didelez and N. Sheehan. Mendelian randomisation as an instrumental variable approach to causal inference. *Statistical Methods in Medical Research*, 16(4):309–330, 2007.

J. Durbin. Errors in variables. *Review of the International Statistical Institute*, 22(1):23–32, 1954.

B. Efron and D. Feldman. Compliance as an explanatory variable in clinical trials. *Journal of the American Statistical Association*, 86(413):9–26, 1991.

E. Gawrilow and M. Joswig. Polymake: a framework for analyzing convex polytopes. In G. Kalai and G. M. Ziegler, editors, *Polytopes - Combinatorics and Computation*, Birkhäuser, Basel, 2000. Available online at URL: http://wwwopt.mathematik.tu-darmstadt.de/polymake/doku.php.

D. Geiger and C. Meek. Graphical models and exponential families. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 156–165, Morgan Kaufmann, San Francisco, California, 1998.

S. Geneletti. Identifying direct and indirect effects in a non-counterfactual framework. *Journal of the Royal Statistical Society*, Ser. B, 69(2):199–215, 2007.

M. Goldszmidt and J. Pearl. Rank based systems: a simple approach to belief revision, belief update, and reasoning about evidence and actions. In *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, California, 1992.

D. Heckerman and R. Shachter. Decision theoretic foundations for causal reasoning. *Journal of Artificial Intelligence Research*, 3:405–430, 1995.

K. Hirano, G. W. Imbens, D. B. Rubin and X.-H. Zhou. Assessing the effect of an influenza vaccine in an encouragement design. *Biostatistics*, 1(1):69–88, 2000.

G. W. Imbens and J. D. Angrist. Identification and estimation of local average treatment effects. *Econometrica*, 62(2):467–475, 1994.

C. Kang and J. Tian. Inequality constraints in causal models with hidden variables. In *Proceedings of the Twenty Second Annual Conference on Uncertainty in Artificial Intelligence*, pages 233–240, AUAI Press, Arlington, Virginia, 2006.

S. Kaufman, J. S. Kaufman and R. F. MacLehose. Analytic bounds on causal risk difference in directed acyclic graphs involving three observed binary variables. *Journal of Statistical Planning and Inference*, 139(10):3473–3487, 2009.

S. L. Lauritzen. *Graphical models*. Oxford University Press, Clarendon, Oxford, UK, 1996.

S. L. Lauritzen. *Causal inference from graphical models*. In O. E. Barndorff-Nielsen, D. R. Cox and C. Klüppelberg, editors, *Complex Stochastic Systems*, CRC Press, London, 2001.

S. L. Lauritzen. Discussion on causality. *Scandinavian Journal of Statistics*, 31(2):189–201, 2004.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen and H. G. Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990.

Lipid Research Clinic Program. The lipid research clinics coronary primary prevention trial results, part I and II. *Journal of the American Medical Association*, 251(3):351–374, 1984.

C. F. Manski. Non-parametric bounds on treatment effects. *American Economic Review, Papers and Proceedings*, 80(2):319–323, 1990.

J. Pearl. Comment: graphical models, causality and interventions. *Statistical Science*, 8(3):266–269, 1993.

J. Pearl. Causal inference from indirect experiments. *Artificial Intelligence in Medicine*, 7(6):561–582, 1995.

J. Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 411–420, Morgan Kaufmann, San Francisco, California, 2001.

R. R. Ramsahai. Causal bounds and instruments. In *Proceedings of the Twenty Third Annual Conference on Uncertainty in Artificial Intelligence*, pages 310–317, AUAI Press, Corvallis, Oregon, 2007.

R. R. Ramsahai and S. L. Lauritzen. Likelihood analysis of the binary instrumental variable model. *Biometrika*, 98(4):987–994, 2011.

T. S. Richardson and J. M. Robins. Analysis of the binary instrumental variable model. In R. Dechter, H. Geffner and J. Y. Halpern, editors, *Heuristics, probability and causality: a tribute to Judea Pearl*, College Publications, UK, 2010.

J. M. Robins. The analysis of randomized and non-randomized AIDS treatment trials using a new approach to causal inference in logitudinal studies. In L. Sechrest, H. Freeman and A. Mulley, editors, *Health Service Research Methodology: a Focus on AIDS*, U.S. Public Health Service, Washington D.C., 1989.

J.M. Robins and S. Greenland. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, 3(2):143–155, 1992.

J. M. Robins and T. S. Richardson. Alternative graphical causal models and the identification of direct effects. In P. Shrout, editor, *Causality and Psychopathology: Finding the Determinants of Disorders and Their Cures*, Oxford University Press, 2010.

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

B. E. Shepherd, P. B. Gilbert, Y. Jemiai and A. Rotnitzky. Sensitivity analyses comparing outcomes only existing in a subset selected post randomization, conditional on covariates, with application to HIV vaccine trial. *Biometrics*, 62(2):332–342, 2006.

A. Sjölander. Bounds on natural direct effects in the presence of confounded intermediate variables. *Statistics in Medicine*, 28(4):558–571, 2009.

A. Sommer and S. L. Zeger. On estimating efficacy from clinical trials. *Statistics in Medicine*, 10(1):45–52, 1991.

P. Spirtes, C. Glymour and R. Scheines. *Causation, Prediction and Search*. Springer-Verlag, New York, 1993.

R. H. Strotz and H. O. A. Wold. Recursive vs non-recursive systems: an attempt at synthesis (part I of a triptych on causal chain systems). *Econometrica*, 28(2):417–427, 1960.

T. Verma and J. Pearl. Causal networks: semantics and expressiveness. In *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pages 352-359, Elsevier Science, New York, 1988.

NIMFA : A Python Library for Nonnegative Matrix Factorization

Marinka Žitnik MARINKA.ZITNIK@STUDENT.UNI-LJ.SI
Blaž Zupan BLAZ.ZUPAN@FRI.UNI-LJ.SI
Faculty of Computer and Information Science
University of Ljubljana
SI-1000 Ljubljana, Tržaška 25, Slovenia

Editor: Mikio Braun

Abstract

NIMFA is an open-source Python library that provides a unified interface to nonnegative matrix factorization algorithms. It includes implementations of state-of-the-art factorization methods, initialization approaches, and quality scoring. It supports both dense and sparse matrix representation. NIMFA's component-based implementation and hierarchical design should help the users to employ already implemented techniques or design and code new strategies for matrix factorization tasks.

Keywords: nonnegative matrix factorization, initialization methods, quality measures, scripting, Python

1. Introduction

As a method to learn parts-based representation, a nonnegative matrix factorization (NMF) has become a popular approach for gaining new insights about complex latent relationships in high-dimensional data through feature construction, selection and clustering. It has recently been successfully applied to many diverse fields such as image and signal processing, bioinformatics, text mining, speech processing, and analysis of multimedia data (Cichocki et al., 2009). NMF's distinguishing feature is imposition of nonnegativity constraints, where only non-subtractive combinations of vectors in original space are allowed (Lee and Seung, 1999, 2001). Specific knowledge of the problem domain can be modelled by further imposing discriminative constraints, locality preservation, network-regularization or constraint on sparsity.

We have developed a Python-based NMF library called NIMFA which implements a wide variety of useful NMF operations and its components at a granular level. Our aim was both to provide access to already published variants of NMF and ease the innovative use of its components in crafting new algorithms. The library intentionally focuses on nonnegative variant of matrix factorization, and in terms of variety of different approaches compares favourably to several popular matrix factorization packages that are broader in scope (PyMF, (`http://code.google.com/p/pymf`), NMF package (`http://nmf.r-forge.r-project.org`), and bioNMF (`http://bionmf.cnb.csic.es`); see Table 1).

	NIMFA	PyMF	NMF	bioNMF
Language	Python	Python	R, C++	PHP, Matlab, C
License/Copyright	GPL3	GPL3	GPL2+	license-free
Hierarchical factorization models	+	–	(+)	–
Sparse format support	+	(+)	–	–
Web based client	–	–	–	+
Quality measures	+	–	+	+
Fitted model and residuals tracking	+	–	+	–
Algorithm specific parameters	+	+	+	+
Advanced initialization methods	+	–	+	(+)
Extensive documentation	+	–	+	+
Support for multiple runs	+	–	+	+
Visualization	(+)	–	+	+
Methods / Shared with NIMFA	11/11	10/3	5/4	3/3

Table 1: Feature comparison of NIMFA and three popular matrix factorization libraries. Symbol $+$ denotes full support, $(+)$ partial support and symbol $-$ no support. Last row reports on a number of different NMF algorithms implemented and a number of these that are shared with NIMFA .

2. Supported Factorization Methods and Approaches

In a standard model of NMF (Lee and Seung, 2001), a data matrix V is factorized to $V \equiv W\ H$ by solving a related optimization problem. Nonnegative matrices W and H are commonly referred to as basis and mixture matrix, respectively. NIMFA implements an originally proposed optimization (Lee and Seung, 2001; Brunet et al., 2004) with Euclidean or Kullback-Leibler cost function, along with Frobenius, divergence or connectivity costs. It also supports alternative optimization algorithms including Bayesian NMF Gibbs sampler (Schmidt et al., 2009), iterated conditional modes NMF (Schmidt et al., 2009), probabilistic NMF (Laurberg et al., 2008) and alternating least squares NMF using projected gradient method for subproblems (Lin, 2007). Sparse matrix factorization is provided either through probabilistic (Dueck and Frey, 2004) or alternating nonnegativity-constrained least squares factorization (Kim and Park, 2007). Fisher local factorization (Wang et al., 2004; Li et al., 2001) may be used when dependency of a new feature is constrained to a given small number of original features. Crisp relations can be revealed by binary NMF (Zhang et al., 2007).

NIMFA also implements several non-standard models. These comprise nonsmooth factorization $V \equiv W\ S(\theta)\ H$ (Pascual-Montano et al., 2006) and multiple model factorization for simultaneous treatment of several input matrices and their factorization with the same basis matrix W (Zhang et al., 2011).

All mentioned optimizations are incremental and start with initial approximation of matrices W and H. Appropriate choice of initialization can greatly speed-up the convergence and increase the overall quality of the factorization results. NIMFA contains implementations of popular initialization methods such as nonnegative double singular value decomposition (Boutsidis and Gallopoulos, 2007), random C and random Vcol algorithms (Albright et al., 2006). User can also completely specify initial factorization by passing fixed factors or choose any inexpensive method of randomly populated factors.

Factorization rank, choice of optimization method, and method-specific parameters jointly define the quality of approximation of input matrix V with the factorized system. NIMFA provides a number of quality measures ranging from standard ones (e.g., Euclidean distance, Kullback-Leibler divergence, and sparseness) to those more specific like feature scoring representing specificity to basis vectors (Kim and Park, 2007).

3. Design and Implementation

NIMFA has hierarchical, modular, and scalable structure which allows uniform treatment of numerous factorization models, their corresponding factorization algorithms and initialization methods. The library enables easy integration into user's code and arbitrary combinations of its factorization algorithms and their components. NIMFA's modules encompass implementations of factorization (`nimfa.methods.factorization`) and initialization algorithms (`nimfa.methods.seeding`), supporting models for factorization, fitted results, tracking and computation of quality and performance measures (`nimfa.models`), and linear algebra helper routines for sparse and dense matrices (`nimfa.utils`).

The library provides access to a set of standard data sets (`nimfa.datasets`), including those from text mining, image processing, bioinformatics, functional genomics, and collaborative filtering. Module `nimfa.examples` stores scripts that demonstrate factorization-based analysis of these data sets and provide examples for various analytic approaches like factorization of sparse matrices, multiple factorization runs, and others.

The guiding principle of constructing NIMFA was a component-oriented architecture. Every block of the algorithms, like data preprocessing, initialization of matrix factors, overall optimization, stopping criteria and quality scoring may be selected from the library or defined in a user-script, thus seamlessly enabling experimentation and construction of new approaches. Optimization process may be monitored, tracking residuals across iterations or tracking fitted factorization model.

NIMFA uses a popular Python matrix computation package `NumPy` for data management and representation. A drawback of the library is that is holds matrix factors and fitted model in main memory, raising an issue with very large data sets. To address this, NIMFA fully supports computations with sparse matrices as implemented in `SciPy`.

4. An Example Script

The sample script below demonstrates factorization of medulloblastoma gene expression data using alternating least squares NMF with projected gradient method for subproblems (Lin, 2007) and Random Vcol (Albright et al., 2006) initialization algorithm. An object returned by `nimfa.mf_run` is fitted factorization model through which user can access matrix factors and estimate quality measures.

```
import nimfa
V = nimfa.examples.medulloblastoma.read(normalize = True)
fctr = nimfa.mf(V, seed='random_vcol', method='lsnmf', rank=40, max_iter=65)
fctr_res = nimfa.mf_run(fctr)

print 'Rss:␣%5.4f,␣Evar:␣%5.4f' % (fctr_res.fit.rss(), fctr_res.fit.evar())
print 'K-L␣divergence:␣%5.4f' % fctr_res.distance(metric = 'kl')
print 'Sparseness,␣W:␣%5.4f,␣H:␣%5.4f' % fctr_res.fit.sparseness()
```

Running this script produces the following output, where slight differences in reported scores across different runs can be attributed to randomness of the Random Vcol initialization method.

```
Rss: 0.1895, Evar: 0.9998
K-L divergence: 38.6581
Sparseness, W: 0.7279, H: 0.8739
```

5. Availability and Requirements

NIMFA is a Python-based package requiring SciPy version 0.9.0 or higher. It is available under the GNU General Public License (GPL) version 3. The latest version with documentation and working examples can be found at http://nimfa.biolab.si.

Acknowledgments

We would like to acknowledge support for this project from the Google Summer of Code 2011 program and from the Slovenian Research Agency grants P2-0209, J2-9699, and L2-1112.

References

Russell Albright, Carl D. Meyer and Amy N. Langville. Algorithms, initializations, and convergence for the nonnegative matrix factorization. NCSU Technical Report Math 81706, NC State University, Releigh, USA, 2006.

Christos Boutsidis and Efstratios Gallopoulos. SVD-based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

Jean-P. Brunet, Pablo Tamayo, Todd R. Golub and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. In *Proceedings of the National Academy of Sciences of the USA*, 101(12):4164–4169, 2004.

Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan and Shun-ichi Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley & Sons Ltd, West Sussex, United Kingdom, 2009.

Delbert Dueck and Brendan J. Frey. Probabilistic sparse matrix factorization. University of Toronto Technical Report PSI-2004-23, Probabilistic and Statistical Inference Group, University of Toronto, 2004.

Hyuonsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

Hans Laurberg, Mads G. Christensen, Mark D. Plumbley, Lars K. Hansen and Soren H. Jensen. Theorems on positive data: on the uniqueness of NMF. *Computational Intelligence and Neuroscience*, doi: 10.1155/2008/764206, 2008.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the Neural Information Processing Systems*, pages 556–562, Vancouver, Canada, 2001.

Stan Z. Li, Xinwen Huo, Hongjiang Zhang and Qian S. Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 207-212, Kauai, USA, 2001.

Chin J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

Alberto Pascual-Montano, J. M. Carazo, Kieko Kochi, Dietrich Lehmann and Roberto D. Pascual-Marqui. Nonsmooth nonnegative matrix factorization (nsnmf). In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):403–415, 2006.

Mikkel N. Schmidt, Ole Winther, and Lars K. Hansen. Bayesian non-negative matrix factorization. In *Proceedings of the 9th International Conference on Independent Component Analysis and Signal Separation*, pages 540–547, Paraty, Brazil, 2009.

Yuan Wang, Yunde Jia, Changbo Hu and Matthew Turk. Fisher non-negative matrix factorization for learning local features. In *Proceedings of the 6th Asian Conference on Computer Vision*, pages 27–30, Jeju, Korea, 2004.

Shihua Zhang, Qingjiao Li and Xianghong J. Zhou. A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules. *Bioinformatics*, 27(13):401–409, 2011.

Zhongyuan Zhang, Tao Li, Chris H. Q. Ding and Xiangsun Zhang. Binary Matrix Factorization with applications. In *Proceedings of 7th IEEE International Conference on Data Mining*, pages 391–400, Omaha, USA, 2007.

Journal of Machine Learning Research 13 (2012) 855-903 Submitted 4/11; Revised 8/11; Published 3/12

Algebraic Geometric Comparison of Probability Distributions

Franz J. Király[*] FRANZ.J.KIRALY@TU-BERLIN.DE
Paul von Bünau PAUL.BUENAU@TU-BERLIN.DE
Frank C. Meinecke FRANK.MEINECKE@TU-BERLIN.DE
Duncan A. J. Blythe[†] DUNCAN.BLYTHE@BCCN-BERLIN.DE
Klaus-Robert Müller[‡] KLAUS-ROBERT.MUELLER@TU-BERLIN.DE
Machine Learning Group, Computer Science
Berlin Institute of Technology (TU Berlin)
Franklinstr. 28/29
10587 Berlin, Germany

Editor: Kenji Fukumizu

Abstract

We propose a novel algebraic algorithmic framework for dealing with probability distributions represented by their cumulants such as the mean and covariance matrix. As an example, we consider the unsupervised learning problem of finding the subspace on which several probability distributions agree. Instead of minimizing an objective function involving the estimated cumulants, we show that by treating the cumulants as elements of the polynomial ring we can directly solve the problem, at a lower computational cost and with higher accuracy. Moreover, the algebraic viewpoint on probability distributions allows us to invoke the theory of algebraic geometry, which we demonstrate in a compact proof for an identifiability criterion.

Keywords: computational algebraic geometry, approximate algebra, unsupervised Learning

1. Introduction

Comparing high dimensional probability distributions is a general problem in machine learning, which occurs in two-sample testing (e.g., Hotelling, 1932; Gretton et al., 2007), projection pursuit (e.g., Friedman and Tukey, 1974), dimensionality reduction and feature selection (e.g., Torkkola, 2003). Under mild assumptions, probability densities are uniquely determined by their cumulants which are naturally interpreted as coefficients of homogeneous multivariate polynomials. Representing probability densities in terms of cumulants is a standard technique in learning algorithms. For example, in Fisher Discriminant Analysis (Fisher, 1936), the class conditional distributions are approximated by their first two cumulants.

In this paper, we take this viewpoint further and work explicitly with polynomials. That is, we treat estimated cumulants not as constants in an objective function but as objects that we manipulate algebraically in order to find the optimal solution. As an example, we consider the problem of finding the linear subspace on which several probability distributions are identical: given D-variate

[*] Also in the Discrete Geometry Group, Institute of Mathematics, FU Berlin.

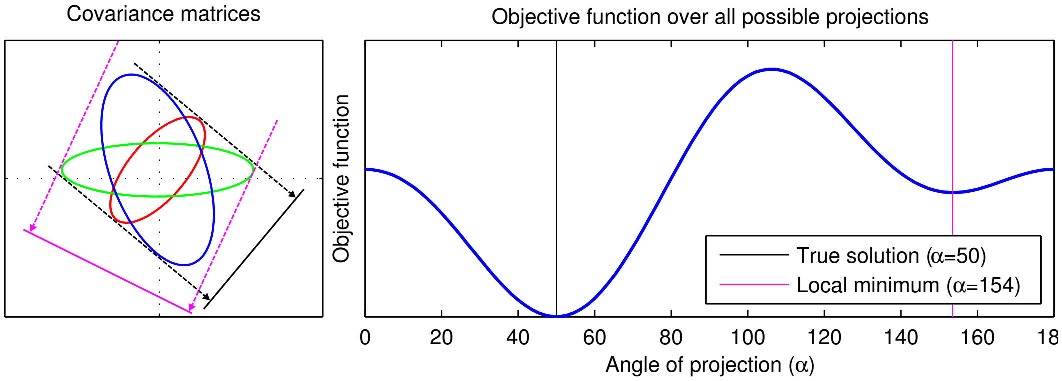
[†] Also in the Bernstein Center for Computational Neuroscience (BCCN), Berlin.

[‡] Also in the Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea.

random variables $X_1, \dots, X_m$, we want to find the linear map $P \in \mathbb{R}^{d \times D}$ such that the projected random variables have the same probability distribution,

$$PX_1 \sim \cdots \sim PX_m.$$

This amounts to finding the directions on which all projected cumulants agree. For the first cumulant, the mean, the projection is readily available as the solution of a set of linear equations. For higher order cumulants, we need to solve polynomial equations of higher degree. We present the first algorithm that solves this problem explicitly for arbitrary degree, and show how algebraic geometry can be applied to prove properties about it.


Covariance matrices

Objective function over all possible projections

Figure 1: Illustration of the optimization approach. The left panel shows the contour plots of three sample covariance matrices. The black line is the true one-dimensional subspace on which the projected variances are exactly equal, the magenta line corresponds to a local minimum of the objective function. The right panel shows the value of the objective function over all possible one-dimensional subspaces, parameterized by the angle α to the horizontal axis; the angles corresponding to the global minimum and the local minimum are indicated by black and magenta lines respectively.

To clarify the gist of our approach, let us consider a stylized example. In order to solve a learning problem, the conventional approach in machine learning is to formulate an objective function, for example, the log likelihood of the data or the empirical risk. Instead of minimizing an objective function that involves the polynomials, we consider the polynomials as *objects in their own right* and then solve the problem by algebraic manipulations. The advantage of the algebraic approach is that it captures the inherent structure of the problem, which is in general difficult to model in an optimization approach. In other words, the algebraic approach actually *solves* the problem, whereas optimization *searches* the space of possible solutions guided by an objective function that is minimal at the desired solution but can give poor directions outside of the neighborhood around its global minimum. Let us consider the problem where we would like to find the direction $v \in \mathbb{R}^2$ on which several sample covariance matrices $\Sigma_1, \dots, \Sigma_m \subset \mathbb{R}^{2\times 2}$ are equal. The usual ansatz would be to formulate an optimization problem such as

$$v^* = \operatorname*{argmin}_{\|v\|=1} \sum_{1 \le i,j \le m} \left(v^\top \Sigma_i v - v^\top \Sigma_j v \right)^2. \tag{1}$$

This objective function measures the deviation from equality for all pairs of covariance matrices; it is zero if and only if all projected covariances are equal and positive otherwise. Figure 1 shows an example with three covariance matrices (left panel) and the value of the objective function for all possible projections $v = \begin{bmatrix}\cos(\alpha) & \sin(\alpha)\end{bmatrix}^\top$. The solution to this non-convex optimization problem can be found using a gradient-based search procedure, which may terminate in one of the local minima (e.g., the magenta line in Figure 1) depending on the initialization.

However, the natural representation of this problem is not in terms of an objective function but rather a system of equations to be solved for v, namely

$$v^\top \Sigma_1 v = \cdots = v^\top \Sigma_m v. \tag{2}$$

In fact, by going from an algebraic description of the set of solutions to a formulation as an optimization problem in Equation 1, we lose important structure. In the case where there is an exact solution, it can be attained explicitly with algebraic manipulations. However, when we estimate a covariance matrix from finite or noisy samples, there exists no exact solution in general. Therefore we present an algorithm which combines the statistical treatment of uncertainty in the coefficients of polynomials with the exactness of algebraic computations to obtain a consistent estimator for v that is computationally efficient.

Note that this approach is not limited to this particular learning task. In fact, it is applicable whenever a set of solutions can be described in terms of a set of polynomial equations, which is a rather general setting. For example, we could use a similar strategy to find a subspace on which the projected probability distribution has another property that can be described in terms of cumulants, for example, independence between variables. Moreover, an algebraic approach may also be useful in solving certain optimization problems, as the set of extrema of a polynomial objective function can be described by the vanishing set of its gradient. The algebraic viewpoint also allows a novel interpretation of algorithms operating in the feature space associated with the polynomial kernel. We would therefore argue that methods from computational algebra and algebraic geometry are useful for the wider machine learning community.

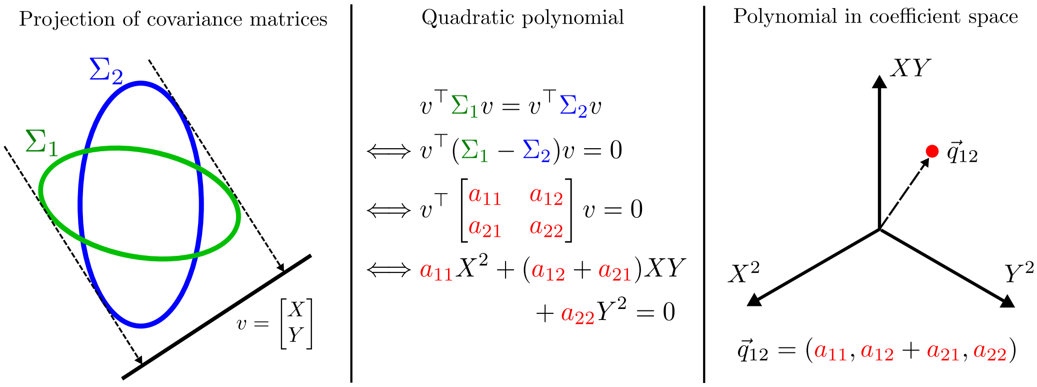

Projection of covariance matrices
Σ_2
Σ_1
$v = \begin{bmatrix} X \\ Y \end{bmatrix}$
Quadratic polynomial
$v^\top \Sigma_1 v = v^\top \Sigma_2 v$
$\iff v^\top (\Sigma_1 - \Sigma_2) v = 0$
$\iff v^\top \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} v = 0$
$\iff a_{11} X^2 + (a_{12} + a_{21}) XY + a_{22} Y^2 = 0$
Polynomial in coefficient space
XY
$\vec{q}_{12}$
X^2
Y^2
$\vec{q}_{12} = (a_{11}, a_{12} + a_{21}, a_{22})$

Figure 2: Representation of the problem: the left panel shows sample covariance matrices Σ_1 and Σ_2 with the desired projection v. In the middle panel, this projection is defined as the solution to a quadratic polynomial. This polynomial is embedded in the vector space of coefficients spanned by the monomials X^2, Y^2 and XY shown in the right panel.

Let us first of all explain the representation over which we compute. We will proceed in the three steps illustrated in Figure 2, from the geometric interpretation of sample covariance matrices in data space (left panel), to the quadratic equation defining the projection v (middle panel), to the representation of the quadratic equation as a coefficient vector (right panel). To start with, we consider the Equation 2 as a set of homogeneous quadratic equations defined by

$$v^\top(\Sigma_i - \Sigma_j)v = 0 \;\; \forall 1 \leq i, j \leq m, \tag{3}$$

where we interpret the components of v as variables, $v = \begin{bmatrix} X & Y \end{bmatrix}^\top$. The solution to these equations is the direction in $\mathbb{R}^2$ on which the projected variance is equal over all covariance matrices. Each of these equations corresponds to a quadratic polynomial in the variables X and Y,

$$\begin{aligned} q_{ij} &= v^\top(\Sigma_i - \Sigma_j)v \\ &= v^\top \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} v \\ &= a_{11}X^2 + (a_{12} + a_{21})XY + a_{22}Y^2, \end{aligned} \tag{4}$$

which we embed into the vector space of coefficients. The coordinate axis are the monomials $\{X^2, XY, Y^2\}$; that is, the three independent entries in the Gram matrix $(\Sigma_i - \Sigma_j)$. That is, the polynomial in Equation 4 becomes the coefficient vector

$$\vec{q}_{ij} = \begin{bmatrix} a_{11} & a_{12} + a_{21} & a_{22} \end{bmatrix}^\top.$$

The motivation for the vector space interpretation is that every linear combination of the Equations 3 is also a characterization of the set of solutions: this will allow us to find a particular set of equations by linear combination, from which we can directly obtain the solution. Note, however, that the vector space representation does not give us all equations which can be used to describe the solution: we can also multiply with arbitrary polynomials. However, for the algorithm that we present here, linear combinations of polynomials are sufficient.

Figure 3 illustrates how the algebraic algorithm works in the vector space of coefficients. The polynomials $\mathcal{Q} = \{q_{ij}\}_{i,j=1}^n$ span a space of constraints which defines the set of solutions. The next step is to find a polynomial of a certain form that immediately reveals the solution. One of these sets is the linear subspace spanned by the monomials $\{XY, Y^2\}$: any polynomial in this span is divisible by Y. Our goal is now to find a polynomial which is contained in both this subspace and the span of $\mathcal{Q}$. Under mild assumptions, one can always find a polynomial of this form, and it corresponds to an equation

$$Y(\alpha X + \beta Y) = 0. \tag{5}$$

Since this polynomial is in the span of $\mathcal{Q}$, our solution v has to be a zero of this particular polynomial: $v_2(\alpha v_1 + \beta v_2) = 0$. Moreover, we can assume[1] that $v_2 \neq 0$, so that we can divide out the variable Y to get the linear factor $(\alpha X + \beta Y)$,

$$0 = \alpha X + \beta Y = \begin{bmatrix} \alpha & \beta \end{bmatrix} v.$$

[1] This is a consequence of the generative model for the observed polynomials which is introduced in Section 2.1. In essence, we use the fact that our polynomials have no special property (apart from the existence of a solution) with probability one.

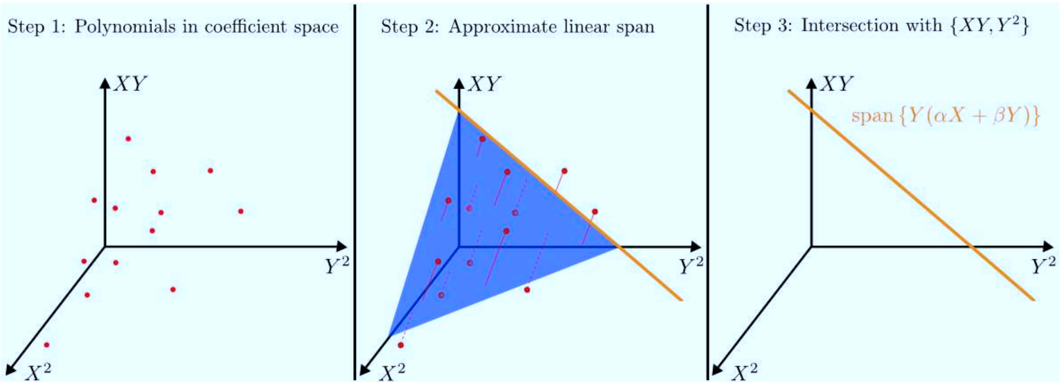

Step 1: Polynomials in coefficient space
XY
Y^2
X^2
Step 2: Approximate linear span
XY
Y^2
X^2
Step 3: Intersection with $\{XY, Y^2\}$
XY
span $\{Y(\alpha X + \beta Y)\}$
Y^2
X^2

Figure 3: Illustration of the algebraic algorithm. The left panel shows the vector space of coefficients where the polynomials corresponding to the Equations 3 are considered as elements of the vector space shown as red points. The middle panel shows the approximate 2-dimensional subspace (blue surface) onto which we project the polynomials. The right panel shows the one-dimensional intersection (orange line) of the approximate subspace with the plane spanned by spanned by $\{XY, Y^2\}$. This subspace is spanned by the polynomial $Y(\alpha X + \beta Y)$, so we can divide by the variable Y.

Hence $v = \begin{bmatrix} -\beta & \alpha \end{bmatrix}^\top$ is the solution up to arbitrary scaling, which corresponds to the one-dimensional subspace in Figure 3 (orange line, right panel). A more detailed treatment of this example can also be found in Appendix A.

In the case where there exists a direction v on which the projected covariances are exactly equal, the linear subspace spanned by the set of polynomials Q has dimension two, which corresponds to the degrees of freedom of possible covariance matrices that have fixed projection on one direction. However, since in practice covariance matrices are estimated from finite and noisy samples, the polynomials Q usually span the whole space, which means that there exists only a trivial solution $v = 0$. This is the case for the polynomials pictured in the left panel of Figure 3. Thus, in order to obtain an approximate solution, we first determine the approximate two-dimensional span of Q using a standard least squares method as illustrated in the middle panel. We can then find the intersection of the approximate two-dimensional span of Q with the plane spanned by the monomials $\{XY, Y^2\}$. As we have seen in Equation 5, the polynomials in this span provide us with a unique solution for v up to scaling, corresponding to the fact that the intersection has dimension one (see the right panel of Figure 3). Alternatively, we could have found the one-dimensional intersection with the span of $\{XY, X^2\}$ and divided out the variable X. In fact, in the final algorithm we will find all such intersections and combine the solutions in order to increase the accuracy. Note that we have found this solution by solving a simple least-squares problem (second step, middle panel of Figure 3). In contrast, the optimization approach (Figure 1) can require a large number of iterations and may converge to a local minimum. A more detailed example of the algebraic algorithm can be found in Appendix A.

The algebraic framework does not only allow us to construct efficient algorithms for working with probability distributions, it also offers powerful tools to prove properties of algorithms that operate with cumulants. For example, we can answer the following central question: how many

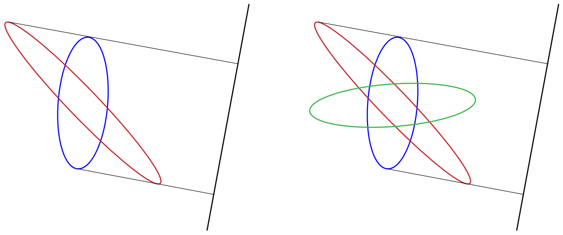


Figure 4: The left panel shows two sample covariance matrices in the plane, along with a direction on which they are equal. In the right panel, a third (green) covariance matrix does not have the same projected variance on the black direction.

distinct data sets do we need such that the subspace with identical probability distributions becomes uniquely identifiable? This depends on the number of dimensions and the cumulants that we consider. Figure 4 illustrates the case where we are given only the second order moment in two dimensions. Unless $\Sigma_1 - \Sigma_2$ is indefinite, there *always* exists a direction on which two covariance matrices in two dimensions are equal (left panel of Figure 4)—irrespective of whether the probability distributions are actually equal. We therefore need at least three covariance matrices (see right panel), or to consider other cumulants as well. We derive a tight criterion on the necessary number of data sets depending on the dimension and the cumulants under consideration. The proof hinges on viewing the cumulants as polynomials in the algebraic geometry framework: the polynomials that define the sought-after projection (e.g., Equations 3) generate an ideal in the polynomial ring which corresponds to an algebraic set that contains all possible solutions. We can then show how many independent polynomials are necessary so that the dimension of the linear part of the algebraic set has smaller dimension in the generic case. We conjecture that these proof techniques are also applicable to other scenarios where we aim to identify a property of a probability distribution from its cumulants using algebraic methods.

Our work is not the first that applies geometric or algebraic methods to Machine Learning or statistics: for example, methods from group theory have already found their application in machine learning, for example, Kondor (2007) and Kondor and Borgwardt (2008); there are also algebraic methods estimating structured manifold models for data points as in Vidal et al. (2005) which are strongly related to polynomial kernel PCA—a method which can itself be interpreted as a way of finding an approximate vanishing set.

The field of Information Geometry interprets parameter spaces of probability distributions as differentiable manifolds and studies them from an information-theoretical point of view (see for example the standard book by Amari and Nagaoka, 2000), with recent interpretations and improvements stemming from the field of algebraic geometry by Watanabe (2009). There is also the nascent field of algebraic statistics which studies the parameter spaces of mainly discrete random variables in terms of commutative algebra and algebraic geometry, see the recent overviews by Sturmfels (2002, Chapter 8) and Drton et al. (2010) or the book by Gibilisco et al. (2010) which also focuses on the interplay between information geometry and algebraic statistics. These approaches have in common that the algebraic and geometric concepts arise naturally when considering distributions in parameter space.

Given samples from a probability distribution, we may also consider algebraic structures in the data space. Since the data are uncertain, the algebraic objects will also come with an inherent uncertainty, unlike the exact manifolds in the case when we have an a-priori family of probability distributions. Coping with uncertainties is one of the main interests of the emerging fields of approximative and numerical commutative algebra, see the book by Stetter (2004) for an overview on numerical methods in algebra, or the treatise by Kreuzer et al. (2009) for recent developments in approximate techniques on noisy data. There exists a wide range of methods; however, to our knowledge, the link between approximate algebra and the representation of probability distributions in terms of their cumulants has not been studied yet.

The remainder of this paper is organized as follows: in the next Section 2, we introduce the algebraic view of probability distribution, rephrase our problem in terms of this framework and investigate its identifiability. The algorithm for the exact case is presented in Section 3, followed by the approximate version in Section 4. The results of our numerical simulations and a comparison against the Stationary Subspace Analysis (SSA) algorithm given in von Bünau et al. (2009), can be found in Section 5. In the last Section 6, we discuss our findings and point to future directions. The appendix contains an example and proof details.

2. The Algebraic View on Probability Distributions

In this section we introduce the algebraic framework for dealing with probability distributions. This requires basic concepts from complex algebraic geometry. A comprehensive introduction to algebraic geometry with a view to computation can be found in the book by Cox et al. (2007). In particular, we recommend to go through the Chapters 1 and 4.

In this section, we demonstrate the algebraic viewpoint of probability distributions on the application that we study in this paper: finding the linear subspace on which probability distributions are equal.

Problem 1 *Let $X_1, \ldots, X_m$ be a set of D-variate random variables, having smooth densities. Find all linear maps $P \in \mathbb{R}^{d \times D}$ such that the transformed random variables have the same distribution,*

$$PX_1 \sim \cdots \sim PX_m.$$

In the first part of this section, we show how this problem can be formulated algebraically. We will first of all review the relationship between the probability density function and its cumulants, before we translate the cumulants into algebraic objects. Then we introduce the theoretical underpinnings for the statistical treatment of polynomials arising from estimated cumulants and prove conditions on identifiability for the problem addressed in this paper.

2.1 From Probability Distributions to Polynomials

The probability distribution of every smooth real random variable X can be fully characterized in terms of its *cumulants*, which are the tensor coefficients of the cumulant generating function. This representation has the advantage that each cumulant provides a compact description of certain aspects of the probability density function.

Definition 2 *Let X be a D-variate random variable. Then by $\kappa_n(X) \in \mathbb{R}^{D^{(\times n)}}$ we denote the n-th cumulant, which is a real tensor of degree n.*

Let us introduce a useful shorthand notation for linearly transforming tensors.

Definition 3 *Let $A \in \mathbb{C}^{d \times D}$ be a matrix. For a tensor $T \in \mathbb{R}^{D^{(\times n)}}$ (i.e., a real tensor T of degree n of dimension $D^n = D \cdot D \cdot \ldots \cdot D$) we will denote by $A \circ T$ the application of A to T along all tensor dimensions, that is,*

$$(A \circ T)_{i_1 \ldots i_n} = \sum_{j_1=1}^{D} \cdots \sum_{j_n=1}^{D} A_{i_1 j_1} \cdot \ldots \cdot A_{i_n j_n} T_{j_1 \ldots j_n}.$$

The cumulants of a linearly transformed random variable are the multilinearly transformed cumulants, which is a convenient property when one is looking for a certain linear subspace.

Proposition 4 *Let X be a real D-dimensional random variable and let $A \in \mathbb{R}^{d \times D}$ be a matrix. Then the cumulants of the transformed random variable AX are the transformed cumulants,*

$$\kappa_n(AX) = A \circ \kappa_n(X).$$

We now want to formulate our problem in terms of cumulants. First of all, note that $PX_i \sim PX_j$ if and only if $vX_i \sim vX_j$ for all row vectors $v \in \operatorname{span} P^\top$.

Problem 5 *Find all d-dimensional linear subspaces in the set of vectors*

$$\begin{aligned} S &= \left\{ v \in \mathbb{R}^D \;\middle|\; v^\top X_1 \sim \cdots \sim v^\top X_m \right\} \\ &= \left\{ v \in \mathbb{R}^D \;\middle|\; v^\top \circ \kappa_n(X_i) = v^\top \circ \kappa_n(X_j),\ n \in \mathbb{N}, 1 \le i, j \le m \right\}. \end{aligned}$$

Note that we are looking for linear subspaces in S; however, S itself is not a vector space in general. Apart from the fact that S is homogeneous, that is, $\lambda S = S$ for all $\lambda \in \mathbb{R}$, there is no additional structure that we make use of.

For the sake of clarity, in the remainder of this paper we restrict ourselves to the first two cumulants. Note, however, that one of the strengths of the algebraic framework is that the generalization to arbitrary degree is straightforward; throughout this paper, we indicate the necessary changes and differences. Thus, from now on, we denote the first two cumulants by $\mu_i = \kappa_1(X_i)$ and $\Sigma_i = \kappa_2(X_i)$ respectively for all $1 \le i \le m$. Moreover, without loss of generality, we can shift the mean vectors and choose a basis such that the random variable X_m has zero mean and unit covariance. Thus we arrive at the following formulation.

Problem 6 *Find all d-dimensional linear subspaces in*

$$S = \{ v \in \mathbb{R}^D \mid v^\top (\Sigma_i - I) v = 0,\ v^\top \mu_i = 0,\ 1 \le i \le (m-1) \}.$$

Note that S is the set of solutions to $m-1$ quadratic and $m-1$ linear equations in D variables. Now it is only a formal step to arrive in the framework of algebraic geometry: let us think of the left hand side of each of the quadratic and linear equations as polynomials $q_1, \ldots, q_{m-1}$ and $f_1, \ldots, f_{m-1}$ in the variables $T_1, \ldots, T_D$ respectively,

$$q_i = \begin{bmatrix} T_1 \cdots T_D \end{bmatrix} \circ (\Sigma_i - I) \quad \text{and} \quad f_i = \begin{bmatrix} T_1 \cdots T_D \end{bmatrix} \circ \mu_i,$$

which are elements of the polynomial ring over the complex numbers in D variables, $\mathbb{C}[T_1, \ldots, T_D]$. Note that in the introduction we have used X and Y to denote the variables in the polynomials, we will now switch to $T_1,\dots,T_D$ in order to avoid confusion with random variables. Thus S can be rewritten in terms of polynomials,

$$S = \left\{v \in \mathbb{R}^D \mid q_i(v) = f_i(v) = 0 \; \forall \, 1 \leq i \leq m-1\right\},$$

which means that S is an algebraic set. In the following, we will consider the corresponding complex vanishing set

$$\begin{aligned} S &= \mathrm{V}(q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}) \\ &:= \left\{v \in \mathbb{C}^D \mid q_i(v) = f_i(v) = 0 \; \forall \, 1 \leq i \leq m-1\right\} \subseteq \mathbb{C}^D \end{aligned}$$

and keep in mind that eventually we will be interested in the real part of S. Working over the complex numbers simplifies the theory and creates no algorithmic difficulties: when we start with real cumulant polynomials, the solution will always be real. Finally, we can translate our problem into the language of algebraic geometry.

Problem 7 *Find all d-dimensional linear subspaces in the algebraic set*

$$S = \mathrm{V}(q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}).$$

So far, this problem formulation does not include the assumption that a solution exists. In order to prove properties about the problem and algorithms for solving it we need to assume that there exist a d-dimensional linear subspace $S' \subset S$. That is, we need to formulate a *generative model* for our observed polynomials $q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}$. To that end, we introduce the concept of a *generic* polynomial, for a technical definition see Appendix B. Intuitively, a generic polynomial is a continuous, polynomial valued random variable which almost surely has no algebraic properties except for those that are logically implied by the conditions on it. An algebraic property is an event in the probability space of polynomials which is defined by the common vanishing of a set of polynomial equations in the coefficients. For example, the property that a quadratic polynomial is a square of linear polynomial is an algebraic property, since it is described by the vanishing of the discriminants. In the context of Problem 7, we will consider the observed polynomials as generic conditioned on the algebraic property that they vanish on a fixed d-dimensional linear subspace S'.

One way to obtain generic polynomials is to replace coefficients with, for example, Gaussian random variables. For example, a generic homogeneous quadric $q \in \mathbb{C}[T_1,T_2]$ is given by

$$q = Z_{11}T_1^2 + Z_{12}T_1T_2 + Z_{22}T_2^2,$$

where the coefficients $Z_{ij} \sim \mathcal{N}(\mu_{ij},\sigma_{ij})$ are independent Gaussian random variables with arbitrary parameters. Apart from being homogeneous, there is no condition on q. If we want to add the condition that q vanishes on the linear space defined by $T_1 = 0$, we would instead consider

$$q = Z_{11}T_1^2 + Z_{12}T_1T_2.$$

A more detailed treatment of the concept of genericity, how it is linked to probabilistic sampling, and a comparison with the classical definitions of genericity can be found in Appendix B.1.

We are now ready to reformulate the genericity conditions on the random variables $X_1,\dots,X_m$ in the above framework. Namely, we have assumed that the X_i are general under the condition that they agree in the first two cumulants when projected onto some linear subspace S'. Rephrased for the cumulants, Problems 1 and 7 become well-posed and can be formulated as follows.

Problem 8 *Let S' be an unknown d-dimensional linear subspace in $\mathbb{C}^D$. Assume that $f_1,\dots,f_{m-1}$ are generic homogenous linear polynomials, and $q_1,\dots,q_{m-1}$ are generic homogenous quadratic polynomials, all vanishing on S'. Find all d-dimensional linear subspaces in the algebraic set*

$$S = \operatorname{V}(q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}).$$

As we have defined "generic" as an implicit "almost sure" statement, we are in fact looking for an algorithm which gives the correct answer with probability one under our model assumptions. Intuitively, S' should be also the only d-dimensional linear subspace in S, which is not immediately guaranteed from the problem description. Indeed this is true if m is large enough, which is the topic of the next section.

2.2 Identifiability

In the last subsection, we have seen how to reformulate our initial Problem 1 about comparison of cumulants as the completely algebraic Problem 8. We can also reformulate identifiability of the true solution in the original problem in an algebraic way: identifiability in Problem 1 means that the projection P can be uniquely computed from the probability distributions. Following the same reasoning we used to arrive at the algebraic formulation in Problem 8, one concludes that identifiability is equivalent to the fact that there exists a unique linear subspace in S.

Since identifiability is now a completely algebraic statement, it can be treated also in algebraic terms. In Appendix B, we give an algebraic geometric criterion for identifiability of the stationary subspace; we will sketch its derivation in the following.

The main ingredient is the fact that, intuitively spoken, every generic polynomials carries one degree of freedom in terms of dimension, as for example the following result on generic vector spaces shows:

Proposition 9 *Let $\mathcal{P}$ be an algebraic property such that the polynomials with property $\mathcal{P}$ form a vector space V. Let $f_1,\dots,f_n \in \mathbb{C}[T_1,\dots T_D]$ be generic polynomials satisfying $\mathcal{P}$. Then*

$$\operatorname{rank}\operatorname{span}(f_1,\dots,f_n) = \min(n, \dim V).$$

Proof This is Proposition 42 in the appendix. ■

On the other hand, if the polynomials act as constraints, one can prove that each one reduces the degrees of freedom in the solution by one:

Proposition 10 *Let Z be a sub-vector space of $\mathbb{C}^D$. Let $f_1,\dots,f_n$ be generic homogenous polynomials in D variables (of fixed but arbitrary degree each), vanishing on Z. Then for their common vanishing set $\operatorname{V}(f_1,\dots,f_n) = \{x \in \mathbb{C}^D \mid f_i(x) = 0\ \forall i\}$, one can write*

$$\operatorname{V}(f_1,\dots,f_n) = Z \cup U,$$

where U is an algebraic set with

$$\dim U \leq \max(D-n,\, 0).$$

Proof This follows from Corollary 61 in the appendix. ■

Proposition 10 can now be directly applied to Problem 8. It implies that $S = S'$ if $2(m-1) \geq D+1$, and that S' is the maximal dimensional component of S if $2(m-1) \geq D-d+1$. That is, if we start with m random variables, then S' can be identified uniquely if

$$2(m-1) \geq D-d+1$$

with classical algorithms from computational algebraic geometry in the noiseless case.

Theorem 11 *Let $X_1,\ldots,X_m$ be random variables. Assume there exists a projection $P \in \mathbb{R}^{d\times D}$ such that the first two cumulants of all $PX_1,\ldots,PX_m$ agree and the cumulants are generic under those conditions. Then the projection P is identifiable from the first two cumulants alone if*

$$m \geq \frac{D-d+1}{2} + 1.$$

Proof This is a direct consequence of Proposition 65 in the appendix, applied to the reformulation given in Problem 8. It is obtained by applying Proposition 10 to the generic forms vanishing on the fixed linear subspace S', and using that S' can be identified in S if it is the biggest dimensional part. ■

We have seen that identifiability means that there is an algorithm to compute P uniquely when the cumulants are known, resp. to compute a unique S from the polynomials f_i, q_i. It is not difficult to see that an algorithm doing this can be made into a consistent estimator when the cumulants are sample estimates. We will give an algorithm of this type in the following parts of the paper.

3. An Algorithm for the Exact Case

In this section we present an algorithm for solving Problem 8, under the assumption that the cumulants are known exactly. We will first fix notation and introduce important algebraic concepts. In the previous section, we derived in Problem 8 an algebraic formulation of our task: given generic quadratic polynomials $q_1,\ldots,q_{m-1}$ and linear polynomials $f_1,\ldots,f_{m-1}$, vanishing on a unknown linear subspace S' of $\mathbb{C}^D$, find S' as the unique d-dimensional linear subspace in the algebraic set $\operatorname{V}(q_1,\ldots,q_{m-1},f_1,\ldots,f_{m-1})$. First of all, note that the linear equations f_i can easily be removed from the problem: instead of looking at $\mathbb{C}^D$, we can consider the linear subspace defined by the f_i, and examine the algebraic set $\operatorname{V}(q'_1,\ldots,q'_{m-1})$, where q'_i are polynomials in $D-m+1$ variables which we obtain by substituting $m-1$ variables. So the problem we need to examine is in fact the modified problem where we have only quadratic polynomials. Secondly, we will assume that $m-1 \geq D$. Then, from Proposition 10, we know that $S = S'$ and Problem 8 becomes the following.

Problem 12 *Let S be an unknown d-dimensional subspace of $\mathbb{C}^D$. Given $m-1 \geq D$ generic homogenous quadratic polynomials $q_1,\ldots,q_{m-1}$ vanishing on S, find the d-dimensional linear subspace*

$$S = \operatorname{V}(q_1,\ldots,q_{m-1}).$$

Of course, we have to say what we mean by *finding* the solution. By assumption, the quadratic polynomials already fully describe the linear space S. However, since S is a linear space, we want a basis for S, consisting of d linearly independent vectors in $\mathbb{C}^D$. Or, equivalently, we want to find linearly independent linear forms $\ell_1, \ldots, \ell_{D-d}$ such that $\ell_i(x) = 0$ for all $x \in S$. The latter is the correct description of the solution in algebraic terms. We now show how to reformulate this in the right language, following the algebra-geometry duality. The algebraic set S corresponds to an ideal in the polynomial ring $C[T_1, \ldots, T_D]$.

Notation 13 *We denote the polynomial ring $\mathbb{C}[T_1, \ldots, T_D]$ by R. The ideal of S is an ideal in R, and we denote it by by $\mathfrak{s} = \mathrm{I}(S)$. Since S is a linear space, there exists a linear generating set $\ell_1, \ldots, \ell_{D-d}$ of $\mathfrak{s}$ which we will fix in the following.*

We can now relate the Problem 12 to a classical problem in algebraic geometry.

Problem 14 *Let $m > D$ and $q_1, \ldots, q_{m-1}$ be generic homogenous quadratic polynomials vanishing on a linear d-dimensional subspace $S \subseteq \mathbb{C}^D$. Then find a linear basis for the radical ideal*

$$\sqrt{\langle q_1, \ldots, q_{m-1} \rangle} = \mathrm{I}(\mathrm{V}(q_1, \ldots, q_{m-1})) = \mathrm{I}(S).$$

The first equality follows from Hilbert's Nullstellensatz. This also shows that solving the problem is in fact a question of computing a radical of an ideal. Computing the radical of an ideal is a classical problem in computational algebraic geometry, which is known to be difficult (for a more detailed discussion see Section 3.3). However, if we assume $m - 1 \geq D(D+1)/2 - d(d+1)/2$, we can dramatically reduce the computational cost and it is straightforward to derive an approximate solution. In this case, the q_i generate the vector space of homogenous quadratic polynomials which vanish on S, which we will denote by $\mathfrak{s}_2$. That this is indeed the case, follows from Proposition 9, and we have $\dim \mathfrak{s}_2 = D(D+1)/2 - d(d+1)/2$, as we will calculate in Remark 23.

Before we continue with solving the problem, we will need to introduce several concepts and abbreviating notations. First we introduce notation to denote sub-vector spaces which contain polynomials of certain degrees.

Notation 15 *Let I be a sub-$\mathbb{C}$-vector space of R, that is, $I = R$, or I is some ideal of R, for example, $I = \mathfrak{s}$. We denote the sub-$\mathbb{C}$-vector space of homogenous polynomials of degree k in I by I_k (in commutative algebra, this is standard notation for homogenously generated R-modules).*

For example, the homogenous polynomials of degree 2 vanishing on S form exactly the vector space $\mathfrak{s}_2$. Moreover, for any I, the equation $I_k = I \cap R_k$ holds. The vector spaces R_2 and $\mathfrak{s}_2$ will be the central objects in the following chapters. As we have seen, their dimension is given in terms of triangular numbers, for which we introduce some notation:

Notation 16 *We will denote the n-th triangular number by $\Delta(n) = \frac{n(n+1)}{2}$.*

The last notational ingredient will capture the structure which is imposed on R_k by the orthogonal decomposition $\mathbb{C}^D = S \oplus S^\perp$.

Notation 17 *Let $S^\perp$ be the orthogonal complement of S. Denote its ideal by $\mathfrak{n} = \mathrm{I}\left(S^\perp\right)$.*

Remark 18 *As $\mathfrak{n}$ and $\mathfrak{s}$ are homogenously generated in degree one, we have the calculation rules*

$$\mathfrak{s}_{k+1} = \mathfrak{s}_k \cdot R_1 \quad \textit{and} \quad \mathfrak{n}_{k+1} = \mathfrak{n}_k \cdot R_1,$$
$$(\mathfrak{s}_1)^k = (\mathfrak{s}^k)_k \quad \textit{and} \quad (\mathfrak{n}_1)^k = (\mathfrak{n}^k)_k$$

where $\cdot$ is the symmetrized tensor or outer product of vector spaces (these rules are canonically induced by the so-called graded structure of R-modules). In terms of ideals, the above decomposition translates to

$$R_1 = \mathfrak{s}_1 \oplus \mathfrak{n}_1.$$

Using the above rules and the binomial formula for ideals, this induces an orthogonal decomposition

$$\begin{aligned} R_2 =& R_1 \cdot R_1 = (\mathfrak{s}_1 \oplus \mathfrak{n}_1) \cdot (\mathfrak{s}_1 \oplus \mathfrak{n}_1) = (\mathfrak{s}_1)^2 \oplus (\mathfrak{s}_1 \cdot \mathfrak{n}_1) \oplus (\mathfrak{n}_1)^2 \\ &= \mathfrak{s}_1 \cdot (\mathfrak{s}_1 \oplus \mathfrak{n}_1) \oplus (\mathfrak{n}^2)_2 = \mathfrak{s}_1 \cdot R_1 \oplus (\mathfrak{n}^2)_2 = \mathfrak{s}_2 \oplus (\mathfrak{n}^2)_2 \end{aligned}$$

(and similar decompositions for the higher degree polynomials R_k).

The tensor products above can be directly translated to products of ideals, as the vector spaces above are each generated in a single degree (e.g., $\mathfrak{s}^k, \mathfrak{n}^k$, are generated homogenously in degree k). To express this, we will define an ideal which corresponds to R_1:

Notation 19 *We denote the ideal of R generated by all monomials of degree* 1 *by*

$$\mathfrak{m} = \langle T_1, \ldots, T_D \rangle.$$

Note that ideal $\mathfrak{m}$ is generated by all elements in R_1. Moreover, we have $\mathfrak{m}_k = R_k$ for all $k \geq 1$. Using $\mathfrak{m}$, one can directly translate products of vector spaces involving some R_k into products of ideals:

Remark 20 *The equality of vector spaces*

$$\mathfrak{s}_k = \mathfrak{s}_1 \cdot (R_1)^{k-1}$$

translates to the equality of ideals

$$\mathfrak{s} \cap \mathfrak{m}^k = \mathfrak{s} \cdot \mathfrak{m}^{k-1},$$

since both the left and right sides are homogenously generated in degree k.

3.1 The Algorithm

$S \subset \mathbb{C}^D$	d-dimensional projection space
$R = \mathbb{C}[T_1, \ldots T_D]$	Polynomial ring over $\mathbb{C}$ in D variables
R_k	$\mathbb{C}$-vector space of homogenous k-forms in $T_1, \ldots, T_D$
$\Delta(n) = \frac{n(n+1)}{2}$	n-th triangular number
$\mathfrak{s} = \langle \ell_1, \ldots, \ell_{D-d} \rangle = \mathrm{I}(S)$	The ideal of S, generated by linear polynomials ℓ_i
$\mathfrak{s}_k = R_k \cap \mathfrak{s}$	$\mathbb{C}$-vector space of homogenous k-forms vanishing on S
$\mathfrak{n} = \mathrm{I}(S^\perp)$	The ideal of $S^\perp$
$\mathfrak{n}_k = R_k \cap \mathfrak{n}$	$\mathbb{C}$-vector space of homogenous k-forms vanishing on $S^\perp$
$\mathfrak{m} = \langle T_1, \ldots, T_D \rangle$	The ideal of the origin in $\mathbb{C}^D$

Table 1: Notation and important definitions

In this section we present an algorithm for solving Problem 14, the computation of the radical of the ideal $\langle q_1,\dots,q_{m-1}\rangle$ under the assumption that

$$m \geq \Delta(D)-\Delta(d)+1.$$

Under those conditions, as we will prove in Remark 23 (iii), we have that

$$\langle q_1,\dots,q_{m-1}\rangle = \mathfrak{s}_2.$$

Using the notations previously defined, one can therefore infer that solving Problem 14 is equivalent to computing the radical $\mathfrak{s}=\sqrt{\mathfrak{s}\cdot\mathfrak{m}}$ in the sense of obtaining a linear generating set for $\mathfrak{s}$, or equivalent to finding a basis for $\mathfrak{s}_1$ when $\mathfrak{s}_2$ is given in an arbitrary basis. $\mathfrak{s}_2$ contains the complete information given by the covariance matrices and $\mathfrak{s}_1$ gives an explicit linear description of the space of projections under which the random variables $X_1,\dots,X_m$ agree.

Algorithm 1 The *input* consists of the quadratic forms $q_1,\dots,q_{m-1}\in R$, generating $\mathfrak{s}_2$, and the dimension d; the *output* is the linear generating set $\ell_1,\dots,\ell_{D-d}$ for $\mathfrak{s}_1$.

```
Let $\pi \leftarrow (1\,2\cdots D)$ be a transitive permutation of the variable indices $\{1,\dots,D\}$
Let $Q \leftarrow \begin{bmatrix} q_1 & \cdots & q_{m-1}\end{bmatrix}^\top$ be the $((m-1)\times\Delta(D))$-matrix of coefficient vectors, where every row corresponds to a polynomial and every column to a monomial $T_iT_j$.
for $k=1,\dots,D-d$ do
    Order the columns of $Q$ according to the lexicographical ordering of monomials $T_iT_j$ with variable indices permuted by $\pi^k$, that is, the ordering of the columns is given by the relation $\succ$ as
        $T^2_{\pi^k(1)} \succ T_{\pi^k(1)}T_{\pi^k(2)} \succ T_{\pi^k(1)}T_{\pi^k(3)} \succ \cdots \succ T_{\pi^k(1)}T_{\pi^k(D)} \succ T^2_{\pi^k(2)}$
        $\succ T_{\pi^k(2)}T_{\pi^k(3)} \succ \cdots \succ T^2_{\pi^k(D-1)} \succ T_{\pi^k(D-1)}T_{\pi^k(D)} \succ T^2_{\pi^k(D)}$
    Transform $Q$ into upper triangular form $Q'$ using Gaussian elimination
    The last non-zero row of $Q'$ is a polynomial $T_{\pi^k(D)}\ell$, where $\ell$ is a linear form in $\mathfrak{s}$, and we set $\ell_k \leftarrow \ell$
end for
```

Algorithm 1 shows the procedure in pseudo-code; a summary of the notation defined in the previous section can be found in Table 1. The algorithm has polynomial complexity in the dimension d of the linear subspace S.

Remark 21 *Algorithm 1 has average and worst case complexity*

$$O\left((\Delta(D)-\Delta(d))^2\Delta(D)\right),$$

In particular, if d is not considered as parameter of the algorithm, the average and the worst case complexity is $O(D^6)$. On the other hand, if $\Delta(D)-\Delta(d)$ is considered a fixed parameter, then Algorithm 1 has average and worst case complexity $O(D^2)$.

Proof This follows from the complexities of the elementary operations: upper triangularization of a generic matrix of rank r with m columns matrix needs $O(r^2m)$ operations. We first perform triangularization of a rank $\Delta(D)-\Delta(d)$ matrix with $\Delta(D)$ columns. The permutations can be obtained efficiently by bringing Q in row-echelon form and then performing row operations. Operations for extracting the linear forms and comparisons with respect to the monomial ordering are negligible. Thus the overall operation complexity to calculate $\mathfrak{s}_1$ is $O((\Delta(D)-\Delta(d))^2\Delta(D))$.

Note that the difference between worst- and average case lies at most in the coefficients, since the inputs are generic and the complexity only depends on the parameter D and not on the q_i. Thus, with probability 1, exactly the worst-case-complexity is attained. ■

There are two crucial facts which need to be verified for correctness of this algorithm. Namely, there are implicit claims made in Line 6 of Algorithm 1: first, it is claimed that the last non-zero row of Q' corresponds to a polynomial which factors into certain linear forms. Second, it is claimed that the ℓ obtained in step 6 generate $\mathfrak{s}$ resp. $\mathfrak{s}_1$. The proofs of these non-trivial claims can be found in Proposition 22 in the next subsection.

Dealing with additional linear forms $f_1,\ldots,f_{m-1}$, is possible by way of a slight modification of the algorithm. Because the f_i are linear forms, they are generators of $\mathfrak{s}$. We may assume that the f_i are linearly independent. By performing Gaussian elimination before the execution of Algorithm 1, we may reduce the number of variables by $m-1$, thus having to deal with new quadratic forms in $D-m+1$ instead of D variables. Also, the dimension of the space of projections is reduced to $\min(d-m+1,-1)$. Setting $D'=D-m+1$ and $d'=\min(d-m+1,-1)$ and considering the quadratic forms q_i with Gaussian eliminated variables, Algorithm 1 can be applied to the quadratic forms to find the remaining generators for $\mathfrak{s}_1$. In particular, if $m-1\geq d$, then there is no need for considering the quadratic forms, since d linearly independent linear forms already suffice to determine the solution.

We can also incorporate forms of higher degree corresponding to higher order cumulants. For this, we start with $\mathfrak{s}_k$, where k is the degree of the homogenous polynomials we get from the cumulant tensors of higher degree. Supposing we start with enough cumulants, we may assume that we have a basis of $\mathfrak{s}_k$. Performing Gaussian elimination on this basis with respect to the lexicographical order, we obtain in the last row a form of type $T_{\pi^k(D)}^{k-1}\ell$, where ℓ is a linear form. Doing this for $D-d$ permutations again yields a basis for $\mathfrak{s}_1$.

Moreover, slight algebraic modifications of this strategy also allow to consider data from cumulants of different degree simultaneously, and to reduce the number of needed polynomials to $O(D)$; however, due to its technicality, this is beyond the scope of the paper. We sketch the idea: in the general case, one starts with an ideal

$$I=\langle f_1,\ldots,f_m\rangle,$$

homogenously generated in arbitrary degrees. such that $\sqrt{I}=\mathfrak{s}$. Proposition 55 in the appendix implies that this happens whenever $m\geq D+1$. One then proves that due to the genericity of the f_i, there exists an N such that

$$I_N=\mathfrak{s}_N,$$

which means that $\mathfrak{s}_1$ can again be obtained by calculating the saturation of the ideal I. When fixing the degrees of the f_i, we will have $N=O(D)$ with a relatively small constant (for all f_i quadratic, this even becomes $N=O(\sqrt{D})$). So algorithmically, one would first calculate $I_N=\mathfrak{s}_N$, which then may be used to compute $\mathfrak{s}_1$ and thus $\mathfrak{s}$ analogously to the case $N=2$, as described above.

3.2 Proof of Correctness

In order to prove the correctness of Algorithm 1, we need to prove the following three statements.

Proposition 22 *For Algorithm 1 it holds that*
(i) *Q is of rank $\Delta(D)-\Delta(d)$.*
(ii) *The last column of Q in step 6 is of the claimed form.*
(iii) *The $\ell_1,\dots,\ell_{D-d}$ generate $\mathfrak{s}_1$.*

Proof This proposition will be proved successively in the following: (i) will follow from Remark 23 (iii); (ii) will be proved in Lemma 24; and (iii) will be proved in Proposition 25. ■

Let us first of all make some observations about the structure of the vector space $\mathfrak{s}_2$ in which we compute. It is the vector space of polynomials of homogenous degree 2 vanishing on S. On the other hand, we are looking for a basis $\ell_1,\dots,\ell_{D-d}$ of $\mathfrak{s}_1$. The following remark will relate both vector spaces:

Remark 23 *The following statements hold:*
(i) *$\mathfrak{s}_2$ is generated by the polynomials $\ell_i T_j, 1\le i\le D-d, 1\le j\le D,$.*
(ii) $\dim_{\mathbb{C}}\mathfrak{s}_2=\Delta(D)-\Delta(d)$
(iii) *Let $q_1,\dots,q_m$ with $m\ge\Delta(D)-\Delta(d)$ be generic homogenous quadratic polynomials in $\mathfrak{s}$. Then $\langle q_1,\dots,q_m\rangle=\mathfrak{s}_2$.*

Proof (i) In Remark 18, we have concluded that $\mathfrak{s}_2=\mathfrak{s}_1\cdot R_1$. Thus the product vector space $\mathfrak{s}_2$ is generated by a product basis of $\mathfrak{s}_1$ and R_1. Since $T_j, 1\le j\le D$ is a basis for R_1, and $\ell_i, 1\le i\le D-d$ is a basis for $\mathfrak{s}_1$, the statement holds. (ii) In Remark 20, we have seen that $R_2=\mathfrak{s}_2\oplus(\mathfrak{n}_1)^2$, thus $\dim\mathfrak{s}_2=\dim R_2-\dim(\mathfrak{n}_1)^2$. The vector space R_2 is minimally generated by the monomials of degree 2 in $T_1,\dots T_D$, whose number is $\Delta(D)$. Similarly, $(\mathfrak{n}_1)^2$ is minimally generated by the monomials of degree 2 in the variables $\ell'_1,\dots,\ell'_d$ that form the dual basis to the ℓ_i. Their number is $\Delta(d)$, so the statement follows. (iii) As the q_i are homogenous of degree two and vanish on S, they are elements in $\mathfrak{s}_2$. Due to (ii), we can apply Proposition 9 to conclude that they generate $\mathfrak{s}_2$ as vector space. ■

Now we continue to prove the remaining claims.

Lemma 24 *In Algorithm 1 the $(\Delta(D)-\Delta(d))$-th row of Q' (the upper triangular form of Q) corresponds to a 2-form $T_{\pi(D)}\ell$ with a linear polynomial $\ell\in\mathfrak{s}_1$.*

Proof Note that every homogenous polynomial of degree k is canonically an element of the vector space R_k in the monomial basis given by the T_i. Thus it makes sense to speak about the coefficients of T_i for an 1-form resp. the coefficients of T_iT_j of a 2-form.

Also, without loss of generality, we can take the trivial permutation $\pi=\mathrm{id}$, since the proof will not depend on the chosen lexicographical ordering and thus will be naturally invariant under permutations of variables. First we remark: since S is a generic d-dimensional linear subspace of $\mathbb{C}^D$, any linear form in $\mathfrak{s}_1$ will have at least $d+1$ non-vanishing coefficients in the T_i. On the other hand, by displaying the generators $\ell_i, 1\le i\le D-d$ in $\mathfrak{s}_1$ in reduced row echelon form with respect to the T_i-basis, one sees that one can choose all the ℓ_i in fact with exactly $d+1$ non-vanishing coefficients in the T_i such that no nontrivial linear combination of the ℓ_i has less then $d+1$ non-vanishing coefficients. In particular, one can choose the ℓ_i such that the biggest (w.r.t. the lexicographical order) monomial with non-vanishing coefficient of ℓ_i is T_i.

Remark 23 (i) states that $\mathfrak{s}_2$ is generated by

$$\ell_i T_j, 1 \leq i \leq D-d, 1 \leq j \leq D.$$

Together with our above reasoning, this implies the following.

Fact 1: There exist linear forms $\ell_i, 1 \leq i \leq D-d$ such that: the 2-forms $\ell_i T_j$ generate $\mathfrak{s}_2$, and the biggest monomial of $\ell_i T_j$ with non-vanishing coefficient under the lexicographical ordering is $T_i T_j$. By Remark 23 (ii), the last row of the upper triangular form Q' is a polynomial which has zero coefficients for all monomials possibly except the $\Delta(d)+1$ smallest,

$$T_{D-d}T_D, T_{D-d+1}^2, T_{D-d+1}T_{D-d+2}, \dots, T_{D-1}T_D, T_D^2.$$

On the other hand, it is guaranteed by our genericity assumption that the biggest of those terms is indeed non-vanishing, which implies the following.

Fact 2: The biggest monomial of the last row with non-vanishing coefficient (w.r.t the lexicographical order) is that of $T_{D-d}T_D$.

Combining Facts 1 and 2, we can now infer that the last row must be a scalar multiple of $\ell_{D-d}T_D$: since the last row corresponds to an element of $\mathfrak{s}_2$, it must be a linear combination of the $\ell_i T_j$. By Fact 1, every contribution of an $\ell_i T_j, (i, j) \neq (D-d, D)$ would add a non-vanishing coefficient lexicographically bigger than $T_{D-d}T_D$ which cannot cancel. So, by Fact 2, T_D divides the last row of the upper triangular form of Q, which then must be $T_D \ell_{D-d}$ or a multiple thereof. Also we have that $\ell_{D-d} \in \mathfrak{s}$ by definition. ■

It remains to be shown that by permutation of the variables we can find a basis for $\mathfrak{s}_1$.

Proposition 25 *The $\ell_1, \dots, \ell_{D-d}$ generate $\mathfrak{s}_1$ as vector space and thus $\mathfrak{s}$ as ideal.*

Proof Recall that π^i was the permutation to obtain ℓ_i. As we have seen in the proof of Lemma 24, ℓ_i is a linear form which has non-zero coefficients only for the $d+1$ coefficients $T_{\pi^i(D-d)}, \dots, T_{\pi^i(D)}$. Thus ℓ_i has a non-zero coefficient where all the $\ell_j, j < i$ have a zero coefficient, and thus ℓ_i is linearly independent from the $\ell_j, j < i$. In particular, it follows that the ℓ_i are linearly independent in R_1. On the other hand, they are contained in the $D-d$-dimensional sub-$\mathbb{C}$-vector space $\mathfrak{s}_1$ and are thus a basis of $\mathfrak{s}_1$, and also a generating set for the ideal $\mathfrak{s}$. ■

Note that all of these proofs generalize to k-forms. For example, one calculates that

$$\dim_{\mathbb{C}} \mathfrak{s}_k = \binom{D+k-1}{k} - \binom{d+k-1}{k},$$

and the triangularization strategy yields a last row which corresponds to $T_{\pi(D)}^{k-1}\ell$ with a linear polynomial $\ell \in \mathfrak{s}_1$

3.3 Relation to Previous Work in Computational Algebraic Geometry

In this section, we discuss how the algebraic formulation of the cumulant comparison problem given in Problem 14 relates to the classical problems in computational algebraic geometry.

Problem 14 confronts us with the following task: given polynomials $q_1, \dots, q_{m-1}$ with special properties, compute a linear generating set for the radical ideal

$$\sqrt{\langle q_1, \dots, q_{m-1} \rangle} = \mathrm{I}(\mathrm{V}(q_1, \dots, q_{m-1})).$$

Computing the radical of an ideal is a classical task in computational algebraic geometry, so our problem is a special case of radical computation of ideals, which in turn can be viewed as an instance of primary decomposition of ideals, see (Cox et al., 2007, Section 4.7).

While it has been known since the work of Hermann (1926) that there exist constructive algorithms to calculate the radical of a given ideal in polynomial rings, only in the recent decades there have been algorithms feasible for implementation in modern computer algebra systems. The best known algorithms are those of Gianni et al. (1988), implemented in AXIOM and REDUCE, the algorithm of Eisenbud et al. (1992), implemented in Macaulay 2, the algorithm of Caboara et al. (1997), currently implemented in CoCoA, and the algorithm of Krick and Logar (1991) and its modification by Laplagne (2006), available in SINGULAR.

All of these algorithms have two points in common. First of all, these algorithms have computational worst case complexities which are doubly exponential in the square of the number of variables of the given polynomial ring, see (Laplagne, 2006, Section 4). Although the worst case complexities may not be approached for the problem setting described in the current paper, these off-the-shelf algorithms do not take into account the specific properties of the ideals in question.

On the other hand, Algorithm 1 can be seen as a homogenous version of the well-known Buchberger algorithm to find a Groebner basis of the dehomogenization of $\mathfrak{s}$ with respect to a degree-first order. Namely, due to our strong assumptions on m, or as is shown in Proposition 55 in the appendix for a more general case, the homogenous saturations of the ideal $\langle q_1,\dots,q_{m-1}\rangle = \mathfrak{m}\cdot\mathfrak{s}$ and the ideal $\mathfrak{s}$ coincide. In particular, the dehomogenizations of the q_i constitute a generating set for the dehomogenization of $\mathfrak{s}$. The Buchberger algorithm now finds a reduced Groebner basis of $\mathfrak{s}$ which consists of exactly $D-d$ linear polynomials. Their homogenizations then constitute a basis of homogenous linear forms of $\mathfrak{s}$ itself. It can be checked that the first elimination steps which the Buchberger algorithm performs for the dehomogenizations of the q_i correspond directly to the elimination steps in Algorithm 1 for their homogenous versions. So our algorithm performs similarly to the Buchberger algorithm in a noiseless setting, since both algorithms compute a reduced Groebner basis in the chosen coordinate system.

However, in our setting which stems from real data, there is a second point which is more grave and makes the use of off-the-shelf algorithms impossible: the computability of an exact result completely relies on the assumption that the ideals given as input are exactly known, that is, a generating set of polynomials is exactly known. This is not a problem in classical computational algebra; however, when dealing with polynomials obtained from real data, the polynomials come not only with numerical error but in fact with statistical uncertainty. In general, the classical algorithms are unable to find any solution when confronted even with minimal noise on the otherwise exact polynomials. Namely, when we deal with a system of equations for which over-determination is possible, any perturbed system will be over-determined and thus have no solution. For example, the exact intersection of $N > D+1$ linear subspaces in complex D-space is always empty when they are sampled with uncertainty; this is a direct consequence of Proposition 10, when using the assumption that the noise is generic. However, if all those hyperplanes are nearly the same, then the result of a meaningful approximate algorithm should be a hyperplane close to all input hyperplanes instead of the empty set.

Before we continue, we would like to stress a conceptual point in approaching uncertainty. First, as in classical numerics, one can think of the input as theoretically exact but with fixed error ε and then derive bounds on the output error in terms of this ε and analyze their asymptotics. We will refer to this approach as *numerical uncertainty*, as opposed to *statistical uncertainty*, which is a view more common to statistics and machine learning, as it is more natural for noisy data. Here, the error is considered as inherently probabilistic due to small sample effects or noise fluctuation, and algorithms may be analyzed for their statistical properties, independent of whether they are themselves deterministic or stochastic. The statistical view on uncertainty is the one the reader should have in mind when reading this paper.

Parts of the algebra community have been committed to the numerical viewpoint on uncertain polynomials: the problem of numerical uncertainty is for example extensively addressed in Stetter's standard book on numerical algebra (Stetter, 2004). The main difficulties and innovations stem from the fact that standard methods from algebra like the application of Groebner bases are numerically unstable, see (Stetter, 2004, Chapter 4.1-2).

Recently, the algebraic geometry community has developed an increasing interest in solving algebraic problems arising from the consideration of real world data. The algorithms in this area are more motivated to perform well on the data, some authors start to adapt a statistical viewpoint on uncertainty, while the influence of the numerical view is still dominant. As a distinction, the authors describe the field as approximate algebra instead of numerical algebra. Recent developments in this sense can be found for example in Heldt et al. (2009) or the book of Kreuzer et al. (2009). We will refer to this viewpoint as the statistical view in order to avoid confusion with other meanings of approximate.

Interestingly, there are significant similarities on the methodological side. Namely, in computational algebra, algorithms often compute primarily over vector spaces, which arise for example as spaces of polynomials with certain properties. Here, numerical linear algebra can provide many techniques of enforcing numerical stability, see the pioneering paper of Corless et al. (1995). Since then, many algorithms in numerical and approximate algebra use linear optimization to estimate vector spaces of polynomials. In particular, least-squares-approximations of rank or kernel are canonical concepts in both numerical and approximate algebra.

However, to the best of our knowledge, there is to date no algorithm which computes an "approximate" (or "numerical") radical of an ideal, or an approximate saturation, and also none in our special case. In the next section, we will use estimation techniques from linear algebra to convert Algorithm 1 into an algorithm which can cope with the inherent statistical uncertainty of the estimation problem.

4. Approximate Algebraic Geometry on Real Data

In this section we show how algebraic computations can be applied to polynomials with inexact coefficients obtained from estimated cumulants on finite samples. Note that our method for computing the approximate radical is not specific to the problem studied in this paper.

The reason why we cannot directly apply our algorithm for the exact case to estimated polynomials is that it relies on the assumption that there exists an exact solution, such that the projected cumulants are equal, that is, we can find a projection P such that the equalities

$$P\Sigma_1 P^\top = \cdots = P\Sigma_m P^\top \quad \text{and} \quad P\mu_1 = \cdots = P\mu_m$$

hold exactly. However, when the elements of $\Sigma_1, \ldots, \Sigma_m$ and $\mu_1, \ldots, \mu_m$ are subject to random fluctuations or noise, there exists no projection that yields exactly the same random variables. In algebraic terms, working with inexact polynomials means that the joint vanishing set of $q_1, \ldots, q_{m-1}$

and $f_1,\dots,f_{m-1}$ consists only of the origin $0 \in \mathbb{C}^D$ so that the ideal becomes trivial:

$$\langle q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}\rangle = \mathfrak{m}.$$

Thus, in order to find a meaningful solution, we need to compute the radical approximately.

In the exact algorithm, we are looking for a polynomial of the form $T_D\ell$ vanishing on S, which is also a $\mathbb{C}$-linear combination of the quadratic forms q_i. The algorithm is based on an explicit way to do so which works since the q_i are generic and sufficient in number. So one could proceed to adapt this algorithm to the approximate case by performing the same operations as in the exact case and then taking the $(\Delta(D)-\Delta(d))$-th row, setting coefficients not divisible by T_D to zero, and then dividing out T_D to get a linear form. This strategy performs fairly well for small dimensions D and converges to the correct solution, albeit slowly.

Instead of computing one particular linear generator as in the exact case, it is advisable to use as much information as possible in order to obtain better accuracy. The least-squares-optimal way to approximate a linear space of known dimension is to use singular value decomposition (SVD): with this method, we may directly eliminate the most insignificant directions in coefficient space which are due to fluctuations in the input. To that end, we first define an approximation of an arbitrary matrix by a matrix of fixed rank.

Definition 26 *Let $A \in \mathbb{C}^{m\times n}$ with singular value decomposition $A = UDV^*$, where $D = \operatorname{diag}(\sigma_1,\dots,\sigma_p) \in \mathbb{C}^{p\times p}$ is a diagonal matrix with ordered singular values on the diagonal,*

$$|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_p| \geq 0.$$

For $k \leq p$, let $D' = \operatorname{diag}(\sigma_1,\dots,\sigma_k,0,\dots,0)$. Then the matrix $A' = UD'V^*$ is called rank k approximation of A. The null space, left null space, row span, column span of A' will be called rank k approximate null space, left null space, row span, column span of A.

For example, if $u_1,\dots,u_p$ and $v_1,\dots,v_p$ are the columns of U and V respectively, the rank k approximate left null space of A is spanned by the rows of the matrix

$$L = \begin{bmatrix} u_{p-k+1} & \cdots & u_p \end{bmatrix}^\top,$$

and the rank k approximate row span of A is spanned by the rows of the matrix

$$S = \begin{bmatrix} v_1 & \cdots & v_p \end{bmatrix}^\top.$$

We will call those matrices the *approximate left null space matrix* resp. the approximate row span matrix of rank k associated to A. The approximate matrices are the optimal approximations of rank k with respect to the least-squares error.

We can now use these concepts to obtain an approximative version of Algorithm 1. Instead of searching for a single element of the form $T_D\ell$, we estimate the vector space of all such elements via singular value decomposition—note that this is exactly the vector space $(\langle T_D\rangle \cdot \mathfrak{s})_2$, that is, the vector space of all homogenous polynomials of degree two which are divisible by T_D. Also note that the choice of the linear form T_D is irrelevant, that is, we may replace T_D above by any variable or even linear form. As a trade-off between accuracy and runtime, we additionally estimate the vector spaces $(\langle T_D\rangle \cdot \mathfrak{s})_2$ for all $1 \leq i \leq D$, and then least-squares average the putative results for $\mathfrak{s}$ to obtain a final estimator for $\mathfrak{s}$ and thus the desired space of projections.

Algorithm 2 The *input* consists of noisy quadratic forms $q_1,\dots,q_{m-1}\in\mathbb{C}[T_1,\dots,T_D]$, and the dimension d; the *output* is an approximate linear generating set $\ell_1,\dots,\ell_{D-d}$ for the ideal $\mathfrak{s}$.

```
1: Let $Q \leftarrow \begin{bmatrix} q_1 & \cdots & q_{m-1}\end{bmatrix}^\top$ be the $(m-1\times\Delta(D))$-matrix of coefficient vectors, where every row corresponds to a polynomial and every column to a monomial $T_iT_j$ in arbitrary order.
2: for $i=1,\dots,D$ do
3:     Let $Q_i$ be the $((m-1)\times\Delta(D)-D)$-sub-matrix of $Q$ obtained by removing all columns corresponding to monomials divisible by $T_i$
4:     Compute the approximate left null space matrix $L_i$ of $Q_i$ of rank $(m-1)-\Delta(D)+\Delta(d)+D-d$
5:     Compute the approximate row span matrix $L'_i$ of $L_iQ$ of rank $D-d$
6:     Let $L''_i$ be the $(D-d\times D)$-matrix obtained from $L'_i$ by removing all columns corresponding to monomials not divisible by $T_i$
7: end for
8: Let $L$ be the $(D(D-d)\times D)$-matrix obtained by vertical concatenation of $L''_1,\dots,L''_D$
9: Compute the approximate row span matrix $A=\begin{bmatrix} a_1 & \cdots & a_{D-d}\end{bmatrix}^\top$ of $L$ of rank $D-d$ and let $\ell_i=\begin{bmatrix} T_1 & \cdots & T_D\end{bmatrix}a_i$ for all $1\le i\le D-d$.
```

We explain the logic behind the single steps: in the first step, we start with the same matrix Q as in Algorithm 1. Instead of bringing Q into triangular form with respect to the term order $T_1\prec\cdots\prec T_D$, we compute the left kernel space row matrix S_i of the monomials not divisible by T_i. Its left image $L_i=S_iQ$ is a matrix whose row space generates the space of possible last rows after bringing Q into triangular form in an arbitrary coordinate system. In the next step, we perform PCA to estimate a basis for the so-obtained vector space of quadratic forms of type T_i times linear form, and extract a basis for the vector space of linear forms estimated via L_i. Now we can put together all L_i and again perform PCA to obtain a more exact and numerically more estimate for the projection in the last step. The rank of the matrices after PCA is always chosen to match the correct ranks in the exact case.

Note that Algorithm 2 is a consistent estimator for the correct space of projections if the covariances are sample estimates. Let us first clarify in which sense consistent is meant here: if each covariance matrix is estimated from a sample of size N or greater, and N goes to infinity, then the estimate of the projection converges in probability to the true projection. The reason why Algorithm 2 gives a consistent estimator in this sense is elementary: covariance matrices can be estimated consistently, and so can their differences, the polynomials q_i. Moreover, the algorithm can be regarded as an almost continuous function in the polynomials q_i; so convergence in probability to the true projection and thus consistency follows from the continuous mapping theorem.

The runtime complexity of Algorithm 2 is $O(D^6)$ as for Algorithm 1. For this note that calculating the singular value decomposition of an $m\times n$-matrix is $O(mn\max(m,n))$.

If we want to consider k-forms instead of 2-forms, we can use the same strategies as above to numerically stabilize the exact algorithm. In the second step, one might want to consider all sub-matrices Q_M of Q obtained by removing all columns corresponding to monomials divisible by some degree $(k-1)$ monomial M and perform the for-loop over all such monomials or a selection of them. Considering D monomials or more gives again a consistent estimator for the projection. Similarly, these methods allow us to numerically stabilize versions with reduced epoch requirements and simultaneous consideration of different degrees.

5. Numerical Evaluation

In this section we evaluate the performance of the algebraic algorithm on synthetic data in various settings. In order to contrast the algebraic approach with an optimization-based method (cf. Figure 1), we compare with the Stationary Subspace Analysis (SSA) algorithm (von Bünau et al., 2009), which solves a similar problem in the context of time series analysis; see Müller et al. (2011) for an open-source implementation. To date, SSA has been successfully applied in the context of biomedical data analysis (von Bünau et al., 2010), domain adaptation (Hara et al., 2010), change-point detection (Blythe et al., 2012) and computer vision (Meinecke et al., 2009).

5.1 Stationary Subspace Analysis

Stationary Subspace Analysis (von Bünau et al., 2009; Müller et al., 2011) factorizes an observed time series according to a linear model into underlying stationary and non-stationary sources. The observed time series $x(t) \in \mathbb{R}^D$ is assumed to be generated as a linear mixture of stationary sources $s^{\mathfrak{s}}(t) \in \mathbb{R}^d$ and non-stationary sources $s^{\mathfrak{n}}(t) \in \mathbb{R}^{D-d}$,

$$x(t) = As(t) = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix},$$

with a time-constant mixing matrix A. The underlying sources $s(t)$ are not assumed to be independent or uncorrelated.

The aim of SSA is to invert this mixing model given only samples from $x(t)$. The true mixing matrix A is not identifiable (von Bünau et al., 2009); only the projection $P \in \mathbb{R}^{d\times D}$ to the stationary sources can be estimated from the mixed signals $x(t)$, up to arbitrary linear transformation of its image. The estimated stationary sources are given by $\hat{s}^{\mathfrak{s}}(t) = Px(t)$, that is, the projection P eliminates all non-stationary contributions: $PA^{\mathfrak{n}} = 0$.

The SSA algorithms (von Bünau et al., 2009; Hara et al., 2010) are based on the following definition of stationarity: a time series X_t is considered stationary if its mean and covariance is constant over time, that is, $\mathbb{E}[X_{t_1}] = \mathbb{E}[X_{t_2}]$ and $\mathbb{E}[X_{t_1}X_{t_1}^\top] = \mathbb{E}[X_{t_2}X_{t_2}^\top]$ for all pairs of time points $t_1, t_2 \in \mathbb{N}$. Following this concept of stationarity, the projection P is found by minimizing the difference between the first two moments of the estimated stationary sources $\hat{s}^{\mathfrak{s}}(t)$ across epochs of the times series. To that end, the samples from $x(t)$ are divided into m non-overlapping epochs of equal size, corresponding to the index sets $\mathcal{T}_1, \ldots, \mathcal{T}_m$, from which the mean and the covariance matrix is estimated for all epochs $1 \leq i \leq m$,

$$\hat{\mu}_i = \frac{1}{|\mathcal{T}_i|}\sum_{t\in\mathcal{T}_i} x(t) \qquad \text{and} \qquad \hat{\Sigma}_i = \frac{1}{|\mathcal{T}_i|-1}\sum_{t\in\mathcal{T}_i} (x(t)-\hat{\mu}_i)(x(t)-\hat{\mu}_i)^\top .$$

Given a projection P, the mean and the covariance of the estimated stationary sources in the i-th epoch are given by $\hat{\mu}_i^{\mathfrak{s}} = P\hat{\mu}_i$ and $\hat{\Sigma}_i^{\mathfrak{s}} = P\hat{\Sigma}_i P^\top$ respectively. Without loss of generality (by centering and whitening[2] the average epoch) we can assume that $\hat{s}^{\mathfrak{s}}(t)$ has zero mean and unit covariance.

The objective function of the SSA algorithm (von Bünau et al., 2009) minimizes the sum of the differences between each epoch and the standard normal distribution, measured by the Kullback-

[2] A whitening transformation is a basis transformation W that sets the sample covariance matrix to the identity. It can be obtained from the sample covariance matrix $\hat{\Sigma}$ as $W = \hat{\Sigma}^{-\frac{1}{2}}$

Leibler divergence D_{KL} between Gaussians: the projection P^* is found as the solution to the optimization problem,

$$\begin{aligned} P^* &= \operatorname*{argmin}_{PP^\top = I} \sum_{i=1}^{m} D_{\mathrm{KL}} \left[\mathcal{N}(\hat{\mu}_i^{\mathfrak{s}}, \hat{\Sigma}_i^{\mathfrak{s}}) \,\middle\|\, \mathcal{N}(0, I) \right] \\ &= \operatorname*{argmin}_{PP^\top = I} \sum_{i=1}^{m} \left(-\log\det \hat{\Sigma}_i^{\mathfrak{s}} + (\hat{\mu}_i^{\mathfrak{s}})^\top \hat{\mu}_i^{\mathfrak{s}} \right), \end{aligned}$$

which is non-convex and solved using an iterative gradient-based procedure.

This SSA algorithm considers a problem that is closely related to the one addressed in this paper, because the underlying definition of stationarity does not consider the time structure. In essence, the m epochs are modeled as m random variables $X_1, \ldots, X_m$ for which we want to find a projection P such that the projected probability distributions $PX_1, \ldots, PX_m$ are equal, up to the first two moments. This problem statement is equivalent to the task that we solve algebraically.

5.2 Results

In our simulations, we investigate the influence of the noise level and the number of dimensions on the performance and the runtime of our algebraic algorithm and the SSA algorithm. We measure the performance using the subspace angle between the true and the estimated space of projections S.

The setup of the synthetic data is as follows: we fix the total number of dimensions to $D = 10$ and vary the dimension d of the subspace with equal probability distribution from one to nine. We also fix the number of random variables to $m = 110$. For each trial of the simulation, we need to choose a random basis for the two subspaces $\mathbb{R}^D = S \oplus S^\perp$, and for each random variable, we need to choose a covariance matrix that is identical only on S. Moreover, for each random variable, we need to choose a positive definite disturbance matrix (with given noise level σ), which is added to the covariance matrix to simulate the effect of finite or noisy samples.

The elements of the basis vectors for S and $S^\perp$ are drawn uniformly from the interval $(-1, 1)$. The covariance matrix of each epoch $1 \leq i \leq m$ is obtained from Cholesky factors with random entries drawn uniformly from $(-1, 1)$, where the first d rows remain fixed across epochs. This yields noise-free covariance matrices $C_1, \ldots, C_m \in \mathbb{R}^{D \times D}$ where the first $(d \times d)$-block is identical. Now for each C_i, we generate a random disturbance matrix E_i to obtain the final covariance matrix

$$C_i' = C_i + E_i.$$

The disturbance matrix E_i is determined as $E_i = V_i D_i V_i^\top$ where V_i is a random orthogonal matrix, obtained as the matrix exponential of an antisymmetric matrix with random elements and D_i is a diagonal matrix of eigenvalues. The noise level σ is the log-determinant of the disturbance matrix E_i. Thus the eigenvalues of D_i are normalized such that

$$\frac{1}{10} \sum_{i=1}^{10} \log D_{ii} = \sigma.$$

In the final step of the data generation, we transform the disturbed covariance matrices $C_1', \ldots, C_m'$ into the random basis to obtain the cumulants $\Sigma_1, \ldots, \Sigma_m$ which are the input to our algorithm.

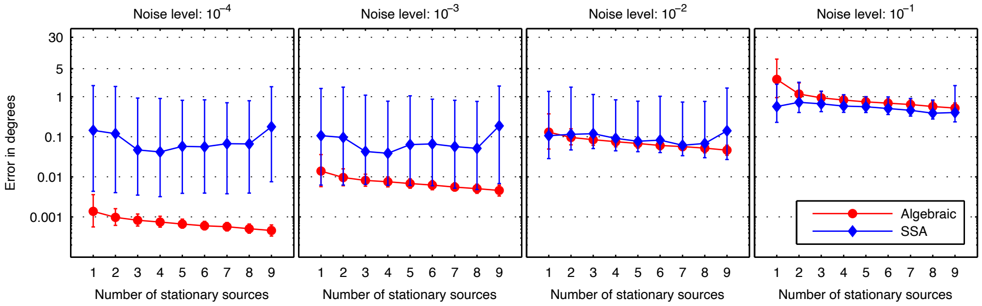


Figure 5: Comparison of the algebraic algorithm and the SSA algorithm. Each panel shows the median error of the two algorithms (vertical axis) for varying numbers of stationary sources in ten dimensions (horizontal axis). The noise level increases from the left to the right panel; the error bars extend from the 25% to the 75% quantile estimated over 2000 random realizations of the data set.

The first set of results is shown in Figure 5. With increasing noise levels (from left to right panel) both algorithms become worse. For low noise levels, the algebraic method yields significantly better results than the optimization-based approach, over all dimensionalities. For medium and high-noise levels, this situation is reversed.

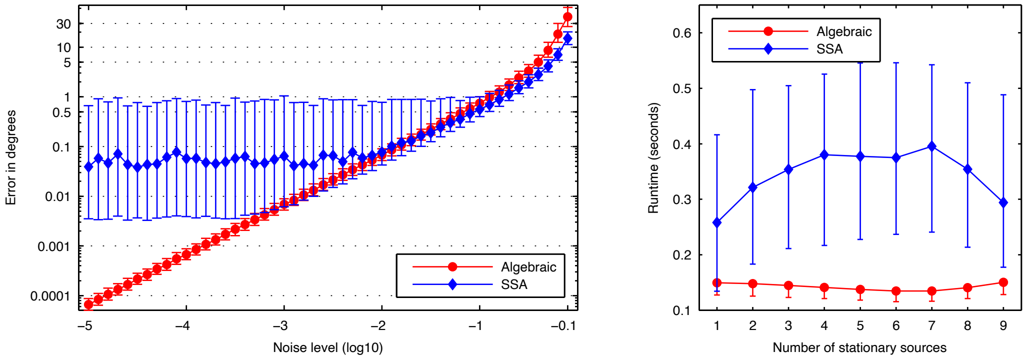


Figure 6: The left panel shows a comparison of the algebraic method and the SSA algorithm over varying noise levels (five stationary sources in ten dimensions), the two curves show the median log error. The right panel shows a comparison of the runtime for varying numbers of stationary sources. The error bars extend from the 25% to the 75% quantile estimated over 2000 random realizations of the data set.

In the left panel of Figure 6, we see that the error level of the algebraic algorithm decreases with the noise level, converging to the exact solution when the noise tends to zero. In contrast, the error of original SSA decreases with noise level, reaching a minimum error baseline which it cannot fall below. In particular, the algebraic method significantly outperforms SSA for low noise levels, whereas SSA is better for high noise. However, when noise is too high, none of the two algorithms can find the correct solution. In the right panel of Figure 6, we see that the algebraic method is significantly faster than SSA.

6. Conclusion

In this paper we have shown how a learning problem formulated in terms of cumulants of probability distributions can be addressed in the framework of computational algebraic geometry. As an example, we have demonstrated this viewpoint on the problem of finding a linear map $P \in \mathbb{R}^{d\times D}$ such that a set of projected random variables $X_1,\ldots,X_m \in \mathbb{R}^D$ have the same distribution,

$$PX_1 \sim \cdots \sim PX_m.$$

To that end, we have introduced the theoretical groundwork for an algebraic treatment of inexact cumulants estimated from data: the concept of polynomials that are *generic* up to a certain property which we aim to recover from the data. In particular, we have shown how we can find an approximate exact solution to this problem using algebraic manipulation of cumulants estimated on samples drawn from $X_1,\ldots,X_m$. Therefore we have introduced the notion of computing an *approximate saturation* of an ideal that is optimal in a least-squares sense. Moreover, using the algebraic problem formulation in terms of generic polynomials, we have presented compact proofs for a condition on the identifiability of the true solution.

In essence, instead of searching the surface of a non-convex objective function involving the cumulants, the algebraic algorithm directly finds the solution by manipulating cumulant polynomials—which is the more natural representation of the problem. This viewpoint is not only theoretically appealing but conveys practical advantages that we demonstrate in a numerical comparison to Stationary Subspace Analysis (von Bünau et al., 2009): the computational cost is significantly lower and the error converges to zero as the noise level goes to zero. However, the algebraic algorithm requires $m \geq \Delta(D)$ random variables with distinct distributions, which is quadratic in the number of dimensions D. This is due to the fact that the algebraic algorithm represents the cumulant polynomials in the vector space of coefficients. Consequently, the algorithm is confined to linearly combining the polynomials which describe the solution. However, the set of solutions is also invariant under multiplication of polynomials and polynomial division, that is, the algorithm does not use all information contained in the polynomial equations. We conjecture that we can construct a more efficient algorithm, if we also multiply and divide polynomials.

The theoretical and algorithmic techniques introduced in this paper can be applied to other scenarios in machine learning, including the following examples.

- **Finding properties of probability distributions.** Any inference problem that can be formulated in terms of polynomials, in principle, is amenable to our algebraic approach; incorporating polynomial constraints is also straightforward.

- **Approximate solutions to polynomial equations.** In machine learning, the problem of solving polynomial equations can, for example, occur in the context of finding the solution to a constrained nonlinear optimization problem by means of setting the gradient to zero.

- **Conditions for identifiability.** Whenever a machine learning problem can be formulated in terms of polynomials, identifiability of its generative model can also be phrased in terms of algebraic geometry, where a wealth of proof techniques stands at disposition.

We argue for a cross-fertilization of approximate computational algebra and machine learning: the former can benefit from the wealth of techniques for dealing with uncertainty and noisy data; the machine learning community may find a novel framework for representing learning problems that can be solved efficiently using symbolic manipulation.

Acknowledgments

We thank Marius Kloft and Jan Saputra Müller for valuable discussions. We are particularly grateful to Gert-Martin Greuel for his insightful remarks. We also thank Andreas Ziehe for proofreading the manuscript. This work has been supported by the Bernstein Cooperation (German Federal Ministry of Education and Science), Förderkennzeichen 01 GQ 0711, the Mathematisches Forschungsinstitut Oberwolfach (MFO) and the World Class University Program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology, under Grant R31-10008. This work was partly done while Klaus-Robert Müller was at IPAM, UCLA, Los Angeles, USA. A preprint version of this manuscript has appeared as part of the Oberwolfach Preprint series (Király et al., 2011).

Appendix A. An Example

In this section, we will show by using a concrete example how the Algorithms 1 and 2 work. The setup will be the similar to the example presented in the introduction. We will use the notation introduced in Section 3.

Example 27 In this example, let us consider the simplest non-trivial case: two random variables X_1, X_2 in $\mathbb{R}^2$ such that there is exactly one direction $w \in \mathbb{R}^2$ such that $w^\top X_1 = w^\top X_2$; that is, the total number of dimensions is $D = 2$, the dimension of the set of projections is $d = 1$. As in the beginning of Section 3, we may assume that $\mathbb{R}^2 = S \oplus S^\perp$ is an orthogonal sum of a one-dimensional space of projections S and its orthogonal complement $S^\perp$. In particular, $S^\perp$ is given as the linear span of a single vector, say $\begin{bmatrix}\alpha & \beta\end{bmatrix}^\top$. The space S is also the linear span of the vector $\begin{bmatrix}\beta & -\alpha\end{bmatrix}^\top$.

Now we partition the sample into $D(D+1)/2 - d(d+1)/2 = 2$ epochs (this is the lower bound needed by Proposition 22). From the two epochs we can estimate two covariance matrices $\hat{\Sigma}_1, \hat{\Sigma}_2$. Suppose we have

$$\hat{\Sigma}_1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

From this matrices, we can now obtain a polynomial

$$\begin{aligned} q_1 &= w^\top (\hat{\Sigma}_1 - I) w \\ &= w^\top \begin{bmatrix} a_{11} - 1 & a_{12} \\ a_{21} & a_{22} - 1 \end{bmatrix} w \\ &= (a_{11} - 1) T_1^2 + (a_{12} + a_{21}) T_1 T_2 + (a_{22} - 1) T_2^2, \end{aligned}$$

where $w = \begin{bmatrix} T_1 & T_2 \end{bmatrix}^\top$. Similarly, we obtain a polynomial q_2 as the Gram polynomial of $\hat{\Sigma}_2 - I$.

First we now illustrate how Algorithm 1, which works with homogenous exact polynomials, can determine the vector space S from these polynomials. For this, we assume that the estimated polynomials are exact; we will discuss the approximate case later. We can also write q_1 and q_2 in coefficient expansion:

$$\begin{aligned} q_1 &= q_{11}T_1^2 + q_{12}T_1T_2 + q_{13}T_2^2, \\ q_2 &= q_{21}T_1^2 + q_{22}T_1T_2 + q_{23}T_2^2. \end{aligned}$$

We can also write this formally in the (2×3) coefficient matrix $Q = (q_{ij})_{ij}$, where the polynomials can be reconstructed as the entries in the vector

$$Q \cdot \begin{bmatrix} T_1^2 & T_1T_2 & T_2^2 \end{bmatrix}^\top .$$

Algorithm 1 now calculates the upper triangular form of this matrix. For polynomials, this is equivalent to calculating the last row

$$\begin{aligned} & q_{21}q_1 - q_{11}q_2 \\ & = [q_{21}q_{12} - q_{11}q_{22}]T_1T_2 + [q_{21}q_{13} - q_{11}q_{23}]T_2^2. \end{aligned}$$

Then we divide out T_2 and obtain

$$P = [q_{21}q_{12} - q_{11}q_{22}]T_1 + [q_{21}q_{13} - q_{11}q_{23}]T_2.$$

The algorithm now identifies $S^\perp$ as the vector space spanned by the vector

$$\begin{bmatrix} \alpha & \beta \end{bmatrix}^\top = \begin{bmatrix} q_{21}q_{12} - q_{11}q_{22} & q_{21}q_{13} - q_{11}q_{23} \end{bmatrix}^\top .$$

This already finishes the calculation given by Algorithm 1, as we now explicitly know the solution

$$\begin{bmatrix} \alpha & \beta \end{bmatrix}^\top .$$

To understand why this strategy works, we need to have a look at the input. Namely, one has to note that q_1 and q_2 are generic homogenous polynomials of degree 2, vanishing on S. That is, we will have $q_i(x) = 0$ for $i = 1,2$ and all points $x \in S$. It is not difficult to see that every polynomial fulfilling this condition has to be of the form

$$(\alpha T_1 + \beta T_2)(aT_1 + bT_2)$$

for some $a, b \in \mathbb{C}$; that is, a multiple of the equation defining S. However we may not know this factorization a priori, in particular we are in general agnostic as to the correct values of α and β. They have to be reconstructed from the q_i via an algorithm. Nonetheless, a correct solution exists, so we may write

$$\begin{aligned} q_1 &= (\alpha T_1 + \beta T_2)(a_1X + b_1T_2), \\ q_2 &= (\alpha T_1 + \beta T_2)(a_2X + b_2T_2), \end{aligned}$$

with a_i, b_i generic, without knowing the exact values a priori. If we now compare to the above expansion in the q_{ij}, we obtain the linear system of equations

$$\begin{aligned} q_{i1} &= \alpha a_i, \\ q_{i2} &= \alpha b_i + \beta a_i, \\ q_{i3} &= \beta b_i \end{aligned}$$

for $i = 1,2$, from which we may reconstruct the a_i, b_i and thus α and β. However, a more elegant and general way of getting to the solution is to bring the matrix Q as above into triangular form. Namely, by assumption, the last row of this triangular form corresponds to the polynomial P which vanishes on S. Using the same reasoning as above, the polynomial P has to be a multiple of $(\alpha T_1 + \beta T_2)$. To check the correctness of the solution, we substitute the q_{ij} in the expansion of P for a_i, b_i, and obtain

$$\begin{aligned}P =&[q_{21}q_{12} - q_{11}q_{22}]T_1T_2 + [q_{21}q_{13} - q_{11}q_{23}]T_2^2\\ =&[\alpha a_2(\alpha b_1 + \beta a_1) - \alpha a_1(\alpha b_2 + \beta a_2)]T_1T_2 + [\alpha a_2\beta b_1 - \alpha a_1\beta b_2]T_2^2\\ =&[\alpha^2 a_2b_1 - \alpha^2 a_1b_2]T_1T_2 + [\alpha\beta a_2b_1 - \alpha\beta a_1b_2]T_2^2\\ =&(\alpha T_1 + \beta T_2)\alpha[a_2b_1 - a_1b_2]T_2.\end{aligned}$$

This is $(\alpha T_1 + \beta T_2)$ times T_2 up to a scalar multiple - from the coefficients of the form P, we may thus directly reconstruct the vector $\begin{bmatrix}\alpha & \beta\end{bmatrix}$ up to a common factor and thus obtain a representation for S, since the calculation of these coefficients did not depend on a priori knowledge about S.

If the estimation of the $\hat{\Sigma}_i$ and thus of the q_i is now endowed with noise, and we have more than two epochs and polynomials, Algorithm 2 provides the possibility to perform this calculation approximately. Namely, Algorithm 2 finds a linear combination of the q_i which is approximately of the form $T_D\ell$ with a linear form ℓ in the variables T_1, T_2. The Young-Eckart Theorem guarantees that we obtain a consistent and least-squares-optimal estimator for P, similarly to the exact case. The reader is invited to check this by hand as an exercise.

Now the observant reader may object that we may have simply obtained the linear form $(\alpha T_1 + \beta T_2)$ and thus S directly from factoring q_1 and q_2 and taking the unique common factor. Note however that this strategy can only be applied in the very special case $D - d = 1$. To illustrate the additional difficulties in the general case, we repeat the above example for $D = 4$ and $d = 2$ for the exact case:

Example 28 In this example, we need already $D(D+1)/2 - d(d+1)/2 = 7$ polynomials $q_1, \dots, q_7$ to solve the problem with Algorithm 1. As above, we can write

$$\begin{aligned}q_i =&q_{i1}T_1^2 + q_{i2}T_1T_2 + q_{i3}T_1T_3 + q_{i4}T_1T_4 + q_{i5}T_2^2\\ &+ q_{i6}T_2T_3 + q_{i7}T_2T_4 + q_{i8}T_3^2 + q_{i9}T_3T_4 + q_{i,10}T_4^2\end{aligned}$$

for $i = 1, \dots, 7$, and again we can write this in a (7×10) coefficient matrix $Q = (q_{ij})_{ij}$. In Algorithm 1, this matrix is brought into triangular form. The last row of this triangular matrix will thus correspond to a polynomial of the form

$$P = p_7T_2T_4 + p_8T_3^2 + p_9T_3T_4 + p_{10}T_4^2$$

A polynomial of this form is not divisible by T_4 in general. However, Proposition 22 guarantees us that the coefficient p_8 is always zero due to our assumptions. So we can divide out T_4 to obtain a linear form

$$p_7T_2 + p_9T_3 + p_{10}T_4.$$

This is one equation defining the linear space S. One obtains another equation in the variables T_1, T_2, T_3 if one, for example, inverts the numbering of the variables $1-2-3-4$ to $4-3-2-1$. Two equations suffice to describe S, and so Algorithm 1 yields the correct solution.

As in the example before, it can be checked by hand that the coefficient p_7 indeed vanishes, and the obtained linear equations define the linear subspace S. For this, one has to use the classical result from algebraic geometry that every q_i can be written as

$$q_i = \ell_1 P_1 + \ell_2 P_2,$$

where the ℓ_i are fixed but arbitrary linear forms defining S as their common zero set, and the P_i are some linear forms determined by q_i and the ℓ_i (this is for example a direct consequence of Hilbert's Nullstellensatz). Caution is advised as the equations involved become very lengthy - while not too complex - already in this simple example. So the reader may want to check only that the coefficient p_8 vanishes as claimed.

Appendix B. Algebraic Geometry of Genericity

In the paper, we have reformulated a problem of comparing probability distributions in algebraic terms. For the problem to be well-defined, we need the concept of genericity for the cumulants. The solution can then be determined as an ideal generated by generic homogenous polynomials vanishing on a linear subspace. In this supplement, we will extensively describe this property which we call genericity and derive some simple consequences.

Since genericity is an algebraic-geometric concept, knowledge about basic algebraic geometry will be required for an understanding of this section. In particular, the reader should be at least familiar with the following concepts before reading this section: polynomial rings, ideals, radicals, factor rings, algebraic sets, algebra-geometry correspondence (including Hilbert's Nullstellensatz), primary decomposition, height resp. dimension theory in rings. A good introduction into the necessary framework can be found in the book of Cox et al. (2007).

B.1 Definition of Genericity

In the algebraic setting of the paper, we would like to calculate the radical of an ideal

$$I = \langle q_1, \ldots, q_{m-1}, f_1, \ldots, f_{m-1} \rangle.$$

This ideal I is of a special kind: its generators are random, and are only subject to the constraints that they vanish on the linear subspace S to which we project, and that they are homogenous of fixed degree. In order to derive meaningful results on how I relates to S, or on the solvability of the problem, we need to model this kind of randomness.

In this section, we introduce a concept called genericity. Informally, a generic situation is a situation without pathological degeneracies. In our case, it is reasonable to believe that apart from the conditions of homogeneity and the vanishing on S, there are no additional degeneracies in the choice of the generators. So, informally spoken, the ideal I is generated by generic homogenous elements vanishing on S. This section is devoted to developing a formal theory in order to address such generic situations efficiently.

The concept of genericity is already widely used in theoretical computer science, combinatorics or discrete mathematics; there, it is however often defined inexactly or not at all, or it is only given as an ad-hoc definition for the particular problem. On the other hand, genericity is a classical concept in algebraic geometry, in particular in the theory of moduli. The interpretation of generic properties as probability-one-properties is also a known concept in applied algebraic geometry, for example, algebraic statistics. However, the application of probability distributions and genericity to the setting of generic ideals, in particular in the context of conditional probabilities, are original to the best of our knowledge, though not being the first one to involve generic resp. general polynomials, see Iarrobino (1984). Generic polynomials and ideals have been also studied by Fröberg and Hollman (1994). A collection of results on generic polynomials and ideals which partly overlap with ours may also be found in the recent paper of Pardue (2010).

Before continuing to the definitions, let us explain what genericity should mean. Intuitively, generic objects are objects without unexpected pathologies or degeneracies. For example, if one studies say n lines in the real plane, one wants to exclude pathological cases where lines lie on each other or where many lines intersect in one point. Having those cases excluded means examining the "generic" case, that is, the case where there are $n(n+1)/2$ intersections, $n(n+1)$ line segments and so forth. Or when one has n points in the plane, one wants to exclude the pathological cases where for example there are three affinely dependent points, or where there are more sophisticated algebraic dependencies between the points which one wants to exclude, depending on the problem.

In the points example, it is straightforward how one can define genericity in terms of sampling from a probability distribution: one could draw the points under a suitable continuous probability distribution from real two-space. Then, saying that the points are "generic" just amounts to examine properties which are true with probability one for the n points. Affine dependencies for example would then occur with probability zero and are automatically excluded from our interest. One can generalize this idea to the lines example: one can parameterize the lines by a parameter space, which in this case is two-dimensional (slope and ordinate), and then sample lines uniformly distributed in this space (one has of course to make clear what this means). For example, lines lying on each other or more than two lines intersecting at a point would occur with probability zero, since the part of parameter space for this situation would have measure zero under the given probability distribution.

When we work with polynomials and ideals, the situation gets a bit more complicated but the idea is the same. Polynomials are uniquely determined by their coefficients, so they can naturally be considered as objects in the vector space of their coefficients. Similarly, an ideal can be specified by giving the coefficients of some set of generators. Let us make this more explicit: suppose first we have given a single polynomial $f \in \mathbb{C}[X_1, \dots X_D]$ of degree k.

In multi-index notation, we can write this polynomial as a finite sum

$$f = \sum_{\alpha \in \mathbb{N}^D} c_\alpha X^\alpha \quad \text{with } c_\alpha \in \mathbb{C}.$$

This means that the possible choices for f can be parameterized by the $\binom{D+k}{k}$ coefficients c_I with $\|I\|_1 \le k$. Thus polynomials of degree k with complex coefficients can be parameterized by complex $\binom{D+k}{k}$-space.

Algebraic sets can be similarly parameterized by parameterizing the generators of the corresponding ideal. However, this correspondence is highly non-unique, as different generators may give rise to the same zero set. While the parameter space can be made unique by dividing out redundancies, which gives rise to the Hilbert scheme, we will instead use the redundant, though pragmatic characterization in terms of a finite dimensional vector space over $\mathbb{C}$ of the correct dimension.

We will now fix notation for the parameter space of polynomials and endow it with algebraic structure. The extension to ideals will then be derived later. Let us write $\mathcal{M}_k$ for complex $\binom{D+k}{k}$-space (we assume D as fixed), interpreting it as a parameter space for the polynomials of degree k as shown above. Since the parameter space $\mathcal{M}_k$ is isomorphic to complex $\binom{D+k}{k}$-space, we may speak about algebraic sets in $\mathcal{M}_k$. Also, $\mathcal{M}_k$ carries the complex topology induced by the topology on $\mathbb{R}^{2k}$ and by topological isomorphy the Lebesgue measure; thus it also makes sense to speak about probability distributions and random variables on $\mathcal{M}_k$. This dual interpretation will be the main ingredient in our definition of genericity, and will allow us to relate algebraic results on genericity to the probabilistic setting in the applications. As $\mathcal{M}_k$ is a topological space, we may view any algebraic set in $\mathcal{M}_k$ as an event if we randomly choose a polynomial in $\mathcal{M}_k$:

Definition 29 *Let X be a random variable with values in $\mathcal{M}_k$. Then an event for X is called algebraic event or algebraic property if the corresponding event set in $\mathcal{M}_k$ is an algebraic set. It is called irreducible if the corresponding event set in $\mathcal{M}_k$ is an irreducible algebraic set.*

If an event A is irreducible, this means that if we write A as the event "A_1 and A_2", for algebraic events A_1, A_2, then $A = A_1$, or $A = A_2$. We now give some examples for algebraic properties.

Example 30 The following events on $\mathcal{M}_k$ are algebraic:

1. The sure event.
2. The empty event.
3. The polynomial is of degree n or less.
4. The polynomial vanishes on a prescribed algebraic set.
5. The polynomial is contained in a prescribed ideal.
6. The polynomial is homogenous.
7. The polynomial is a square.
8. The polynomial is reducible.

Properties 1-5 are additionally irreducible.

We now show how to prove these claims: 1-2 are clear, we first prove that properties 3-5 are algebraic and irreducible. By definition, it suffices to prove that the subset of $\mathcal{M}_k$ corresponding to those polynomials is an irreducible algebraic set. We claim: in any of those cases, the subset in question is moreover a linear subspace, and thus algebraic and irreducible. This can be easily verified by checking directly that if f_1, f_2 fulfill the property in question, then $f_1 + \alpha f_2$ also fulfills the property.

Property 6 is algebraic, since it can be described as the disjunction of the properties "The polynomial is homogenous and of degree n" for all $n \leq k$. Those single properties can be described by linear subspaces of $\mathcal{M}_k$ as above, thus property 6 is parameterized by the union of those linear subspaces. In general, these are orthogonal, so property 6 is not irreducible.

Property 7 is algebraic, as we can check it through the vanishing of a system of generalized discriminant polynomials. One can show that it is also irreducible since the subset of $\mathcal{M}_k$ in question corresponds to the image of a Veronese map (homogenization to degree k is a strategy); however, since we will not need such a result, we do not prove it here.

Property 8 is algebraic, since factorization can also be checked by sets of equations. One has to be careful here though, since those equations depend on the degrees of the factors. For example, a polynomial of degree 4 may factor into two polynomials of degree 1 and 3, or in two polynomials of degree 2 each. Since in general each possible combination defines different sets of equations and thus different algebraic subsets of $\mathcal{M}_k$, property 8 is in general not irreducible (for $k \leq 3$ it is).

The idea defining a choice of polynomial as generic follows the intuition of the affirmed non-sequitur: a generic, resp. generically chosen polynomial should not fulfill any algebraic property. A generic polynomial, having a particular simple (i.e., irreducible) algebraic property, should not fulfill any other algebraic property which is not logically implied by the first one. Here, algebraic properties are regarded as the natural model for restrictive and degenerate conditions, while their logical negations are consequently interpreted as generic, as we have seen in Example 30. These considerations naturally lead to the following definition of genericity in a probabilistic context:

Definition 31 *Let X be a random variable with values in $\mathcal{M}_k$. Then X is called generic, if for any irreducible algebraic events A,B, the following holds:*
The conditional probability $P_X(A|B)$ exists and vanishes if and only if B does not imply A.

In particular, B may also be the sure event.

Note that without giving a further explication, the conditional probability $P_X(A|B)$ is not well-defined, since we condition on the event B which has probability zero. There is also no unique way of remedying this, as for example the Borel-Kolmogorov paradox shows. In Section B.2, we will discuss the technical notion which we adopt to ensure well-definedness.

Intuitively, our definition means that an event has probability zero to occur unless it is logically implied by the assumptions. That is, degenerate dependencies between events do not occur.

For example, non-degenerate multivariate Gaussian distributions or Gaussian mixture distributions on $\mathcal{M}_k$ are generic distributions. More general, any positive continuous probability distribution which can be approximated by Gaussian mixtures is generic (see Example 37). Thus we argue that non-generic random variables are very pathological cases. Note however, that our intention is primarily not to analyze the behavior of particular fixed generic random variables (this is part of classical statistics). Instead, we want to infer statements which follow not from the particular structure of the probability function but solely from the fact that it is generic, as these statements are intrinsically implied by the conditional postulate in Definition 31 alone. We will discuss the definition of genericity and its implications in more detail in Section B.2.

With this definition, we can introduce the terminology of a generic object: it is a generic random variable which is object-valued.

Definition 32 *We call a generic random variable with values in $\mathcal{M}_k$ a generic polynomial of degree k. When the degree k is arbitrary but fixed (and still ≥ 1), we will say that f is a generic polynomial, or that f is generic, if it is clear from the context that f is a polynomial. If the degree k is zero, we will analogously say that f is a generic constant.*

We call a set of constants or polynomials $f_1,\ldots,f_m$ generic if they are generic and independent.

We call an ideal generic if it is generated by a set of m generic polynomials.

We call an algebraic set generic if it is the vanishing set of a generic ideal.

Let $\mathcal{P}$ be an algebraic property on a polynomial, a set of polynomials, an ideal, or an algebraic set (e.g., homogenous, contained in an ideal et.). We will call a polynomial, a set of polynomials, or an ideal, a generic $\mathcal{P}$ polynomial, set, or ideal, if it the conditional of a generic random variable with respect to $\mathcal{P}$.

If $\mathcal{A}$ is a statement about an object (polynomial, ideal etc), and $\mathcal{P}$ an algebraic property, we will say briefly "A generic $\mathcal{P}$ object is $\mathcal{A}$" instead of saying "A generic $\mathcal{P}$ object is $\mathcal{A}$ with probability one".

Note that formally, these objects are all polynomial, ideal, algebraic set etc -valued random variables. By convention, when we state something about a generic object, this will be an implicit probability-one statement. For example, when we say

"A generic green ideal is blue",

this is an abbreviation for the by definition equivalent but more lengthy statement

"Let $f_1, \dots, f_m$ be independent generic random variables with values in $\mathcal{M}_{k_1}, \dots, \mathcal{M}_{k_m}$. If the ideal $\langle f_1, \dots, f_m \rangle$ is green, then with probability one, it is also blue - this statement is independent of the choice of the k_i and the choice of which particular generic random variables we use to sample.

On the other hand, we will use the verb "generic" also as a qualifier for "constituting generic distribution". So for example, when we say

"The Z of a generic red polynomial is a generic yellow polynomial",

this is an abbreviation of the statement

"Let X be a generic random variable on $\mathcal{M}_k$, let X' be the yellow conditional of X. Then the Z of X' is the red conditional of some generic random variable - in particular this statement is independent of the choice of k and the choice of X."

It is important to note that the respective random variables will not be made explicit in the following subsections, since the statements will rely only on its property of being generic, and not on its particular structure which goes beyond being generic.

As an application of these concepts, we may now formulate the problem of comparing cumulants in terms of generic algebra:

Problem 33 *Let $\mathfrak{s} = \operatorname{I}(S)$, where S is an unknown d-dimensional subspace of $\mathbb{C}^D$. Let*

$$I = \langle f_1, \dots, f_m \rangle$$

with $f_i \in \mathfrak{s}$ generic of fixed degree each (in our case, one and two), such that $\sqrt{I} = \mathfrak{s}$.

Then determine a reduced Groebner basis (or another simple generating system) for $\mathfrak{s}$.

As we will see, genericity is the right concept to model random sampling of polynomials, as we will derive special properties of the ideal I which follow from the genericity of the f_i.

B.2 Zero-Measure Conditionals, and Relation to Other Types of Genericity

In this section, se will discuss the definition of genericity in Definition 31 and ensure its well-definedness. Then we will invoke alternative definitions for genericity and show their relation to our probabilistic intuitive approach from section B.1. As this section contains technical details and is not necessary for understanding the rest of the appendix, the reader may opt to skip it.

An important concept in our definition of genericity in Definition 31 is the conditional probability $P_X(A|B)$. As B is an algebraic set, its probability $P_X(B)$ is zero, so the Bayesian definition of conditional cannot apply. There are several ways to make it well-defined; in the following, we explain the Definition of conditional we use in Definition 31. The definition of conditional we use is one which is also often applied in this context.

Remark 34 *Let X be a real random variable (e.g., with values in $\mathcal{M}_k$) with probability measure μ. If μ is absolutely continuous, then by the theorem of Radon-Nikodym, there is a unique continuous density p such that*

$$\mu(U) = \int_U p\, d\lambda$$

for any Borel-measurable set U and the Lebesgue measure λ. If we assume that p is a continuous function, it is unique, so we may define a restricted measure μ_B on the event set of B by setting

$$\nu(U) = \int_U p\, dH,$$

for Borel subsets of U and the Hausdorff measure H on B. If $\nu(B)$ is finite and non-zero, that is, ν is absolutely continuous with respect to H, then it can be renormalized to yield a conditional probability measure $\mu(.)|_B = \nu(.)/\nu(B)$. The conditional probability $P_X(A|B)$ has then to be understood as

$$P_X(A|B) = \int_B \mathbb{1}(A \cap B)\, d\mu\, |_B,$$

whose existence in particular implies that the Lebesgue integrals $\nu(B)$ are all finite and non-zero.

As stated, we adopt this as the definition of conditional probability for algebraic sets A and B. It is important to note that we have made implicit assumptions on the random variable X by using the conditionals $P_X(A|B)$ in Remark 34 (and especially by assuming that they exist): namely, the existence of a continuous density function and existence, finiteness, and non-vanishing of the Lebesgue integrals. Similarly, by stating Definition 31 for genericity, we have made similar assumptions on the generic random variable X, which can be summarized as follows:

Assumption 35 *X is an absolutely continuous random variable with continuous density function p, and for every algebraic event B, the Lebesgue integrals*

$$\int_B p\, dH,$$

where H is the Hausdorff measure on B, are non-zero and finite.

This assumption implies the existence of all conditional probabilities $P_X(A|B)$ in Definition 31, and are also necessary in the sense that they are needed for the conditionals to be well-defined. On the other hand, if those assumptions are fulfilled for a random variable, it is automatically generic:

Remark 36 *Let X be a $\mathcal{M}_k$-valued random variable, fulfilling the Assumptions in 35. Then, the probability density function of X is strictly positive. Moreover, X is a generic random variable.*

Proof Let X be a $\mathcal{M}_k$-valued random variable fulfilling the Assumptions in 35. Let p be its continuous probability density function.

We first show positivity: if X would not be strictly positive, then p would have a zero, say x. Taking $B = \{x\}$, the integral $\int_B p\, dH$ vanishes, contradicting the assumption.

Now we prove genericity, that is, that for arbitrary irreducible algebraic properties A, B such that B does not imply A, the conditional probability $P_X(A|B)$ vanishes. Since B does not imply A, the algebraic set defined by B is not contained in A. Moreover, as B and A are irreducible and algebraic, $A \cap B$ is also of positive codimension in B. Now by assumption, X has a positive continuous probability density function f which by assumption restricts to a probability density on B, being also positive and continuous. Thus the integral

$$P_X(A|B) = \int_B \mathbb{1}_A f(x)\, dH,$$

where H is the Hausdorff measure on B, exists. Moreover, it is zero, as we have derived that A has positive codimension in B. ■

This means that already under mild assumptions, which merely ensure well-definedness of the statement in the Definition 31 of genericity, random variables are generic. The strongest of the comparably mild assumptions are the convergence of the conditional integrals, which allow us to renormalize the conditionals for all algebraic events. In the following example, a generic and a non-generic probability distribution are presented.

Example 37 Gaussian distributions and Gaussian mixture distributions are generic, since for any algebraic set B, we have

$$\int_B \mathbb{1}_{\mathcal{B}(t)}\, dH = O(t^{\dim B}),$$

where $\mathcal{B}(t) = \{x \in \mathbb{R}^n \ ; \ \|x\| < t\}$ is the open disc with radius t. Note that this particular bound is false in general and may grow arbitrarily large when we omit B being algebraic, even if B is a smooth manifold. Thus $P_X(A|B)$ is bounded from above by an integral (or a sum) of the type

$$\int_0^\infty \exp(-t^2) t^a\, dt \quad \text{with } a \in \mathbb{N}$$

which is known to be finite.

Furthermore, sums of generic distributions are again generic; also, one can infer that any continuous probability density dominated by the distribution of a generic density defines again a generic distribution.

An example of a non-generic but smooth distribution is given by the density function

$$p(x,y) = \frac{1}{\mathcal{N}} e^{-x^4 y^4}$$

where $\mathcal{N}$ is some normalizing factor. While p is integrable on $\mathbb{R}^2$, its restriction to the coordinate axes $x = 0$ and $y = 0$ is constant and thus not integrable.

Now we will examine different known concepts of genericity and relate them briefly to the one we have adopted.

A definition of genericity in combinatorics and geometry which can be encountered in different variations is that there exist no degenerate interpolating functions between the objects:

Definition 38 *Let $P_1,\dots,P_m$ be points in the vector space $\mathbb{C}^n$. Then $P_1,\dots,P_m$ are general position (or generic, general) if no $n+1$ points lie on a hyperplane. Or, in a stronger version: for any $d\in\mathbb{N}$, no (possibly inhomogenous) polynomial of degree d vanishes on $\binom{n+d}{d}+1$ different P_i.*

As $\mathcal{M}_k$ is a finite dimensional $\mathbb{C}$-vector space, this definition is in principle applicable to our situation. However, this definition is deterministic, as the P_i are fixed and no random variables, and thus preferable when making deterministic statements. Note that the stronger definition is equivalent to postulating general position for the points $P_1,\dots,P_m$ in any polynomial kernel feature space.

Since not lying on a hyperplane (or on a hypersurface of degree d) in $\mathbb{C}^n$ is a non-trivial algebraic property for any point which is added beyond the n-th (resp. the $\binom{n+d}{d}$-th) point P_i (interpreted as polynomial in $\mathcal{M}_k$), our definition of genericity implies general position. This means that generic polynomials $f_1,\dots,f_m\in\mathcal{M}_k$ (almost surely) have the deterministic property of being in general position as stated in Definition 38. A converse is not true for two reasons: first, the P_i are fixed and no random variables. Second, even if one would define genericity in terms of random variables such that the hyperplane (resp. hypersurface) conditions are never fulfilled, there are no statements made on conditionals or algebraic properties other than containment in a hyperplane, also Lebesgue zero sets are not excluded from occurring with positive probability.

Another example where genericity classically occurs is algebraic geometry, where it is defined rather general for moduli spaces. While the exact definition may depend on the situation or the particular moduli space in question, and is also not completely consistent, in most cases, genericity is defined as follows: general, or generic, properties are properties which hold on a Zariski-open subset of an (irreducible) variety, while very generic properties hold on a countable intersection of Zariski-open subsets (which are thus paradoxically ”less” generic than general resp. generic properties in the algebraic sense, as any general resp. generic property is very generic, while the converse is not necessarily true). In our special situation, which is the affine parameter space of tuples of polynomials, these definitions can be rephrased as follows:

Definition 39 *Let $B\subseteq\mathbb{C}^k$ be an irreducible algebraic set, let $P=(f_1,\dots,f_m)$ be a tuple of polynomials, viewed as a point in the parameter space B. Then a statement resp. property A of P is called very generic if it holds on the complement of some countable union of algebraic sets in B. A statement resp. property A of P is called general (or generic) if it holds on the complement of some finite union of algebraic sets in B.*

This definition is more or less equivalent to our own; however, our definition adds the practical interpretation of generic/very generic/general properties being true with probability one, while their negations are subsequently true with probability zero. In more detail, the correspondence is as follows: If we restrict ourselves only to algebraic properties A, it is equivalent to say that the property A is very generic, or general for the P in B, and to say with our original definition that a generic P fulfilling B is also A; since if A is by assumption an algebraic property, it is both an algebraic set and a complement of a finite (countable) union of algebraic sets in an irreducible algebraic set, so A must be equal to an irreducible component of B; since B is irreducible, this implies equality of

A and B. On the other hand, if A is an algebraic property, it is equivalent to say that the property not-A is very generic, or general for the P in B, and to say with our original definition that a generic P fulfilling B is not A - this corresponds intuitively to the probability-zero condition $P(A|B) = 0$ which states that non-generic cases do not occur. Note that by assumption, not-A is then always the complement of a finite union of algebraic sets.

B.3 Arithmetic of Generic Polynomials

In this subsection, we study how generic polynomials behave under classical operations in rings and ideals. This will become important later when we study generic polynomials and ideals.

To introduce the reader to our notation of genericity, and since we will use the presented facts and similar notations implicitly later, we prove the following

Lemma 40 *Let $f \in \mathbb{C}[X_1, \dots, X_D]$ be generic of degrees k. Then:*
(i) *The product αf is generic of degree k for any fixed $\alpha \in \mathbb{C} \setminus \{0\}$.*
(ii) *The sum $f + g$ is generic of degree k for any $g \in \mathbb{C}[X_1, \dots, X_D]$ of degree k or smaller.*
(iii) *The sum $f + g$ is generic of degree k for any generic $g \in \mathbb{C}[X_1, \dots, X_D]$ of degree k or smaller.*

Proof (i) is clear since the coefficients of g_1 are multiplied only by a constant. (ii) follows directly from the definitions since adding a constant g only shifts the coefficients without changing genericity. (iii) follows since f, g are independently sampled: if there were algebraic dependencies between the coefficients of $f + g$, then either f or g was not generic, or the f, g are not independent, which both would be a contradiction to the assumption. ■

Recall again what this Lemma means: for example, Lemma 40 (i) does not say, as one could think:

"Let X be a generic random variable with values in the vector space of degree k polynomials. Then $X = \alpha X$ for any $\alpha \in \mathbb{C} \setminus \{0\}$."

The correct translation of Lemma 40 (i) is:

"Let X be a generic random variable with values in the vector space of degree k polynomials. Then $X' = \alpha X$ for any fixed $\alpha \in \mathbb{C} \setminus \{0\}$ is a generic random variable with values in the vector space of degree k polynomials"

The other statements in Lemma 40 have to be interpreted similarly.

The following remark states how genericity translates through dehomogenization:

Lemma 41 *Let $f \in \mathbb{C}[X_1, \dots, X_D]$ be a generic homogenous polynomial of degree d.
Then the dehomogenization $f(X_1, \dots, X_{D-1}, 1)$ is a generic polynomial of degree d in the polynomial ring $\mathbb{C}[X_1, \dots, X_{D-1}]$.*

Similarly, let $\mathfrak{s} \trianglelefteq \mathbb{C}[X_1,\dots,X_D]$ be a generic homogenous ideal. Let $f \in \mathfrak{s}$ be a generic homogenous polynomial of degree d.
Then the dehomogenization $f(X_1,\dots,X_{D-1},1)$ is a generic polynomial of degree d in the dehomogenization of $\mathfrak{s}$.

Proof For the first statement, it suffices to note that the coefficients of a homogenous polynomial of degree d in the variables $X_1,\dots,X_D$ are in bijection with the coefficients of a polynomial of degree d in the variables $X_1,\dots,X_{D-1}$ by dehomogenization. For the second part, recall that the dehomogenization of $\mathfrak{s}$ consists exactly of the dehomogenizations of elements in $\mathfrak{s}$. In particular, note that the homogenous elements of $\mathfrak{s}$ of degree d are in bijection to the elements of degree d in the dehomogenization of $\mathfrak{s}$. The claims then follows from the definition of genericity. ■

B.4 Generic Spans and Generic Height Theorem

In this subsection, we will derive the first results on generic ideals. We will derive an statement about spans of generic polynomials, and generic versions of Krull's principal ideal and height theorems which will be the main tool in controlling the structure of generic ideals. This has immediate applications for the cumulant comparison problem.

Now we present the first result which can be easily formulated in terms of genericity:

Proposition 42 *Let P be an algebraic property such that the polynomials with property P form a vector space V. Let $f_1,\dots,f_m \in \mathbb{C}[X_1,\dots X_D]$ be generic polynomials satisfying P. Then*

$$\operatorname{rank}\operatorname{span}(f_1,\dots,f_m) = \min(m,\dim V).$$

Proof It suffices to prove: if $i \le M$, then f_i is linearly independent from $f_1,\dots f_{i-1}$ with probability one. Assuming the contrary would mean that for some i, we have

$$f_i = \sum_{k=0}^{i-1} f_k c_k \quad \text{for some } c_k \in \mathbb{C},$$

thus giving several equations on the coefficients of f_i. But these are fulfilled with probability zero by the genericity assumption, so the claim follows. ■

This may be seen as a straightforward generalization of the statement: the span of n generic points in $\mathbb{C}^D$ has dimension $\min(n,D)$.

We now proceed to another nontrivial result which will now allow us to formulate a generic version of Krull's principal ideal theorem:

Proposition 43 *Let $Z \subseteq \mathbb{C}^D$ be a non-empty algebraic set, let $f \in \mathbb{C}[X_1,\dots X_D]$ generic. Then f is no zero divisor in $O(Z) = \mathbb{C}[X_1,\dots X_D]/\mathrm{I}(Z)$.*

Proof We claim: being a zero divisor in $O(Z)$ is an irreducible algebraic property. We will prove that the zero divisors in $O(Z)$ form a linear subspace of $\mathcal{M}_k$, and linear spaces are irreducible.

For this, one checks that sums and scalar multiples of zero divisors are also zero divisors: if g_1, g_2 are zero divisors, there must exist h_1, h_2 such that $g_1 h_1 = g_2 h_2 = 0$. Now for any $\alpha \in \mathbb{C}$, we have that

$$(g_1 + \alpha g_2)(h_1 h_2) = (g_1 h_1) h_2 + (g_2 h_2) \alpha h_1 = 0.$$

This proves that $(g_1 + \alpha g_2)$ is also a zero divisor, proving that the zero divisors form a linear subspace and thus an irreducible algebraic property.

To apply the genericity assumption to argue that this event occurs with probability zero, we must exclude the possibility that being a zero divisor is trivial, that is, always the case. This is equivalent to proving that the linear subspace has positive codimension, which is true if and only if there exists a non-zero divisor in $O(Z)$. But a non-zero divisor always exists since we have assumed Z is non-empty: thus $\mathrm{I}(Z)$ is a proper ideal, and $O(Z)$ contains $\mathbb{C}$, which contains a non-zero divisor, for example, the one element.

So by the genericity assumption, the event that f is a zero divisor occurs with probability zero, that is, a generic f is not a zero divisor. Note that this does not depend on the degree of f. ■

Note that this result is already known, compare Conjecture B in Pardue (2010).

A straightforward generalization using the same proof technique is given by the following

Corollary 44 *Let $I \trianglelefteq \mathbb{C}[X_1, \ldots, X_D]$, let P be a non-trivial algebraic property. Let $f \in \mathbb{C}[X_1, \ldots X_D]$ be a generic polynomial with property P. If one can write $f = f' + c$, where f' is a generic polynomial subject to some property P', and c is a generic constant, then f is no zero divisor in $\mathbb{C}[X_1, \ldots, X_D]/I$.*

Proof First note that f is a zero divisor in $\mathbb{C}[X_1, \ldots, X_D]/I$ if and only if f is a zero divisor in $\mathbb{C}[X_1, \ldots, X_D]/\sqrt{I}$. This allows us to reduce to the case that $I = \mathrm{I}(Z)$ for some algebraic set $Z \subseteq \mathbb{C}^D$.

Now, as in the proof of Proposition 43, we see that being a zero divisor in $O(Z)$ is an irreducible algebraic property and corresponds to a linear subspace of $\mathcal{M}_k$, where $k = \deg f$. The zero divisors with property P are thus contained in this linear subspace. Now let f be generic with property P as above. By assumption, we may write $f = f' + c$. But c is (generically) no zero divisor, so f is also not a zero divisor, since the zero divisors form a linear subspace of $\mathcal{M}_k$. Thus f is no zero divisor. This proves the claim. ■

Note that Proposition 43 is actually a special case of Corollary 44, since we can write any generic polynomial f as $f' + c$, where f' is generic of the same degree, and c is a generic constant.

The major tool to deal with the dimension of generic intersections is Krull's principal ideal theorem:

Theorem 45 (Krull's principal ideal theorem) *Let R be a commutative ring with unit, let $f \in R$ be non-zero and non-invertible. Then*

$$\mathrm{ht}\langle f \rangle \leq 1,$$

with equality if and only if f is not a zero divisor in R.

The reader unfamiliar with height theory may take

$$\mathrm{ht}\, I = \mathrm{codim}\, \mathrm{V}(I)$$

as the definition for the height of an ideal (caveat: codimension has to be taken in R).

Reformulated geometrically for our situation, Krull's principal ideal theorem implies:

Corollary 46 *Let Z be a non-empty algebraic set in $\mathbb{C}^D$.Then*

$$\operatorname{codim}(Z \cap \mathrm{V}(f)) \le \operatorname{codim} Z + 1.$$

Proof Apply Krull's principal ideal theorem to the ring $R = O(Z) = \mathbb{C}[X_1, \dots, X_D]/\mathrm{I}(Z)$. ■

Together with Proposition 43, one gets a generic version of Krull's principal ideal theorem:

Theorem 47 (Generic principal ideal theorem) *Let Z be a non-empty algebraic set in $\mathbb{C}^D$, let $R = O(Z)$, and let $f \in \mathbb{C}[X_1, \dots, X_D]$ be generic. Then we have*

$$\operatorname{ht}\langle f \rangle = 1.$$

In its geometric formulation, we obtain the following result.

Corollary 48 *Consider an algebraic set $Z \subseteq \mathbb{C}^D$, and the algebraic set $\mathrm{V}(f)$ for some generic $f \in \mathbb{C}[X_1, \dots, X_D]$. Then*

$$\operatorname{codim}(Z \cap \mathrm{V}(f)) = \min(\operatorname{codim} Z + 1,\ D + 1).$$

Proof This is just a direct reformulation of Theorem 47 in the vein of Corollary 46. The only additional thing that has to be checked is the case where $\operatorname{codim} Z = D + 1$, which means that Z is the empty set. In this case, the equality is straightforward. ■

The generic version of the principal ideal theorem straightforwardly generalizes to a generic version of Krull's height theorem. We first mention the original version:

Theorem 49 (Krull's height theorem) *Let R be a commutative ring with unit, let $I = \langle f_1, \dots, f_m \rangle \trianglelefteq R$ be an ideal. Then*

$$\operatorname{ht} I \le m,$$

with equality if and only if $f_1, \dots, f_m$ is an R-regular sequence, that is, f_i is not invertible and not a zero divisor in the ring $R/\langle f_1, \dots, f_{i-1} \rangle$ for all i.

The generic version can be derived directly from the generic principal ideal theorem:

Theorem 50 (Generic height theorem) *Let Z be an algebraic set in $\mathbb{C}^D$, let $I = \langle f_1, \dots, f_m \rangle$ be a generic ideal in $\mathbb{C}[X_1, \dots, X_D]$. Then*

$$\operatorname{ht}(\mathrm{I}(Z) + I) = \min(\operatorname{codim} Z + m,\ D + 1).$$

Proof We will write $R = O(Z)$ for abbreviation.

First assume $m \le D + 1 - \operatorname{codim} Z$. It suffices to show that $f_1, \dots, f_m$ forms an R-regular sequence, then apply Krull's height theorem. In Proposition 43, we have proved that f_i is not a zero divisor in the ring $O(Z \cap \mathrm{V}(f_1, \dots, f_{i-1}))$ (note that the latter ring is nonzero by Krull's height theorem). By Hilbert's Nullstellensatz, this is the same as the ring $R/\sqrt{\langle f_1, \dots, f_{i-1} \rangle}$. But by the definition of radical, this implies that f_i is no zero divisor in the ring $R/\langle f_1, \dots, f_{i-1} \rangle$, since if $f_i \cdot h = 0$ in the first ring, we have

$$(f_i \cdot h)^N = f_i \cdot (f_i^{N-1} h^N) = 0$$

in the second. Thus the f_i form an R-regular sequence, proving the theorem for the case $m \leq D+1-\operatorname{codim} Z$.

If now $m > k := D+1-\operatorname{codim} Z$, the above reasoning shows that the radical of $\mathrm{I}(Z)+\langle f_1,\dots,f_k\rangle$ is the module $\langle 1\rangle$, which means that those are equal. Thus

$$\mathrm{I}(Z)+\langle f_1,\dots,f_k\rangle = \mathrm{I}(Z)+\langle f_1,\dots,f_m\rangle = \langle 1\rangle,$$

proving the theorem.

Note that we could have proved the generic height theorem also directly from the generic principal ideal theorem by induction. ■

Again, we give the geometric interpretation of Krull's height theorem:

Corollary 51 *Let Z_1 be an algebraic set in $\mathbb{C}^D$, let Z_2 be a generic algebraic set in $\mathbb{C}^D$. Then one has*

$$\operatorname{codim}(Z_1 \cap Z_2) = \min(\operatorname{codim} Z_1 + \operatorname{codim} Z_2,\ D+1).$$

Proof This follows directly from two applications of the generic height theorem 50: first for $Z = \mathbb{C}^D$ and $Z_2 = \mathrm{V}(I)$, showing that $\operatorname{codim} Z_2$ is equal to the number m of generators of I; then, for $Z = Z_1$ and $Z_2 = \mathrm{V}(I)$, and substituting $m = \operatorname{codim} Z_2$. ■

We can now immediately formulate a homogenous version of Proposition 51:

Corollary 52 *Let Z_1 be a homogenous algebraic set in $\mathbb{C}^D$, let Z_2 be a generic homogenous algebraic set in $\mathbb{C}^D$. Then one has*

$$\operatorname{codim}(Z_1 \cap Z_2) = \min(\operatorname{codim} Z_1 + \operatorname{codim} Z_2,\ D).$$

Proof Note that homogenization and dehomogenization of a non-empty algebraic set do not change its codimension, and homogenous algebraic sets always contain the origin. Also, one has to note that by Lemma 41, the dehomogenization of Z_2 is a generic algebraic set in $\mathbb{C}^{D-1}$. ■

Finally, using Corollary 44, we want to give a more technical variant of the generic height theorem, which will be of use in later proofs. First, we introduce some abbreviating notations:

Definition 53 *Let $f \in \mathbb{C}[X_1,\dots X_D]$ be a generic polynomial with property P. If one can write $f = f' + c$, where f' is a generic polynomial subject to some property P', and c is a generic constant, we say that f has independent constant term. If c is generic and independent with respect to some collection of generic objects, we say that f has independent constant term with respect to that collection.*

In this terminology, Corollary 44 rephrases as: a generic polynomial with independent constant term is no zero divisor. Using this, we can now formulate the corresponding variant of the generic height theorem:

Lemma 54 *Let Z be an algebraic set in $\mathbb{C}^D$. Let $f_1,\dots,f_m \in \mathbb{C}[X_1,\dots,X_D]$ be generic, possibly subject to some algebraic properties, such that f_i has independent constant term with respect to Z and $f_1,\dots,f_{i-1}$. Then*

$$\operatorname{ht}(\mathrm{I}(Z)+I) = \min(\operatorname{codim} Z + m,\ D+1).$$

Proof Using Corollary 44, one obtains that f_i is no zero divisor modulo $\mathrm{I}(Z)+\langle f_1,\dots,f_{i+1}\rangle$. Using Krull's height theorem yields the claim. ■

B.5 Generic Ideals

The generic height theorem 50 has allowed us to make statements about the structure of ideals generated by generic elements without constraints. However, the ideal I in our the cumulant comparison problem is generic subject to constraints: namely, its generators are contained in a prescribed ideal, and they are homogenous. In this subsection, we will use the theory developed so far to study generic ideals and generic ideals subject to some algebraic properties, for example, generic ideals contained in other ideals. We will use these results to derive an identifiability result on the marginalization problem which has been derived already less rigorously in the supplementary material of von Bünau et al. (2009) for the special case of Stationary Subspace Analysis.

Proposition 55 *Let $\mathfrak{s} \trianglelefteq \mathbb{C}[X_1,\dots,X_D]$ be an ideal, having an H-basis $g_1,\dots,g_n$. Let*

$$I = \langle f_1,\dots,f_m\rangle, \quad m \geq \max(D+1,n)$$

with generic $f_i \in \mathfrak{s}$ such that

$$\deg f_i \geq \max_j (\deg g_j) \quad \textit{for all} \quad 1 \leq i \leq m.$$

Then $I = \mathfrak{s}$.

Proof First note that since the g_i form a degree-first Groebner basis, a generic $f \in \mathfrak{s}$ is of the form

$$f = \sum_{k=1}^{n} g_k h_k \quad \text{with generic } h_k,$$

where the degrees of the h_k are appropriately chosen, that is, $\deg h_k \leq \deg f - \deg g_k$.

So we may write

$$f_i = \sum_{k=1}^{n} g_k h_{ki} \quad \text{with generic } h_{ki},$$

where the h_{ki} are generic with appropriate degrees, and independently chosen. We may also assume that the f_i are ordered increasingly by degree.

To prove the statement, it suffices to show that $g_j \in I$ for all j. Now the height theorem 50 implies that

$$\langle h_{11},\dots h_{1m}\rangle = \langle 1\rangle,$$

since the h_{ki} were independently generic, and $m \geq D+1$. In particular, there exist polynomials $s_1,\dots,s_m$ such that

$$\sum_{i=1}^{m} s_i h_{1i} = 1.$$

Thus we have that

$$\begin{aligned}\sum_{i=1}^{m} s_i f_i &= \sum_{i=1}^{m} s_i \sum_{k=1}^{n} g_k h_{ki} = \sum_{k=1}^{n} g_k \sum_{i=1}^{m} s_i h_{ki} \\ &= g_1 + \sum_{k=2}^{n} g_k \sum_{i=1}^{m} s_i h_{ki} =: g_1 + \sum_{k=2}^{n} g_k h'_k.\end{aligned}$$

Subtracting a suitable multiple of this element from the $f_1,\dots,f_m$, we obtain

$$f'_i = \sum_{k=2}^{n} g_k (h_{ki} - h_{1i} h'_k) =: \sum_{k=2}^{n} g_k h'_{ki}.$$

We may now consider $h_{1i}h'_k$ as fixed, while the h_{ki} are generic. In particular, the h'_{ki} have independent constant term, and using Lemma 54, we may conclude that

$$\langle h'_{21},\dots,h'_{2m}\rangle = \langle 1\rangle,$$

allowing us to find an element of the form

$$g_2 + \sum_{k=3}^{n} g_k \cdot \dots$$

in I. Iterating this strategy by repeatedly applying Lemma 54, we see that g_k is contained in I, because the ideals I and $\mathfrak{s}$ have same height. Since the numbering for the g_j was arbitrary, we have proved that $g_j \in I$, and thus the proposition. ■

The following example shows that we may not take the degrees of the f_i completely arbitrary in the proposition, that is, the condition on the degrees is necessary:

Example 56 Keep the notations of Proposition 55. Let $\mathfrak{s} = \langle X_2 - X_1^2, X_3\rangle$, and $f_i \in \mathfrak{s}$ generic of degree one. Then

$$\langle f_1,\dots,f_m\rangle = \langle X_3\rangle.$$

This example can be generalized to yield arbitrarily bad results if the condition on the degrees is not fulfilled.

However note that when $\mathfrak{s}$ is generated by linear forms, as in the marginalization problem, the condition on the degrees vanishes.

We may use Proposition 55 also in another way to derive a more detailed version of the generic height theorem for constrained ideals:

Proposition 57 *Let V be a fixed complete intersection set in $\mathbb{C}^D$, i.e. an algebraic set of codimension d such that there exist d generators $g_1,\dots,g_d$ for $\operatorname{I}(V)$. Let $f_1,\dots,f_m$ be generic forms in $\operatorname{I}(V)$ such that $\deg f_i \geq \max_j(\deg g_j)$ for $1 \leq i \leq m$. Then we can write $\operatorname{V}(f_1,\dots,f_m) = V \cup U$ with U an algebraic set of*

$$\operatorname{codim} U \geq \min(m,\, D+1),$$

the equality being strict for $m < d$.

Proof If $m \geq D+1$, this is just a direct consequence of Proposition 55.

First assume $m = d$. Consider the situation modulo $X_m, \ldots, X_D$. This corresponds to looking at the situation

$$\mathrm{V}(f_1, \ldots, f_m) \cap H \subseteq H \cong \mathbb{C}^{m-1},$$

where H is the linear subspace given by $X_m = \cdots = X_D = 0$. Since the coordinate system was generic, the elements f_i will be also generic modulo $X_{,} \ldots, X_D$, and we have by Proposition 55 that $\mathrm{V}(f_1, \ldots, f_m) \cap H = V \cap H$. Also, the H can be regarded as a generic linear subspace, thus by Corollary 51, we see that $\mathrm{V}(f_1, \ldots, f_m)$ consists of V and possibly components of equal or higher codimension. This proves the claim for $m = \operatorname{codim} V$.

The case $m < d$ follows from Krull's principal ideal theorem 45: it states that the codimension of $\mathrm{V}(f_1, \ldots, f_i)$ increases at most by one when increasing i by one; above, we have proved equality for $i = d$. Thus, the codimension of $\mathrm{V}(f_1, \ldots, f_i)$ must have been i for every $i \leq d$. This yields the claim.

Now we prove the remaining case $m \geq d$. We will assume that $m = D+1$ and prove the statement for the sets $\mathrm{V}(f_1, \ldots, f_i), d \leq i \leq m$. By the Lasker-Noether-Theorem, we may write

$$\mathrm{V}(f_1, \ldots, f_d) = V \cup Z_1 \cup \cdots \cup Z_N$$

for finitely many irreducible components Z_j with $\operatorname{codim} Z_j \geq d$. Proposition 55 states that

$$\mathrm{V}(f_1, \ldots, f_m) = V.$$

For $i \geq d$, write now

$$Z_{ji} = Z_j \cap \mathrm{V}(f_1, \ldots, f_i) = Z_j \cap \mathrm{V}(f_{d+1}, \ldots, f_i).$$

With this, we have the equalities

$$\begin{aligned} \mathrm{V}(f_1, \ldots, f_i) &= \mathrm{V}(f_1, \ldots, f_d) \cap \mathrm{V}(f_{d+1}, \ldots, f_i) \\ &= V \cup (Z_1 \cap \mathrm{V}(f_{d+1}, \ldots, f_i)) \cup \cdots \cup (Z_N \cap \mathrm{V}(f_{d+1}, \ldots, f_i)) \\ &= V \cup Z_{1i} \cup \cdots \cup Z_{Ni}. \end{aligned}$$

for $i \geq d$. Thus, reformulated, Proposition 55 states that $Z_{jm} = \varnothing$ for any j. We can now infer by Krull's principal ideal theorem 45 that

$$\operatorname{codim} Z_{ji} \leq \operatorname{codim} Z_{j,i-1} + 1$$

for any i, j. But since $\operatorname{codim} Z_{jm} = D+1$, and $\operatorname{codim} Z_{jd} \geq d$, this can only happen when $\operatorname{codim} Z_{ji} \geq i$ for any $d \leq i \leq m$. Thus we may write

$$\mathrm{V}(f_1, \ldots, f_i) = V \cup U \quad \text{with } U = Z_{1i} \cup \cdots \cup Z_{Ni}$$

with $\operatorname{codim} U \geq i$, which proves the claim for $m \geq \operatorname{codim} V$. ■

Note that depending on V and the degrees of the f_i, it may happen that even in the generic case, the equality in Proposition 57 is not strict for $m \geq \operatorname{codim} V$:

Example 58 Let V be a generic linear subspace of dimension d in $\mathbb{C}^D$, let $f_1,\dots,f_m \in \mathrm{I}(V)$ be generic with degree one. Then $\mathrm{V}(f_1,\dots,f_m)$ is a generic linear subspace of dimension $\max(D-m,d)$ containing V. In particular, if $m \geq D-d$, then $\mathrm{V}(f_1,\dots,f_m) = V$. In this example, $U = \mathrm{V}(f_1,\dots,f_m)$, if $m < \operatorname{codim} V$, with codimension m, and $U = \varnothing$, if $m \geq \operatorname{codim} V$, with codimension $D+1$.

Similarly, one may construct generic examples with arbitrary behavior for $\operatorname{codim} U$ when $m \geq \operatorname{codim} V$, by choosing V and the degrees of f_i appropriately.

Algebraic sets which are not complete intersection sets are still contained in a complete intersection set of same dimension, so the following similar result holds for arbitrary algebraic sets:

Corollary 59 *Let V be a fixed algebraic set in $\mathbb{C}^D$, of codimension d; let $g_1,\dots,g_d$ be a regular sequence in $\mathrm{I}(V)$, let n be the cardinality of some H-basis of $\mathrm{I}(V)$. Let $f_1,\dots,f_m$ be generic forms in $\mathrm{I}(V)$ such that $\deg f_i \geq \max_j(\deg g_j)$ for $1 \leq i \leq m$. Then we can write $\mathrm{V}(f_1,\dots,f_m) = V \cup U$ with an algebraic set U whose codimension satisfies*

$$\begin{aligned} \operatorname{codim} U &= m \quad \textit{if} \quad m \leq d \\ \operatorname{codim} U &\geq \min(D+1+m-n, m, D+1) \quad \textit{if} \quad m \geq d. \end{aligned}$$

Proof This follows in analogy to Proposition 57. ■

Similarly as in the geometric version for the height theorem, we may derive the following geometric interpretation of this result:

Corollary 60 *Let $V \subseteq Z_1$ be fixed algebraic sets in $\mathbb{C}^D$. Let Z_2 be a generic algebraic set in $\mathbb{C}^D$ containing V. Then*

$$\operatorname{codim}(Z_1 \cap Z_2 \setminus V) \geq \min(\operatorname{codim}(Z_1 \setminus V) + \operatorname{codim}(Z_2 \setminus V),\ D+1).$$

Informally, we have derived a height theorem type result for algebraic sets under the constraint that they contain another prescribed algebraic set V.

We also give a homogenous version of Proposition 57, since the ideals we will consider are homogenous complete intersection:

Corollary 61 *Let V be a fixed homogenous complete intersection set in $\mathbb{C}^D$. Let $f_1,\dots,f_m$ be generic homogenous forms in $\mathrm{I}(V)$, satisfying the degree condition as in Proposition 57. Then $\mathrm{V}(f_1,\dots,f_m) = V + U$ with U an algebraic set fulfilling*

$$\operatorname{codim} U \geq \min(m,\ D).$$

In particular, if $m > D$, then $\mathrm{V}(f_1,\dots,f_m) = V$. Also, the maximal dimensional part of $\mathrm{V}(f_1,\dots,f_m)$ equals V if and only if $m > D - \dim V$.

Proof This follows immediately by dehomogenizing, applying Proposition 57, and homogenizing again. ■

From this Corollary, we now can directly derive a statement on the necessary number of epochs for the identifiability of the projection making several random variables appear identical. For the convenience of the reader, we recall the setting and then explain what identifiability means. The problem we consider in the main part of the paper can be described as follows:

Problem 62 *Let $X_1,\dots,X_m$ be random variables, let*

$$q_i = [T_1,\dots,T_D]\circ(\kappa_2(X_i)-\kappa_2(X_m)),\ 1\le i\le m-1$$

and

$$f_i = [T_1,\dots,T_D]\circ(\kappa_1(X_i)-\kappa_1(X_m)),\ 1\le i\le m-1$$

be the corresponding cumulant polynomials in the formal variables $T_1,\dots,T_D$. What can one say about the set

$$S' = \operatorname{V}(q_1,\dots,q_{m-1},f_1,\dots,f_{m-1}).$$

If there is a linear subspace S on which the cumulants agree, then the q_i, f_i vanish on S. If we assume that this happens generically, the problem reformulates to

Problem 63 *Let S be a d-dimensional linear subspace of $\mathbb{C}^D$, let $\mathfrak{s} = \operatorname{I}(S)$, and let $f_1,\dots,f_N$ be generic homogenous quadratic or linear polynomials in $\mathfrak{s}$. How does $S' = \operatorname{V}(f_1,\dots,f_N)$ relate to S?.*

Before giving bounds on the identifiability, we first begin with a direct consequence of Corollary 61:

Remark 64 *The highest dimensional part of $S' = \operatorname{V}(f_1,\dots,f_N)$ is S if and only if*

$$N > D-d.$$

For this, remark that $\operatorname{I}(S)$ is generated in degree one, and thus the degree condition in Corollary 61 becomes empty.

We can now also get an identifiability result for S:

Proposition 65 *Let $f_1,\dots,f_N$ be generic homogenous polynomials of degree one or two, vanishing on a linear space S of dimension $d>0$. Then S is identifiable from the f_i alone if*

$$N \ge D-d+1.$$

Moreover, if all f_i are quadrics, then S is identifiable from the f_i alone only if

$$N \ge 2.$$

Proof Note that the $f_1,\dots,f_N$ are generic polynomials contained in $\mathfrak{s} := \operatorname{I}(S)$.

First assume $N \ge D-d+1$. We prove that S is identifiable: using Corollary 61, one sees now that the common vanishing set of the f_i is S up to possible additional components of dimension $d-1$ or less; that is, the radical of the ideal generated by the f_i has a prime decomposition

$$\sqrt{\langle f_1,\dots,f_N\rangle} = \mathfrak{s}\cap\mathfrak{p}_1\cap\cdots\cap\mathfrak{p}_k,$$

where the $\mathfrak{p}_i$ are of dimension $d-1$ or less, while $\mathfrak{s}$ has dimension d. So one can use one of the existing algorithms calculating primary decomposition to identify $\mathfrak{s}$ as the unique component of the highest dimensional part, which proves identifiability if $N \geq D-d+1$.

Now we prove the only if part: assume that $N=1$, that is, we have only a single f_1. Since f_1 is generic with the property of vanishing on S, we have

$$f_1 = \sum_{i=1}^{D-d} g_i h_i,$$

where $g_1, \dots, g_{D-d}$ is some homogenous linear generating set for $\mathrm{I}(S)$, and $h_1, \dots, h_{D-d}$ are generic homogenous linear forms. Thus, the zero set $\mathrm{V}(f_1)$ also contains the linear space $S' = \mathrm{V}(h_1, \dots, h_{D-d})$ which is a generic d-dimensional linear space in $\mathbb{C}^D$ and thus different from S; no algorithm can decide whether S or S' is the correct solution, so S is not identifiable. ■

Note that there is no obvious reason for the lower bound $N \geq D-d+1$ given in Proposition 65 to be strict. While it is most probably the best possible bound which is in D and d, in general it may happen that S can be reconstructed from the ideal $\langle f_1, \dots, f_N \rangle$ directly. The reason for this is that a generic homogenous variety of high enough degree and dimension does not need to contain a linear subspace of fixed dimension d in general.

References

Shun'ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of mathematical monographs*. American Mathematical Society, 2000. ISBN 9780821805312.

Duncan A. J. Blythe, Paul von Bünau, Frank C. Meinecke, and Klaus-Robert Müller. Feature extraction for change-point detection using stationary subspace analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):631–643, 2012. doi: 10.1109/TNNLS.2012.2185811.

Massimo Caboara, Pasqualina Conti, and Carlo Traverse. Yet another ideal decomposition algorithm. In Teo Mora and Harold Mattson, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1255 of *Lecture Notes in Computer Science*, pages 39–54. Springer Berlin / Heidelberg, 1997.

Robert M. Corless, Patrizia M. Gianni, Barry M. Trager, and Steven M. Watt. The singular value decomposition for polynomial systems. *Proc. ISSAC '95*, pages 195–207, 1995.

David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 0387356509.

Mathias Drton, Bernd Sturmfels, and Seth Sullivant. *Lectures on Algebraic Statistics*. Oberwolfach Seminars. Birkhauser Basel, 2010. ISBN 9783764389048.

David Eisenbud, Craig Huneke, and Wolmer Vasconcelos. Direct methods for primary decomposition. *Inventiones Mathematicae*, 110:207–235, 1992. ISSN 0020-9910.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7: 179–188, 1936.

Jerome H. Friedman and John W. Tukey. A projection pursuit algorithm for exploratory data analysis. *Computers, IEEE Transactions on*, C-23(9):881 – 890, 9 1974. ISSN 0018-9340. doi: 10.1109/T-C.1974.224051.

Ralf Fröberg and Joachim Hollman. Hilbert series for ideals generated by generic forms. *Journal of Symbolic Computation*, 17(2):149 – 157, 1994. ISSN 0747-7171. doi: DOI:10.1006/jsco.1994.1008.

Patrizia Gianni, Barry Trager, and Gail Zacharias. Groebner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(2-3):149 – 167, 1988. ISSN 0747-7171. doi: DOI:10.1016/S0747-7171(88)80040-3.

Paolo Gibilisco, Eva Riccomagno, Maria Piera Rogantin, and Henry P. Wynn. *Algebraic and Geometric Methods in Statistics*. Cambridge University Press, 2010. ISBN 9780521896191.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel method for the two sample problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.

Satoshi Hara, Yoshinobu Kawahara, Takashi Washio, and Paul von Bünau. Stationary subspace analysis as a generalized eigenvalue problem. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ICONIP'10, pages 422–429, Berlin, Heidelberg, 2010. Springer-Verlag.

Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, and Hennie Poulisse. Approximate computation of zero-dimensional polynomial ideals. *Journal of Symbolic Computation*, 44(11):1566 – 1591, 2009. ISSN 0747-7171. doi: DOI:10.1016/j.jsc.2008.11.010. In Memoriam Karin Gatermann.

Grete Hermann. Die Frage der endlich vielen Schritte in der Theorie der Polynomideale - unter Benutzung nachgelassener Sätze von K. Hentzelt. *Mathematische Annalen*, 95(1):736 – 788, 1926. ISSN 0747-7171. doi: DOI:10.1007/BF01206635.

Harold Hotelling. The generalization of student's ratio. *Annals of Mathematical Statistics*, 2(3): 360–378, 1932.

Anthony Iarrobino. Compressed algebras: Artin algebras having given socle degrees and maximal length. *Transactions of the American Mathematical Society*, 285(1):337 – 378, 1984.

Franz J. Király, Paul von Bünau, Frank Meinecke, Duncan Blythe, and Klaus-Robert Müller. Algebraic geometric comparison of probability distributions, 2011. Oberwolfach Preprint 2011-30.

Risi Kondor. The skew spectrum of functions on finite groups and their homogeneous spaces, 2007. Eprint arXiv:0712.4259.

Risi Kondor and Karsten M. Borgwardt. The skew spectrum of graphs. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 496–503, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390219.

Martin Kreuzer, Hennie Poulisse, and Lorenzo Robbiano. *Approximate Commutative Algebra*, chapter From Oil Fields to Hilbert Schemes. Springer-Verlag Berlin Heidelberg, 2009.

Teresa Krick and Alessandro Logar. An algorithm for the computation of the radical of an ideal in the ring of polynomials. In Harold Mattson, Teo Mora, and T. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 539 of *Lecture Notes in Computer Science*, pages 195–205. Springer Berlin / Heidelberg, 1991.

Santiago Laplagne. An algorithm for the computation of the radical of an ideal. In *Proceedings of the 2006 international symposium on Symbolic and algebraic computation*. ACM, 2006.

Frank C. Meinecke, Paul von Bünau, Motoaki Kawanabe, and Klaus-Robert Müller. Learning invariances with stationary subspace analysis. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 87 –92, 2009. doi: 10.1109/ICCVW.2009.5457715.

Jan S. Müller, Paul von Bünau, Frank C. Meinecke, Frank J. Király, and Klaus-Robert Müller. The stationary subspace analysis toolbox. *Journal of Machine Learning Research*, 12:3065–3069, 2011.

Keith Pardue. Generic sequences of polynomials. *Journal of Algebra*, 324(4):579 – 590, 2010. ISSN 0021-8693. doi: DOI:10.1016/j.jalgebra.2010.04.018.

Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics, 2004. ISBN 0898715571.

Bernd Sturmfels. *Solving Systems of Polynomial Equations*, volume 97 of *CBMS Regional Conferences Series*. Amer. Math. Soc., Providence, Rhode Island, 2002.

Kari Torkkola. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.*, 3:1415–1438, March 2003. ISSN 1532-4435. URL `http://portal.acm.org/citation.cfm?id=944919.944981`.

René Vidal, Yi Ma, and Sastry Shankar. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

Paul von Bünau, Frank C. Meinecke, Franz J. Király, and Klaus-Robert Müller. Finding stationary subspaces in multivariate time series. *Phys. Rev. Lett.*, 103(21):214101, Nov 2009. doi: 10.1103/PhysRevLett.103.214101.

Paul von Bünau, Frank C. Meinecke, Simon Scholler, and Klaus-Robert Müller. Finding stationary brain sources in EEG data. In *Proceedings of the 32nd Annual Conference of the IEEE EMBS*, pages 2810–2813, 2010.

Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, United Kingdom, 2009. ISBN 9780521864671.

Journal of Machine Learning Research 13 (2012) 905-948 Submitted 11/10; Revised 1/12; Published 4/12

Stability of Density-Based Clustering

Alessandro Rinaldo ARINALDO@CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Aarti Singh AARTI@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Rebecca Nugent RNUGENT@STAT.CMU.EDU
Larry Wasserman[*] LARRY@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Sanjoy Dasgupta

Abstract

High density clusters can be characterized by the connected components of a level set $L(\lambda) = \{x : p(x) > \lambda\}$ of the underlying probability density function p generating the data, at some appropriate level $\lambda \geq 0$. The complete hierarchical clustering can be characterized by a cluster tree $\mathcal{T} = \bigcup_\lambda L(\lambda)$. In this paper, we study the behavior of a density level set estimate $\widehat{L}(\lambda)$ and cluster tree estimate $\widehat{\mathcal{T}}$ based on a kernel density estimator with kernel bandwidth h. We define two notions of instability to measure the variability of $\widehat{L}(\lambda)$ and $\widehat{\mathcal{T}}$ as a function of h, and investigate the theoretical properties of these instability measures.

Keywords: clustering, density estimation, level sets, stability, model selection

1. Introduction

A common approach to identifying high density clusters is based on using level sets of the density function (see, for instance, Hartigan, 1975; Rigollet and Vert, 2009). Let $X_1,\ldots,X_n$ be a random sample from a distribution P on $\mathbb{R}^d$ with density p. For $\lambda > 0$ define the level set $L(\lambda) = \{x : p(x) > \lambda\}$. Assume that $L(\lambda)$ can be decomposed into disjoint, connected sets $L(\lambda) = \bigcup_{j=1}^{N(\lambda)} C_j$. Following Hartigan (1975), we refer to $\mathcal{C}_\lambda = \{C_1,\ldots,C_{N(\lambda)}\}$ as the *density clusters* at level λ. We call the collection of clusters

$$\mathcal{T} = \bigcup_{\lambda \geq 0} \mathcal{C}_\lambda$$

the *cluster tree* of the density p. Note that $\mathcal{T}$ does indeed have a tree structure: if $A, B \in \mathcal{T}$ then either, $A \subset B$, or $B \subset A$ or $A \cap B = \emptyset$. The cluster tree summarizes the cluster structure of the distribution; see Stuetzle and Nugent (2009).

[*] Also in the Machine Learning Department.

It is also possible to index the level sets by probability content. For $0 < \alpha < 1$, define the level set $M(\alpha) = L(\lambda_\alpha)$, where

$$\lambda_\alpha = \sup\{\lambda : \ P(L(\lambda)) \geq \alpha\}.$$

If the density does not contain any jumps or flat parts, then there is a one-to-one correspondence between the level sets indexed by the density level and the probability content. The cluster tree obtained from the clusters of $M(\alpha)$ for $0 \leq \alpha \leq 1$ is equivalent to $\mathcal{T}$. Relabeling the tree in terms of α may be convenient because α is more interpretable than λ, but the tree is the same. Figure 1 shows the cluster tree for a density estimate of a mixture of three normals (using a reference rule bandwidth). The cluster tree's two splits and subsequent three leaves correspond to the density estimate's modes. The tree is also indexed by λ, the density estimate's height, on the left and α, the probability content, on the right. For example, the second split corresponds to $\lambda = 0.086$ and $\alpha = 0.257$. We note here that determining the true clusters for even this seemingly simple univariate distribution is not trivial for all λ; in particular, values of λ near 0.04 and 0.09 will give ambiguous results.

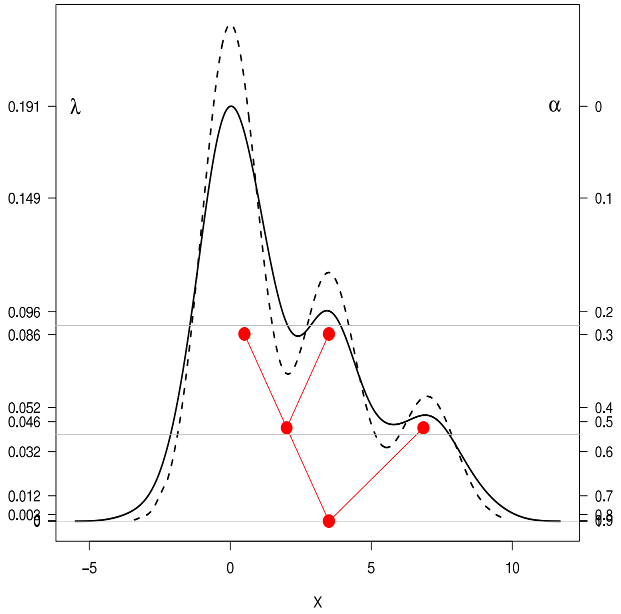


Figure 1: The cluster tree for a Gaussian kernel density estimate (normal reference rule bandwidth) of a sample from the mixture $(4/7)N(0,1) + (2/7)N(3.5,1) + (1/7)N(7,1)$; the tree is indexed by both λ (left) and α (right). The dashed curve indicates the true underlying density. The gray lines indicate $L(0.04), L(0.09)$.

In this paper we study some properties of clusters defined by density level sets and cluster trees. In particular, we consider their estimators based on a kernel density estimate and show how the bandwidth h of the kernel affects the risk of these estimators. Then we investigate the notion of stability for density-based clustering. Specifically, we propose two measures of instability. The first, denoted by $\Xi_{\lambda,n}(h)$, measures the instability of a given level set. The second, denoted by $\Gamma_n(h)$, is a more global measure of instability.

Investigation of the stability properties of density clusters is the main focus of the paper. Stability has become an increasingly popular tool for choosing tuning parameters in clustering; see von Luxburg (2009), Lange et al. (2004), Ben-David et al. (2006), Ben-Hur et al. (2002), Carlsson and Memoli (2010), Meinshausen and Bühlmann (2010), Fischer and Buhmann (2003), and Rinaldo and Wasserman (2010). The basic idea is this: clustering procedures inevitably depend on one or more tuning parameters. If we choose a good value of the tuning parameter, then we expect that the clusters from different subsets of the data should be similar. While this idea sounds simple, the reality is rather complex. Figure 2 shows a plot of Ξ_n and Γ_n for our example. We see that $\Xi_{\lambda,n}(h)$ is a complicated function of h while $\Gamma_n(h)$ is much simpler. Our results will explain this behavior.

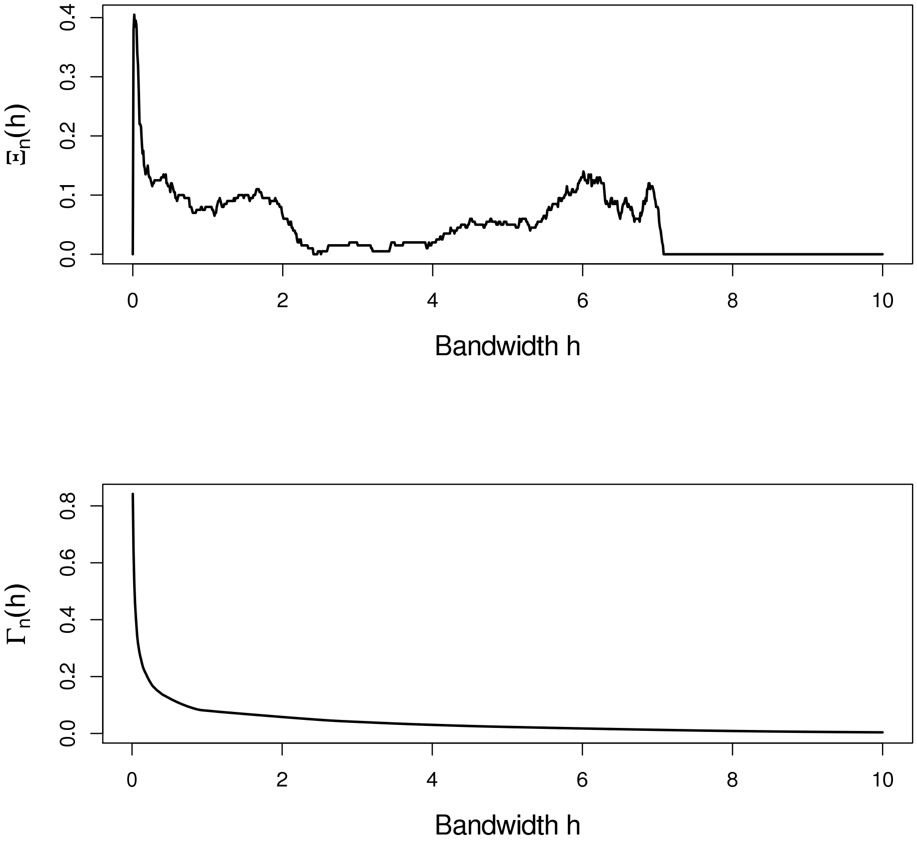


Figure 2: Plots of the fixed-λ instability (top) $\Xi_{\lambda,n}(h)$ for $\lambda = 0.09$ and of the total variation instability $\Gamma_n(h)$ (bottom) for the mixture distribution in Figure 1 as functions of the bandwidth h.

Below we briefly describe our contributions.

- We consider plug-in estimates of the level sets $L(\lambda)$ corresponding to fixed density levels λ and also to the level sets $L(\lambda_\alpha)$ corresponding to fixed probability contents α using kernel density estimators. We analyze the statistical properties of these plug-in estimates and formulate conditions on the density of the data generating distribution and on the kernel that guarantee accurate recovery of the level sets as n becomes large.

- We formulate a notion of cluster stability of the level sets based on a splitting of the the data that quantifies the variability of the level set estimators we consider. We construct an estimator of the cluster instability and analyze its performance as n become large, and argue that stability can provide a guidance on the optimal choice of the bandwidth parameter. As a result of our analysis, we are able to provide a rigorous characterization of the levels sets for which the the uncertainty is larger and, therefore, for which the cluster tree can be estimated with a smaller degree of accuracy. Our results suggest that the sample complexity for successful reconstruction of the cluster tree may vary significantly depending on whether we estimating a portion of the tree that is far removed from a branching region or not, and for those portion of the tree we provide some rates.

- We formulate and analyze a stronger notion of cluster stability that is based on the total variation distance between kernel density estimates computed over different data subsamples. This second kind of stability is more global and has natural and interesting connections with the problem of optimally estimating a density in L_1 norm.

After the writing of the first draft of this paper we learned of the interesting and relevant contributions by Chaudhuri and Dasgupta (2010), Kpotufe and von Luxburg (2011) and Steinwart (2011) who all consider the problem of estimating the cluster tree. Our results provide a different perspective on this issue as we concern ourselves with quantifying, based on stability criteria, the uncertainty of the cluster tree estimate. Furthermore, these papers only characterize the optimal scaling of parameters to guarantee cluster tree recovery and do not provide a data-driven way to choose these parameters. In this paper, we investigate stability as a means for data-adaptive choice of parameters such as the kernel bandwidth.

The paper is organized as follows. In Section 2 we describe the assumptions on the density and recall some facts about kernel density estimation. In Section 3 we construct plug-in estimates $\widehat{L}(\lambda)$ of the level set $L(\lambda)$, $\widehat{\mathcal{T}}$ of the cluster tree $\mathcal{T}$, and $\widehat{M}(\alpha)$ of the level set indexed by probability content $M(\alpha)$. In Section 4 we define and study a notion of the stability of $\widehat{L}(\lambda)$ and extend it to $\widehat{\mathcal{T}}$. We also consider an alternative version of our results when the level sets are indexed by probability content. We then describe another notion of stability of cluster trees based on total variation that leads to a constructive procedure for selecting the kernel bandwidth. In Section 5 we consider some numerical examples. Section 6 contains a discussion of the results and the proofs are in Section A. Throughout, we use symbols like $c, c_1, c_2, \ldots, C, C_1, C_2, \ldots$, to denote various positive constants whose value can change in different expressions.

2. Preliminaries

In this section we introduce some notation, state the assumptions on the density we will be using throughout and review some useful facts about kernel density estimation.

2.1 Notation

For $x \in \mathbb{R}^d$, let $||x||$ denote its Euclidean norm. Let $B(x,\varepsilon) = \{y : \; ||x-y|| \leq \varepsilon\} \subset \mathbb{R}^d$ denote a ball centered at x with radius ε. For two sets A and B in $\mathbb{R}^d$, their Hausdorff distance is

$$d_\infty(A,B) = \inf\{\varepsilon : \; A \subset (B \oplus \varepsilon) \; \text{ and } \; B \subset (A \oplus \varepsilon)\},$$

where $A \oplus \varepsilon = \bigcup_{x \in A} B(x, \varepsilon)$, and

$$A \Delta B = (A \cap B^c) \bigcup (A^c \cap B)$$

denotes the symmetric set difference. Finally, we let $v_d = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}$ be the volume of the d-dimensional Euclidean unit ball.

For sequences of real numbers $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if there exists a $C > 0$ such that $|a_n| \leq C|b_n|$ for all n large enough, and we will write $a_n = \omega(b_n)$ if there exists a constant $C > 0$ such that $|a_n| \geq C|b_n|$ for all n large enough. When $\{a_n\}$ and $\{b_n\}$ are sequences of random variables described by a probability measure P, we will write $a_n = O_P(b_n)$ if, for any $\varepsilon > 0$, there exists a constant $C > 0$ such that $|a_n| \leq C|b_n|$ with P-probability at least $1 - \varepsilon$ for all n large enough.

We will be considering samples of n independent and identically distributed random vectors from an unknown probability measure P on $\mathbb{R}^d$ with Lebesgue density p. If X and Y are such samples, we will denote with $\mathbb{P}_{X,Y}$ the probability measures associated to them and with $\mathbb{E}_{X,Y}$ the corresponding expectation operator. Thus, if $\mathcal{A}$ is an event depending on X and Y, we will write $\mathbb{P}_{X,Y}(\mathcal{A})$ for its probability. Finally, for a sample $X = (X_1, \ldots, X_n)$, we will denote with $\widehat{P}_X$ the empirical measure associated with it; explicitly, for any measurable set $A \subset \mathbb{R}^d$,

$$\widehat{P}_X(A) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \in A).$$

2.2 Assumptions

We will use the following assumptions on the density p and its local behavior around a given density level λ.

(A0) *Compact Support* - The support S of p is compact.

(A1) *Lipschitz Density* - Assume that

$$p \in \Sigma(A) \equiv \left\{ p: \ |p(x) - p(y)| \leq A||x - y||, \ \text{ for all } x, y \in S \right\}$$

for some $A > 0$.

(A2) *Local density regularity at* λ- For a given density level of interest λ, there exist constants $0 < \kappa_1 \leq \kappa_2 < \infty$ and $0 < \varepsilon_0$ such that, for all $\varepsilon < \varepsilon_0$,

$$\kappa_1 \varepsilon \leq P(\{x\colon |p(x) - \lambda| \leq \varepsilon\}) \leq \kappa_2 \varepsilon.$$

It is possible to formulate condition (A2) more generally in terms of powers of ε, that is ε^a. However, as argued in Rinaldo and Wasserman (2010), the above statement typically holds with $a = 1$ for almost all λ.

Assumptions (A1) and (A2) impose some mild regularity conditions on the density: (A1) implies that the density cannot change drastically anywhere, while (A2) implies that the density cannot be too flat or steep locally around the level set. In particular, (A2) is necessary to ensure that small error in estimating the density level does not translate into a huge error in localizing the level set.

We remark that this assumption is an extension of the Tsybakov noise-margin condition for classification (see Mammen and Tsybakov, 1999; Tsybakov, 2004) to the density level set context and has been used in other work on density level-set estimation, such as Polonik (1995), Tsybakov (1997), Cuevas et al. (2006), Rigollet and Vert (2009), Singh et al. (2009) and Rinaldo and Wasserman (2010). Finally notice that (A0) and (A1) together imply that the density p is bounded by some positive constant $p_{\max} < \infty$. These assumptions are stronger than necessary, but they simplify the proofs. Notice in particular, that assumptions (A1) and (A2) each rule out the case of sharp clusters, in which S is the disjoint union of a finite number of compact sets over which p is bounded from below by a positive constant. Finally, we remark that some of our results will only require a subset of these assumptions.

2.3 Estimating the Density

To estimate the density p based on the i.i.d. sample $X = (X_1, \dots, X_n)$, we use the kernel density estimator

$$\widehat{p}_{h,X}(u) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^d} K\left(\frac{u - X_i}{h}\right), \quad u \in \mathbb{R}^d,$$

where the kernel K is a symmetric, non-negative function with compact support such that $\int_{R^d} K(z)dz = 1$ and $h > 0$ is the bandwidth. In some results we will consider specifically the *spherical kernel* $K(u) = \frac{I_{B(0,1)}(u)}{v_d}$, $u \in \mathbb{R}^d$, where $I_{B(0,1)}(\cdot)$ denotes the indicator function of the Euclidean ball $B(0,1)$.

For $h > 0$, let $p_h(u) = \mathbb{E}_X[\widehat{p}_{h,X}(u)]$. Note that p_h is the Lebesgue density of the probability measure

$$P_h = P * \mathbb{K}_h,$$

where $*$ denotes convolution of probability measures and $\mathbb{K}_h$ denotes the probability measure of a random variable with density $K_h(z) = h^{-d}K(z/h), z \in \mathbb{R}^d$.

We note that the compactness of K and assumption (A0) on p imply that the support of P_h is compact, while assumption (A1) on p further yields that $p_h \in \Sigma(A)$, both statements holding for all $h \geq 0$ (for a formal proof of the second claim, see the end of the proof of Lemma 5). Below, we will be concerned with given values of the density level λ and of the probability parameter $\alpha \in (0,1)$ and will impose the following assumptions.

(B2) *Local density regularity at* λ- For a given density level λ, there exist positive constants $\kappa_1' \leq \kappa_2'$, ε_0 and H bounded away from 0 and ∞, such that, for all $0 \leq \varepsilon < \varepsilon_0$,

$$\kappa_1'\varepsilon \leq \inf_{0 \leq h \leq H} P(\{x \colon |p_h(x) - \lambda| \leq \varepsilon\}) \leq \sup_{0 \leq h \leq H} P(\{x \colon |p_h(X) - \lambda| \leq \varepsilon\}) \leq \kappa_2'\varepsilon.$$

(B3) *Local density regularity at* α- For a given probability value α, there exist positive constants κ_3, η_0 and H bounded away from 0 and ∞, such that, for all $0 \leq \eta < |\eta_0|$,

$$\sup_{0 \leq h \leq H} d_\infty(M_h(\alpha), M_h(\alpha + \eta)) \leq \kappa_3 |\eta|,$$

where $M_h(\alpha) = \{u : p_h(u) > \lambda_\alpha\}$.

Conditions (B2) and (B3) are used only for some specific results from Section 4.1 and Section 3.2, respectively. This will be explicitly mentioned in the statement of such results. In particular, condition (B2) is needed in order to explicitly state the behavior of the instability measure we define below. We conjecture that (B2) follows from condition (A2) on the true density p and using kernels with compact support. This assumption holds for all density levels that are not too close to a local maxima or minima of the density. Assumption (B3) characterizes the regularity of the level sets of p_h and essentially states that the boundary of these level sets is well-behaved and not space-filling (see Tsybakov, 1997; Singh et al., 2009, for analogous conditions). Both assumptions (B2) and (B3) could be stated more generally by assuming some uniformity over λ and α respectively, but for the sake of readability we state them as point-wise conditions.

Our analysis depends crucially on the quantity $\|\widehat{p}_{h,X} - p_h\|_\infty = \sup_{u \in \mathbb{R}^d} |\widehat{p}_{h,X}(u) - p_h(u)|$, for which we use a probabilistic upper established by Giné and Guillou (2002), to which the reader is referred for details. To this end, we will make the following assumption on the kernel K:

(VC) The class of functions

$$\mathcal{F} = \left\{ K\left(\frac{x - \cdot}{h}\right), x \in \mathbb{R}^d, h > 0 \right\}$$

satisfies, for some positive numbers V and v,

$$\sup_P N(\mathcal{F}_h, L_2(P), \varepsilon \|F\|_{L_2(P)}) \leq \left(\frac{V}{\varepsilon}\right)^v,$$

where $N(T; d; \varepsilon)$ denotes the ε-covering number of the metric space (T, d), F is the envelope function of $\mathcal{F}$ and the supremum is taken over the set of all probability measures P on $\mathbb{R}^d$. The quantities V and v are called the VC characteristics of $\mathcal{F}$.

Assumption (VC) holds for a large class of kernels, including, any compactly supported polynomial kernel and the Gaussian kernel. The lemma below follows from Giné and Guillou (2002) (see also Rinaldo and Wasserman, 2010).

Lemma 1 *Assume that the kernel satisfies the VC property, and that*

$$\sup_{t \in \mathbb{R}^d} \sup_{h > 0} \int_{\mathbb{R}^d} K_h^2(t - x) dP(x) < B < \infty.$$

There exist positive constants K_1, K_2 and C, which depends on B and the VC characteristic of K such that the following hold:

1. *For every $\varepsilon > 0$ and $h > 0$, there exists $n(\varepsilon, h)$ such that, for all $n \geq n(\varepsilon, h)$*

$$\mathbb{P}_X\left(\|\widehat{p}_{h,X} - p_h\|_\infty > \varepsilon\right) \leq K_1 e^{-K_2 n \varepsilon^2 h^d}. \tag{1}$$

2. *Let $h_n \to 0$ as $n \to \infty$ in such a way that*

$$\frac{n h_n^d}{\log n} \to \infty. \tag{2}$$

Then, there exist a constant K_3 and a number $n_0 \equiv n_0(d, K_3)$ such that, setting $\varepsilon_n = \sqrt{\frac{K_3 \log n}{n h_n^d}}$,

$$\mathbb{P}_X\left(\|\widehat{p}_{h_n,X} - p_{h_n}\|_\infty > \varepsilon_n\right) \leq \frac{1}{n}, \tag{3}$$

for all $n \geq n_0(d, K_3)$.

The numbers $n(\varepsilon,h)$ and n_0 depend also on the VC characteristic of K and on B. Furthermore, $n(\varepsilon,h)$ is decreasing in both ε and h.

This result requires virtually no assumptions on p and only minimal assumptions on the kernel, which are satisfied by the most commonly used kernels.

The constraint in Equation (2), which in general cannot be dispensed with, has a subtle but important implication for our later results on instability. In fact, it implies that the bandwidth parameter h_n is only allowed to vanish at a slower rate than $\left(\frac{\log n}{n}\right)^{1/d}$. As a result, our measures of instability defined in Sections 4.1 and 3.2 can be reliably estimated for values of the bandwidth $h \gg \left(\frac{\log n}{n}\right)^{1/d}$. Indeed, the threshold value $\left(\frac{\log n}{n}\right)^{1/d}$ is of the same order of magnitude of the maximal spacing among the points in a sample of size n from P (see, for instance, Penrose, 2003).

3. Estimating the Level Set and Cluster Tree

For a given density level λ and kernel bandwidth h, the estimated level set is $\widehat{L}_{h,X}(\lambda) = \{x\colon\ \widehat{p}_{h,X}(x) > \lambda\}$. The clusters (connected components) of $\widehat{L}_{h,X}(\lambda)$ are denoted by $\widehat{\mathcal{C}}_{h,\lambda}$ and the estimated cluster tree is

$$\widehat{\mathcal{T}}_h = \bigcup_{\lambda\geq 0} \widehat{\mathcal{C}}_{h,\lambda}.$$

3.1 Fixed λ

We measure the quality of $\widehat{L}_{h,X}(\lambda)$ as an estimator of $L(\lambda)$ using the loss function

$$\mathcal{L}(h,X,\lambda) = \int_{L(\lambda)\Delta\widehat{L}_{h,X}(\lambda)} p(u)du,$$

where we recall that Δ denotes the symmetric set difference. The performance of plug-in estimators of density level sets has been studied earlier, but we state the results here in a form that provides insights into the performance of instability measures proposed in the next section.

Theorem 2 *Assume that the density p satisfies conditions (A0) and (A1) and let $D = \int \|z\|K(z)dz$ (which is finite by compactness of K). For any sequence $h_n = \omega((\log n/n)^{1/d})$, let*

$$\varepsilon_n = \sqrt{\frac{K_3 \log n}{nh_n^d}}$$

and

$$r_{h_n,\varepsilon_n,\lambda} = P\left(\{u\colon\ |p(u)-\lambda| < ADh_n + \varepsilon_n\}\right).$$

Then, for all $n \geq n(n_0,\lambda,A,D,d)$,

$$\mathbb{P}_X\left(\mathcal{L}(h_n,X,\lambda) \leq r_{h_n,\varepsilon_n,\lambda}\right) \geq 1-\frac{1}{n}.$$

If assumption (A2) holds for the density level λ, then for all $n \geq n(n_0,\lambda,A,D,\varepsilon_0,d)$,

$$\mathbb{P}_X\Big(\mathcal{L}(h_n,X,\lambda) \leq \kappa_2(ADh_n+\varepsilon_n)\Big) \geq 1-\frac{1}{n}.$$

The following corollary characterizes the optimal scaling of the bandwidth parameter h_n that balances the approximation and estimation errors.

Corollary 3 *The value of h that minimizes the bound on $\mathcal{L}$ is*

$$h_n^* = c\left(\frac{n}{\log n}\right)^{-\frac{1}{d+2}},$$

where $c > 0$ is an appropriate constant.

3.2 Fixed α

Often it is more natural to index the density clusters by the probability mass contained in the corresponding high-density regions, instead of the associated density levels. The level set estimator indexed by the probability content $\alpha \in (0,1)$ is given as

$$\widehat{M}_{h,X}(\alpha) = \widehat{L}_{h,X}(\widehat{\lambda}_{h,\alpha,X}),$$

where

$$\widehat{\lambda}_{h,\alpha,X} = \sup\left\{\lambda : \widehat{P}_X(\{u : \widehat{p}_{h,X}(u) > \lambda\}) \geq \alpha\right\} \tag{4}$$

and $\widehat{p}_{h,X}$ is the kernel density estimate computed using the data X with bandwidth h. This estimator was studied by Cadre et al. (2009), though using different techniques and in different settings than ours.

Let $\alpha \in (0,1)$ be fixed and define

$$\lambda_{h,\alpha} = \sup\{\lambda : P(p_h(X) > \lambda) \geq \alpha\}.$$

We first show that the deviation $|\lambda_{h,\alpha} - \lambda_\alpha|$ is of order h, uniformly over α, under the very general assumption that the true density p is Lipschitz.

Lemma 4 *Assume the true density p satisfies the conditions (A0) and (A1). Then, for any $h > 0$,*

$$\sup_{\alpha \in (0,1)} |\lambda_{h,\alpha} - \lambda_\alpha| \leq ADh,$$

where $D = \int_{\mathbb{R}^d} \|z\| K(z) dz$.

Remark: More generally, if p is assumed to be Hölder continuous with parameter β then, under additional mild integrability conditions on K, it can be shown that $|\lambda_{h,\alpha} - \lambda_\alpha| = O(h^\beta)$, uniformly in α.

The following lemma bounds the deviation of $|\widehat{\lambda}_{h,\alpha,X} - \lambda_{h,\alpha}|$.

Lemma 5 *Assume that the true density satisfies (A0)-(A1) and the density level sets of p_h corresponding to probability content α satisfy (B3). Then, for any $0 < h \leq H$, any $\varepsilon < \eta_0 - 1/n$, and all $n \geq n(\varepsilon, h)$,*

$$\mathbb{P}_X\left(|\widehat{\lambda}_{h,\alpha,X} - \lambda_{h,\alpha}| \geq \varepsilon(A\kappa_3 + 1) + A\kappa_3/n\right) \leq K_1 e^{-K_2 n h^d \varepsilon^2} + 8ne^{-n\varepsilon^2/32}, \tag{5}$$

where A is the Lipschitz constant and κ_3 is the constant in (B3).

Using Lemma 4 and Lemma 5, we immediately obtain the following bound on the deviation of the estimated level $\widehat{\lambda}_{h,\alpha,X}$ from the true density level λ_α corresponding to probability content α.

Corollary 6 *Under the same conditions of Lemma 5,*

$$\mathbb{P}_X\left(|\widehat{\lambda}_{h,\alpha,X}-\lambda_\alpha| \geq ADh+\varepsilon(A\kappa_3+1)+A\kappa_3/n\right) \leq K_1 e^{-K_2 nh^d\varepsilon^2}+8ne^{-n\varepsilon^2/32}.$$

We now study the performance of the level set estimator indexed by probability content using the following loss function

$$\mathcal{L}^*(h,X,\alpha)=P(M(\alpha)\Delta\widehat{M}_{h,X}(\alpha))=\int_{M(\alpha)\Delta\widehat{M}_{h,X}(\alpha)} p(u)du.$$

Theorem 7 *Assume that the density p satisfies conditions (A0) and (A1) and the level set of p_h indexed by probability content α satisfies (B3). For any sequence $h_n=\omega((\log n/n)^{1/d})$, let*

$$\varepsilon_n=\sqrt{\frac{K_3\log n}{nh_n^d}}$$

and set

$$C_{1,n}=ADh_n+\varepsilon_n, \quad C_{2,n}=ADh_n+(A\kappa_3+1)\varepsilon_n+A\kappa_3/n$$

and

$$r_{h_n,\varepsilon_n,\alpha}=P\left(\{u\colon |p(u)-\lambda_\alpha|\leq C_{1,n}+C_{2,n}\}\right).$$

Then, for $h_n=\omega((\log n/n)^{1/d})$ and $h_n\leq H$, we have for all $n\geq n(n_0,\eta_0,K_3,d)$,

$$\mathbb{P}_X(\mathcal{L}^*(h_n,X,\alpha)\leq r_{h_n,\varepsilon_n,\alpha})\geq 1-\frac{2}{n}.$$

In particular, if assumption (A2) also holds for density level λ_α, then, for all $n\geq n(n_0,\eta_0,K_3)$,

$$\mathbb{P}_X\left(\mathcal{L}^*(h_n,X,\alpha)\leq \kappa_2(C_{1,n}+C_{2,n})\right)\geq 1-\frac{2}{n}.$$

Corollary 8 *The value of h that minimizes the upper bound on $\mathcal{L}$ is*

$$h^*_{n,\alpha}=c\left(\frac{n}{\log n}\right)^{-\frac{1}{d+2}}$$

where $c>0$ is a constant.

4. Stability

The loss $\mathcal{L}$ is a useful theoretical measure of clustering accuracy. Balancing the terms in the upper bound on the loss gives an indication of the optimal scaling behavior of h. But estimating the loss is difficult and the value of the constant c in the expression for h^*_n is unknown. Thus, in practice, we need an alternative method to determine h. Instead of minimizing the loss, we consider using the stability of $\widehat{L}_{h,X}(\lambda)$ and $\widehat{\mathcal{T}}_h$ to choose h. As we discussed in the introduction, stability ideas have been used for clustering before. But the behavior of stability measures can be quite complicated. For example, in the context of k-means clustering and related methods, Ben-David et al. (2006) showed that minimizing instability leads to poor clustering. On the other hand, Rinaldo and Wasserman (2010) showed that, for density-based clustering, stability-based methods can sometimes lead to good results. This motivates us to take a deeper look at stability for density clustering. In this section, we investigate two measures of cluster stability.

The first measure of cluster stability we analyze is the *level set stability*, which we denote, for a fixed density level λ and a varying bandwidth value h, with $\Xi_{\lambda,n}(h)$. Assuming for convenience that the sample size is $3n$, we randomly split the data into three pieces (X,Y,Z) each of size n. Let $\widehat{p}_{h,X}$ be the density estimator constructed from X and $\widehat{p}_{h,Y}$ be the density estimator constructed from Y. The sample instability statistic is

$$\Xi_{\lambda,n}(h) = \widehat{P}_Z(\widehat{L}_{h,X}(\lambda)\Delta\widehat{L}_{h,Y}(\lambda)), \tag{6}$$

where $\widehat{P}_Z$ denote the empirical measure induced by Z. The measure $\Xi_{\lambda,n}(h)$ is the stability of a fixed level set, as a function of h. We will see that Ξ_n has surprisingly complex behavior. See Figure 2. First of all, $\Xi_n(0) = 0$. This is an artifact and is due to the fact that the level sets get small as $h \to 0$. As h increases, $\Xi_{\lambda,n}(h)$ first increases and then gets smaller. Once it gets small enough, the level sets have become stable and we have reached a good value of h. However, after this point, $\Xi_{\lambda,n}(h)$ continues to rise and fall. The reason is that, as h gets larger, $p_h(x)$ decreases. Every time we reach a value of h such that a mode of p_h has height λ, $\Xi_{\lambda,n}(h)$ will increase. $\Xi_{\lambda,n}(h)$ is thus a non-monotonic function whose mean and variance become large at particular values of h. This behavior will be described explicitly in the theory and simulations that follow. As a practical matter, since $\Xi_{\lambda,n}(h)$ vanishes for very small values of h, we recommend to exclude all values of h before the first local maximum of $\Xi_{\lambda,n}(h)$. Then, a reasonable choice of h is the smallest value h^* for which $\Xi_{\lambda,n}(h)$ remains less than some maximal pre-specified probability value β for the empirical instability, such as 5% or 10%, for all $h \geq h^*$. The parameter β is an entirely subjective quantity to be chosen by the practitioner, akin to the type-I-error parameter in standard hypothesis testing, and quantifies the maximal amount of empirical instability that one is willing to accept.

The second measure of cluster stability we consider is the *total variation* stability, denoted, for a varying value of the bandwidth h, as $\Gamma_n(h)$. Assuming again for simplicity that the sample is of size $2n$, we randomly split the data into two parts (X,Y) of equal sizes n. Then, for a given bandwidth h, we compute separately on each of the two samples X and Y the kernel density estimates $\widehat{p}_{h,X}$ and $\widehat{p}_{h,Y}$, respectively. The total variation stability is defined to be the quantity

$$\Gamma_n(h) \equiv \sup_{B\in\mathcal{B}} \left| \int_B \widehat{p}_{h,X}(u)du - \int_B \widehat{p}_{h,Y}(u)du \right| = \frac{1}{2}\int |\widehat{p}_{h,X}(u) - \widehat{p}_{h,Y}(u)|\, du, \tag{7}$$

where the supremum is over all Borel sets B. Note that the total variation stability is a function of h. Unlike the level set stability, the total variation stability is a global measure of cluster stability in the sense that it takes into account the difference between $\widehat{p}_{h,X}$ and $\widehat{p}_{h,Y}$ overall all measurable sets, not just over the level sets. Thus, total variation stability is a much stronger notion of cluster stability. In fact, when $\Gamma_n(h)$ is small, the whole cluster tree is stable. It turns out that the behavior of $\Gamma_n(h)$ is much simpler. It is monotonically decreasing as a function of h. In this case we recommend choosing h to be the smallest bandwidth value h^* for which the instability is no larger than a pre-specified probability values $\beta \in (0,1)$, that is $\Gamma_n(h^*) \leq \beta$.

The motivation for choosing the bandwidth parameter h in the way described above is as follows. We cannot estimate loss exactly. But we can use the instability to estimate variability. Our choice of

h corresponds to making the bias as small as possible while maintaining control over the variability. This is very much in the spirit of the Neyman-Pearson approach to hypothesis testing where one tries to make the power of a test as large as possible while controlling the probability of false positives. Put another way, $P_h = P * \mathbb{K}_h$ has a blurred version of the shape information in P. *We are choosing the smallest h such that the shape information in P_h can be reliably recovered.*

Before getting into the details, which turn out to be somewhat technical, here is a very loose description of the results. For large h, $\Gamma_n(h) \approx 1/\sqrt{nh^d}$. On the other hand, $\Xi_{\lambda,n}(h)$ tends to oscillate up and down corresponding to the presence of modes of the density. In regions where it is small, it also behaves like $1/\sqrt{nh^d}$.

4.1 Level Set Stability

For the analysis of the level set stability we focus on a single level set indexed by some density level value $\lambda \geq 0$. Consider two independent samples $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$ and set

$$\xi_{\lambda,n}(h) \quad = \quad \mathbb{E}_{XY}\left(P\left(\widehat{L}_{h,X}(\lambda)\Delta\widehat{L}_{h,Y}(\lambda)\right)\right).$$

The quantity $\xi_{\lambda,n}(h)$ measures the expected disagreement between level sets based on two samples as a function of the bandwidth h.

The definition of $\xi_{\lambda,n}$ depends on P which, of course, we do not know. To estimate $\xi_{\lambda,n}(h)$ we use the sample instability statistic defined above in Equation (6), where it was assumed for simplicity that the sample size is $3n$ and the data were randomly split into three pieces (X,Y,Z) each of size n. It is immediate to see that the expectation of the sample instability statistic is precisely $\xi_{\lambda,n}(h)$, that is

$$\xi_{\lambda,n}(h) = \mathbb{E}_{X,Y,Z}[\Xi_{\lambda,n}(h)].$$

Note that since we are using the empirical distribution $\widehat{P}_Z$, the sample instability can be rewritten as

$$\begin{aligned}\Xi_{\lambda,n}(h) \quad &= \quad \frac{1}{n}\sum_{i=1}^{n} I(Z_i \in (\widehat{L}_{h,X}(\lambda)\Delta\widehat{L}_{h,Y}(\lambda)))\\ &= \quad \frac{1}{n}\sum_{i=1}^{n} I(\text{sign}(\widehat{p}_{h,X}(Z_i)-\lambda) \neq \text{sign}(\widehat{p}_{h,Y}(Z_i)-\lambda)).\end{aligned}$$

The above equation show that, for a fixed λ, $\Xi_{\lambda,n}(h)$ is obtained as the fraction of the observations in Z where $\widehat{p}_{h,X}(Z_i) < \lambda < \widehat{p}_{h,Y}(Z_i)$ or $\widehat{p}_{h,X}(Z_i) > \lambda > \widehat{p}_{h,Y}(Z_i)$. This representation is closely tied to the use of the *sample level sets* to construct the cluster tree (Stuetzle and Nugent, 2009) where each level set is represented only by the observations associated with its connected components rather than the feature space. Using the empirical distribution $\widehat{P}_Z$ also removes the need to determine the exact shape of the level sets of the density estimate. The top graph of Figure 2 shows the sample instability as a function of h for $\lambda = 0.09$ for our example distribution depicted in Figure 1. Note that the instability initially drops and then oscillates before dropping to zero at $h = 7.08$, indicating the multi-modality seen in Figure 1. More discussion of this example is in Section 5.

As mentioned at the end of section 2.3, for values of $h \ll \left(\frac{\log n}{n}\right)^{1/d}$, the kernel density estimate $\widehat{p}_h$ is no longer a reliable estimate of p_h. The following simple but important boundary properties of Ξ_n and ξ describes the behavior of the empirical and expected instability when h is either too small or too large.

Lemma 9 *For fixed n and* $\lambda > 0$,

$$\lim_{h\to 0} \xi_{\lambda,n}(h) = \lim_{h\to\infty} \xi_{\lambda,n}(h) = \lim_{h\to 0} \Xi_{\lambda,n}(h) = \lim_{h\to\infty} \Xi_{\lambda,n}(h) = 0,$$

where the last two limits occurs almost surely. In particular, $\xi_{\lambda,n}(h) = O(h^d)$, *as* $h \to 0$.

We now study the behavior of the mean function $\xi_{\lambda,n}(h)$. Let $u \in \mathbb{R}^d$, $h > 0$ and $\varepsilon > 0$, and define

$$\pi_h(u) = \mathbb{P}_X(\widehat{p}_{h,X}(u) > \lambda) \quad \text{and} \quad U_{h,\varepsilon} = \{u \colon |p_h(u) - \lambda| < \varepsilon\}. \tag{8}$$

Theorem 10 *Let* $u \in \mathbb{R}^d$, $h > 0$ *and* $\varepsilon > 0$.

1. *The following identity holds:*

$$\xi_{\lambda,n}(h) = 2\int_{\mathbb{R}^d} \pi_h(u)(1-\pi_h(u))dP(u).$$

2. *Also, for all* $n \geq n(\varepsilon, h)$,

$$r_{h,\varepsilon}\, \underline{A}_{h,\varepsilon} \leq \xi_{\lambda,n}(h) \leq r_{h,\varepsilon}\, \overline{A}_{h,\varepsilon} + 2K_1 e^{-K_2 n h^d \varepsilon^2},$$

where $r_{h,\varepsilon} = P(U_{h,\varepsilon})$,

$$\overline{A}_{h,\varepsilon} = \sup_{u \in U_{h,\varepsilon}} 2\pi_h(u)(1-\pi_h(u))$$

and

$$\underline{A}_{h,\varepsilon} = \inf_{u \in U_{h,\varepsilon}} 2\pi_h(u)(1-\pi_h(u)).$$

Part 2 of the previous theorem implies that the behavior of ξ is essentially captured by the behavior of the probability content $r_{h,\varepsilon}$. This quantity is, in general, a complicated function of both h and ε. While it is easy to see that, for fixed h and a sufficiently well-behaved density p, $r_{h,\varepsilon} \to 0$ as $\varepsilon \to 0$, for fixed ε, $r_{h,\varepsilon}$ can instead be a non-monotonic function of h. See, for example, the bottom right plot in Figure 3, which displays the values $r_{h,\varepsilon}$ as a function of $h \in [0, 4.5]$ and for ε equal to 0.02, 0.05 and 0.1 for the mixture density of Figure 1. In particular, the fluctuations of $r_{h,\varepsilon}$ as a function of h are related to the values of h for which the critical points of p_h are in the interval $[\lambda - \varepsilon, \lambda + \varepsilon]$. The main point to notice is that $r_{h,\varepsilon}$ is a complicated, non-monotonic function of h. This explains why $\Xi_n(h)$ is non-monotonic in h.

We now provide an upper and lower bound on the values of $\overline{A}_{h,\varepsilon}$ and $\underline{A}_{h,\varepsilon}$, respectively, under the simplifying assumption that K is the spherical kernel. Notice that, while $\overline{A}_{h,\varepsilon}$ remains bounded away from ∞ for any sequence $\varepsilon_n \to 0$ and $h_n = \omega(n^{-1/d})$, the same is not true for $\underline{A}_{h,\varepsilon}$, which remains bounded away from 0 as long as $\varepsilon_n = \Theta(\frac{1}{nh_n^d})$ and $h_n = \omega(n^{-1/d})$.

Lemma 11 *Assume that K is the spherical kernel and let* $0 < \varepsilon \leq \lambda/2$. *For a given* $\delta \in (0,1)$, *let*

$$h(\delta, \varepsilon) = \sup\left\{h \colon \sup_{u \in U_{h,\varepsilon}} P(B(u,h)) \leq 1 - \delta\right\}.$$

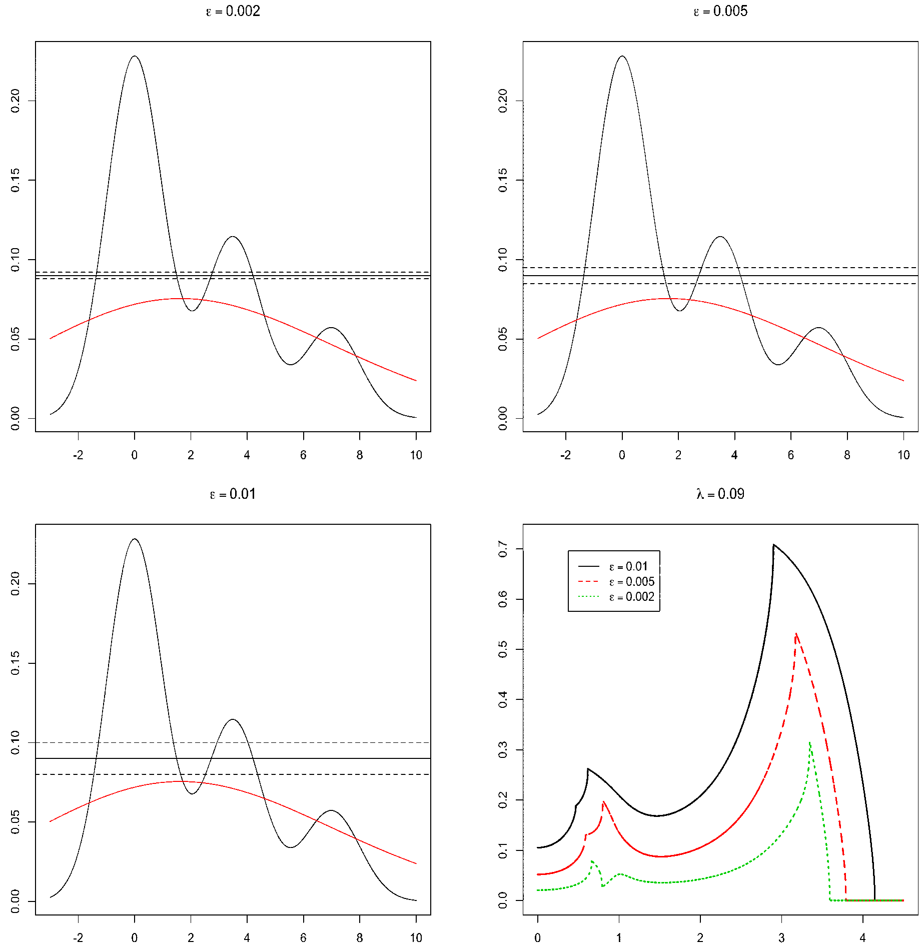


Figure 3: Top plots and left bottom plot: two densities p_h corresponding to the mixture distribution of Figure 1 for $h = 0$, the true density (in black) and $h = 4.5$ (in red); the horizontal lines indicate the level set value of $\lambda = 0.09$, $\lambda + \varepsilon$ and $\lambda - \varepsilon$, for ε equal to $0.02, 0.05$ and 0.1. Right bottom plot: probability content values $r_{h,\varepsilon}$ as a function of $h \in [0, 4.5]$ for the three values of ε.

Then, for all $h \leq h(\delta,\varepsilon)$,

$$\overline{A}_{h,\varepsilon} \leq 2\left(1 - \Phi\left(-\sqrt{nh^d}\,\varepsilon\frac{2v_d}{3\lambda}\right) + \frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

and

$$\underline{A}_{h,\varepsilon} \geq 2\left(1 - \Phi\left(\sqrt{nh^d}\,\varepsilon\frac{2v_d}{\delta\lambda}\right) - \frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

where Φ denote the cumulative distribution function of a standard normal random variable and

$$C(\delta,\lambda) = \frac{33}{4}\sqrt{\frac{2}{\delta v_d \lambda}}.$$

The dips in Figure 2 correspond to values for which p_h does not have a mode at height λ. In this case, (B2) holds and we have $r_{h,\varepsilon} = \Theta(\varepsilon)$. Now choosing $\varepsilon \approx \sqrt{\log n/(nh^d)}$ for the upper bound and $\varepsilon \approx \sqrt{1/(nh^d)}$ for the lower bound, we have that $\overline{A}_{h,\varepsilon}$ and $\underline{A}_{h,\varepsilon}$ are bounded, and the theorem yields

$$\sqrt{\frac{C_1}{nh^d}} \leq \xi_{\lambda,n}(h) \leq \sqrt{\frac{C_2 \log n}{nh^d}}.$$

Next we investigate the extent to which $\Xi_{\lambda,n}(h)$ is concentrated around its mean $\xi_{\lambda,n}(h) = \mathbb{E}[\Xi_{\lambda,n}(h)]$. We first point out that, for any fixed h, the variance of the instability can be bounded by $\xi_{\lambda,n}(h)(1/2 - \xi_{\lambda,n}(h))$.

Lemma 12 *For any $h > 0$,*

$$\mathrm{Var}[\Xi_{\lambda,n}(h)] \leq \xi_{\lambda,n}(h)\left(\frac{n+1}{2n} - \xi_{\lambda,n}(h)\right) \approx \xi_{\lambda,n}(h)\left(\frac{1}{2} - \xi_{\lambda,n}(h)\right).$$

The previous results highlight the interesting feature that the empirical instability will be less variable around the values of h for which the expected instability is very small (close to 0) or very large (close to $1/2$).

Lemma 13 *Suppose that $h > 0$, $\varepsilon > 0$, $\eta \in (0,1)$ and $t > 0$ are such that*

$$t(1-\eta) \geq r_{h,\varepsilon} + 2K_1 e^{-K_2 n \varepsilon^2 h^d}, \tag{9}$$

where $r_{h,\varepsilon} = P(U_{h,\varepsilon})$. Then, for all $n \geq n(\varepsilon, h)$,

$$\mathbb{P}_{X,Y,Z}\left(\left|\Xi_{\lambda,n}(h) - \xi_{\lambda,n}(h)\right| > t\right) \leq e^{-ntC_\eta} + 2K_1 e^{-nK_2 h^d \varepsilon^2}$$

where

$$C_\eta = 9(1-\eta)\left(\frac{3-2\eta}{3(1-\eta)} - \sqrt{\frac{3-\eta}{3(1-\eta)}}\right).$$

4.2 Stability of Level Sets Indexed by Probability Content

As in the fixed-λ case, we assume for simplicity that the sample has size $3n$ and split it equally in three parts: X, Y and Z. We now define the fixed-α instability as

$$\Xi_{\alpha,n}(h) = \widehat{P}_Z(\widehat{M}_{h,X}(\alpha) \Delta \widehat{M}_{h,Y}(\alpha)),$$

where

$$\widehat{M}_{h,X}(\alpha) = \{x \colon \widehat{p}_{h,X}(x) > \widehat{\lambda}_{h,\alpha,X}\},$$

with $\widehat{\lambda}_{h,\alpha,X}$ estimated as in (4) using the points in X; we similarly estimate $\widehat{M}_{h,Y}(\alpha)$. As before, $\widehat{P}_Z$ denote the empirical measure arising from Z. Again, we use the observations to represent $\widehat{M}_{h,X}$, $\widehat{M}_{h,Y}$ as done for $\Xi_{\lambda,n}(h)$ for a fixed λ. Examples of $\Xi_{\alpha,n}(h)$ as a function of h, α can be seen in Section 5.

The expected instability is

$$\xi_{\alpha,n}(h) = \mathbb{E}_{X,Y,Z}[\Xi_{\alpha,n}(h)].$$

We begin by studying the behavior of the expected instability.

Theorem 14 *Let $u \in \mathbb{R}^d$, $h > 0$ and $\varepsilon > 0$, and set*

$$\pi_{h,\alpha}(u) = \mathbb{P}_X(\widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}) \quad \text{and } U_{h,2\varepsilon,\alpha} = \{u\colon |p_h(u) - \lambda_{\alpha,h}| \leq 2\varepsilon\}.$$

1. *The expected instability can be expressed as*

$$\xi_{\alpha,n}(h) = \mathbb{E}_{X,Y,Z}[\Xi_{\alpha,n}(h)] = 2\int_{\mathbb{R}^d} \pi_{h,\alpha}(u)(1-\pi_{h,\alpha}(u))dP(u).$$

2. *Let $\varepsilon < \eta_0 - 1/n$ and $\widetilde{\varepsilon} = \varepsilon(A\kappa_3 + 1) + A\kappa_3/n$. Then, for all $n \geq n(\varepsilon, h)$,*

$$P(U_{h,2\widetilde{\varepsilon},\alpha})\underline{A}_{h,\varepsilon,\alpha} \leq \xi_{\lambda,n}(h) \leq P(U_{h,2\widetilde{\varepsilon},\alpha})\overline{A}_{h,\varepsilon,\alpha} + 4K_1 e^{-K_2 nh^d \varepsilon^2} + 16ne^{-n\varepsilon^2/32},$$

where

$$\overline{A}_{h,\varepsilon,\alpha} = \sup_{u \in U_{h,2\widetilde{\varepsilon},\alpha}} 2\pi_{h,\alpha}(u)(1-\pi_{h,\alpha}(u))$$

and

$$\underline{A}_{h,\varepsilon,\alpha} = \inf_{u \in U_{h,2\widetilde{\varepsilon},\alpha}} 2\pi_{h,\alpha}(u)(1-\pi_{h,\alpha}(u)).$$

3. *Assume in addition that K is the spherical kernel and that $\widetilde{\varepsilon} \leq \inf_h \frac{\lambda_{h,\alpha}}{4}$. For a given $\delta \in (0,1)$, let*

$$h(\delta,\varepsilon,\alpha) = \sup\Big\{h : \sup_{u \in U_{h,\widetilde{\varepsilon},\alpha}} P(B(u,h)) \leq 1-\delta\Big\}.$$

Then, for all $h \leq h(\delta,\varepsilon,\alpha)$,

$$\overline{A}_{h,\varepsilon,\alpha} \leq 2\left(1 - \Phi\left(-3\sqrt{nh^d}\widetilde{\varepsilon}\frac{2v_d}{3\lambda_{h,\alpha}}\right) + \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} + 4K_1 e^{-K_2 nh^d \varepsilon^2} + 16ne^{-n\varepsilon^2/32}\right)^2,$$

and

$$\underline{A}_{h,\varepsilon,\alpha} \geq 2\left(1 - \Phi\left(3\sqrt{nh^d}\widetilde{\varepsilon}\frac{2v_d}{\delta\lambda_{h,\alpha}}\right) - \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} - 4K_1 e^{-K_2 nh^d \varepsilon^2} - 16ne^{-n\varepsilon^2/32}\right)^2,$$

where Φ denote the cumulative distribution function of a standard normal random variable and

$$C(\delta,\lambda_{h,\alpha}) = \frac{33}{4}\sqrt{\frac{2}{\delta v_d \lambda_{h,\alpha}}}.$$

As for the fluctuations of $\Xi_{\alpha,n}(h)$ around its mean, we can easily obtain a result similar to the one we obtain in Lemma 13.

Lemma 15 *Let $h > 0$, $\varepsilon > 0$, $\eta \in (0,1)$ and t be such that*

$$t(1-\eta) \geq r_{h,\varepsilon,\alpha} + 4K_1 e^{-K_2 nh^d \varepsilon^2} + 16ne^{-n\varepsilon^2/32},$$

where $r_{h,\varepsilon,\alpha} = P(\{u\colon |p_h(u) - \lambda_{h,\alpha}| \leq 2\widetilde{\varepsilon}\})$, with $\widetilde{\varepsilon} = \varepsilon(A\kappa_3+1) + A\kappa_3/n$. Then, for all $n \geq n(\varepsilon,h)$,

$$\mathbb{P}_{X,Y,Z}(|\Xi_{\alpha,n}(h) - \xi_{\alpha,n}(h)| > t) \leq e^{-ntC_\eta} + 4K_1 e^{-K_2 nh^d \varepsilon^2} + 16ne^{-n\varepsilon^2/32},$$

where

$$C_\eta = 9(1-\eta)\left(\frac{3-2\eta}{3(1-\eta)} - \sqrt{\frac{3-\eta}{3(1-\eta)}}\right).$$

The proof is basically the same as the proof of Lemma 13, except that we have to restrict our analysis to the event described in Equation (29). We omit the details.

4.3 Stability for Density Cluster Trees

The stability properties of the cluster tree can be easily derived from the results we have established so far. To this end, for a fixed $h > 0$, define the level set of p_h

$$L_h(\lambda) = \{u : p_h(u) > \lambda\}$$

and recall its estimator based on the kernel density estimator $\widehat{p}_{h,X}$:

$$\widehat{L}_{h,X}(\lambda) = \{u\colon\, \widehat{p}_{h,X}(u) > \lambda\}.$$

Let $N_h(\lambda)$, $\widehat{N}_{h,X}(\lambda)$ be the number of connected components of the sets $L_h(\lambda)$ and $\widehat{L}_{h,X}(\lambda)$, respectively. Notice that $\widehat{L}_{h,X}(\lambda)$ is a random set. Also, denote with $C_1,\ldots,C_{N_h(\lambda)}$ and $\widehat{C}_1,\ldots,\widehat{C}_{\widehat{N}_{h,X}(\lambda)}$ the connected components of $L_h(\lambda)$ and $\widehat{L}_{h,X}(\lambda)$, respectively.

When building cluster trees, the value of the bandwidth h is kept fixed and the values of the level λ vary instead. It has been observed empirically (see, for instance Stuetzle and Nugent, 2009) that the uncertainty of cluster tree estimators depend on the particular value of λ at which the tree is observed. In order to characterize the behavior of the cluster tree, we propose the following definition, which formalize the case in which the clusters $C_1,\ldots,C_{N_h(\lambda')}$ persist for each λ' in a neighborhood of λ.

Definition 16 *A level set value λ is (h,ε)-stable, with $\varepsilon > 0$ and $h > 0$, if*

$$N_h(\lambda) = N_h(\lambda'), \quad \forall \lambda' \in (\lambda - \varepsilon, \lambda + \varepsilon)$$

and, for any $\lambda - \varepsilon < \lambda_1 < \lambda_2 < \lambda + \varepsilon$,

$$C_i(\lambda_2) \subseteq C_i(\lambda_1), \quad \forall i = 1,\ldots,N_h(\lambda).$$

If the level λ is (h,ε)-stable, then the cluster tree estimate at level λ is an accurate estimate of the true cluster tree, in a sense made precise by the following result, whose proof follows easily from the proofs of our previous results and Lemma 2 in Rinaldo and Wasserman (2010).

Lemma 17 *If λ is (h,ε)-stable, then, for all large $n \geq n(\varepsilon,\lambda)$, with probability at least $1 - \frac{1}{n}$,*

1. $N_h(\lambda) = \widehat{N}_{h,X}(\lambda)$;
2. *there exists a permutation σ on $\{1,\ldots,N_h(\lambda)\}$ such that, for every connected component C_j of $L_h(\lambda - \varepsilon)$ there exists one $\widehat{C}_{\sigma(j)}$ for which*

$$C_j \subseteq \widehat{C}_{\sigma(j)};$$

3. $P(\widehat{L}_{h,X}(\lambda)\Delta L_h(\lambda)) \leq P(\{u : |p_h(u) - \lambda| < \varepsilon\})$.

Remarks.

1. If p_h is smooth (which is the case if, for instance, the kernel or p are smooth), the values of λ which are not (h, ε)-stable are values for which the set $U_{\lambda',h,\varepsilon}$ contains critical points of p_h, that is
$$\inf_{u \in U_{\lambda',h,\varepsilon}} \|\nabla p_h(u)\| = 0 \quad \text{for some } \lambda' \in (\lambda - \varepsilon, \lambda + \varepsilon),$$
where ∇p_h denotes the gradient of p. For those values, the probability of $N_h(\lambda) \neq \widehat{N}_{h,X}(\lambda)$ can be quite large, since the set $\widehat{L}_{h,X} \Delta L_h(\lambda)$ may have a relatively large P-mass.

2. Conversely, if p_h is smooth (which is the case if, for instance, the kernel or p are smooth) and $\inf_{u \in U_{\lambda,h,\varepsilon}} \|\nabla p_h(u)\| > \delta$, then λ is (h, ε)-stable for a small enough ε.

The above result has a somewhat limited practical value, because the notion of a (h, ε)-stable λ depends on the unknown density p_h. In order to get a better sense of which λ's are (h, ε)-stable or not, we once again resort to evaluate the instability of the clustering solution via data splitting. In fact, essentially all of our previous results about instability from section 4.1 carry over to these new settings by treating h fixed and letting λ vary. To express this changes explicitly, we will adopt a slightly different notation for quantities we have already considered. In particular, we let

$$\begin{array}{rcl} U_{\lambda,\varepsilon} & = & \{u \colon |p_h(u) - \lambda| < \varepsilon\}, \\ r_{\lambda,\varepsilon} & = & P(U_{\lambda,\varepsilon}), \\ \pi_\lambda(u) & = & \mathbb{P}_X(\widehat{p}_{h,X}(u) > \lambda), \\ \overline{A}_{\lambda,\varepsilon} & = & \sup_{u \in U_{\lambda,\varepsilon}} 2\pi_\lambda(u)(1 - \pi_\lambda(u)) \end{array}$$

and

$$\underline{A}_{\lambda,\varepsilon} = \inf_{u \in U_{\lambda,\varepsilon}} 2\pi_\lambda(u)(1 - \pi_\lambda(u)).$$

We divide the sample size into three distinct groups, X, Y and Z, of equal sizes n. For a fixed bandwidth h, we define the instability of the density cluster tree as the random function $T_{h,n} \colon \mathbb{R}_{\geq 0} \mapsto [0,1]$ given by

$$\lambda \rightarrow \widehat{\mathbb{P}}_Z(\widehat{L}_{h,X}(\lambda) \Delta \widehat{L}_{h,Y}(\lambda))$$

and denote its expectation by

$$\tau_{h,n}(\lambda) = \mathbb{E}_{X,Y,Z}[T_{h,n}(\lambda)].$$

For any fixed h, the behavior of $T_{h,n}(\lambda)$ and $\tau_{h,n}(\lambda)$ is essentially governed by $r_{\lambda,\varepsilon}$. The following result describes some of the properties of the density tree instability. We omit its proof, because it relies essentially on the same arguments from the proofs of the results described in section 4.1.

Corollary 18

1. *For any $\lambda > 0$, the expected cluster tree instability can be expressed as*
$$\tau_{h,n}(\lambda) = 2 \int \pi_\lambda(u)(1 - \pi_\lambda(u)) dP(u).$$

2. *For any $\varepsilon > 0$ and $\lambda > 0$,*
$$\underline{A}_{\lambda,\varepsilon} r_{\lambda,\varepsilon} \leq \tau_{h,n}(\lambda) \leq \underline{A}_{\lambda,\varepsilon} r_{\lambda,\varepsilon} + 2K_1 e^{-K_2 n h^d \varepsilon^2},$$
for all n large enough.

3. *Assume that K is the spherical kernel. For any* $\lambda > 0$, *let* $0 < \varepsilon \leq \frac{\lambda}{2}$ *and let*

$$\delta = 1 - \sup_u P(B(u,h)).$$

Then,

$$\overline{A}_{\lambda,\varepsilon} \leq 2\left(1 - \Phi\left(-\sqrt{nh^d}\varepsilon\frac{2v_d}{3\lambda}\right) + \frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

and

$$\underline{A}_{\lambda,\varepsilon} \geq 2\left(1 - \Phi\left(\sqrt{nh^d}\varepsilon\frac{2v_d}{\delta\lambda}\right) - \frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

where Φ *denote the cumulative distribution function of a standard normal random variable and*

$$C(\delta,\lambda) = \frac{33}{4}\sqrt{\frac{2}{\delta v_d \lambda}}.$$

4. *For any* $h > 0$, $\varepsilon > 0$, $\eta \in (0,1)$ *let t by such that*

$$t(1-\eta) \geq r_{\lambda,\varepsilon} + 2K_1 e^{-K_2 n\varepsilon^2 h^d},$$

Then, for all $n \geq n(\varepsilon,h)$,

$$\mathbb{P}_{X,Y,Z}\left(|T_{h,n}(\lambda) - \tau_{h,n}(\lambda)| > t\right) \leq e^{-ntC_\eta} + 2K_1 e^{-nK_2 h^d \varepsilon^2}.$$

with

$$C_\eta = 9(1-\eta)\left(\frac{3-2\eta}{3(1-\eta)} - \sqrt{\frac{3-\eta}{3(1-\eta)}}\right).$$

Collectively, the results above results show that the cluster tree of p_h can be estimated more accurately for values of λ for which the quantity $r_{\lambda,\varepsilon}$ remain small, with ε a term vanishing in n. In particular, the level sets λ with larger instability are then the ones that are close to a critical level of p_h or for which the gradient of p_h is not defined, vanishes of has infinite norm for some points in $\{x\colon\, p_h(x) = \lambda\}$. This suggests that the sample complexity for accurately reconstructing of the cluster tree may vary significantly depending on the particular level of the tree, with levels closer to a branching point exhibiting a higher degree of uncertainty and, therefore, requiring larger sample sizes.

4.4 Total Variation Stability

In the previous section, we established stability of the cluster tree for a fixed h and all levels λ that are (h,ε)-stable. A more complete measure of stability would be to establish stability of the entire cluster tree. However, it appears that this is not feasible. Here we investigate an interesting alternative: we compare the entire distribution $\widehat{p}_{h,X}$ to the entire distribution $\widehat{p}_{h,Y}$. The idea is that if these two distributions are stable over all measurable sets, then this implies it is stable over any class of subsets, including all clusters.

More precisely, we consider the stronger notion of instability corresponding to the total variation stability as defined in (7). Recall that we assume that the data have sample size $2n$ and we randomly split them into two sets of size n, X and Y, with which we compute the he kernel density estimates $\widehat{p}_{h,X}$ and $\widehat{p}_{h,Y}$, for a given value of the bandwidth h. Then, the total variation stability is defined as

$$\Gamma_n(h) \equiv \sup_{B \in \mathcal{B}} \left| \int_B \widehat{p}_{h,X}(u)du - \int_B \widehat{p}_{h,Y}(u)du \right| = \frac{1}{2} \int |\widehat{p}_{h,X}(u) - \widehat{p}_{h,Y}(u)|\, du$$

where where the supremum is over all Borel sets B and the second equality is a standard identity. Requiring $\Gamma_n(h)$ to be small is a more demanding type of stability. In particular, $\mathcal{B}$ includes all level sets for all λ. Thus, when $\Gamma_n(h)$ is small, the entire cluster tree is stable. Note that $\Gamma_n(h)$ is easy to interpret: it is the maximum difference in probability between the two density estimators. And of course $0 \leq \Gamma_n(h) \leq 1$. The bottom graph in Figure 2 shows the total variation instability for our example distribution in Figure 1. Note that $\Gamma_n(h)$ first drops drastically as h increases and then continues to smoothly decrease.

We now discuss the properties of $\Gamma_n(h)$. Note first that $\Gamma_n(h) \approx 1$ for small h so the behavior as h gets large is most relevant.

Theorem 19 *Let $\mathcal{H}_n$ be a finite set of bandwidths such that $|\mathcal{H}_n| = Hn^a$, for some positive H and $a \in (0,1)$. Fix a $\delta \in (0,1)$.*

1. *(Upper bound.) There exists a constant C such that, for all $n \geq n_0 \equiv n_0(\delta, H, a)$, and such that $\delta > H/n$,*

$$\mathbb{P}_{X,Y}\left(\Gamma_n(h) \leq t_h \text{ for all } h \in \mathcal{H}_n\right) > 1 - \delta,$$

where $t_h = \sqrt{\frac{C \log n}{nh^d}}$.

2. *(Lower bound.) Suppose that K is the spherical kernel and that the probability distribution P satisfies the conditions*

$$a_1 h^d v_d \leq \inf_{u \in S} P(B(u,h)) \leq \sup_{u \in S} P(B(u,h)) \leq h^d v_d a_2, \quad \forall h > 0, \tag{10}$$

for some positive constants $a_1 < a_2$, where S denotes the support of P. Let h_* be such that $\sup_u P(B(u,h_*)) < 1 - \delta$. There exists a t, depending on δ but not on h, such that, for all $h < h_*$ and for $n \geq n_0 \equiv n_0(a, a_1, a_2, h, delta)$

$$\mathbb{P}_{X,Y}\left(\Gamma_n(h) \geq t\sqrt{\frac{1}{nh^d}}\right) > 1 - \delta.$$

3. *$\Gamma_n(0) = 1$ and $\Gamma_n(\infty) = 0$.*

Remarks.

1. Note that the upper bound is uniform in h while the lower bound is pointwise in h. Making the lower bound uniform is an open problem. However, if we place a nonzero lower bound on the bandwidths in $\mathcal{H}_n$ then the bound could be made uniform. The latter approach was used in Chaudhuri and Marron (2000).

2. Conditions (10) are quite standard in support set estimation. In particular, when the lower bound holds, the support S is said to be *standard*. See, for instance, Cuevas and Rodríguez-Casal (2004).

In low dimensions, we can compute $\Gamma_n(h)$ by numerically evaluating the integral

$$\frac{1}{2}\int |\widehat{p}_{h,X}(u)-\widehat{p}_{h,Y}(u)|\,du.$$

In high dimensions it may be easier to use importance sampling as follows. Let $g(u) = (1/2)(\widehat{p}_{h,X}(u)+\widehat{p}_{h,Y}(u))$. Then,

$$\Gamma_n(h) = \frac{1}{2}\int \frac{|\widehat{p}_{h,X}(u)-\widehat{p}_{h,Y}(u)|}{g(u)} g(u)du \approx \frac{1}{N}\sum_{i=1}^{N}\frac{|\widehat{p}_{h,X}(U_i)-\widehat{p}_{h,Y}(U_i)|}{|\widehat{p}_{h,X}(U_i)+\widehat{p}_{h,Y}(U_i)|},$$

where $U_1,\dots,U_N$ is a random sample from g. We can thus estimate $\Gamma_n(h)$ with the following algorithm:

1. Draw Bernoulli(1/2) random variables $Z_1,\dots,Z_N$.
2. Draw $U_1,\dots,U_N$ as follows:
   (a) If $Z_i = 1$: draw X randomly from $X_1,\dots,X_n$. Draw $W \sim K$. Set $U_i = X + hW$.
   (b) If $Z_i = 0$: draw Y randomly from $Y_1,\dots,Y_n$. Draw $W \sim K$. Set $U_i = Y + hW$.
3. Set
$$\widehat{\Gamma}_n(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{|\widehat{p}_{h,X}(U_i)-\widehat{p}_{h,Y}(U_i)|}{|\widehat{p}_{h,X}(U_i)+\widehat{p}_{h,Y}(U_i)|}.$$

It is easy to see that U_i has density g and that $\widehat{\Gamma}_n(h)-\Gamma_n(h) = O_P(1/\sqrt{N})$ which is negligible for large N.

5. Examples

We present results for two examples where, although the dimensionality is low, estimating the connected components of the true level sets is surprisingly difficult. For the first example, we begin by illustrating how the instability changes for given values of λ,α and then split each data set 200 times to find point-wise confidence bands for $\Xi_{\lambda,n}(h)$ for fixed λ,α and for $\Gamma_n(h)$. We then present selected results for a bivariate example.

5.1 Instability as Function of h for Fixed λ

Returning to the example distribution in Section 1, 600 observations were sampled from the following mixture of normals: $(4/7)N(0,1)+(2/7)N(3.5,1)+(1/7)N(7,1)$. The original sample is randomly split into three samples of 200. All kernel density estimates use the Epanechnikov kernel

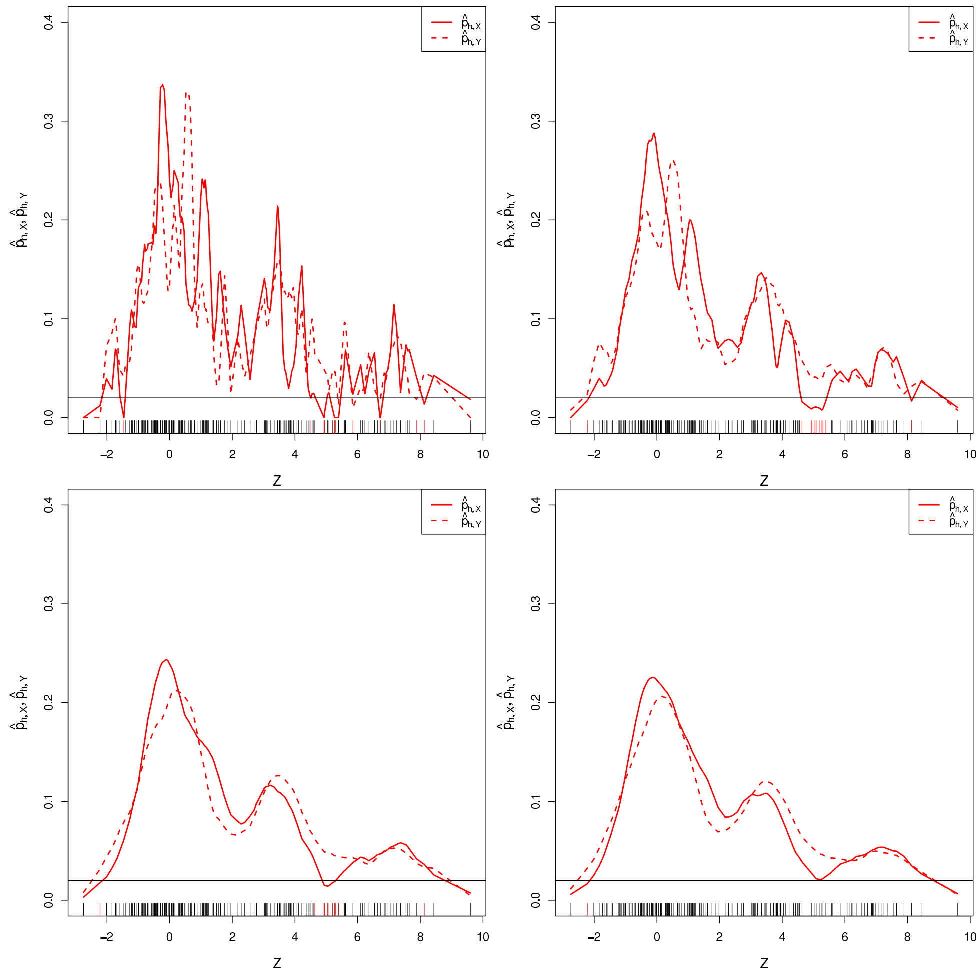


Figure 4: Comparing $\widehat{L}_{h,X}(0.02)$ and $\widehat{L}_{h,Y}(0.02)$ with $h = 0.15$ (top left), $h = 0.35$ (top right), $h = 0.75$ (bottom left) and $h = 0.95$ (bottom right) for data sampled from the mixture distribution of Figure 1. The two kernel density estimates are obtained using the X sample (solid line) and the Y sample (dotted line). Points in the Z sample are showed as short vertical lines on the x-axis, and are colored in red when they belong to $\widehat{L}_{h,X}(\lambda)\Delta\widehat{L}_{h,Y}(\lambda)$.

(Scott, 1992). We examine the stability at $\lambda = 0.02$, a height at which the true density's connected components should be unambiguous, and $\lambda = 0.09$, the height used in our earlier motivating graphs.

We start by illustrating the instability for selected values of h in Figures 4, 5. In each subfigure, $\widehat{p}_{h,X}, \widehat{p}_{h,Y}$ are graphed for the Z set of observations. Levels $\lambda = 0.02, 0.09$ are marked respectively with a horizontal line. Those observations in Z that belong to $\widehat{L}_{h,X}(\lambda)$ and not to $\widehat{L}_{h,Y}(\lambda)$ (or vice versa) are marked in red; the overall fraction of these observations is $\Xi_{\lambda,n}(h)$. In general, we can see that as h increases, the number of the red Z observations decreases. For $\lambda = 0.02$, note that the location that most contributes to the instability is the valley around $Z = 5$. Once h is large enough to smooth this valley to have height above $\lambda = 0.02$, the instability is negligible. Turning to $\lambda = 0.09$ (Figure 5), even for larger values of h, the differences between the two density estimates can be

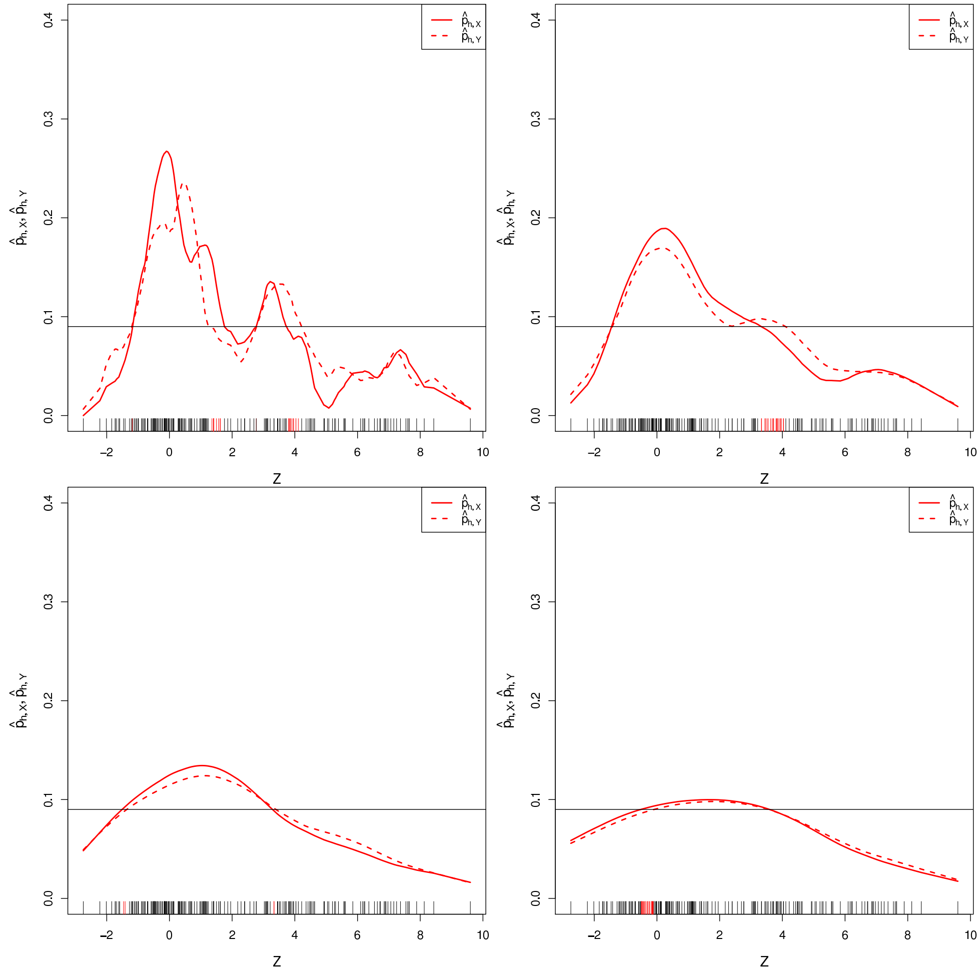


Figure 5: Comparing $\widehat{L}_{h,X}(0.09)$ and $\widehat{L}_{h,Y}(0.09)$ for $h = 0.5$ (top left), $h = 1.75$ (top right), $h = 3.75$ (bottom left) and $h = 6$ (bottom right) for data sampled from the mixture distribution of Figure 1. The two kernel density estimates are obtained using the X sample (solid line) and the Y sample (dotted line). Points in the Z sample are showed as short vertical lines on the x-axis, and are colored in red when they belong to $\widehat{L}_{h,X}(\lambda)\Delta\widehat{L}_{h,Y}(\lambda)$.

quite large. When h is large enough such that both density estimates lie entirely below $\lambda = 0.09$, our instability drops to and remains at zero.

Figure 6 shows the overall behavior of $\Xi_{\lambda,n}(h)$ as a function of h. As expected, for $\lambda = 0.02$, $\Xi_{\lambda,n}(h)$ jumps for the first non-zero h and then quickly drops to almost zero by $h = 1$ (Figure 6, left). At $\lambda = 0.09$, a height with a wide range of possible level sets (depending on the density estimate and the value of h), $\Xi_{\lambda,n}(h)$ first drops and then oscillates as previously described as h increases, indicating multi-modality (Figure 6, right).

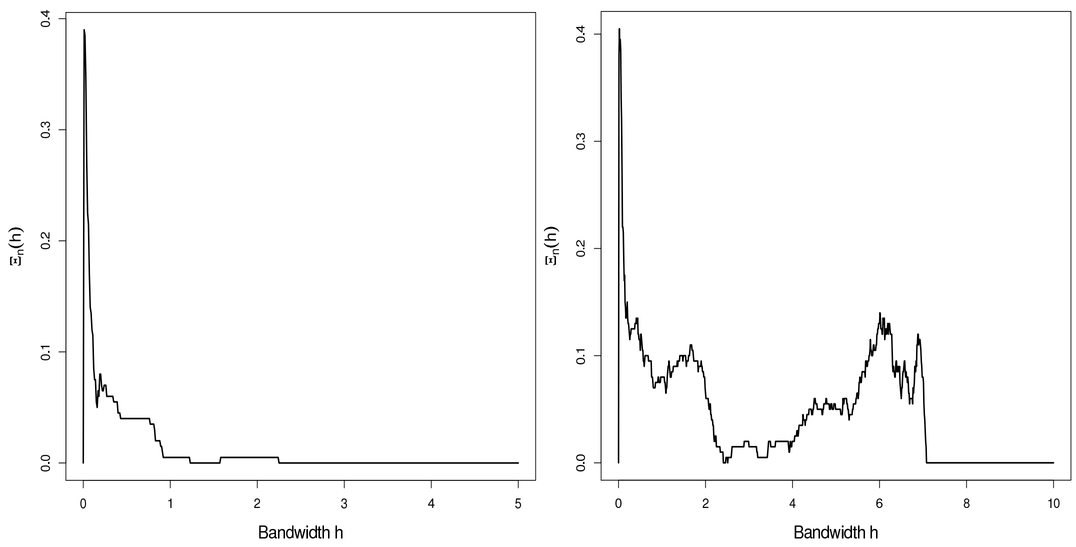


Figure 6: $\Xi_{\lambda,n}(h)$ as a function of the bandwidth h for $\lambda = 0.02$ (left) and 0.09 (right) for data sampled from the mixture distribution of Figure 1.

5.2 Instability as Function h for Fixed α

In Section 4.2 we consider the sample instability $\Xi_{\alpha,n}(h)$ as a function of h and α. As done before, we show $\Xi_{\alpha,n}(h)$ for selected values of h and $\alpha = 0.50$ and 0.95 in Figure 7. In each subfigure, $\widehat{p}_{h,X}, \widehat{p}_{h,Y}$ again are graphed for the Z set of observations. The probability content of the density estimates are respectively indicated on the left and right axes. The values $\alpha = 0.50, 0.95$ are also marked with solid and dashed horizontal lines for the two density estimates. Those observations in Z that belong to $\widehat{M}_{h,X}(\alpha)$ and not to $\widehat{M}_{h,Y}(\alpha)$ (or vice versa) are marked in red; the overall fraction of these observations is $\Xi_{\alpha,n}(h)$. In general, we can see that as h increases (for both values of α), the number of red Z observations decreases. This decrease happens more quickly for higher values of α (as expected).

In Figure 8, we display $\Xi_n(h,\alpha)$ as a function of h for $\alpha = 0.50, 0.95$. For level sets that contain at least 50% probability content, such as $\widehat{M}_{h,X}(0.50)$, the instability quickly drops as h increases and then oscillates as h approaches values that correspond to density estimates with uncertainty at those levels. Again, this ambiguity occurs due to the presence of the second mode (we would see similar behavior with respect to the smallest mode if $\alpha \approx 0.80$). As h continues to increase, the density estimates become smooth enough that there is very little difference between $M_{h,X}(0.50)$, $M_{h,Y}(0.50)$. This behavior also occurs when $\alpha = 0.95$ albeit more quickly (Figure 8, top right) since level sets that contain at least 95% probability content occur at lower heights and are more stable.

Figure 8c is the corresponding heat map for $\alpha = 0, 0.01, \dots, 1.0$ and $h = 0, 0.01, \dots, 10$. White sections indicate $\Xi_{\alpha,n}(h) \approx 0$; black sections indicate higher instability values. In this particular example, the maximum instability of 0.425 is found at $h = 0.03, \alpha = 0.46$. Note that around $h = 3$, we have very low instability values for almost all values of α, and hence this value of kernel bandwidth would be a good choice that yields stable clustering.

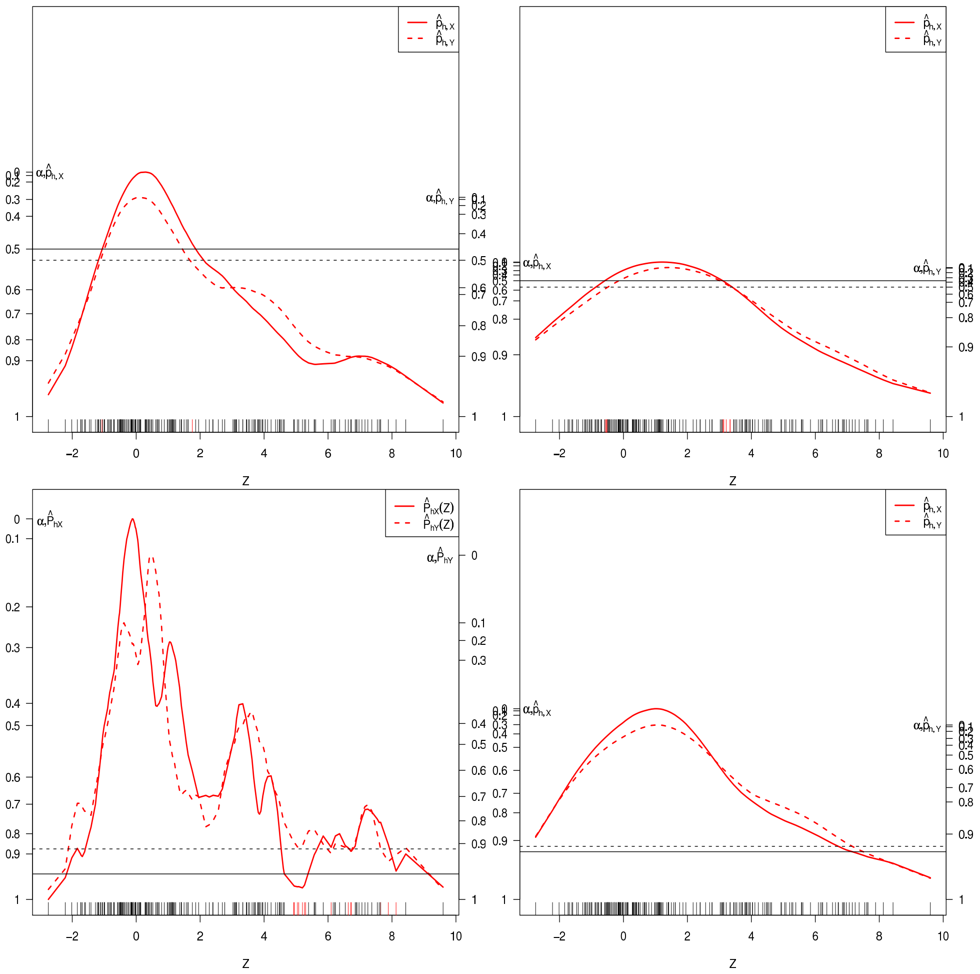


Figure 7: Top: comparing $\widehat{M}_{h,X}(0.50)$ and $\widehat{M}_{h,Y}(0.50)$ for $h = 2$ (left) and $h = 5$ right). Bottom: comparing $\widehat{M}_{h,X}(0.95)$ and $\widehat{M}_{h,Y}(0.95)$ for $h = 0.4$ (left) and $h = 3.5$ (right). The data were sampled from the mixture distribution of Figure 1. The two kernel density estimates are obtained using the X sample (solid line) and the Y sample (dotted line). Points in the Z sample are showed as short vertical lines on the x-axis, and are colored in red when they belong to $\widehat{M}_{h,X}(\alpha)\Delta\widehat{M}_{h,Y}(\alpha)$.

5.3 Instability Confidence Bands

The results in the previous subsections were for splitting the original sample one time into three groups of 200 observations. Here we briefly include a snapshot of what the distribution of our instability measures look like over repeated splits. For computational reasons, we used the binned kernel density estimate, again with the Epanechnikov kernel, and discretize the feature space over 200 bins; see Wand (1994). Increasing the number of bins improves the approximation to the kernel density estimate; the use of two hundred bins was found to give almost identical results to the original kernel density estimate (results not shown). We split the original sample 200 times and find 95% point-wise confidence intervals for $\Xi_{\lambda,n}(h)$, $\Gamma_n(h)$, and $\Xi_{\alpha,n}(h)$ for $\alpha = 0.50, 0.95$ and as a function of h. The results are depicted in Figure 9. The confidence bands are plotted in red, the

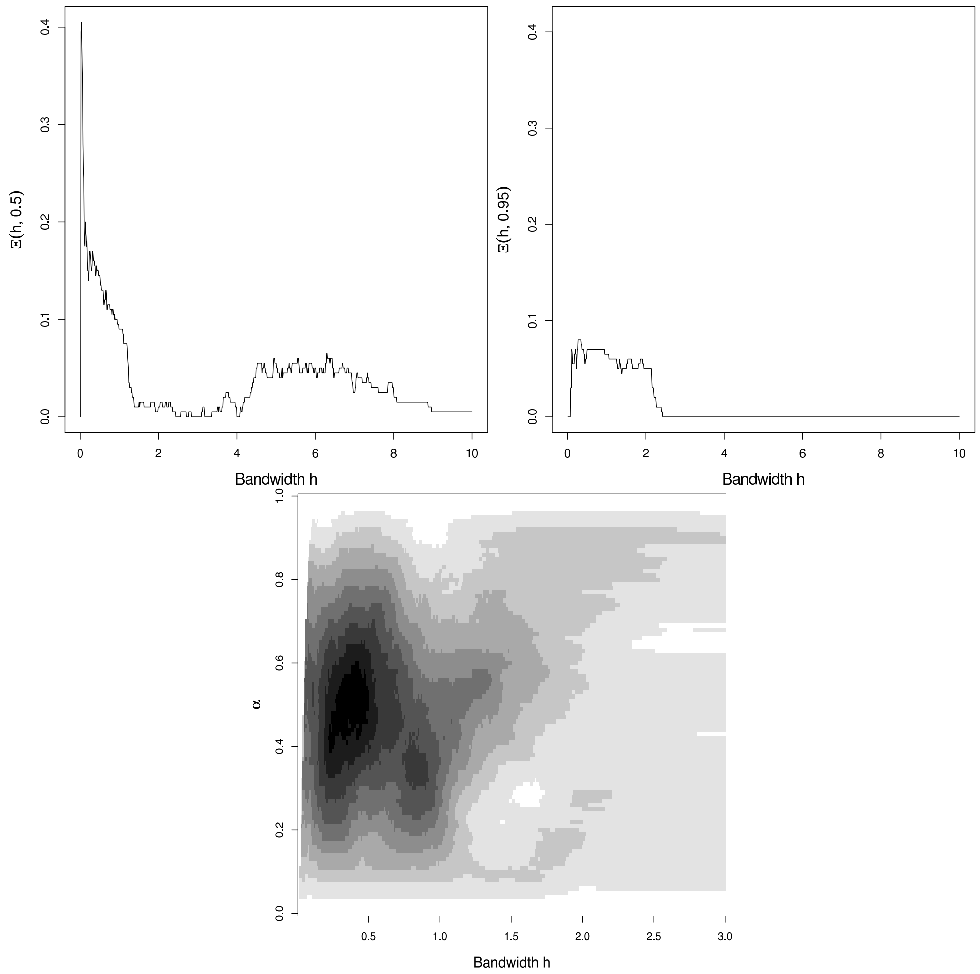


Figure 8: Top: $\Xi_n(h, \alpha = 0.50)$ (left) and $\Xi_n(h, \alpha = 0.95)$ (right) as a function of h. Bottom: heat map of $\Xi_{\alpha,n}(h)$ as function of h, α for the example of Figure 1. The data were sampled from the mixture distribution of Figure 1.

medians in black. The distribution of the instability measures for each value of h is also plotted using density strips (see Jackson, 2008); on the grey-scale, darker colors indicate more common instability values. The density strips allow us to see how the distribution changes (not just the 50, 95% percentiles). For example, for the plot on the top left in Figure 9, note that right before $h = 2$, the upper half of the distribution of $\Xi_{\lambda,n}(h)$ is more concentrated. This shift corresponds to the increase in instability in the presence of the additional modes.

5.4 Bivariate Moons

We also include a bivariate example with two equal-sized moons; this data set with seemingly simple structure can be quite difficult to analyze. The scatterplot of the data on the left in Figure 10 show two clusters, each shaped like a half moon. Each cluster contains 300 data points. The plot on the right in Figure 10b shows a two-dimensional kernel density estimate using a Epanechnikov kernel with $h = 0.60$ (for illustrative purposes) and 10,000 evaluation points. We can see that while levels

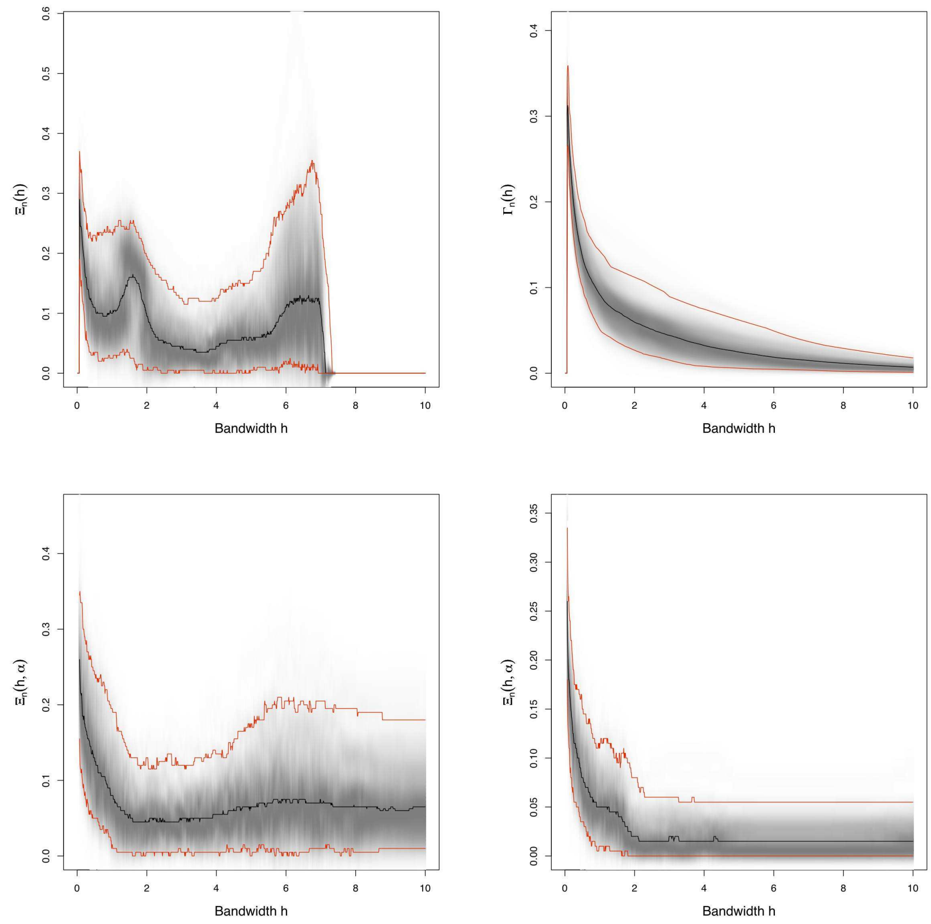


Figure 9: 95% point-wise confidence bands for $\Xi_{\lambda,n}(h)$ (top left), $\Gamma_n(h)$ (top right), $\Xi_n(h,\alpha = 0.50)$ (bottom left) and $\Xi_n(h,\alpha = 0.95)$ (bottom right) for data sampled from the mixture distribution of Figure 1.

around $\lambda = 0.012$ show clear multi-modality, the connectedness of the level sets around $\lambda = 0.01$ is less clear.

To examine instability, we use a product Epanechnikov kernel density estimate with the same bandwidth h for both dimensions. Figure 11 shows the sample instability $\Xi_{\lambda,n}(h)$ as a function of h for $\lambda = 0.10, 0.20, 0.30$ as well as the total variation instability $\Gamma_n(h)$ as a function of h. As expected, the higher the λ, the more quickly the sample instability drops. We also see the possible presence of multi-modality for all three values of λ in $\Xi_{\lambda,n}(h)$. On the other hand, the total variation instability drops smoothly as h increases.

Figure 12 contains the instability as a function of h and probability content α for all values of h, α (Figure 12d) and specifically for $\alpha = 0.50, 0.075, 0.95$. Again, as expected, $\Xi_n(h,\alpha)$ drops as h increases for smaller values of α. Note that for $\alpha = 0.95$, the instability remains relatively low regardless of the value of h. When examining the heat map, we see that for small values of h, level sets corresponding to probability content around 0.4-0.6 are very unstable. This behavior

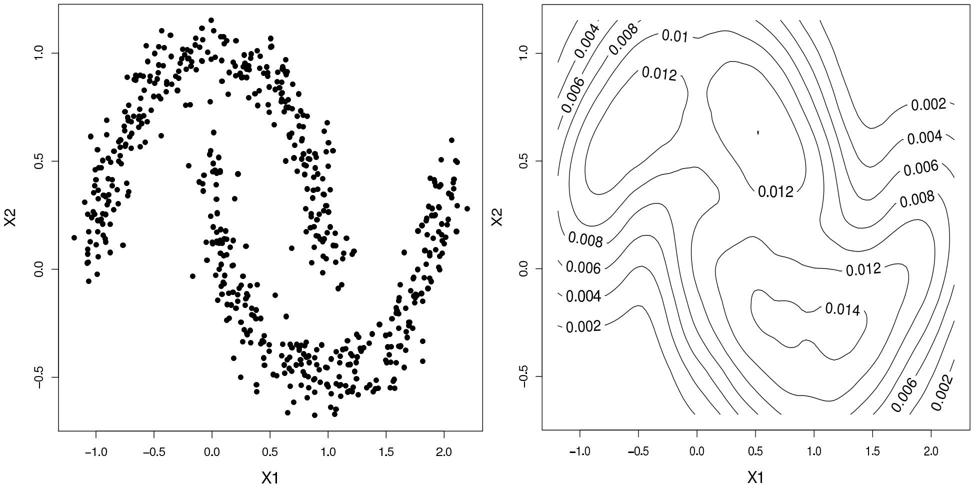


Figure 10: Bivariate moons (left) and contours of a Epanechnikov kernel density estimate (right) for the example discussed in Section 5.4.

is not unexpected given that the moons are of equal sizes and difficult to separate due to sampling variability. We would expect to have difficulty finding stable level sets “in the middle”.

6. Discussion

We have investigated the properties of the density level set and cluster tree estimator based on kernel density estimates, and we have proposed and analyzed various measures of instability for these quantities. We believe these measures of instability can be of guidance in choosing the bandwidth parameter and also as exploratory tools to gain insights into the properties and shape of the data-generating distribution.

Our analysis leaves some some open questions that we think deserve further attention. First, we have focused on kernel density estimators but the same ideas can be used with other density estimators or more, generally, with other clustering methods for which underlying tuning parameters have to be chosen in a data-driven fashion. See, for instance, Meinshausen and Bühlmann (2010) for a related stability-based approach to clustering.

We have assumed the existence of the Lebesgue density p but this assumption can be relaxed using methods in Rinaldo and Wasserman (2010) to allow for distributions supported on lower-dimensional, well-behaved subsets. This extension is potentially important because it would allows us to include cases where the distribution has positive mass on lower dimensional structures such as points and manifolds.

We have formulated our assumptions and results about stability of the level sets and of the cluster tree in a point-wise manner, for given values of λ and α. As suggested by a reviewer, it would be desirable to extend them to hold uniformly across level sets. This can be achieved by requiring (A2), (B2) and (B3) to hold uniformly over values of λ and α. In fact, we believe that it is likely that, for most densities, such uniform assumptions hold for a wide range of λ's but certainly they cannot hold for *all* λ's. Indeed, our results indicate that these uniformity assumptions are reasonable only for level sets λ for which the function $r_{h,\varepsilon}(\lambda)$ remains small and does not fluctuate too wildly.

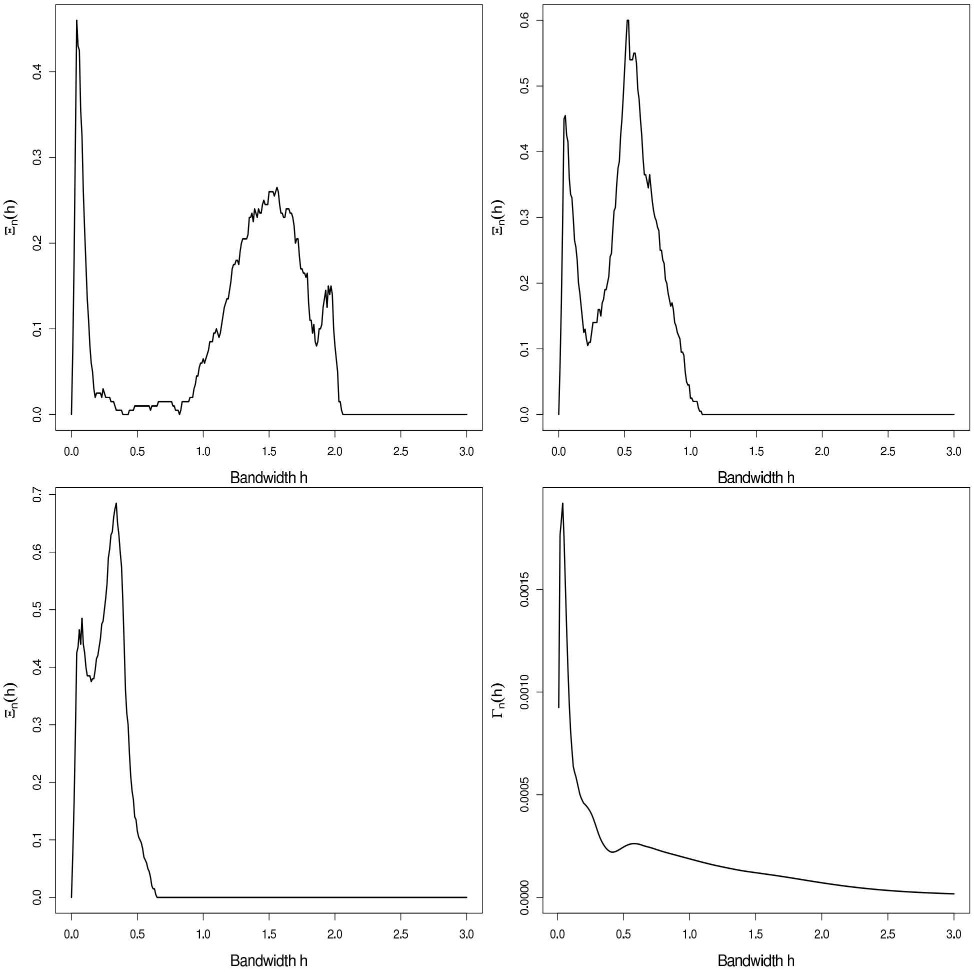


Figure 11: $\Xi_{\lambda,n}(h)$ as a function of h for $\lambda = 0.10$ (top left) 0.20 (top right) and 0.30 (bottom left). $\Gamma_n(h)$ as a function of h (bottom right) for the data depicted in Figure 10.

Finally, in computing the various measures of instability, we have considered just a single split of the data into non-overlapping sub-samples. In fact, one can randomly repeat the splitting process and combine over many splits, which is how we obtained the confidence bands of Figure 9. Though the increase in the computational costs may be significant, repeated sub-sampling would yield a reliable estimate of the uncertainty of the chosen instability measures and would therefore be highly informative about the sample. We believe that the properties of Ξ_n can be established using the theory of U-statistics.

Acknowledgments

Research supported by NSF grant CCF-0625879 and AFOSR contract FA9550-09-1-0373. The authors thank the referees for helpful comments and suggestions.

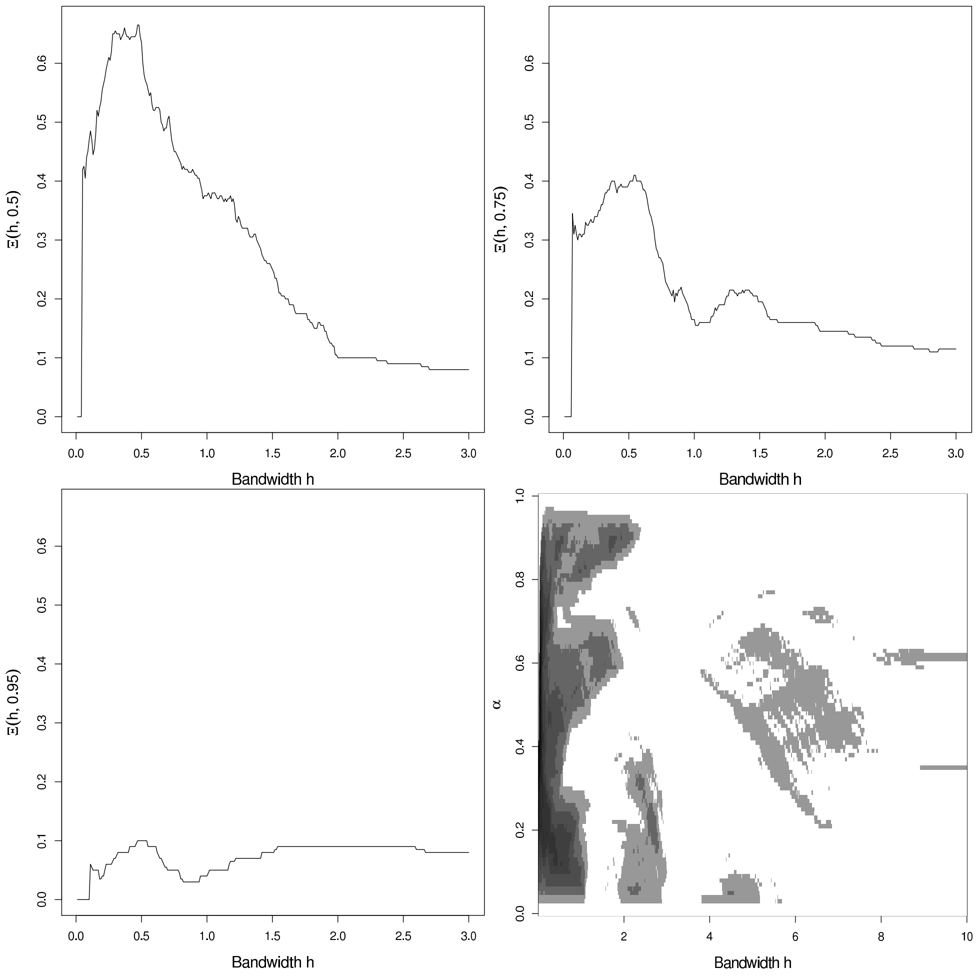


Figure 12: $\Xi_{\alpha,n}(h)$ as a function of h for $\alpha = 0.50$ (top left) 0.75 (top right) and 0.95 (bottom left). Heat Map of $\Xi_{\alpha,n}(h)$ as function of h and α; for readability, values of $\Xi_{\alpha,n}(h)$ smaller than 0.045 are displayed in white (bottom right).

Appendix A. Proofs

Proof of Theorem 2: Let $\mathcal{A}_{h_n,\varepsilon_n}$ denote the event that $\|\widehat{p}_{h_n,X} - p_{h_n}\|_\infty \leq \varepsilon_n$. Then, for all $n \geq n_0$, by Equation (3), $\mathbb{P}_X(\mathcal{A}_{h_n,\varepsilon_n}) \geq 1 - \frac{1}{n}$. Also observe that Assumption (A1) implies that, for any $h > 0$, the sup-norm density approximation error can be bounded as

$$\begin{aligned} \|p_h - p\|_\infty &= \sup_x \left| \int \frac{1}{h^d} K\left(\frac{x-y}{h}\right) p(y)dy - p(x) \right| \\ &\leq \sup_x \int \frac{1}{h^d} K\left(\frac{x-y}{h}\right) A\|x-y\|dy \\ &= ADh. \end{aligned} \tag{11}$$

The second step in the previous display follows since $\int K(z)dz = 1$ and using the Lipschitz assumption (A1) on the density, and the last step since $\int \|z\|K(z)dz = D$. Putting the estimation and approximation error together, and using the triangle inequality, we obtain that, on the event $\mathcal{A}_{h_n,\varepsilon_n}$,

$$\|\widehat{p}_{h_n,X} - p\|_\infty \leq ADh_n + \varepsilon_n, \tag{12}$$

for all $n \geq n_0$. Using Equation (12), we have that, on $\mathcal{A}_{h_n,\varepsilon_n}$ and for all $n \geq n_1(n_0,\lambda)$ so that $ADh_n + \varepsilon_n < \lambda$, the set

$$L(\lambda)\Delta\widehat{L}_{h_n,X}(\lambda) = \{u\colon\, p(u) > \lambda, \widehat{p}_{h_n,X}(u) \leq \lambda\} \cup \{u\colon\, p(u) \leq \lambda, \widehat{p}_{h_n,X}(u) > \lambda\}$$

is contained in

$$\{u\colon\, p(u) > \lambda, p(u) \leq \lambda + ADh_n + \varepsilon_n\} \cup \{u\colon\, p(u) \leq \lambda, p(u) > \lambda - ADh_n - \varepsilon_n\},$$

which is equal to

$$\{u\colon\, |p(u) - \lambda| < ADh_n + \varepsilon_n\}.$$

Then, on $\mathcal{A}_{h_n,\varepsilon_n}$ and for all $n \geq n_1(n_0,\lambda)$ large enough

$$\mathcal{L}(h_n,X,\lambda) = P(L(\lambda)\Delta\widehat{L}_{h_n,X}(\lambda)) \leq r_{h_n,\varepsilon_n,\lambda},$$

so that, $\mathbb{P}_X\left(\mathcal{L}(h_n,X,\lambda) \leq r_n\right) \geq \mathbb{P}_X\left(\mathcal{A}_{h_n,\varepsilon_n}\right) \geq 1 - \frac{1}{n}$, as claimed.

If (A2) is in force for the density level λ, then for all $n \geq n_2(n_0,\lambda,A,D,\varepsilon_0)$ so that $ADh_n + \varepsilon_n \leq \varepsilon_0$, we have $r_{h_n,\varepsilon_n,\lambda} \leq \kappa_2(ADh_n + \varepsilon_n)$, which proves the second claim.

Proof of Lemma 4: Using (A1) and the fact that $\int_{\mathbb{R}^d} K(z)dz = 1$, Equation (11) states that for any $h > 0$

$$\|p_h - p\|_\infty \leq ADh.$$

Then, for any $\alpha \in (0,1)$ and $h > 0$,

$$\{u\colon\, p(u) > \lambda_{h,\alpha} + ADh\} \subseteq \{u\colon\, p_h(u) > \lambda_{h,\alpha}\} \subseteq \{u\colon\, p(u) > \lambda_{h,\alpha} - ADh\}.$$

And as a result,

$$P(\{u\colon\, p(u) > \lambda_{h,\alpha} + ADh\}) \leq P(\{u\colon\, p_h(u) > \lambda_{h,\alpha}\}) \leq P(\{u\colon\, p(u) > \lambda_{h,\alpha} - ADh\}).$$

Since $P(\{u\colon\, p(u) > \lambda_\alpha\}) = \alpha = P(\{u\colon\, p_h(u) > \lambda_{h,\alpha}\})$, we have

$$P(\{u\colon\, p(u) > \lambda_{h,\alpha} + ADh\}) \leq P(\{u\colon\, p(u) > \lambda_\alpha\}) \leq P(\{u\colon\, p(u) > \lambda_{h,\alpha} - ADh\}).$$

Consequently,

$$\lambda_{h,\alpha} + ADh \geq \lambda_\alpha \geq \lambda_{h,\alpha} - ADh.$$

It follows that for any $\alpha \in (0,1)$ and $h > 0$

$$|\lambda_{h,\alpha} - \lambda_\alpha| \leq ADh.$$

Proof of Lemma 5: Let $\mathcal{C}_h = \left\{\{u\colon\, p_h(u) > \lambda\}, \lambda > 0\right\}$ denote the class of level sets of p_h and define the events

$$\mathcal{P}_{h,\varepsilon} = \left\{\sup_{C \in \mathcal{C}_h} |\widehat{P}_X(C) - P(C)| \leq \varepsilon\right\} \quad \text{and} \quad \mathcal{A}_{h,\varepsilon} = \{||\widehat{p}_{h,X} - p_h||_\infty \leq \varepsilon\}.$$

Then, since the n-th shatter coefficients (see, for instance, Devroye et al., 1996) of $\mathcal{C}_h$ is n,

$$\mathbb{P}_X(\mathcal{P}^c_{h,\varepsilon}) \leq 8ne^{-n\varepsilon^2/32} \quad \text{and} \quad \mathbb{P}_X(\mathcal{A}^c_{h,\varepsilon}) \leq K_1 e^{-K_2 n\varepsilon^2 h^d}, \tag{13}$$

where the first inequality follows from the VC inequality (see, for instance, Devroye et al., 1996) and the second inequality is just (1). Then, on $\mathcal{A}_{h,\varepsilon}$, we obtain

$$\{u\colon\, p_h(u) > \lambda + \varepsilon\} \subseteq \{u\colon\, \widehat{p}_{h,X}(u) > \lambda\} \subseteq \{u\colon\, p_h(u) > \lambda - \varepsilon\}, \quad \forall \lambda > 0.$$

Thus, on $\mathcal{A}_{h,\varepsilon}$,

$$\widehat{P}_X(\{u\colon\, p_h(u) > \lambda + \varepsilon\}) \leq \widehat{P}_X(\{u\colon\, \widehat{p}_{h,X}(u) > \lambda\}) \leq \widehat{P}_X(\{u\colon\, p_h(u) > \lambda - \varepsilon\}),$$

uniformly over all $\lambda > 0$. In particular, the previous inequality hold also for $\widehat{\lambda}_{\alpha,h,X}$ (which is positive with probability one) for any $\alpha \in (0,1)$ and $h > 0$.

Recalling that, by definition,

$$|\widehat{P}_X(\{u\colon\, \widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}\}) - \alpha| \leq 1/n,$$

we obtain, on the events $\mathcal{P}_{h,\varepsilon}$ and $\mathcal{A}_{h,\varepsilon}$,

$$P(\{u\colon\, p_h(u) > \widehat{\lambda}_{h,\alpha,X} + \varepsilon\}) - \frac{1}{n} - \varepsilon \leq \alpha \leq P\{u\colon\, p_h(u) > \widehat{\lambda}_{h,\alpha,X} - \varepsilon\}) + \frac{1}{n} + \varepsilon. \tag{14}$$

Since $\alpha = P(\{u\colon\, p_h(u) > \lambda_{h,\alpha}\})$, the first inequality in (14) can be written as

$$\alpha + \frac{1}{n} + \varepsilon = P(\{u\colon\, p_h(u) > \lambda_{h,\alpha+\frac{1}{n}+\varepsilon}\}) \geq P(\{u\colon\, p_h(u) > \widehat{\lambda}_{h,\alpha,X} + \varepsilon\})$$

and the second one as

$$\alpha - \frac{1}{n} - \varepsilon = P(\{u\colon\, p_h(u) > \lambda_{h,\alpha-\frac{1}{n}-\varepsilon}\}) \leq P\{u\colon\, p_h(u) > \widehat{\lambda}_{h,\alpha,X} - \varepsilon\}),$$

both holding on the events $\mathcal{P}_{h,\varepsilon}$ and $\mathcal{A}_{h,\varepsilon}$. Combining the last two expressions, we obtain, on the same events, for any $\alpha \in (0,1)$ and $h > 0$,

$$\lambda_{h,\alpha+\frac{1}{n}+\varepsilon} - \varepsilon \leq \widehat{\lambda}_{h,\alpha,X} \leq \lambda_{h,\alpha-\frac{1}{n}-\varepsilon} + \varepsilon. \tag{15}$$

We will now show that, for level sets of p_h indexed by α satisfying (B3), and for any $\eta \in (-\eta_0, \eta_0)$ and $0 < h \leq H$,

$$|\lambda_{h,\alpha+\eta} - \lambda_{h,\alpha}| \leq A\kappa_3|\eta|. \tag{16}$$

Recalling that $\varepsilon + 1/n < \eta_0$, Equations (15) and (16) will then imply

$$\lambda_{h,\alpha} - A\kappa_3\left(\varepsilon + \frac{1}{n}\right) - \varepsilon \leq \widehat{\lambda}_{h,\alpha,X} \leq \lambda_{h,\alpha} + A\kappa_3\left(\varepsilon + \frac{1}{n}\right) + \varepsilon,$$

on the events $\mathcal{P}_{h,\varepsilon}$ and $\mathcal{A}_{h,\varepsilon}$, for level sets of p_h indexed by α satisfying (B3) and with $0 < h \leq H$. Finally, using (13), the claim will follow.

In order to show (16), for a set $A \subset \mathbb{R}^d$, let ∂A denote its boundary. Then, notice that, because p_h is Lipschitz and hence continuous, for every $x \in \partial M_h(\alpha)$, $p_h(x) = \lambda_{h,\alpha}$ and, for every $y \in \partial M_h(\alpha+\eta)$, $p_h(y) = \lambda_{h,\alpha+\eta}$. Furthermore, for any point $x \in \partial M_h(\alpha)$, there exists a point $y = y(x) = \inf_{z\in\partial M_h(\alpha+\eta)} \|x-z\|$. Thus, for $|\eta| < \eta_0$,

$$\|x-y\| \leq d_\infty(M_h(\alpha), M_h(\alpha+\eta)) \leq \kappa_3|\eta|,$$

where the last inequality follows for level sets of p_h indexed by α that satisfy (B3) and $0 < h \leq H$. Therefore,

$$|\lambda_{h,\alpha+\eta} - \lambda_{h,\alpha}| = |p_h(y) - p_h(x)| \leq A\|x-y\| \leq A\kappa_3|\eta|,$$

where in the first inequality we used the fact that, by (A1), p_h is Lipschitz with constant A. Indeed, for any $x \neq y$, using the Lipschitz assumption (A1) on p,

$$|p_h(x) - p_h(y)| \leq \int_{\mathbb{R}^d} |p(x+zh) - p(y+zh)| K(z)dz \leq A\|x-y\| \int_{\mathbb{R}^d} K(z)dz = A\|x-y\|.$$

Proof of Theorem 7: Let $\mathcal{A}_{h_n,\varepsilon_n}$ be event defined in the proof of Theorem 2, and recall that for all $n \geq n_0$, by Equation (3), $\mathbb{P}_X(\mathcal{A}^c_{h_n,\varepsilon_n}) \leq 1/n$ and that, Equation (12) states that

$$\|\widehat{p}_{h,X} - p\|_\infty \leq C_{1,n} \tag{17}$$

on that event, for all $n \geq n_0$. Also, let $\mathcal{P}_{h_n,\varepsilon_n}$ be the event defined in Lemma 5 such that $\mathbb{P}_X(\mathcal{P}^c_{h_n,\varepsilon_n}) \leq 8ne^{-n\varepsilon_n^2/32}$. Then from the proof of Lemma 5, we have that on the event $\mathcal{A}_{h_n,\varepsilon_n} \cap \mathcal{P}_{h_n,\varepsilon_n}$, for $h_n = \omega((\log n/n)^{1/d})$ and $h_n \leq H$,

$$|\widehat{\lambda}_{h_n,\alpha,X} - \lambda_\alpha| \leq C_{2,n} \tag{18}$$

for all $n \geq n_3(n_0, \eta_0, K_3)$. Also, since n is large enough, we have

$$8ne^{-n\varepsilon_n^2/32} \leq \frac{1}{n}.$$

Therefore, for all such large n, both (17) and (18) hold with probability at least

$$\mathbb{P}_X\left(\mathcal{A}_{h_n,\varepsilon_n} \cap \mathcal{P}_{h_n,\varepsilon_n}\right) \geq 1 - \frac{2}{n}.$$

Thus, on $\mathcal{A}_{h_n,\varepsilon_n} \cap \mathcal{P}_{h_n,\varepsilon_n}$, for $h_n = \omega((\log n/n)^{1/d})$ and $h_n \leq H$, we have that, for all $n \geq n_3(n_0, \eta_0, K_3)$, the set

$$M(\alpha)\Delta\widehat{M}_{h,X}(\alpha) = \{u\colon\, p(u) > \lambda_\alpha, \widehat{p}_{h,X}(u) \leq \widehat{\lambda}_{h,\alpha,X}\} \cup \{u\colon\, p(u) \leq \lambda_\alpha, \widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}\}.$$

is contained in

$$\{u\colon\, p(u) > \lambda_\alpha, p(u) \leq \widehat{\lambda}_{h,\alpha,X} + C_{1,n}\} \cup \{u\colon\, p(u) \leq \lambda_\alpha, p(u) > \widehat{\lambda}_{h,\alpha,X} - C_{1,n}\}.$$

which, in turn, is a subset of

$$\{u\colon\, p(u) > \lambda_\alpha, p(u) \leq \lambda_\alpha + C_{1,n} + C_{2,n}\} \cup \{u\colon\, p(u) \leq \lambda_\alpha, p(u) > \lambda_\alpha - C_{1,n} - C_{2,n}\}.$$

The final set is just $\{u\colon |p(u)-\lambda_\alpha| \le C_{1,n}+C_{2,n}\}$. Therefore, for for $h_n = \omega((\log n/n)^{1/d})$ and $h_n \le H$, we have, for all $n \ge n_3(n_0,\eta_0,K_3)$,

$$\mathbb{P}_X\left(\mathcal{L}^*(h_n,X,\alpha) \le r_{h_n,\varepsilon_n,\alpha}\right) \ge \mathbb{P}_X\left(\mathcal{A}_{h_n,\varepsilon_n} \cap \mathcal{P}_{h_n,\varepsilon_n}\right) \ge 1-\frac{2}{n}.$$

Proof of Lemma 9: We only prove the second claim, since the proof of the limits is straightforward. For simplicity, we will provide the proof for the case of a spherical kernel: $K(x) = 1_{\|x\|\le 1}$, $x \in \mathbb{R}^d$. The extension to other compactly supported kernels is analogous.

Let h be strictly smaller than

$$\min\left\{\min_{i\neq j}||X_i-X_j||, \min_{i\neq j}||Y_i-Y_j||, \min_{i,j}||X_i-Y_j||\right\}.$$

For many distributions, this occurs almost surely for $h = O\left(1/n^d\right)$ (see, e.g., Penrose, 2003; Deheuvels et al., 1988). By the compactness of the support of K, for any such h, the sets

$$B(X_1,h),\ldots,B(X_n,h),B(Y_1,h),\ldots,B(Y_n,h)$$

are disjoint. Therefore, $\widehat{p}_{h,X}(u) = 1/(nh^d)$ if and only if $u \in B(X_i,h)$ for one i and, similarly, $\widehat{p}_{h,Y}(u) = 1/(nh^d)$ if and only if $u \in B(Y_j,h)$ for one j. Furthermore,

$$\widehat{L}_{h,X}\Delta\widehat{L}_{h,Y} = \left(\bigcup_i B(X_i,h)\right)\bigcup\left(\bigcup_j B(Y_j,h)\right).$$

As a result, $\Xi_{\lambda,n}(h)$ is the fraction of Z_i's contained in $(\cup_i B(X_i,h))\bigcup(\cup_i B(Y_i,h))$. Thus,

$$\Xi_{\lambda,n}(h) = \widehat{P}_Z(\widehat{L}_{h,X}\Delta\widehat{L}_{h,Y}|X,Y) \overset{d}{=} B/n,$$

where $\overset{d}{=}$ denotes equality in distribution and $B \sim \text{Binomial}(n,p_0)$, with $0 \le p_0 \le 2n\ p_{\max} v_d h^d$ and $p_{\max} = \|p\|_\infty$. Therefore, $\mathbb{E}_Z[\Xi_{\lambda,n}(h)|X,Y] \le 2p_{\max} v_d n h^d$ and hence it follows that

$$\xi_{\lambda,n}(h) = \mathbb{E}_{X,Y,Z}[\Xi_{\lambda,n}(h)] \le 2p_{\max} v_d n h^d = O(h^d),$$

as $h \to 0$.

Proof of Theorem 10:

1. Since X, Y and Z are independent samples from the same distribution, $\widehat{p}_{h,X}(u)$ and $\widehat{p}_{h,Y}(u)$ are independent and identically distributed, for any $u \in \mathbb{R}^d$ and $h > 0$. Also, notice that for every measurable set A, $\mathbb{E}_Z(\widehat{P}_Z(A)) = P(A)$. Thus,

$$\begin{aligned}
\xi_{\lambda,n}(h) &= \mathbb{E}_{X,Y,Z}[\widehat{P}_Z(\{u\colon \widehat{p}_{h,X}(u) > \lambda\}\Delta\{u\colon \widehat{p}_{h,Y}(u) > \lambda\})] \\
&= \mathbb{E}_{X,Y}[P(\{u\colon \widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \le \lambda\}) + P(\{u\colon \widehat{p}_{h,X}(u) \le \lambda, \widehat{p}_{h,Y}(u) > \lambda\})] \\
&= 2\mathbb{E}_{X,Y}\left[P(\{u\colon \widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \le \lambda\})\right] \\
&= 2\int_{\mathbb{R}^d} \mathbb{P}_{X,Y}\left(\widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \le \lambda\right) dP(u), \qquad (19)
\end{aligned}$$

where the last identity follows from Fubini theorem. The integrand in the last equation can be written as

$$\begin{aligned}\mathbb{P}_{X,Y}\left(\widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \leq \lambda\right) &= \mathbb{P}_X\left(\widehat{p}_{h,X}(u) > \lambda\right)\mathbb{P}_Y\left(\widehat{p}_{h,Y}(u) \leq \lambda\right)\\ &= \mathbb{P}_X\left(\widehat{p}_{h,X}(u) > \lambda\right)\mathbb{P}_X\left(\widehat{p}_{h,X}(u) \leq \lambda\right)\\ &= \pi_h(u)(1-\pi_h(u)),\end{aligned}$$

from which (8) follows.

2. Let $\mathcal{A}_{h,\varepsilon}$ denote the event

$$\|p_h - \widehat{p}_{h,X}\|_\infty \vee \|p_h - \widehat{p}_{h,Y}\|_\infty \leq \varepsilon. \tag{20}$$

By (1), $\mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\varepsilon}) \leq 2K_1 e^{-K_2 n h^d \varepsilon^2}$. Letting $1_{\mathcal{A}_{h,\varepsilon}}$ denote the indicator function of the event $\mathcal{A}_{h,\varepsilon}$,

$$\xi_{\lambda,n}(h) \leq \mathbb{E}_{X,Y,Z}[\widehat{P}_Z(\{u\colon \widehat{p}_{h,X}(u) > \lambda\}\Delta\{u\colon \widehat{p}_{h,Y}(u) > \lambda\})1_{\mathcal{A}_{h,\varepsilon}}(X,Y)] + \mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\varepsilon}),$$

and, using the same reasoning that led to (19),

$$\xi_{\lambda,n}(h) \leq 2\int_{\mathbb{R}^d} \mathbb{P}_{X,Y}\left(\{\widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \leq \lambda\} \cap \mathcal{A}_{h,\varepsilon}\right) dP(u) + \mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\varepsilon})$$

Notice that, on $\mathcal{A}_{h,\varepsilon}$,

$$\{u\colon \widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \leq \lambda\} \subseteq \{u\colon \lambda - \varepsilon \leq p_h(u) \leq \lambda + \varepsilon\} = U_{h,\varepsilon},$$

and therefore, $\text{sign}(\widehat{p}_{h,X}(u) - \lambda) = \text{sign}(p_h(u) - \lambda)$ for all $u \notin U_{h,\varepsilon}$. Thus, the previous expression for $\xi_{\lambda,n}(h)$ is upper bounded by

$$2\int_{U_{h,\varepsilon}} \mathbb{P}_{X,Y}\left(\{\widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \leq \lambda\} \cap \mathcal{A}_{h,\varepsilon}\right) dP(u) + 2K_1 e^{-K_2 n h^d \varepsilon^2}$$

which, using independence, is no larger than

$$2\int_{U_{h,\varepsilon}} \pi_h(u)(1-\pi_h(u))dP(u) + 2K_1 e^{-K_2 n h^d \varepsilon^2} \leq P(U_{h,\varepsilon})\overline{A}_{h,\varepsilon} + 2K_1 e^{-K_2 n h^d \varepsilon^2}.$$

As for the lower bound, from (19) we obtain, trivially,

$$\xi_{\lambda,n}(h) \geq 2\int_{U_{h,\varepsilon}} \pi_h(u)(1-\pi_h(u))dP(u) \geq P(U_{h,\varepsilon})\overline{A}_{h,\varepsilon}.$$

Proof of Lemma 11. If K is the spherical kernel, note that $\widehat{p}_{h,X}(u) = n^{-1}\sum_{i=1}^n B_i(u)$, where

$$B_i = h^{-d}K\left(\frac{u - X_i}{h}\right) = \frac{I_{B(u,h)}(X_i)}{(h^d v_d)},$$

with $I_{B(u,h)}(\cdot)$ denoting the indicator function of the ball $B(u,h)$. Let $\sigma^2(u,h) = \text{Var}(B_i(u))$ and $\mu_3(u,h) = \mathbb{E}|B_i(u) - \mu(u,h)|^3$ where $\mu(u,h) = \mathbb{E}(B_i(u)) = p_h(u)$. Finally, let $p_{u,h} = P(B(u,h))$. Then,

$$\sigma^2(u,h) = \frac{p_{u,h}(1-p_{u,h})}{(h^d v_d)^2} \tag{21}$$

and

$$\mu_3(u,h) = \frac{p_{u,h}(1-p_{u,h})\left[(1-p_{u,h})^2 + p_{u,h}^2\right]}{(h^d v_d)^3} \leq \frac{p_{u,h}(1-p_{u,h})}{(h^d v_d)^3},$$

where the last inequality holds since $(1-p_{u,h})^2 + p_{u,h}^2 \leq 1$, for all u and h. As a result,

$$\frac{\mu_3(u,h)}{\sigma^3(u,h)} \leq (p_{u,h}(1-p_{u,h}))^{-1/2}.$$

By assumption, $h < h(\delta,\varepsilon)$ and $\varepsilon \leq \lambda/2$. In order to avoid trivialities, we further assume that $P(U_{h,\varepsilon}) > 0$. Then, uniformly over all u in $U_{h,\varepsilon}$,

$$(\lambda-\varepsilon)v_d h^d \leq p_{u,h} \leq (\lambda+\varepsilon)v_d h^d$$

and

$$(1-p_{u,h}) \geq \delta.$$

Thus,

$$\frac{\mu_3(u,h)}{\sigma^3(u,h)} \leq \sqrt{\frac{1}{\delta v_d h^d(\lambda-\varepsilon)}} \leq \sqrt{\frac{2}{h^d \delta v_d \lambda}},$$

with the last inequality holding because of our assumption $\varepsilon \leq \lambda/2$. From (21), we then obtain

$$\frac{\delta(\lambda-\varepsilon)}{v_d h^d} \leq \sigma^2(u,h) \leq \frac{(\lambda+\varepsilon)}{v_d h^d}.$$

Thus,

$$\frac{a_1}{h^d} \leq \sigma^2(u,h) \leq \frac{a_2}{h^d},$$

where

$$a_1 = \frac{\delta\lambda}{2v_d} \quad \text{and} \quad a_2 = \frac{3\lambda}{2v_d}, \tag{22}$$

uniformly over $u \in U_{h,\varepsilon}$.

Writing $\sigma^2(u,h) = a(u,h)/h^d$ and using the Berry-Esséen bound (Wasserman, 2004, p. 78), we obtain

$$\sup_t \left| P\left(\frac{\sqrt{nh^d}(\widehat{p}_{h,X}(u) - p_h(u))}{a(u,h)} \leq t\right) - \Phi(t)\right| \leq \frac{33}{4}\frac{\mu_3(u,h)}{\sigma^3(u,h)\sqrt{n}} = \sqrt{\frac{C(\delta,\lambda)}{nh^d}},$$

where Φ is the cumulative distribution function of the standard Normal distribution.

Now,

$$\pi_h(u) = \mathbb{P}_X(\widehat{p}_{h,X}(u) > \lambda) = \mathbb{P}_X\left(\frac{\sqrt{nh^d}(\widehat{p}_{h,X}(u) - p_h(u))}{a(u,h)} > \frac{\sqrt{nh^d}(\lambda - p_h(u))}{a(u,h)}\right).$$

Hence,

$$1 - \Phi\left(\frac{\sqrt{nh^d}(\lambda - p_h(u))}{a(u,h)}\right) - \frac{C(\delta,\lambda)}{\sqrt{nh^d}} \leq \pi_h(u) \leq 1 - \Phi\left(\frac{\sqrt{nh^d}(\lambda - p_h(u))}{a(u,h)}\right) + \frac{C(\delta,\lambda)}{\sqrt{nh^d}}.$$

Using the fact that $u \in U_{h,\varepsilon}$, and taking advantage of the uniform bounds $a_1 \leq a(u,h) \leq a_2$, the previous inequalities imply

$$1-\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)-\frac{C(\delta,\lambda)}{\sqrt{nh^d}} \leq \pi_h(u) \leq 1-\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)+\frac{C(\delta,\lambda)}{\sqrt{nh^d}}.$$

Using the inequalities

$$1-\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)=\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)\geq\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)$$

and

$$1-\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)=\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)\leq\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right),$$

we obtain the bounds

$$\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)-\frac{C(\delta,\lambda)}{\sqrt{nh^d}} \leq \pi_h(u) \leq 1-\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)+\frac{C}{\sqrt{nh^d}} \tag{23}$$

and

$$1-\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)-\frac{C(\delta,\lambda)}{\sqrt{nh^d}} \leq \pi_h(u) \leq \Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)+\frac{C}{\sqrt{nh^d}}, \tag{24}$$

respectively. Thus, uniformly over all $\varepsilon \leq \lambda/2$ and all $h < h(\delta,\varepsilon)$, Equations (23) and (24) yield

$$\overline{A}_{h,\varepsilon}=2\sup_{u\in U_{h,\varepsilon}}\pi_h(u)(1-\pi_h(u)) \quad\leq\quad 2\left(1-\Phi\left(-\frac{\sqrt{nh^d}\varepsilon}{a_2}\right)+\frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

and

$$\underline{A}_{h,\varepsilon}=2\inf_{u\in U_{h,\varepsilon}}\pi_h(u)(1-\pi_h(u)) \quad\geq\quad 2\left(1-\Phi\left(\frac{\sqrt{nh^d}\varepsilon}{a_1}\right)-\frac{C(\delta,\lambda)}{\sqrt{nh^d}}\right)^2,$$

respectively, where a_1 and a_2 are given in (22).

Proof of Lemma 12. Letting $1_i = 1_{\{Z_i\in\widehat{L}_{X,h}\Delta\widehat{L}_{Y,h}\}}$, we have

$$\Xi_{\lambda,n}(h)=\frac{1}{n}\sum_{i=1}^{n}1_i.$$

where, conditionally on X and Y, the 1_i's are independent and identically distributed Bernoulli random variables with $\mathbb{E}_Z[1_i|X,Y]=P(\widehat{L}_{h,X}\Delta\widehat{L}_{h,Y})$. Thus

$$\begin{array}{rcl}
\mathbb{V}\left[\Xi_{\lambda,n}(h)\right] & = & \mathbb{E}_{X,Y,Z}\left[\Xi_n^2(h)\right]-\xi^2(h)\\
& = & \frac{1}{n^2}\mathbb{E}_{XY}[\mathbb{E}_Z\left[(\sum_{i=1}^n 1_i+\sum_{j\neq k}1_j1_k)|X,Y\right]]-\xi^2(h)\\
& = & \frac{\xi_{\lambda,n}(h)}{n}+\frac{n-1}{2n}\mathbb{E}_{X,Y}\left[P^2(\widehat{L}_{h,X}\Delta\widehat{L}_{h,Y})\right]-\xi^2(h)\\
& \leq & \frac{\xi_{\lambda,n}(h)}{n}+\frac{n-1}{2n}\mathbb{E}_{X,Y}\left[P(\widehat{L}_{h,X}\Delta\widehat{L}_{h,Y})\right]-\xi^2(h)\\
& = & \frac{\xi_{\lambda,n}(h)}{n}+\frac{n-1}{2n}\xi_{\lambda,n}(h)-\xi^2(h)\\
& = & \xi_{\lambda,n}(h)\left(\frac{n+1}{2n}-\xi_{\lambda,n}(h)\right).
\end{array}$$

Proof of Lemma 13.

Let $\xi(h,X,Y) = \mathbb{E}_Z[\Xi_{\lambda,n}(h)|X,Y]$ and let $A_{h,\varepsilon}$ be the event given in (20), where $\varepsilon, h > 0$, so that $\mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\varepsilon}) \leq 2K_1 \exp\left\{-nK_2 h^d \varepsilon^2\right\}$ by (1). Then, we can write

$$\mathbb{P}_{X,Y,Z}\left(\left|\Xi_{\lambda,n}(h) - \xi_{\lambda,n}(h)\right| > t\right) = \mathbb{P}_{X,Y,Z}\left(\left|\Xi_{\lambda,n}(h) - \xi(h,X,Y) + \xi(h,X,Y) - \xi_{\lambda,n}(h)\right| > t\right),$$

which is therefore upper bounded by

$$\mathbb{P}_{X,Y,Z}\left(\left|\Xi_{\lambda,n}(h) - \xi(h,X,Y) + \xi(h,X,Y) - \xi_{\lambda,n}(h)\right| > t; \mathcal{A}_{h,\varepsilon}\right) + 2K_1 \exp\left\{-nK_2 h^d \varepsilon^2\right\}.$$

The first term in the previous expression is no larger than the sum of

$$\mathbb{E}_{X,Y}\left[\mathbb{P}_Z\left(\left|\Xi_{\lambda,n}(h) - \xi(h,X,Y)\right| > t\eta \middle| X,Y\right); \mathcal{A}_{h,\varepsilon}\right], \tag{25}$$

and

$$\mathbb{P}_{X,Y}\left(\left|\xi(h,X,Y) - \xi_{\lambda,n}(h)\right| > t(1-\eta); \mathcal{A}_{h,\varepsilon}\right), \tag{26}$$

for any $\eta \in (0,1)$. We will first show that, if (9) is satisfied, the probability (26) is zero. Indeed, first observe that

$$\mathbb{E}_Z[\Xi_{\lambda,n}(h)|X,Y] = P(\widehat{L}_{h,X} \Delta \widehat{L}_{h,Y})$$

and that, on $\mathcal{A}_{h,\varepsilon}$,

$$\begin{aligned}
\widehat{L}_{h,X} \Delta \widehat{L}_{h,Y} &= \{u\colon \widehat{p}_{h,X}(u) > \lambda, \widehat{p}_{h,Y}(u) \leq \lambda\} \cup \{u\colon \widehat{p}_{h,X}(u) \leq \lambda, \widehat{p}_{h,Y}(u) > \lambda\} \\
&\subseteq \{u\colon p_h(u) > \lambda - \varepsilon, p_h(u) \leq \lambda + \varepsilon\} \\
&= \{u\colon |p_h(u) - \lambda| \leq \varepsilon\} \\
&= U_{h,\varepsilon},
\end{aligned}$$

Therefore, on $\mathcal{A}_{h,\varepsilon}$,

$$\xi(h,X,Y) = \mathbb{E}_Z[\Xi_{\lambda,n}(h)|X,Y] \leq r_{h,\varepsilon} \leq t(1-\eta). \tag{27}$$

By part 2 of Theorem 10, (9) further implies that $t(1-\eta) \geq \xi_{\lambda,n}(h)$. As a result, on $\mathcal{A}_{h,\varepsilon}$, $\left|\xi(h,X,Y) - \xi_{\lambda,n}(h)\right| \leq t(1-\eta)$, which yields

$$\mathbb{P}_{X,Y}\left(\left|\xi(h,X,Y) - \xi_{\lambda,n}(h)\right| > t(1-\eta); \mathcal{A}_{h,\varepsilon}\right) = 0,$$

as claimed.

We now proceed to bound from above (25). Since

$$\Xi_{\lambda,n}(h) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{Z_i \in \widehat{L}_{h,X} \Delta \widehat{L}_{h,Y}\}},$$

Bernstein's inequality (see, for instance, Massart, 2006, Proposition 2.9) yields that, for any $t > 0$ and conditionally on X and Y,

$$\mathbb{P}_Z\left(\left|\Xi_{\lambda,n}(h) - \xi(h,X,Y)\right| > t\eta \middle| X,Y\right) \leq \exp\left\{-9\sigma^2(X,Y,h) g\left(\frac{nt\eta}{3\sigma^2(X,Y,h)}\right)\right\} \tag{28}$$

where $g(u) = 1 + u - \sqrt{1+2u}$ for all $u > 0$, and

$$\sigma^2(X,Y,h) = \mathrm{Var}_Z[\Xi_{\lambda,n}(h)|X,Y].$$

It is easy to see that

$$\sigma^2(X,Y,h) \leq \mathbb{E}_Z\left[\Xi_{\lambda,n}(h)|X,Y\right] = n\xi(h,X,Y)$$

and, therefore, restricting to the event $\mathcal{A}_{h,\varepsilon}$, $\sigma^2(X,Y,h) \leq nt(1-\eta)$, just like in (27).

Using the fact that $e^{-9xg\left(\frac{nt}{3x}\right)}$ is increasing in x for $x > 0$, we conclude that, on the event $\mathcal{A}_{h,\varepsilon}$, the right hand side of (28) is bounded from above by

$$\exp\left\{-9nt(1-\eta)g\left(\frac{\eta}{3(1-\eta)}\right)\right\},$$

which is independent of X and Y. Thus, the previous expression is an upper bound for (25) and, therefore, for $\mathbb{P}_{X,Y,Z}\left(\left|\Xi_{\lambda,n}(h) - \xi_{\lambda,n}(h)\right| > t\right)$. The claim now follows from simple algebra.

Proof of Theorem 14.

1. The proof is almost the same as the proof of part 1 of Theorem 10 and is therefore omitted.

2. Let $\mathcal{A}_{h,\widetilde{\varepsilon}}$ denote the event

$$\max\left\{||\widehat{p}_{h,X} - p_h||_\infty, |\lambda_{h,\alpha} - \widehat{\lambda}_{h,\alpha,X}|, ||\widehat{p}_{h,Y} - p_h||_\infty, |\lambda_{h,\alpha} - \widehat{\lambda}_{h,\alpha,Y}|\right\} \leq \widetilde{\varepsilon}, \tag{29}$$

where $\widetilde{\varepsilon} = \varepsilon(A\kappa_3 + 1) + A\kappa_3/n$. Then, using (1), (5) and the fact that $\varepsilon < \widetilde{\varepsilon}$, the union bound yields

$$\mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\widetilde{\varepsilon}}) \leq 4K_1 e^{-K_2 nh^d \varepsilon^2} + 16ne^{-n\varepsilon^2/32} \equiv C(h,\varepsilon,n) \tag{30}$$

Now, on $\mathcal{A}_{h,\widetilde{\varepsilon}}$, $\{u : \widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}, \widehat{p}_{h,Y}(u) \leq \widehat{\lambda}_{h,\alpha,Y}\}$ is a subset of

$$\{u : p_h(u) > \widehat{\lambda}_{h,\alpha,X} - \widetilde{\varepsilon}, p_h(u) \leq \widehat{\lambda}_{h,\alpha,Y} + \widetilde{\varepsilon}\},$$

which is equal to

$$\{u : |p_h(u) - \lambda_{h,\alpha}| \leq 2\widetilde{\varepsilon}\} = U_{h,\widetilde{\varepsilon},\alpha}.$$

Therefore, $\mathrm{sign}(\widehat{p}_{h,X}(u) - \widehat{\lambda}_{h,\alpha,X}) = \mathrm{sign}(p_h(u) - \lambda_{h,\alpha})$ for all $u \notin U_{h,2\widetilde{\varepsilon},\alpha}$. Next, just like in the proof of part 2 of theorem 10, using this fact and the result of the first part we have that $\xi_{\alpha,n}(h)$ is no larger than

$$\mathbb{E}_{X,Y,Z}[\widehat{P}_Z(\{u\colon \widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}\}\Delta\{u\colon \widehat{p}_{h,Y}(u) > \widehat{\lambda}_{h,\alpha,Y}\})1_{\mathcal{A}_{h,\widetilde{\varepsilon}}}(X,Y)] + \mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\widetilde{\varepsilon}}).$$

The previous expression can be written as

$$2\int_{\mathbb{R}^d} \mathbb{P}_{X,Y}(\{\widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}, \widehat{p}_{h,Y}(u) \leq \widehat{\lambda}_{h,\alpha,Y}\} \cap \mathcal{A}_{h,\widetilde{\varepsilon}})dP(u) + \mathbb{P}_{X,Y}(\mathcal{A}^c_{h,\widetilde{\varepsilon}}),$$

which is less than

$$2\int_{U_{h,2\widetilde{\varepsilon},\alpha}} \mathbb{P}_{X,Y}(\{\widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}, \widehat{p}_{h,Y}(u) \leq \widehat{\lambda}_{h,\alpha,Y}\} \cap \mathcal{A}_{h,\widetilde{\varepsilon}})dP(u) + C(h,\varepsilon,n).$$

This quantity is bounded from above by

$$2\int_{U_{h,2\tilde{\varepsilon},\alpha}} \mathbb{P}_{X,Y}(\widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}, \widehat{p}_{h,Y}(u) \le \widehat{\lambda}_{h,\alpha,Y})dP(u) + C(h,\varepsilon,n),$$

which is finally smaller than

$$2\int_{U_{h,2\tilde{\varepsilon},\alpha}} \pi_{h,\alpha}(u)(1-\pi_{h,\alpha}(u))dP(u) + C(h,\varepsilon,n) \le P(U_{h,2\tilde{\varepsilon},\alpha})\overline{A}_{h,\varepsilon,\alpha} + C(h,\varepsilon,n).$$

As for the lower bound, from the result of first part we obtain, trivially,

$$\begin{array}{rcl} \xi_{\alpha,n}(h) & \ge & 2\int_{U_{h,2\tilde{\varepsilon},\alpha}} \pi_{h,\alpha}(u)(1-\pi_{h,\alpha}(u))dP(u) \\ & \ge & P(U_{h,2\tilde{\varepsilon},\alpha})\overline{A}_{h,\varepsilon,\alpha}. \end{array}$$

3. To compute an upper bound for $\overline{A}_{h,\varepsilon,\alpha}$ and a lower bound for $\underline{A}_{h,\varepsilon,\alpha}$, we use the Berry-Esséen bound and the stated assumptions. The proof is very similar to the proof of lemma 11, except that the result holds only on the event $\mathcal{A}_{h,\tilde{\varepsilon}}$. Therefore, we only provide a sketch of the arguments.

   The assumptions that $\tilde{\varepsilon} \le \inf_h \frac{\lambda_{\alpha,h}}{4}$, implies that, for any $u \in U_{h,2\tilde{\varepsilon},\alpha}$,

$$\frac{1}{h^d}\frac{\delta\lambda_{\alpha,h}}{2v_d} \le \frac{\delta(\lambda_{\alpha,h} - 2\tilde{\varepsilon})}{h^d v_d} \le \sigma^2(u,h) \le \frac{(\lambda_{\alpha,h} + 2\tilde{\varepsilon})}{h^d v_d} \le \frac{1}{h^d}\frac{3\lambda_{\alpha,h}}{2v_d}.$$

   Because of this and the fact that, on $\mathcal{A}_{h,\tilde{\varepsilon}}$, $|p_h(u) - \widehat{\lambda}_{h,\alpha,X}| \le 3\tilde{\varepsilon}$ for all $u \in U_{h,2\tilde{\varepsilon},\alpha}$, the same Berry-Esseen arguments used in the proof of lemma 11 yield

$$1 - \Phi\left(\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_1}\right) - \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} \le \pi_{h,\alpha,\tilde{\varepsilon}}(u) \le 1 - \Phi\left(-\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_2}\right) + \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}}.$$

   where $\pi_{h,\alpha,\tilde{\varepsilon}}(u) = \mathbb{P}_X\left(\{\widehat{p}_{h,X}(u) > \widehat{\lambda}_{h,\alpha,X}\} \cap \mathcal{A}_{h,\tilde{\varepsilon}}\right)$, $a_1 = \delta\lambda_{h,\alpha}/(2v_d)$, $a_2 = 3\lambda_{h,\alpha}/(2v_d)$, and $C(\delta,\lambda_{h,\alpha}) = \frac{33}{4}\sqrt{\frac{2}{\delta v_d \lambda_{h,\alpha}}}$. Now notice that

$$\pi_{h,\alpha}(u) \ge \pi_{h,\alpha,\tilde{\varepsilon}}(u) \ge 1 - \Phi\left(\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_1}\right) - \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}}$$

   and

$$\pi_{h,\alpha}(u) \le \pi_{h,\alpha,\tilde{\varepsilon}}(u) + P(\mathcal{A}^c_{h,\tilde{\varepsilon}}) \le 1 - \Phi\left(-\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_2}\right) + \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} + C(h,\varepsilon,n).$$

   where $C(h,\varepsilon,n)$ is defined in (30). Therefore,

$$\overline{A}_{h,\varepsilon,\alpha} \le 2\left(1 - \Phi\left(-\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_2}\right) + \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} + C(h,\varepsilon,n)\right)^2,$$

   and

$$\underline{A}_{h,\varepsilon,\alpha} \ge 2\left(1 - \Phi\left(\frac{3\tilde{\varepsilon}\sqrt{nh^d}}{a_1}\right) - \frac{C(\delta,\lambda_{h,\alpha})}{\sqrt{nh^d}} - C(h,\varepsilon,n)\right)^2.$$

Proof of Theorem 19. (1) Since the sample space is compact, $\mu(S) < \infty$, where S denotes the support of P and μ denotes the Lebesgue measure. Therefore, we obtain the inequality

$$\begin{aligned}\Gamma_n(h) &\leq \frac{\mu(S)}{2}||\widehat{p}_{h,X} - \widehat{p}_{h,Y}||_\infty \leq \frac{\mu(S)}{2}||\widehat{p}_{h,X} - p_h||_\infty + \frac{\mu(S)}{2}||\widehat{p}_{h,Y} - p_h||_\infty \\ &\overset{d}{=} \mu(S)||\widehat{p}_{h,X} - p_h||_\infty.\end{aligned}$$

Next, let $C = \frac{(\mu(S))^2(a+2)}{K_2}$, so that for $n > K_1$

$$t_h > \sqrt{\frac{\mu(S)^2 \log(n^{a+1}K_1)}{K_2 n h^d}}.$$

Then,

$$\begin{aligned}\mathbb{P}_{X,Y}\left(\Gamma_n(h) > t_h \text{ for some } h \in \mathcal{H}_n\right) &\leq \mathbb{P}_X\left(||\widehat{p}_{h,X} - p_h||_\infty > \frac{t_h}{\mu(S)} \text{ for some } h \in \mathcal{H}_n\right) \\ &\leq \sum_{h \in \mathcal{H}_n} \mathbb{P}_X\left(||\widehat{p}_{h,X} - p_h||_\infty > \frac{t_h}{\mu(S)}\right) \\ &\leq \sum_{h \in \mathcal{H}_n} K_1 \exp\{-K_2 n t_h^2 h^d/(\mu(S)^2)\} \\ &\leq H n^a \frac{1}{n^{a+1}} = \frac{H}{n} \\ &\leq \delta,\end{aligned}$$

where the third inequality stems from (1) and the assumption that $n \geq n_0$ is large enough, and the last inequality follows from the assumed condition on δ.
(2) Consider any $h \leq h_*$. Note that

$$\Gamma_n(h) \geq \Gamma_{n,S}(h) \equiv \frac{1}{2}\int_S |\widehat{p}_{h,X}(u) - \widehat{p}_{h,Y}(u)|du.$$

Let

$$D(u) = \sqrt{nh^d}(\widehat{p}_{h,X}(u) - \widehat{p}_{h,Y}(u)).$$

The variance of $D(u)$ is

$$\begin{aligned}\mathrm{Var}\left(\sqrt{nh^d}(\widehat{p}_{h,X}(u) - \widehat{p}_{h,Y}(u))\right) &= nh^d\left(\mathrm{Var}(\widehat{p}_{h,X}(u)) + \mathrm{Var}(\widehat{p}_{h,Y}(u))\right) \\ &= 2nh^d\mathrm{Var}(\widehat{p}_{h,X}(u)) \\ &= 2nh^d\mathrm{Var}\left(\frac{1}{nh^d v_d}\sum_{i=1}^n I(||X_i - u|| \leq h)\right) \\ &= \frac{2n^2h^d}{n^2h^{2d}v_d^2}\mathrm{Var}(I(||X_i - u|| \leq h)) \\ &= \frac{2}{v_d^2 h^d}P(B(u,h))(1 - P(B(u,h))).\end{aligned}$$

Now, for $u \in S$, by (10),

$$P(B(u,h))(1-P(B(u,h))) \leq P(B(u,h)) \leq a_2 h^d v_d$$

and

$$P(B(u,h))(1-P(B(u,h))) \geq P(B(u,h))\delta \geq a_1 h^d v_d \delta.$$

Hence,

$$2a_1 v_d \delta \leq \mathrm{Var}(D(u)) \leq 2a_2 v_d, \quad \forall u \in S,$$

which shows that the variance of $D(u)$ is bounded above and below by positive functions that do not depend on h. By a similar calculation, $\mathrm{Cov}(D(u),D(v))$ is bounded above and below by functions that do not depend on h, for all $u,v \in S$.

Now, for any u,

$$D(u) = D_1(u) - D_2(u) \equiv \sqrt{nh^d}(P_n - P)(f_u) - \sqrt{nh^d}(Q_n - P)(f_u)$$

where P_n is the empirical measure based on $X_1,\ldots,X_n$, Q_n is the empirical measure based on $Y_1,\ldots,Y_n$, and $f_u(\cdot) = h^{-d}K(||u-\cdot||/h)$. Note that D_1 and D_2 are independent, mean 0 stochastic processes. We can regard $\{\sqrt{nh^d}(P_n - P)(f) : f \in \mathcal{F}\}$ as an empirical process, where $\mathcal{F} = \{f_u : u \in S\}$ and similarly for $\{\sqrt{nh^d}(Q_n - P)(f) : f \in \mathcal{F}\}$. For fixed h, the collection $\mathcal{F}$ is a Donsker class. Hence, for every $u \in S$, $D_1(u)$ and $D_2(u)$ converge to two independent mean 0 Gaussian processes. By the continuous mapping theorem, for every $u \in S$, $D(u)$ converges to a mean 0 Gaussian process $\mathbb{G}$ with some covariance kernel κ. By the calculations above, there exist positive bounded functions $r(u,v) \leq s(u,v)$ such that $r(u,v) \leq \kappa(u,v) \leq s(u,v)$ and such that neither r nor s depend on h. Hence

$$\begin{aligned}
\mathbb{P}_{X,Y}\left(\Gamma_n(h) \geq t\sqrt{\frac{1}{nh^d}}\right) &\geq \mathbb{P}_{X,Y}\left(\Gamma_{n,S}(h) \geq t\sqrt{\frac{1}{nh^d}}\right) = \mathbb{P}_{X,Y}\left(\sqrt{nh^d}\Gamma_{n,S}(h) \geq t\right) \\
&= \mathbb{P}_{X,Y}\left(\frac{1}{2}\int_S |D(u)|du \geq t\right) \\
&= \mathbb{P}\left(\frac{1}{2}\int |\mathbb{G}(u)|du \geq t\right) + o(1),
\end{aligned}$$

where the last probability is the law of the Gaussian process $\mathbb{G}$. The $o(1)$ term is less than $\delta/2$ when $n \geq n_0$. Since $\mathbb{G}$ has strictly positive variance, $\mathbb{P}(\int |\mathbb{G}| \geq 0) = 1$. Clearly, $\mathbb{P}(\int |\mathbb{G}| \geq 2t)$ is decreasing in t. Hence, for each δ, there is a positive t such that $\mathbb{P}\left(\frac{1}{2}\int |\mathbb{G}| \geq t\right) \geq 1 - \delta/2$.
(3) The proof of this part is straightforward and is omitted.

References

S. Ben-David, U. von Luxburg, and D. Pál. A sober look at clustering stability. In *COLT*, pages 5–19, 2006.

A. Ben-Hur, A. Elisseef, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

B. Cadre, B. Pelletier, and P. Pudlo. Clustering by estimation of density level sets at a fixed probability, 2009. Manuscript available at `http://w3.bretagne.ens-cachan.fr/math/people/benoit.cadre/fichiers/tlevel.pdf`.

G. Carlsson and F. Memoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.

K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In *Neural Information Processing Systems (NIPS)*, December 2010.

P. Chaudhuri and S.J. Marron. Scale space view of curve estimation. *Annals of Statistics*, 28(2): 408–428, 2000.

A. Cuevas and A. Rodríguez-Casal. On boundary estimation. *Advances in Applied Probability*, 36: 340–354, 2004.

A. Cuevas, W. González-Manteiga, and A. Rodríguez-Casal. Plug-in estimation of general level sets. *Australian and New Zealand Journal of Statistics*, 48(1):7–19, 2006.

P. Deheuvels, J. Einmahl, D. Mason, and F. F. Ruymgaart. The almost sure behavior of maximal and minimal multivariate kn-spacings. *Journal of Multivariate Analysis*, 24:155–176, 1988.

L. Devroye, , L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

B. Fischer and J. M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1411–1415, 2003.

E. Giné and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'institut Henri Poincaré (B),* Probabilités et Statistiques, 38:907–921, 2002.

J. Hartigan. *Clustering Algorithms*. Wiley, 1975.

C. Jackson. Displaying uncertainty with shading. *The American Statistician*, 62(4):340–347, 2008.

S. Kpotufe and U. von Luxburg. Pruning nearest neighbor cluster trees. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. Omnipress, 2011.

T. Lange, V. Roth, M. Braun, and J.Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16:1299–1323, 2004.

E. Mammen and A. B. Tsybakov. Smooth discrimination analysis. *Annals of Statistics*, 27(6): 1808–1829, 1999.

P. Massart. *Concentration Inequalities and Model Selection*. Number 1896 in Springer Lecture Notes in Mathematics. Springer, 2006.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72:417–473, 2010.

M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.

W. Polonik. Measuring mass concentration and estimating density contour clusters–an excess mass approach. *Annals of Statistics*, 32(3):855–881, 1995.

P. Rigollet and R. Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15: 1154–1178, 2009.

A. Rinaldo and L. Wasserman. Generalized density clustering. *The Annals of Statistics*, 38(5): 2678–2722, 2010.

D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley, 1992.

A. Singh, C. Scott, and R. Nowak. Adaptive hausdorff estimation of level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.

I. Steinwart. Adaptive density level set clustering. In *Proceedings of the Twenty-Fourth Annual Conference on Learning Theory (COLT'11)*. Omnipress, 2011.

W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19:1–22, 2009.

A. B. Tsybakov. On nonparametric estimation of density level sets. *Annals of Statistics*, 25(3): 948–969, 1997.

A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 32 (1):135–166, 2004.

U. von Luxburg. Clustering stability: An overview. *Foundations and Trends in Machine Learning*, 2:235–274, 2009.

M. P. Wand. Fast computation of multivariate kernel estimators. *Journal of Computational and Graphical Statistics*, 3(4):433–445, 1994.

L. Wasserman. *All of Statistics*. Springer, New York, N.Y., 2004.

Journal of Machine Learning Research 13 (2012) 949-979 Submitted 8/11; Revised 11/11; Published 4/12

Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features

Gil Tahan GILTA@BGU.AC.IL
Lior Rokach LIORRK@BGU.AC.IL
Yuval Shahar YSHAHAR@BGU.AC.IL
Department of Information Systems Engineering
Ben-Gurion University of the Negev
P.O.B. 653
Beer-Sheva, Israel 84105

Editor: Charles Elkan

Abstract

This paper proposes several novel methods, based on machine learning, to detect malware in executable files without any need for preprocessing, such as unpacking or disassembling. The basic method (Mal-ID) is a new static (form-based) analysis methodology that uses common segment analysis in order to detect malware files. By using common segment analysis, Mal-ID is able to discard malware parts that originate from benign code. In addition, Mal-ID uses a new kind of feature, termed meta-feature, to better capture the properties of the analyzed segments. Rather than using the entire file, as is usually the case with machine learning based techniques, the new approach detects malware on the segment level. This study also introduces two Mal-ID extensions that improve the Mal-ID basic method in various aspects. We rigorously evaluated Mal-ID and its two extensions with more than ten performance measures, and compared them to the highly rated boosted decision tree method under identical settings. The evaluation demonstrated that Mal-ID and the two Mal-ID extensions outperformed the boosted decision tree method in almost all respects. In addition, the results indicated that by extracting meaningful features, it is sufficient to employ one simple detection rule for classifying executable files.

Keywords: computer security, malware detection, common segment analysis, supervised learning

1. Introduction

Nowadays the use of the Internet has become an integral part of modern life and Internet browsers are downloading to users a wide variety of content, including new computer software. One consequence of this widespread use is that many computer systems are vulnerable to and infected with malware—malicious software. Malware can be categorized into several groups:

1. Viruses—computer programs that are able to replicate themselves and infect files including the operating systems (OS);

2. Worms—self-replicating computer software that is able to send itself to other computers on a network or the Internet;

3. Trojans—a software that appears to perform the desired functionally but is actually implementing other hidden operations such as facilitating unauthorized access to a computer system;

4. Spyware—a software installed on a computer system without the user's knowledge to collect information about the user.

The rate of malware attacks and infections is not yet leveling. In fact, according to O'Farrell (2011) and Symantec Global Internet Security Threat Report Trends for 2010 (Symantec, 2010), attacks against Web browsers and malicious code variants installed by means of these attacks have increased.

There are many ways to mitigate malware infection and spread. Tools such as anti-virus and anti-spyware are able to identify and block or identify malware based on its behavior (Franc and Sonnenburg, 2009) or static features (see Table 1 below). A static feature may be a rule or a signature that uniquely identifies a malware or malware group. While the tools mitigating malware may vary, at their core there must be some classification method to distinguish malware files from benign files.

Warrender et al. (1999) laid the groundwork for using machine learning for intrusions detection. In particular, machine learning methods have been used to analyze binary executables. For example, Wartell el al. (2011) introduce a machine learning-based disassembly algorithm that segments binaries into subsequences of bytes and then classifies each subsequence as code or data. In this paper, the term segment refers to a sequence of bytes of certain size that was extracted from an executable file. While it sequentially scans an executable, it sets a breaking point at each potential code-to-code and code-to-data/data-to-code transition. In addition, in recent years many researchers have been using machine learning (ML) techniques to produce a binary classifier that is able to distinguish malware from benign files.

The techniques use three distinct stages:

1. Feature Extraction for file representation—The result of the feature extraction phase is a vector containing the features extracted. An executable content is reduced or transformed into a more manageable form such as:

   (a) Strings—a file is scanned sequentially and all plain-text data is selected.

   (b) Portable Executable File Format Fields—information embedded in Win32 and Win64-bit executables. The information is necessary for the Windows OS loader and application itself. Features extracted from PE executables may include all or part of the following pieces of information: attribute certificate—similar to checksum but more difficult to forge; date/time stamp; file pointer—a position within the file as stored on disk; linker information; CPU type; Portable Executable (PE) logical structure (including section alignment, code size, debug flags); characteristics—flags that indicate attributes of the image file; DLL import section—list of DLLs and functions the executable uses; export section—which functions can be imported by other applications; resource directory—indexed by a multiple-level binary-sorted tree structure (resources may include all kinds of information. For example, strings for dialogs, images, dialog structures; version information, build information, original filename, etc.); relocation table; and many other features.

   (c) n-gram—segments of consecutive bytes from different locations within the executables of length *n*. Each n-gram extracted is considered a feature (Rokach et al., 2008).

   (d) Opcode n-gram—Opcode is a CPU specific operational code that performs specific machine instruction. Opcode n-gram refers to the concatenation of Opcodes into segments.

2. Feature Selection (or feature reduction)—During this phase the vector created in phase 1 is evaluated and redundant and irrelevant features are discarded. Feature selection has many benefits including: improving the performance of learning modules by reducing the number of computations and as a result the learning speed; enhancing generalization capability; improving the interpretability of a model, etc. Feature selection can be done using a wrapper approach or a correlation-based filter approach (Mitchell, 1997). Typically, the filter approach is faster than the wrapper approach and is used when many features exist. The filter approach uses a measure to quantify the correlation of each feature, or a combination of features, to a class. The overall expected contribution to the classification is calculated and selection is done according to the highest value. The feature selection measure can be calculated using many techniques, such as gain ratio (GR); information-gain (IG); Fisher score ranking technique (Golub et al., 1999) and hierarchical feature selection (Henchiri and Japkowicz, 2006).

3. The last phase is creating a classifier using the reduced features vector created in phase 2 and a classification technique. Among the many classification techniques, most of which have been implemented in the Weka platform (Witten and Frank, 2005), the following have been used in the context of benign/malware files classification: artificial neural networks (ANNs) (Bishop, 1995) , decision tree (DT) learners (Quinlan, 1993), nave-Bayes (NB) classifiers (John and Langley, 1995), Bayesian networks (BN) (Pearl, 1987), support vector machines (SVMs) (Joachims, 1999), k-nearest neighbor (KNN) (Aha et al., 1991), voting feature intervals (VFI) (Demiröz and Güvenir, 1997), OneR classifier (Holte, 1993), Adaboost (Freund and Schapire, 1999), random forest (Breiman, 2001), and other ensemble methods (Menahem et al., 2009; Rokach, 2010).

To test the effectiveness of ML techniques, in malware detection, the researchers listed in Table 1 conducted experiments combining various feature extraction methods along with several feature selection and classification algorithms.

Ye et al. (2009) suggested using a mixture of features in the malware-detection process. The features are called Interpretable Strings as they include both programs' strings and strings representing the API execution calls used. The assumption is that the strings capture important semantics and can reflect an attacker's intent and goal. The detection process starts with a feature parser that extract the API function calls and looks for a sequence of consecutive bytes that forms the strings used. Strings must be of the same encoding and character set. The feature-parser uses a corpus of natural language to filter and remove non-interpretable strings. Next, the strings are ranked using the Max-Relevance algorithm. Finally, a classification model is constructed from SVM ensemble with bagging.

Ye et al. (2010) presented a variation of the method, presented above, that uses Hierarchical Associative Classifier (HAC) to detect malware from a large imbalanced list of applications. The malware in the imbalanced list were the minority class. The HAC methodology also uses API calls as features. Again, the associative classifiers were chosen due to their interpretability and their capability to discover interesting relationships among API calls. The HAC uses two stages:

to achieve high recall, in the first stage, high precision rules for benign programs (majority class) and low precision rules for minority class are used, then, in the second stage, the malware files are ranked and precision optimization is performed.

Instead of relying on unpacking methods that may fail, Dai et al. (2009) proposed a malware-detection system, based on a virtual machine, to reveal and capture the needed features. The system constructs classification models using common data mining approaches. First, both malware and benign programs are executed inside the virtual machine and the instruction sequences are collected during runtime. Second, the instruction sequence patterns are abstracted. Each sequence is treated as a feature. Next, a feature selection process in performed. In the last stage a classification model is built. In the evaluation the SVM model performed slightly better then the C4.5 model.

Yu et al. (2011) presented a simple method to detect malware variants. First, a histogram is created by iterating over the suspected file binary code. An additional histogram is created for the base sample (the known malware). Then, measures are calculated to estimate the similarity between the two histograms. Yu et al. (2011) showed that when the similarity is high, there is a high probability that the suspected file is a malware variant.

The experiments definitely proved that is possible to use ML techniques for malware detection. Short n-gram were most commonly used as features and yielded the best results. However, the researchers listed did not use the same file sets and test formats and therefore it is very difficult or impossible to compare the results and to determine what the best method under various conditions is. Table 2 presents predictive performance results from various researches.

When we examined the techniques, several insights emerged:

1. All applications (i.e., software files tested in the studies) that were developed using a higher level development platforms (such as Microsoft Visual Studio, Delphi, Microsoft.Net) contain common code and resources that originate from common code and resource libraries. Since most malware are also made of the same common building blocks, we believe it would be reasonable to discard the parts of a malware that are common to all kinds of software, leaving only the parts that are unique to the malware. Doing so should increase the difference between malware files and benign files and therefore should result in a lower misclassification rate.

2. Long n-gram create huge computational loads due to the number of features. This is known as the curse of dimensionality (Bellman et al., 1966). All surveyed n-gram experiments were conducted with n-gram length of up to 8 bytes (in most cases 3-byte n-gram) despite the fact that short n-gram cannot be unique by themselves. In many cases 3- to 8-byte n-gram cannot represent even one line of code composed with a high level language. In fact, we showed in a previous paper (Tahan et al., 2010) that an n-gram should be at least 64 bytes long to uniquely identify a malware. As a result, current techniques using short n-gram rely on complex conditions and involve many features for detecting malware files.

The goal of this paper is to develop and evaluate a novel methodology and supporting algorithms for detecting malware files by using common segment analysis. In the proposed methodology we initially detect and nullify, by zero patching, benign segments and therefore resolve the deficiency of analyzing files with segments that may not contribute or even hinder classification. Note that, when a segment represents at least one line of code developed using a high level language; it can address the second deficiency of using short features that may be meaningless when considered alone. Additionally, we suggest using meta-features instead of using plain features such as n-gram.

Study	Feature Representation	Feature Selection	Classifiers
Schultz et al. (2001)	PE, Strings, n-gram	NA	RIPPER, Nave Bayes, and Multi-Nave Bayes
Kolter and Maloof (2004)	n-gram	NA	TFIDF, Nave Bayes, SVM, Decision Trees, Boosted Decision Trees, Boosted Nave Bayes, and Boosted SVM
Abou-Assaleh et al. (2004)	n-gram	NA	K-Nearest Neighbors
Kolter and Maloof (2006)	n-gram	Information-Gain	K-Nearest Neighbors, Nave Bayes, SVM, Decision Trees, Boosted Decision Trees, Boosted Nave Bayes, and Boosted SVM.
Henchiri and Japkowicz (2006)	n-gram	Hierarchical feature selection	Decision Trees, Nave Bayes, and SVM
Zhang et al. (2007)	n-gram	Information-Gain	Probabilistic Neural Network
Elovici et al. (2007)	PE and n-gram	Fisher Score	Bayesian Networks, Artificial Neural Networks, and Decision Trees
Ye et al. (2008)	PE	Max-Relevance	Classification Based on Association (CBA)
Dai et al. (2009)	instruction sequence	Contrast measure	SVM
Ye et al. (2009)	PE (API)	Max-Relevance	SVM ensemble with bagging
Ye et al. (2010)	PE (API)	Max-Relevance	Hierarchical Associative Classifier (HAC)
Yu et al. (2011)	histogram	NA	Nearest Neighbors

Table 1: Recent research in static analysis malware detection in chronological order.

A meta-feature is a feature that captures the essence of plain feature in a more compact form. Using those meta-features, we are able to refer to relatively long sequences (64 bytes), thus avoiding the curse of dimensionality.

2. Methods

As explained in Section 1, our basic insight is that almost all modern computer applications are developed using higher level development platforms such as: Microsoft Visual Studio, Embarcadero Delphi, etc. There are a number of implications associated with using these development platforms:

Method	**Study**	**Features**	**Feature selection**	**FPR**	**TPR**	**Acc**	**AUC**
Artificial Neural Network	Elovici et al. (2007)	5grams	Fisher Score top 300	0.038	0.89	0.94	0.96
Bayesian Network	Elovici et al. (2007)	5grams	Fisher Score top 300	0.206	0.88	0.81	0.84
Bayesian Network	Elovici et al. (2007)	PE	n/a	0.058	0.93	0.94	0.96
Decision Tree	Elovici et al. (2007)	5grams	Fisher Score top 300	0.039	0.87	0.93	0.93
Decision Tree	Elovici et al. (2007)	PE	n/a	0.035	0.92	0.95	0.96
Classification Based on Association	Ye et al. (2008)	PE	Max-Relevance	0.125	0.97	0.93	——
Boosted Decision Tree	Kolter and Maloof (2006)	4grams	Gain Ratio	——	——	——	0.99

Table 2: Comparison of several kinds of machine learning methods. FPR, TPR, ACC and AUC refers to False Positive Rate, True Positive Rate, Accuracy and the Area Under Receiver Operating Characteristic (ROC) Curve as defined in Section 3.2.

1. Since application development is fast with these platforms, both legitimate developers and hackers tend to use them. This is certainly true for second-stage malware.

2. Applications share the same libraries and resources that originated from the development platform or from third-party software companies. As a result, malware that has been developed with these tools generally resembles benign applications. Malware also tends, to a certain degree, to use the same specialized libraries to achieve a malicious goal (such as attachment to a different process, hide from sight with root kits, etc). Therefore it may be reasonable to assume that there will be resemblances in various types of malware due to sharing common malware library code or even similar specific method to perform malicious action. Of course such malware commonalities cannot be always guaranteed.

3. The size of most application files that are being produced is relatively large. Since many modern malware files are in fact much larger than 1 MB, analysis of the newer applications is much more complex than previously when the applications themselves were smaller as well as the malware attacking them.

The main idea presented in this paper is to use a new static analysis methodology that uses common segment analysis in order to detect files containing malware. As noted above, many applications and malware are developed using the development platforms that include large program language libraries. The result is that large portions of executable code originate from the program language libraries. For example, a worm malware that distributes itself via email may contain a benign code for sending emails. Consequently, since the email handling code is not malicious and can be found in many legitimate applications, it might be a good idea to identify code portions that originate from a benign source and disregard them when classifying an executable file. In other words, when given an unclassified file, the first step would be to detect the file segments that originated from the development platform or from a benign third party library (termed here the Common Function) and then disregard those segments. Finally, the remaining segments would be compared to determine their degree of resemblance to a collection of known malwares. If the resemblance measure satisfies a predetermined threshold or rule then the file can be classified as malware.

To implement the suggested approach, two kinds of repositories are defined:

1. **CFL—Common Function Library.** The CFL contains data structures constructed from benign files.

2. **TFL—Threat Function Library**. The TFL contains data structures constructed from malware without segments identified as benign (i.e., segments that appears in benign files).

Figure 1 presents the different stages required to build the needed data structures and to classify an application file. As can be seen in this figure, our Mal-ID methodology uses two distinct stages to accomplish the malware detection task: setup and detection. The setup stage builds the CFL. The detection phase classifies a previously unseen application as either malware or benign. Each stage and each sub-stage is explained in detail in the following subsections. The Mal-ID pseudo code is presented in Figure 2.

2.1 The Setup Phase

The setup phase involves collecting two kinds of files: benign and malware files. The benign files can be gathered, for example, from installed programs, such as programs located under Windows XP program files folders. The malware files can, for example, be downloaded from trusted dedicated Internet sites, or by collaborating with an anti-virus company. In this study the malware collection was obtained from trusted sources. In particular, Ben-Gurion University Computational Center provided us malware that were detected by them over time. Each and every file from the collection is first broken into 3-grams (three consecutive bytes) and then an appropriate repository is constructed from the 3-grams. The CFL repository is constructed from benign files and the TFL repository is constructed from malware files. These repositories are later used to derive the meta-features—as described in Section 2.2.

Note that in the proposed algorithm, we are calculating the distribution of 3-grams within each file and across files, to make sure that a 3-gram belongs to the examined segment and thus associate the segment to either benign (CFL) or malware (TFL). Moreover, 3-grams that seem to appear approximately within the same offset in all malware can be used to characterize the malware. Before calculating the 3-grams, the training files are randomly divided into 64 groups.

The CFL and TFL repositories share the same data structure:

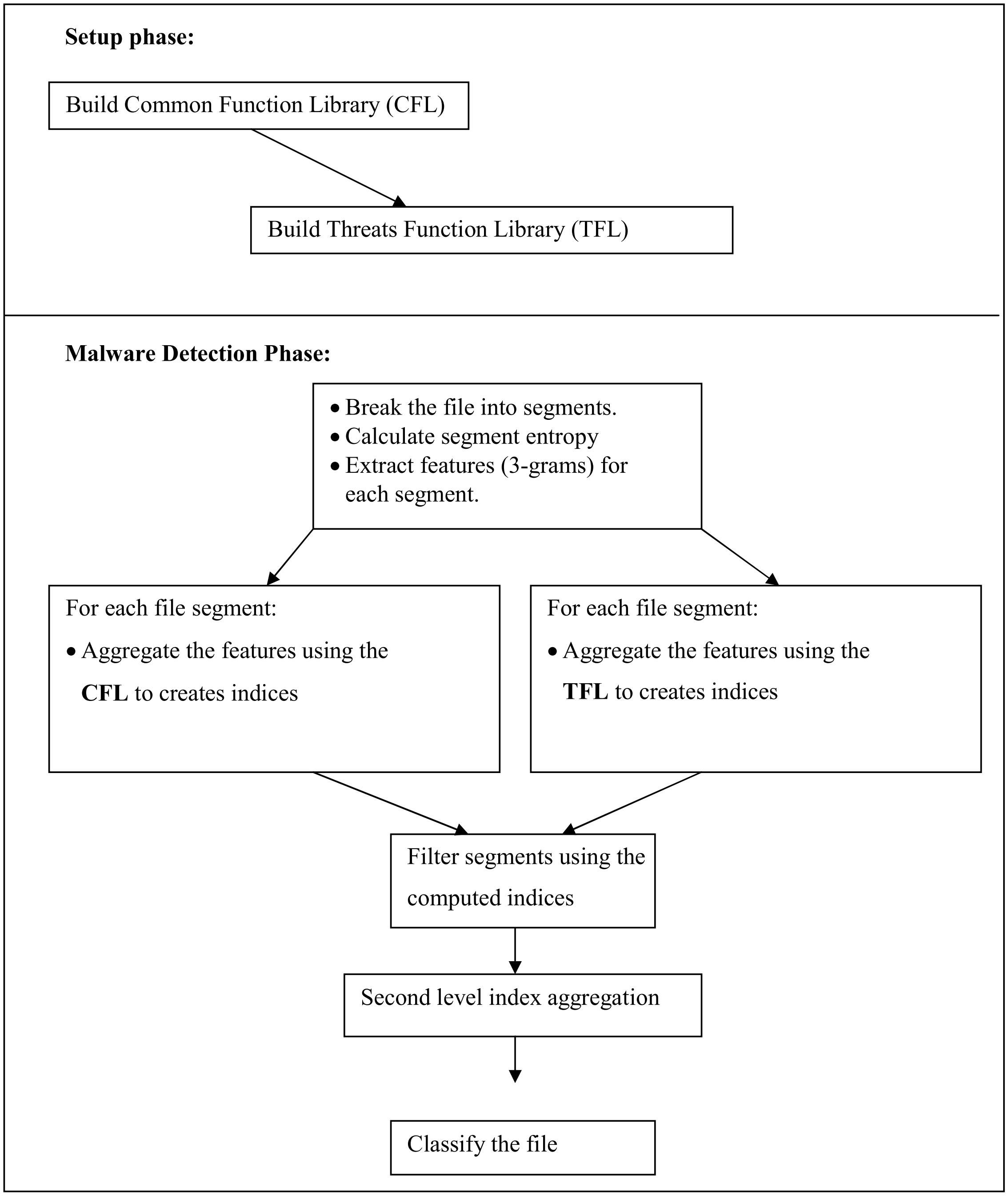

Setup phase:
Build Common Function Library (CFL)
Build Threats Function Library (TFL)
Malware Detection Phase:
• Break the file into segments.
• Calculate segment entropy
• Extract features (3-grams) for each segment.
For each file segment:
• Aggregate the features using the CFL to creates indices
For each file segment:
• Aggregate the features using the TFL to creates indices
Filter segments using the computed indices
Second level index aggregation
Classify the file

Figure 1: The Mal-ID method for detecting new malware applications.

1. 3-gram-files-association: 2^{24} entries, each of 64 bits. A bit value of 1 in a cell (i, j) indicates the appearance of a specific 3-gram i in the j^{th} group of files. The 64-bit entry size was selected since a previous study showed that this size is the most cost effective in terms of

detection performance vs. storage complexity (Tahan et al., 2010). Other implementations may use larger entries.

1. 3-gram-relative-position-within-file: 2^{24} entries, each of 64 bits. A bit value of 1 in a cell (i, j) indicates the appearance of 3-gram i in the j^{th} internal segment of a file (assuming the file is divided into 64 equal length segments).

The CFL is constructed first and then the TFL:

1. Each file from the malware collection is broken into segments. The Mal-ID implementation has used 64-byte segments.

2. Each segment is broken into 3grams and then tested against the CFL using the algorithm and features described next. Segments that are not in the CFL are added to the TFL.

It is important to note that the end result is the TFL, a repository made of segments found only in malware and not in benign files.

2.2 The Detection Phase

The Mal-ID basic is a feature extraction process followed by a simple static decision rule.

It operates by analyzing short segments extracted from the file examined. Each segment comprises a number of 3-grams depending on the length of the segment (e.g., a segment of length 4 bytes is comprised from two 3-grams that overlap by two bytes). Three features can be derived for each segment: Spread, MFG, and Entropy. The Spread and the MFG features are derived using the data structures prepared in the setup stage described in Section 2.1 above.

The definition and motivation behind the new features are hereby provided:

1. **Spread**: Recall that in the Mal-ID setup phase each file in the training set has been divided into 64 relative-position-areas. The Spread feature represents the spread of the signature's 3-grams along the various areas for all the files in a given repository. The Spread feature can be calculated as follows: for each 3-gram, first retrieve the 3-gram-relative-position-within-file bit-field, and then perform 'And' operations over all the bit-fields and count the resulting number of bits that are equal to 1. In other words, spread approximates the maximum number of occurrences of a segment within different relative locations in train sets. For example, a Spread equal to 1 means that the segment appears (at most) in one relative location in all the files.

2. **MFG:** the maximum total number of file-groups that contain the segment. The MFG is calculated using the 3-gram-files-association bit-field, in the same manner that spread is calculated.

3. **Entropy:** the entropy measure of the bytes within a specific segment candidate. In addition to the new estimators presented above, the entropy feature is also used to enable to identification of compressed areas (such as embedded JPEG images) and long repeating sequences that contain relatively little information.

Note that the features, as described above, are in fact **meta-features** as they are used to represent *features of features* (features of the basic 3-grams). As explained next, using these meta-features, Mal-ID can refer to relatively long sequences (64 bytes), thus avoiding the data mining problem known as "the curse of dimensionality", and other problems caused when using short n-gram as features. The advantages of using Mal-ID meta-features will be demonstrated in the evaluation results section and in the discussion section.

2.3 The Mal-ID Basic Detection Algorithm

The input for the Mal-ID method is an unclassified executable file of any size. Once the setup phase has constructed the CFL and the TFL, it is possible to classify a file F as benign or as malware using the algorithm presented in Figure 2.

1. Line 1. Divide file F into S segments of length L. All segments are inserted into a collection and any duplicated segments are removed. The end result is a collection of unique segments. The Mal-ID implementation uses 2000 segments that are 64-bytes in length.

2. Line 3. For each segment in the collection:

   (a) Line 5. Calculate the entropy for the bytes within the segment.

   (b) Line 6. The algorithm gets two parameters EntropyLow and EntropyHigh. The entropy thresholds are set to disregard compressed areas (such as embedded JPEG images) and long repeating sequences that contain relatively little information. In this line we check if the entropy is smaller than EntropyLow threshold or entropy is larger than EntropyHigh. Is so then discard the segment and continue segment iteration. Preliminary evaluation has found the values of EntropyLow=0.5 and EntropyHigh=0.675 maximize the number of irrelevant segments that can be removed.

   (c) Line 9. Extract all 3-grams using 1 byte shifts.

   (d) Line 11. Using the CFL, calculate the CFL-MFG index.

   (e) Line 12. If the CFL-MFG index is larger than zero, then discard the segment and continue segment iteration. The segment is disregarded since it may appear in benign files.

   (f) Line 14. Using the TFL, calculate the TFL-MFG index

   (g) Line 15. The algorithm gets the ThreatThreshold parameter which indicates the minimum occurrences a segment should appear in the TFL in order to be qualified as malware indicator. In this line we check if the TFL-MFG index is smaller or equal to the ThreatThreshold. If so then discard the segment and continue with segment iteration. In the Mal-ID implementation only segments that appear two times or more remain in the segment collection. Obviously a segment that does not appear in any malware cannot be used to indicate that the file is a malware.

   (h) Line 17. Using the TFL calculate the TFL-Spread index

   (i) Line 18. The algorithm gets the SR parameter which indicates the Spread Range required. If the TFL-Spread index equals zero or if it is larger than what we term SR threshold, then discard the segment and continue segment iteration. The purpose of these conditions is to make sure that all 3-grams are located in at least 1 segment in at least 1 specific relative location. If a segment is present in more than SR relative locations it is less likely to belong to a distinct library function and thus should be discarded. In our Mal-ID implementation, SR was set to 9.

(j) Lines 21-25 (optional stage, aimed to reduce false malware detection). A segment that meets all of the above conditions is tested against the malware file groups that contain all 3-gram segments. As a result, only segments that actually reside in the malware are left in the segment collection. Preliminary evaluation showed that there is no significant performance gain performing this stage more than log (SegmentLen) * NumberOfMalwareInTraining iterations.

3. Lines 28-30. Second level index aggregation—Count all segments that are found in malware and not in the CFL.

4. Line 32. Classify—If there are at least X segments found in the malware train set (TFL) and not in the CFL then the file is malware; otherwise consider the file as benign. We have implemented Mal-Id with X set to 1.

Please note that the features used by Mal-ID algorithm described above are in fact meta-features that describe the 3-grams features. The advantages of using Mal-ID meta-features will be described in the following sections.

2.3.1 MAL-ID COMPLEXITY

Proposition 1 *The computational complexity of the algorithm in Figure 2 is* $O(SN + log(SL) \cdot M \cdot MaxMalSize)$ *where SN denotes the number of segments; SL denotes segment length; M denotes the number of malware in the training set; and MaxMalSize denotes the maximum length of a malware.*

Proof The computational complexity of the algorithm in Figure 2 is computed as follows: the GenerateSegmentCollection complexity is $O(SN)$; the complexity of loop number 1 (lines 3-26) is $O(SN + log(SL) \cdot M \cdot MaxMalSize)$; the complexity of loop number 2 (lines 29-30) is $O(SN)$. Thus, the overall complexity is $O(SN + log(SL) \cdot M \cdot MaxMalSize)$. ■

2.4 Combining Mal-ID With ML Generated Models

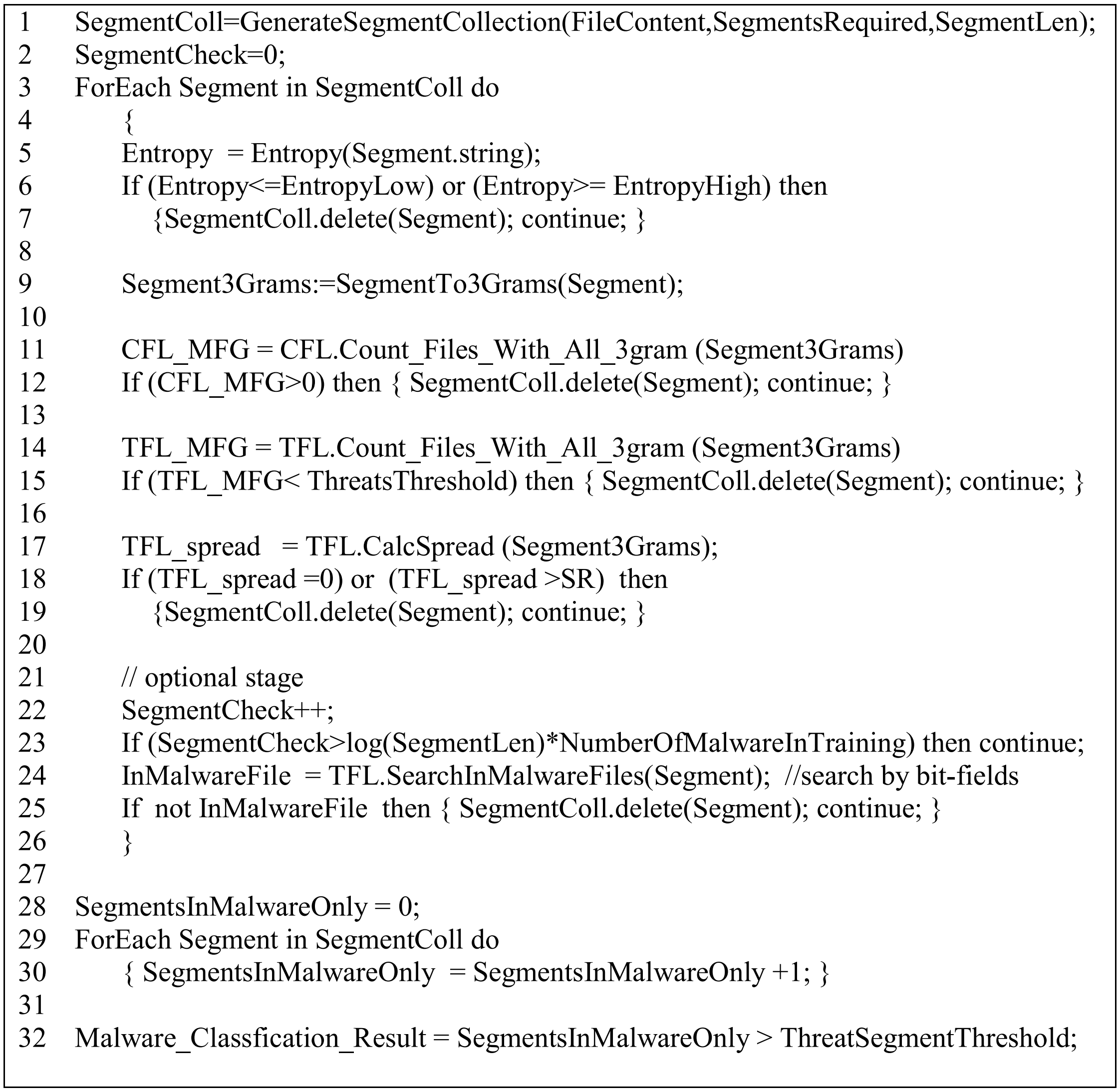
We attempted to improve the *Mal-ID basic* method by using Mal-ID features with various classifiers, but instead of using the Mal-ID decision model described in Section 2, we let various ML algorithms build the model using the following procedure:

1. We apply the common segment analysis method on the training set and obtain a collection of segments for both the CFL and the TFL as explained in Section 2.

2. For each file's segment, we calculated the CFL-MFG, TFL-MFG and the TFL-spread based on the CFL and TFL. The entropy measure is calculated as well.

3. We discretized the numeric domain of the above features using the supervised procedure of Fayyad and Irani (1993). Thus for each feature we found the most representative sub-domains (bins).

4. For each file we count the number of segments associated with each bin. Each frequency count is represented twice: once as absolute numbers (number of segments) and then as a proportional distribution.



```
SegmentColl=GenerateSegmentCollection(FileContent,SegmentsRequired,SegmentLen);
SegmentCheck=0;
ForEach Segment in SegmentColl do
    {
    Entropy  = Entropy(Segment.string);
    If (Entropy<=EntropyLow) or (Entropy>= EntropyHigh) then
        {SegmentColl.delete(Segment); continue; }

    Segment3Grams:=SegmentTo3Grams(Segment);

    CFL_MFG = CFL.Count_Files_With_All_3gram (Segment3Grams)
    If (CFL_MFG>0) then { SegmentColl.delete(Segment); continue; }

    TFL_MFG = TFL.Count_Files_With_All_3gram (Segment3Grams)
    If (TFL_MFG< ThreatsThreshold) then { SegmentColl.delete(Segment); continue; }

    TFL_spread   = TFL.CalcSpread (Segment3Grams);
    If (TFL_spread =0) or  (TFL_spread >SR)  then
        {SegmentColl.delete(Segment); continue; }

    // optional stage
    SegmentCheck++;
    If (SegmentCheck>log(SegmentLen)*NumberOfMalwareInTraining) then continue;
    InMalwareFile  = TFL.SearchInMalwareFiles(Segment);  //search by bit-fields
    If  not InMalwareFile  then { SegmentColl.delete(Segment); continue; }
    }

SegmentsInMalwareOnly = 0;
ForEach Segment in SegmentColl do
    { SegmentsInMalwareOnly  = SegmentsInMalwareOnly +1; }

Malware_Classfication_Result = SegmentsInMalwareOnly > ThreatSegmentThreshold;
```

Figure 2: Mal-ID pseudo code.

5. An induction algorithm is trained over the training set to generate a classifier.

We compare the following three machine learning induction algorithms:

1. C4.5—Single Decision Tree

2. RF—Rotation Forest (Rodriguez et al., 2006) using J48 decision tree as base classifier. The algorithm was executed with 100 iterations and the PCA method for projecting the data in every iteration.

3. NN—A multilayer perception with one hidden layer trained over 500 epochs using backpropagation.

Finally, using the model is used to detect the malware among the files in the test set.

2.5 Combining Mal-ID With ML Models Post Processing

We have attempted to improve the *Mal-ID basic* method by using the following procedure:

1. First, the *Mal-ID basic* method is used to construct the CFL and TFL. This stage is performed only once before the file classification starts.

2. Next, zero patch each malware in the training set as follows: Iterate over all of the file segments and perform common segment analysis to detect the segments that appear in the CFL. The benign segments (the segments that appear in the CFL) are zero patched in an attempt to reduce the number of n-gram that are clearly not relevant for detecting segments that appear only in malware. The end result is a new file with the same length that has zeros in the benign segments.

3. Finally, construct a classification model using Rotation Forest using J48 decision tree as base classifier. The patched malware collection and the unchanged benign file collection are used for training.

To classify a file we first have to zero-patch the file as explained above then use the classification model created earlier.

3. Experimental Evaluation

In order to evaluate the performance of the proposed methods for detecting malwares, a comparative experiment was conducted on benchmark data sets. The proposed methods were compared with the method presented in the research of Kolter and Maloof (2004). The research of Kolter and Maloof (2006) found that the combination of 500 4-grams with gain ratio feature selection and boosted decision tree provides the best performance over many other evaluated method variations. We will refer to our variation of Kolter and Maloof method as *GR500BDT* as it uses Gain Ratio feature selection, **500** 4-grams, and Boosted Decision Tree classifier. The *GR500BDT* method was specifically selected because it was the best method known to us.

The following terms will be used when referring to the various methods:

1. *GR500BDT*—Our baseline method, which is described above.

2. *Mal-IDP+GR500BDT*—As explained in Section 2.5, we use Mal-ID to zero patch common segments in the test files, and then use *GR500BDT* as usual.

3. *Mal-ID basic*—*Mal-ID basic* method as explained in Section 2.

4. *Mal-IDF*+<induction algorithm>—as detailed in Section 2.4, Mal-ID features will be used by induction algorithm.

    (a) *Mal-IDF+RF*—Mal-ID features with Rotation Forest classification
    (b) *Mal-IDF+C4.5*—Mal-ID features with C4.5
    (c) *Mal-IDF+NN*—Mal-ID features with a multilayer perception.

Specifically, the experimental study had the following goals:

1. To examine whether the proposed basic methods, could detect malware while keeping the false alarm rate as small as possible.
2. Compare the performance of the various *Mal-ID basic* extensions.
3. To analyze the effect of the common library size (benign and malware) on performance.

The following subsections describe the experimental set-up and the results that were obtained.

3.1 Experimental Process

The main aim of this process was to estimate the generalized detection performance (i.e., the probability that a malware was detected correctly). The files repository was randomly partitioned into training and test sets. The process was repeated 10 times and we report the average result. The same train-test partitioning was used for all algorithms.

For evaluating the proposed methodology 2627 benign files were gathered from programs installed under Windows XP program files folders, with lengths ranging from 1Kb to 24MB. An additional 849 malware files were gathered from the Internet with lengths ranging from 6Kb to 4.25MB (200 executables were above 300KB). The detailed list of examined executables can be obtained in the following URL: `http://www.ise.bgu.ac.il/faculty/liorr/List.rar`. The malware and benign file sets were used without any decryption, decompression or any other preprocessing. The malware types and frequencies are presented in Figure 3. The evaluation computer used an Intel Q6850 CPU with 4GB of RAM. The processing time was measured using only 1 CPU core, although the implemented algorithm natively supported multiple cores.

3.2 Evaluation Measures

We used the following performance measures:

- TP = true positive
- FP = false positive
- TN = true negative
- FN = false negative
- FPR = FP / N = FP / (FP + TN) = false positive rate
- TPR = TP / P = TP / (TP + FN) = true positive rate (also known as sensitivity)
- PPV = TP / (TP + FP) = positive predictive value
- NPV = TN / (TN + FN) = negative predictive value
- ACC = (TP + TN) / (P + N) = accuracy
- BER = 0.5(FN/P + FP/N) = balanced error rate

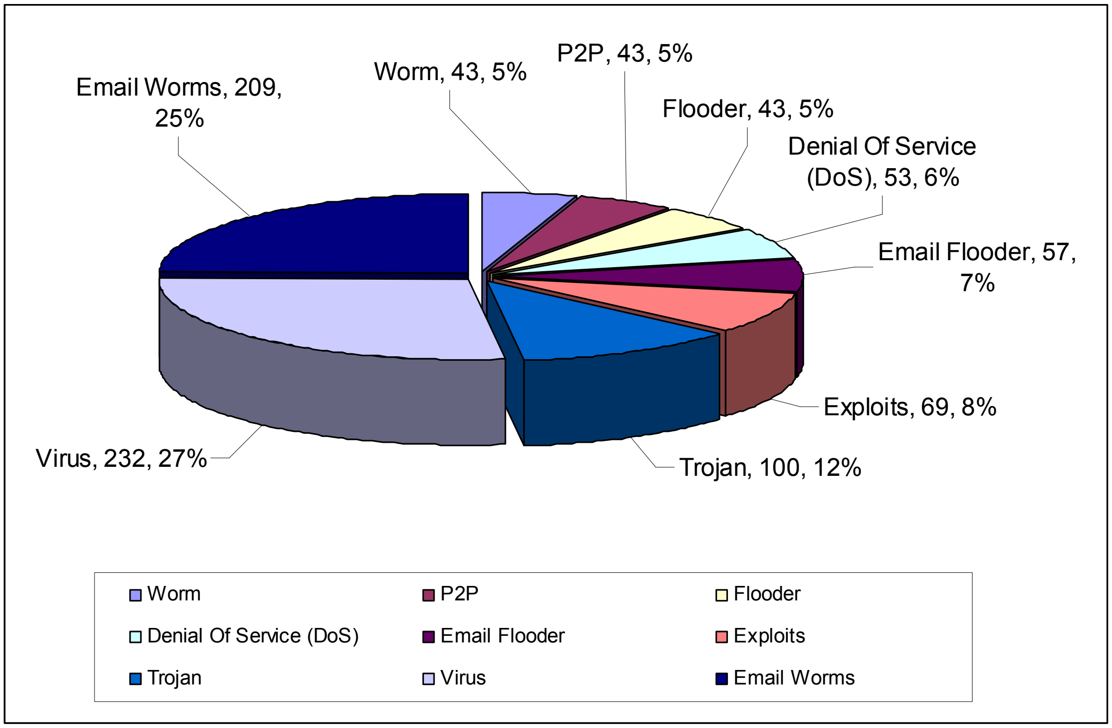


Figure 3: Distribution of malware types in data set.

- BCR = 1- BER = balanced correctness rate
- AUC = area under receiver operating characteristic (ROC) curve

Our measures, such as PPV versus NPV, as well as BER or BCR, try to address the important case of an unbalanced positive/negative instance case mix, which is often ignored in the literature. Given the low rate of malware versus benign code, accuracy might be a misleading measure. For example, a "*Maximal Class Probability*" (MPC) classifier is a classifier that always predicts the most frequent class. Thus, an MPC predicting "BENIGN" for every instance in an environment where 99% of the files are benign would, indeed, be 99% accurate. That would also be its NPV, since there is a 99% probability that the MPC is right when it predicts that the file is benign. However, its PPV would be 0, or rather, undefined, since it never predicts a positive class; in other words, its sensitivity to positive examples is 0.

Furthermore, unlike many studies in the information security literature, we use the cross-entropy as one of our major performance measures. The cross-entropy described by Caruana et al. (2004). It is also referred in the literature by the terms *negative log-likelihood* or *log-loss*. Let $p(x_i)$ represents the posterior probability of the instance x_i to be associated with the malware class according to the classifier. The *average cross-entropy* is defined as the average over all m test instances:

$$Entropy = \frac{1}{m}\sum_{i=1}^{m} I(x_i)$$

where the *cross-entropy* for a certain case is defined as:

$$I(x_i) = \begin{cases} -logP(x_i) & \text{if } x_i \text{ is malware,} \\ -log(1-P(x_i)) & \text{otherwise.} \end{cases}$$

The use of cross-entropy as a measure of knowledge gain allows us to plot the improvement in a learning process, given an increasing number of examples, by noting whether there is a positive information gain (i.e., a reduction in the entropy after learning, compared to the entropy of the previous learning phase). In particular, we would expect an algorithm that really learns something about the classification of both the positive and negative cases to demonstrate a positive monotonic improvement in the cross-entropy measure. It is important to show this positive monotonic improvement since we would prefer an algorithm that generates classifiers in a stable fashion. Such an algorithm can be considered as more trustworthy than an algorithm whose learning curve might be chaotic.

3.3 Results

The following sections describe various Mal-ID evaluation results starting with the *Mal-ID basic* model followed by the results of two enhancements aimed to improve Mal-ID performance.

3.3.1 RESULTS OF MAL-ID BASIC MODEL

Table 3 presents the detection performance of the proposed method for 70% of the benign files and 90% of the malware files that are used for training.

TPR	FPR	PPV	NPV	Accuracy	AUC	BCR	BER
0.909	0.006	0.944	0.99	0.986	0.951	0.952	0.048

Table 3: Predictive Performance of *Mal-ID basic*.

Kolter and Maloof (2006) conducted rigorous research to find the best combination of n-gram length, n-gram number, features selection and classification method. They reported that the combination of five hundred 4-grams, gain ratio feature selection and boosted decision tree (AdaBoost.M1 with J48 as a base classifier) produced excellent results where the AUC was over 0.99. As you recall, we reproduced the work of Kolter and Maloof (gain ratio, 500 4-grams with boosted decision tree; referred to as *GR500BDT*) to objectively compare the performance of our methods and theirs under the same conditions such as data set content, data set training size, etc. A preliminary evaluation indicated that Rotation Forest (RF) boosting method (Rodriguez et al., 2006) performed better than AdaBoost.M1 and many other non-boosting methods such as J48, therefore RF was selected for our evaluation. The results of the evaluation are presented in Table 4 below.

Method	Features	Feature selection	FPR	TPR	Acc	AUC
GR500BDT	4grams	Gain Ratio	0.094	0.959	0.948	0.929
Mal-ID	Mal-ID	-	0.006	0.909	0.986	0.951

Table 4: Comparison between Mal-ID and *GR500BDT*.

3.3.2 RESULTS OF COMBINING MAL-ID WITH ML GENERATED MODELS

As you recall we attempted to improve the *Mal-ID basic* method by using Mal-ID features with various classifiers. The following figures show comparison of various detection performance measures. Many detection performance measures were recorded and reported as presented in the figures below. Please note that ”TrainPercentage” refers to the percentage of benign data sets and ranges from 30 to 70 percent. Malware data set percentages range from 40 to 90 percent. The ratio between malware and benign was kept fixed for all cases.

Figure 4 reports the average cross-entropy for a classifier by averaging the entropy of the posteriori probability that it outputs to all test instances. As expected, we see that the cross-entropy decreases as the training set size increases. For the largest training set, Mal-ID basic shows the best decrease in a posteriori cross-entropy.

Figure 5 presents the accuracy of the *Mal-ID basic* model as well that of the *Mal-IDF+NN* and *Mal-IDF+RF* models. As expected, the accuracy increases almost linearly as the training set size increases. For small training set sizes, *Mal-IDF+RF* outperforms the other methods. However, for the largest training set, the *Mal-ID basic* model eventually achieves the best results.

Figure 6 presents the TPR of all methods. *Mal-IDF+C4.5* demonstrates the lowest TPR. The *Mal-IDF+NN* and *Mal-IDF+RF* models perform the best. The *Mal-ID basic* model benefits the most from increasing the training set size. In small training sets, the difference between the *Mal-ID basic* model and either *Mal-IDF+NN* or *Mal-IDF+RF* are statistically significant. However, for larger training sets the differences are no longer significant.

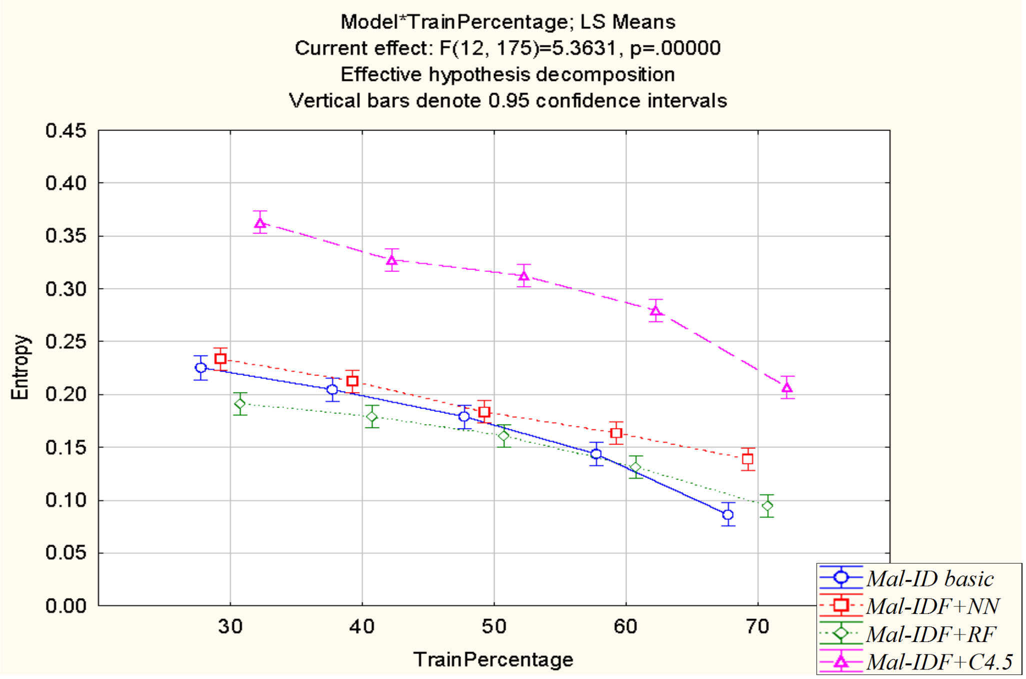


Figure 4: Comparing the a posteriori cross-entropy of various detection modules as a function of training set percentage increase.

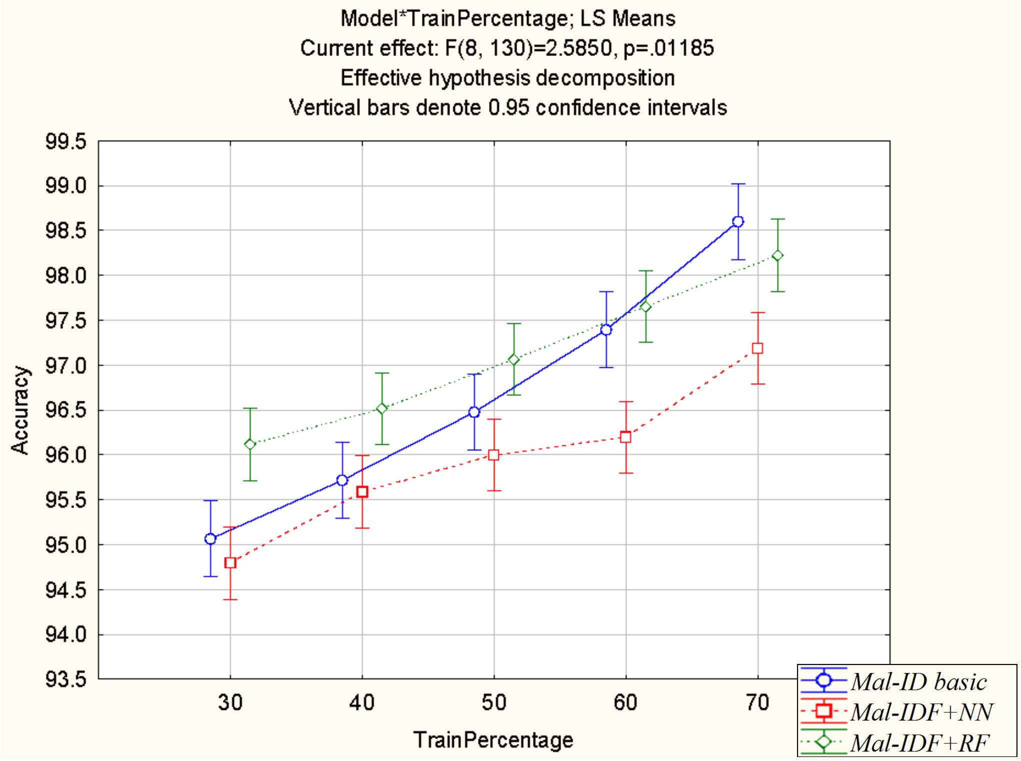


Figure 5: Comparing the accuracy performance of the *Mal-ID basic* model with the machine learning methods on various training set size percentages.

Figure 7 presents the FPR of all methods. The *Mal-ID basic* model demonstrates the best performance. *Mal-IDF+C4.5*, on the other hand, demonstrates the lowest FPR. The performance of *Mal-IDF+NN* does not improve as the training set increases. The *Mal-ID basic* model significantly outperforms *Mal-IDF+C4.5* and *Mal-IDF+NN*. Additionally, a paired t-test indicates the *Mal-ID basic*'s FPR is significantly lower than the FPR of *Mal-IDF+RF* with $p < 0.0001$.

Figure 8 presents the area under the ROC curve for the *Mal-ID basic* model, *Mal-IDF+NN* and *Mal-IDF+RF*. All models improve as the training set increases. The *Mal-ID basic* model shows the lowest AUC but also benefits the most from increasing the training set size. The lower AUC of the *Mal-ID basic* model can be explained by the fact that contrary to the other models, the *Mal-ID basic* model is a *discrete* classifier. Discrete classifiers produce only a single point in ROC space (Fawcett, 2004) and therefore their calculated AUC appears lower.

When we examined the balanced error rate (BER) for *Mal-ID basic*, *Mal-IDF+NN* and *Mal-IDF+RF* Models, we noticed that the BER measure decreases for all models as the training set increases. *Mal-ID basic* demonstrated a significant and sharp decline in the BER as the training set increases. In almost all cases, the *Mal-IDF+RF* achieved the lowest BER. With the largest training set there is no significant difference between the *Mal-ID basic* model and the *Mal-IDF+RF* model.

When we compared the NPV of the *Mal-ID basic* model with the NPV of the *Mal-IDF+NN* and *Mal-IDF+RF*, we noticed, as expected, that the NPV increases almost linearly as the training set

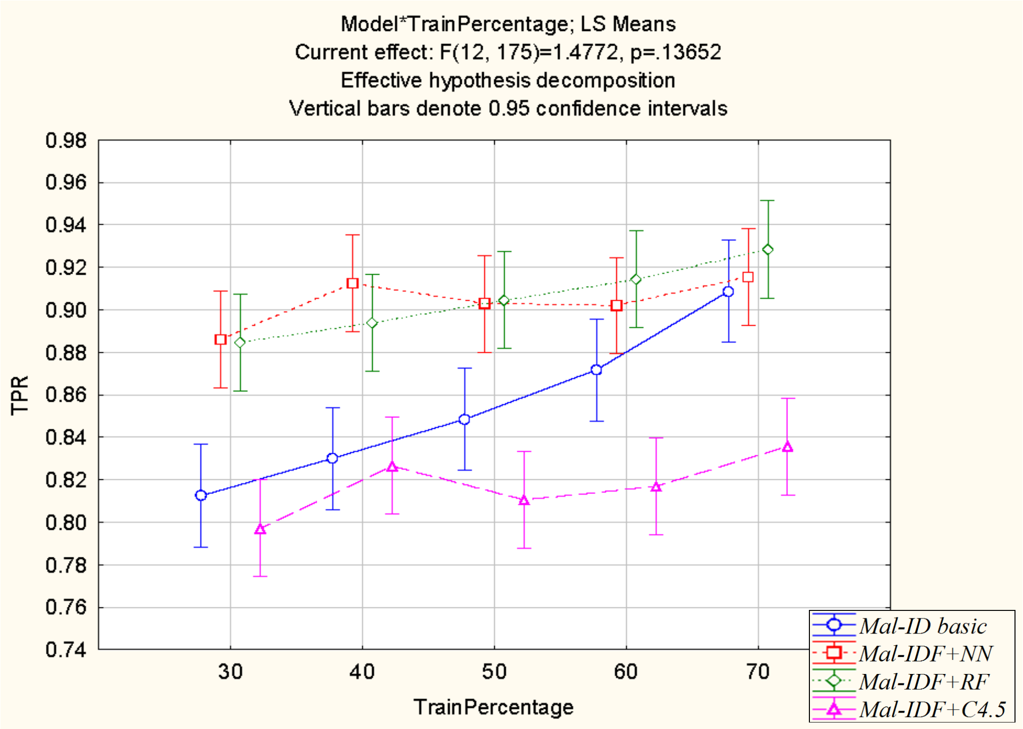


Figure 6: Comparing the true positive rate of the *Mal-ID basic* model with the machine learning methods on various training set size percentages.

size increases. For small training set sizes, *Mal-IDF+RF* and *Mal-IDF+NN* outperform the other methods. Eventually, however, there is no statistically significant difference for the largest training set.

When we compared the PPV of the *Mal-ID basic* model with the PPV of the *Mal-IDF+NN*, *Mal-IDF+C4.5* and *Mal-IDF+RF*, we found out that *Mal-ID basic* has the best PPV for all training set sizes. The *Mal-IDF+RF* performed better than the *Mal-IDF+NN* and the *Mal-IDF+NN* performed better than *Mal-IDF+C4.5*.

To sum up, in many cases Mal-ID basic outperforms the methods that use Mal-ID features combined with a ML classifier and we conclude that a simple decision rule is sufficient.

3.3.3 COMBINING MAL-ID WITH ML MODELS POST PROCESSING

As you recall, we have attempted to improve the *Mal-ID basic* method by using the method to zero-patch the benign common library parts. To measure and compare the effect of the Mal-ID patching prior to classifying, we preformed an evaluation using four methods: *GR500BDT*, *Mal-IDP+GR500BDT*, *Mal-ID basic*, and *Mal-IDF+RF.*

Figure 9 compares the accuracy performance using various training set sizes. The results show that with *Mal-IDP+GR500BDT* we were able to improve performance but only on relatively small training sets. However, compared to the known *GR500BDT*, *Mal-IDP+GR500BDT* show significant and consistent improvements in accuracy by about 2%. All Mal-ID variations were able to

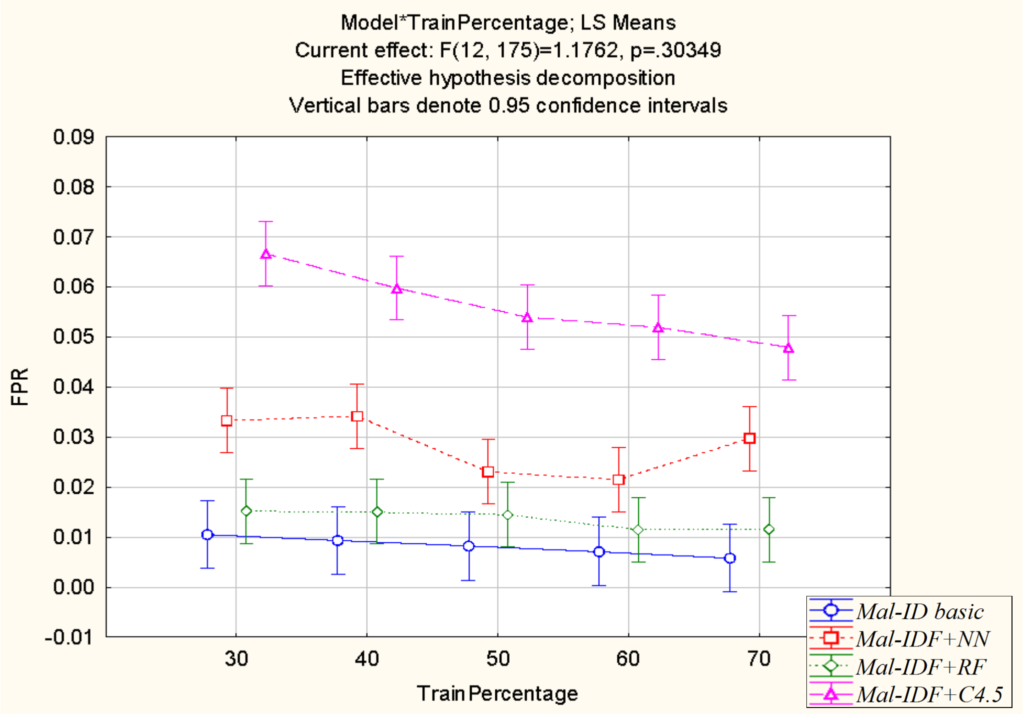


Figure 7: Comparing the false positive rate of the *Mal-ID basic* model with the machine learning methods on various training set size percentages.

outperform *GR500BDT* regardless of training set size. It should be noted that on the one hand we should have expected to an improvement in the predictive performance when the training set size increases. On the other hand because we also increase the imbalance ratio between benign and malware therefore we should have expected to a decrease in the predictive performance. Eventually we observe that accuracy of GR*500BDT* remains almost constant.

Figure 10 compares FPR performance under various training set sizes. The results indicate that there is slight but constant improvement in terms of FPR when first performing a patch with Mal-ID (*Mal-IDP+GR500BDT*) instead of using n-gram without patching (*GR500BDT*). The performance of all n-gram-based methods decreases sharply when the training set consists of more than 50% benign files. The graph shows that in terms of FPR, the *Mal-ID basic* method always performs slightly better than the *Mal-IDF+RF* method and both methods perform significantly better than n-gram based methods. In other words, the graph shows that in terms of FPR, there is a significant difference between methods that use n-gram features and those that use the Mal-ID meta-features.

Table 5 summarizes the detection performance results for the various Mal-ID methods and the *GR500BDT* baseline and can help in choosing the best method when considering detection performance only. Other important considerations will be discussed below. The results demonstrate that *Mal-IDP+GR500BDT* always outperforms *GR500BDT* baseline and *Mal-IDP+GR500BDT* should be used when the highest TPR is desired and a high FPR is acceptable. However *Mal-ID basic* and

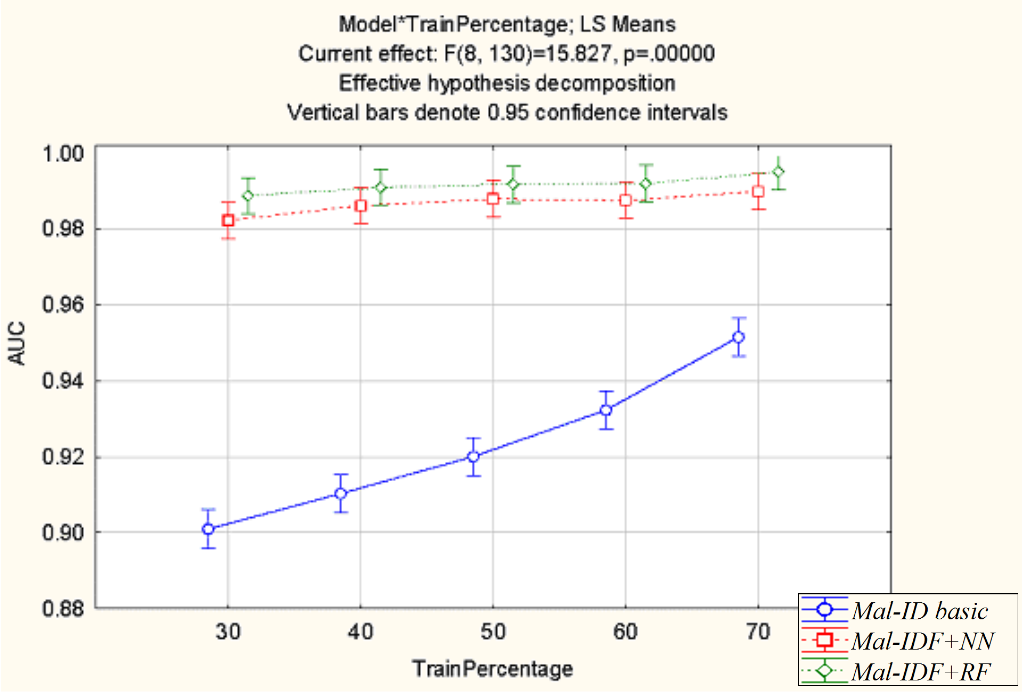


Figure 8: Comparing the AUC of the *Mal-ID basic* model with the machine learning methods on various training set size percentages.

Mal-IDF+RF seems to be the best choice for more balanced performance with extremely low FPR and for achieving the highest accuracy.

Method	**Feature selection**	**FPR**	**TPR**	**Acc**	**AUC**
GR500BDT (un-patched + RF)	Gain Ratio	0.094	0.959	0.948	0.929
Mal-IDP+GR500BDT (patched + RF)	Gain Ratio	0.093	**0.977**	0.963	0.946
Mal-ID basic	Mal-ID	**0.006**	0.909	**0.986**	0.951
Mal-IDF+RF (Mal-ID features + RF)	None	**0.006**	0.916	0.985	**0.995**

Table 5: A comparison of various Mal-ID methods and RF when using maximum training size.

Table 6 presents the training time (in seconds) and detection time (in ms) of all examined methods. The evaluation computer used an Intel Q6850 CPU with 4GB of RAM. All times were measured using only 1 CPU core. The training time of Mal-ID based methods does not include building the CFL and TFL which took around 30 seconds. As expected the training time increases with the training size. In addition, GR500BDT training time does not include the n-gram feature extraction and selection (which took more than ten minutes). The Mal-*ID basic* and Mal-IDF+C4.5 methods demonstrated the best training time performance with less than one second. The detection time

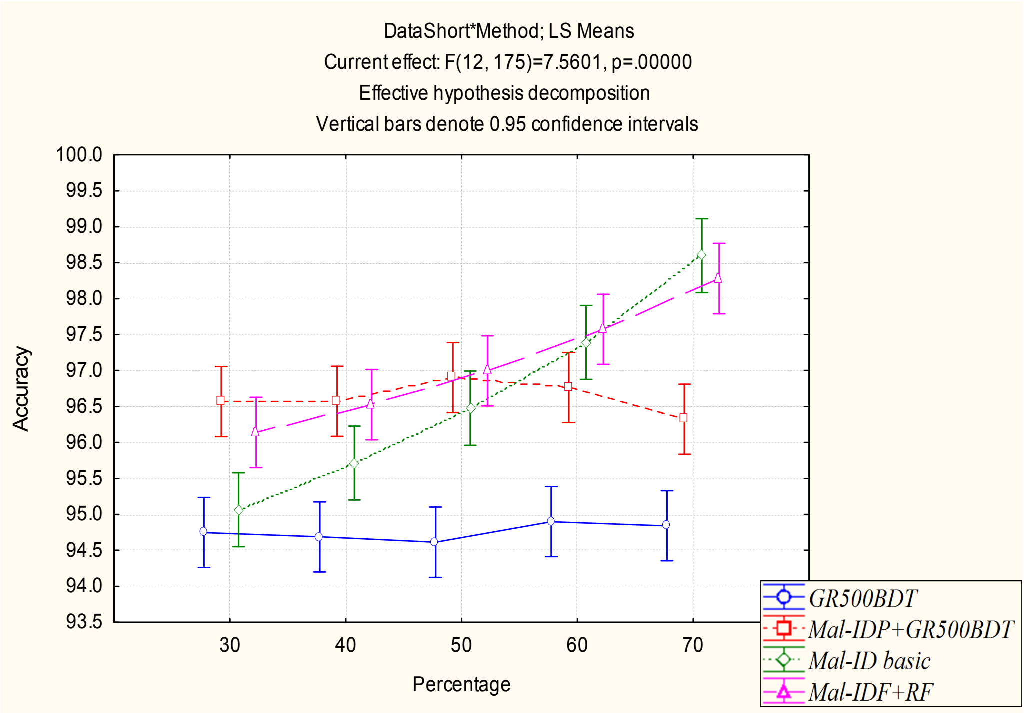


Figure 9: Comparing the accuracy of various Mal-ID-based methods and the n-gram method on various training set size percentages.

seems almost constant regardless of training set size. The only exception is Mal-IDF+RF in which detection time increases almost linearly as the training set increases. Note that the size of the trees (number of nodes) which constitute the rotation forest usually increases with the training set. This can be explained by the fact that the number of leaves in the tree is bounded by the training set size. Larger trees require a longer traversal time and features calculation. Recall that in rotation forest, the features used in the various nodes are linear combination of the original features.

Table 7 reports the mean TPR of *Mal-ID basic* for small malwares (size<=350K) and large malware (size>350K) using the largest training set. Note that the FPR is kept as reported in Table 5 (i.e., FPR=0.006). The results show that the TPR for both small and large group is very similar indicating that MAL ID is not affected by the size of the examined malware.

In order to estimate the effect of obfuscation on detection rate, we have divided the tested malware into two groups—obfuscated and non-obfuscated. Because we were not informed which executable was obfuscated, we have used the following method. We compressed the executables using Zip and sorted them according to the compression ratio. We used a threshold of 50% compression ratio to decide which executable is probably obfuscated. The selection of this threshold was based on experiments of compressing non- obfuscated executables. According to this threshold, about 37.5% of the malware are considered to be obfuscated. Table 8 reports the mean TPR of *Mal-ID basic* for obfuscated and non-obfuscated groups using the largest training set. Note that the FPR is

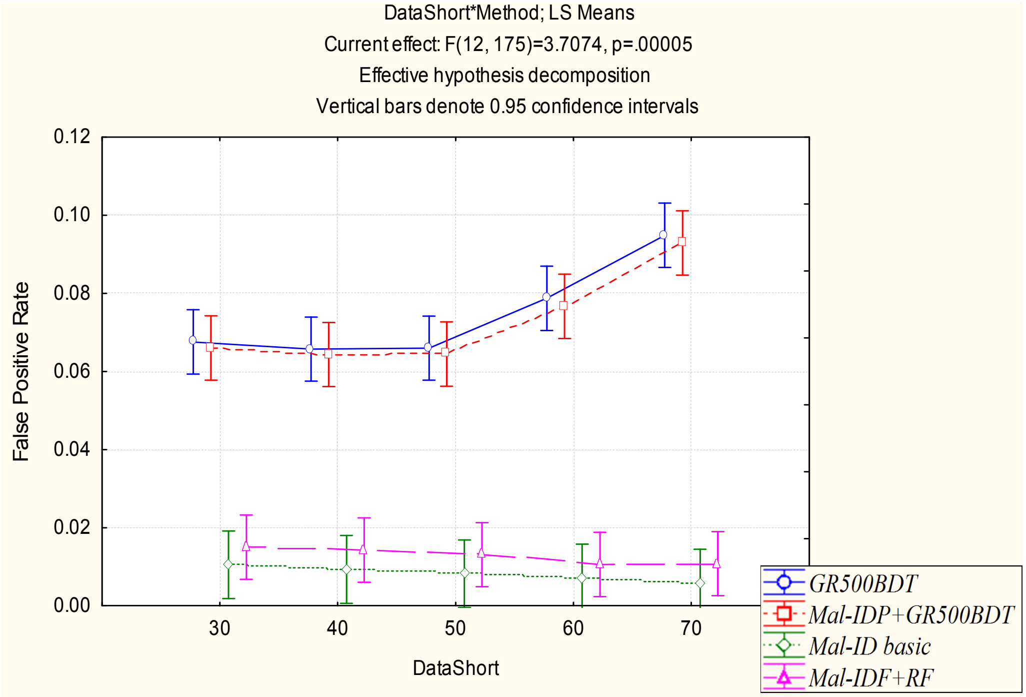


Figure 10: Comparing the FPR of various Mal-ID-based methods and the n-gram method on various training set size percentages.

kept as reported in Table 5 (i.e., FPR=0.006). The results show that the TPR for both obfuscated and non-obfuscated group is very similar with slight advantage to detecting obfuscated malwares.

4. Discussion

This paper proposes a new approach for automatically detecting executable malwares of all kinds and sizes. The results show that using the *Mal-ID basic* and other Mal-ID variants are useful in detecting malware. As can be seen from Table 3, the *Mal-ID basic* method performs very well in respect to all measures. Not only is the accuracy very high (0.986), but the FPR is remarkably low (0.006). In addition, the low Mal-ID BER indicates that the errors are almost uniformly distributed among the malicious and benign files.

As explained in Section 3.3.1, we choose to implement *GR500BDT* as a baseline for comparing the performance of the *Mal-ID basic* method. *GR500BDT* is very similar to the method proposed by Kolter and Maloof (2006). The evaluation shows that *GR500BDT* performed well, but was unable to achieve the AUC of 0.995 that Kolter and Maloof reported. This was probably due to differences in data set content, training size, the benign and malware ratio and possibly other factors. As can be seen from Table 4, under identical conditions the Mal-ID methodology was able to outperform *GR500BDT* in terms of FPR, accuracy and AUC. The FPR of *GR500BDT* method came to almost 10%; Mal-ID FPR was more than 15 times lower.

		Train Percentage				
	Method	**30**	**40**	**50**	**60**	**70**
Training Time (in sec)	MalID-Basic	0.05	0.08	0.11	0.15	0.21
	Mal-IDF+RF	17.19	26.00	36.35	45.78	83.50
	Mal-IDF+C4.5	0.12	0.17	0.22	0.33	0.43
	Mal-IDF+NN	24.33	32.16	40.33	48.37	56.93
	GR500BDT	21.74	34.91	59.86	64.88	75.19
	Mal-IDP+GR500BDT	20.93	31.42	42.96	55.65	63.43
Detection Time per file (in ms)	MalID-Basic	27.86	27.86	27.86	27.86	27.86
	Mal-IDF+RF	49.17	54.69	63.66	73.95	95.82
	Mal-IDF+C4.5	27.86	27.86	27.86	27.86	27.86
	Mal-IDF+NN	27.92	27.92	27.90	27.89	27.88
	GR500BDT	29.63	29.83	29.83	29.85	29.83
	Mal-IDP+GR500BDT	29.01	29.01	29.02	28.98	28.97

Table 6: Training and Detection Time.

Malware Size	**TPR**	**Number of Malwares**	**Mean Size**
Small	0.909	675	96K
Large	0.908	174	554K

Table 7: A comparison of TPR (True Positive Rate) Mal-ID basic for small and large malwares when using maximum training size.

Malware type	**TPR**	**Mean Compression Ratio**
Obfuscated	0.932	41%
Non-obfuscated	0.893	62%

Table 8: A comparison of TPR (True Positive Rate) Mal-ID basic for obfuscated and non-obfuscated malware when using maximum training size.

Once it was established that the *Mal-ID basic* method performs well (in fact better than the best baseline method) we wanted to examine Mal-ID behavior with different train sizes to test if *Mal-ID basic* performs in a stable and “trustworthy” manner. In addition, it was interesting to determine if combining *Mal-ID basic* with ML-generated models, as explained in Section 3.3.2, would yield a better performing malware detection method.

The results presented in Figure 4 to Figure 8 show that combining Mal-ID with ML-based models enabled us to improve many aspects of the *Mal-ID basic* method when training sets are not maximal. However, as training set size increases, the benefit of combining *Mal-ID basic* with ML-based models diminishes. At maximal training set size, the *Mal-ID basic* method almost always demonstrates the best performance or a performance that is statistically equal to the combined methods. It is also important to note that, contrary to the other methods, all measures that we implemented indicated that the *Mal-ID basic* method benefited the most from training set increase and always performed in an expected manner. Thus, it may be considered more stable and "trustworthy" than the other methods.

It is interesting to note that while the performance of non-n-gram methods (*Mal-ID basic* and *Mal-IDF+RF*) continues to improve as more training data become available, the n-gram based methods show a sharp decrease in performance in terms of FPR (see Figure 10). This can be explained by the fact that n-gram methods induce relatively simple patterns that can be learned with comparatively small training sets (30%). The potential benefit of additional training data is nullified by the undesirable increase in the probability that relevant n-gram will be mistakenly considered as non-contributing features. In fact, it is well known that decision trees increase their chances of overfitting when they have more nodes. But in order to have more nodes, they need a larger training set. Thus a larger data set might increase the chance of overfitting especially in cases were there are many irrelevant and noisy features.

The comparison of our two additional methods, *Mal-IDF+RF* and *Mal-IDP+GR500BDT*, with a *GR500BDT* baseline is very important in proving the validity of Mal-ID itself and explaining its excellent performance:

1. (a) Under identical conditions, boosted decision tree, operating on *Mal-ID basic* meta-features (*Mal-IDF+RF*), outperformed boosted decision tree operating on n-gram (*GR500BDT*). The comparison suggests that Mal-ID meta-features are useful in contributing to malware detection and probably more meaningful than simple n-gram in capturing a file's essence.

   (b) Under identical conditions, boosted decision tree operating on *Mal-ID basic* patched files (*Mal-IDP+GR500BDT*) outperformed boosted decision tree operating on non-patched files (*GR500BDT*). The comparison suggests that the novel Mal-ID common segment analysis approach is better than the common approach that treats files as black boxes or which interprets files PE header only.

Since *Mal-ID basic* and *Mal-IDF+RF* methods benefit from both more meaningful features and common segment analysis, they are able to achieve a better overall performance than state-of-the-art *GR500BDT*.

Considering detection performance only when choosing a malware detection method may not be enough; it is important to consider other aspects as well.

4.1 Model Interpretability

Mal-ID basic uses only one static interpretable classification model and therefore experts in the field can be more confident when accepting or rejecting a classification. For instance, once *Mal-ID basic* has detected a yet unknown malware, it is possible to support or reject the classification. The reason is that each detected segment, that passed the Mal-ID filter stage as explained in Section 2, can be tracked back to a specific malware or malware group. Moreover, the specific offset location were the segments appear can be examined to determine the precise nature of the threat, if any exists. Disassembly or reverse engineering of the whole malware is no longer required. Even without examining the segment code, one can make an educated guess about the nature of the threat by examining the list of known malwares that the segment appears in. The other methods do not provide such benefits.

4.2 Incremental

As more malwares are discovered, it is important to update the models from time to time. With *Mal-ID basic* it is particularly easy. Since the model is static, no reconstruction is necessary; all that is required is to just to add or subtract files from the TFL. The CFL can be updated in a similar manner.

4.3 Anytime Detection

Recall that both *Mal-ID basic* and *Mal-IDF+RF* operates on segments. Because *Mal-ID basic* and *Mal-IDF+RF* use relatively large segments and the model is not comprised of combined features from the whole file, it is possible to stop detection at anytime during file scan and determine if the scanned part is malicious. n-gram-based methods are not designed to diagnose part of file but rather whole files only.

4.4 Default Signature For Real-time Malware Detection Hardware

The end result of applying *Mal-ID basic* method is a file segment or segments that appear in malware files only and thus may be used as a signature for anti-virus tools. The detected malware segments can be used, as described by Filiol (2006), to generate signatures resistant against black-box analysis. Moreover, because *Mal-ID basic* produces a simple signature and has *anytime detection traits*, the signature can be used with commercially available real-time intrusion prevention systems (IPS). IPSs require the *anytime detection trait* to act as real-time malware filtering devices and thus promote and provide users with default protection. Having both malware detection and signature generation could help shorten the window of vulnerability. Tahan et al. (2010) have presented a methodology with complete implementation for automatic signature generation, using similar and compatible techniques, which archived excellent results in the evaluation. Thus, the method presented by Tahan et al. (2010) can be easily adopted to produce signature upon detection for the solution presented in this paper.

4.5 Large Files Scalability

Nowadays it's quite common to embed large resources such as JPEG pictures and small animations into executables. This inflation is also true for malware. It is estimated[1] that the mean malware size has increased from 150K (in 2005) to 350K (in 2010). As files become larger, the effectiveness of classification with small n-gram should decrease due to the increase in file entropy. In other words, the more n-gram with equal appearance probability, the greater the misclassification probability becomes. Since *Mal-ID basic* and *Mal-IDF+RF* use relatively large segments (64 bytes) and in addition filter-out high entropy parts, they should be less susceptible to misclassification caused by large files or files with high entropy traits. Figure 10 shows that the Mal-ID methods that operate on large segments (of 64 bytes) has less FPR misclassification then the method that operated on small n-gram (of 4 bytes). We further examined this hypothesis in Table 7.

[1] See `http://nakedsecurity.sophos.com/2010/07/27/large-piece-malware/`.

4.6 Analysis of Mal-ID Performance on Obfuscated Malware

Based on the results presented so far, we hypothesize that the proposed Mal-ID method performs well in a mixed environment where both obfuscated (including compressed or encrypted) and plain executable files exist. In this sense, we referred to malware as they are found "in the Wild".

There might be several reasons that can explain why the TPR of obfuscated binaries appears to be higher than the TPR of non-obfuscated binaries. One reason can be that many obfuscated malwares are generated by automated tools that have distinctive properties. For example, malware developers are sharing tools for facilitating the generation of new malwares. For example, in the web site `http://vx.netlux.org/`, one can find many tools (such as Falckon Encrypter that is used for obfuscation) that can be used by the malware developers but are not used by benign software developers. All malware that use the Falckon Encrypter, share the same decryption segment.

The results of Table 8 agree with the previously-made observation that ML techniques can classify malware that are obfuscated (compressed or encrypted or both). For example, Kolter and Maloof (2006) have noted that ML can detect obfuscated malware. In this paper, we have independently reconfirmed the validity of the above observation using our method. In this experiment, we succeeded to keep FPR relatively low (FPR=0.006), however it should be noted that this value was obtained when our corpus contained 2,627 benign files and 849 malware files (i.e., a benign to malware ratio of 3:1). In reality this ratio can be much higher and therefore one should expect to obtain elevated FPR values.

There seem to be previously suggested explanations to this phenomenon. According to Kolter and Maloof (2006), the success in detecting obfuscated malware relies on learning certain forms of obfuscation such as run-time decompression. Kolter and Maloof (2006) conclude that "...this does not seem problematic as long as those forms are correlated with malicious executables".

Additional explanations can be suggested to the ability to identify obfuscated malware. Studies such as that presented by Newsome and Song, or by Newsome et al. (2005) noticed that in many cases malware requires fixed sequences to be used in the body of the malware (which must exist before self-decryption or self-decompression) in order to exploit a specific vulnerability and self-propagate. Such fixed sequences can be used for detection. This might explain the success in detecting obfuscated malware.

Because the performance of MAL ID is achieved with no disassembly, Op-Code analysis, executable header analysis, unpacking nor any other preprocessing, we hypothesize that the method should be scalable to other Operating Systems and hardware types. Still one can think on cases where preprocessing will be required. Theoretically an attacker can specifically design a malware that will make it hard for MAL ID to detect it. In particular, if a malware is designed such that the entropy measure will be high for all segments, it will be undiscovered by the Mal-ID basic method. In this case Mal-ID can be extended by incorporating an unpacker operating before it, such as those that are incorporated into anti-viruses tools (Kasparsky). However, similar to Kolter and Maloof (2006), we decided to evaluate the raw power of our methods without any use of an unpacker.

5. Summary and Future Work

In this paper we have described novel methods based on machine learning to detect malware in executable files without any need for preprocessing the executables. The basic method that we presented works on the segment level for detecting new malware instead of using the entire file as usually done in machine learning based techniques. The *Mal-ID basic* method and its derived variants were rigorously tested to evaluate their effectiveness under many conditions using a wide variety of measures. The results demonstrate the effectiveness of the methods. In all cases, most of the performance measures showed that the proposed methods significantly outperformed the baseline method *GR500BDT* which is known for its excellent performance (Kolter and Maloof, 2004, 2006). For each method we have pinpointed its strong points and suggested cases where it should be preferred over the others.

We believe this study has made several contributions to malware detection research, including the introduction of:

1. a new and effective method for malware detection based on common segment analysis and supporting algorithms. The importance of common segment analysis to the process of malware detection was identified and demonstrated. The results suggest the method can boost performance for many methods that use n-gram.

2. new kinds of features—*Mal-ID basic* meta-features. The results suggest that the meta-features are much more effective than the commonly used n-gram and probably more meaningful in terms of file representation. We believe that *Mal-ID basic* meta-features could inspire many kinds of additional meta-features that could prove useful.

3. BCR, BER, PPV, NPV and entropy decrease for measuring the performance of malware detection methods. Using these measures, in addition to the commonly used performance measures (TPR, FPR, accuracy and AUC), is not generally practiced. However, these features are helpful in describing the behavior of a new method, particularly when it is not possible to compare results under identical settings and data set imbalance.

The results also indicate that by extracting meaningful features, it is sufficient to employ one simple detection rule for classifying unknown executables.

In the future, we aim to examine the effect of systematically collecting and choosing the benign file set on the performance of the proposed methods. In the evaluations that were conducted for this study, the benign file set was collected randomly and the files used may have had a large degree of similarity. It is our assumption that systematically collecting and choosing common segments will provide a better representation of benign common segments and a more robust and lower FPR. A robust and low FPR will enable the use of more sensitive malware detection methods (or parameters that affect malware detection) without increasing the FPR too much. As a result, we hope to see further increase in the AUC measure. Finally the Mal-ID basic method was developed as a crisp classifier. Additional research is required for developing a method for ranking the examined files according to their presumed threat level. One straightforward measure is the ratio between the segments found in the TFL and the segments found in the CFL. In addition, it will be interesting to test the proposed method on live network data and on an institutional network and determine if it detects malware that is not detected by other means. Finally, future work may repeat the evaluation Mal-ID on a larger scale with thousands of malware samples and tens of thousands of non-malware samples. For this purpose, we might need to upscale software components to accommodate large data set and suitable hardware. In addition, in order to use the proposed method in practice by the industry, fine tuning of the various parameters might be required.

Additional studies might be needed to fully evaluate the performance of Mal-ID under various obfuscation scheme, including use of recursive unpacking. In this paper we focused only on "pure" Mal-ID methods and therefore we did not investigate the proper means to incorporate unpacker.

Acknowledgments

The authors gratefully thank the action editor, Prof. Charles Elkan and the anonymous reviewers whose constructive comments considerably strengthened this manuscript.

References

T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan. N-gram-based detection of new malicious code. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, volume 2, pages 41–42. IEEE, 2004.

D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6 (1):37–66, 1991.

R.E. Bellman, R.E. Bellman, R.E. Bellman, and R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1966.

C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon press Oxford, 1995.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

R. Caruana, T. Joachims, and L. Backstrom. Kdd-cup 2004: Results and analysis. *ACM SIGKDD Explorations Newsletter*, 6(2):95–108, 2004.

J. Dai, R. Guha, and J. Lee. Efficient virus detection using dynamic instruction sequences. *Journal of Computers*, 4(5):405–414, 2009.

G. Demiröz and H. Güvenir. Classification by voting feature intervals. In Maarten van Someren and Gerhard Widmer, editors, *Proceedings of the 9th European Conference on Machine Learning*, Lecture Notes in Computer Science, pages 85–92. Springer Berlin / Heidelberg, 1997.

Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer. Applying machine learning techniques for detection of malicious code in network traffic. *KI 2007: Advances in Artificial Intelligence*, pages 44–50, 2007.

T. Fawcett. Roc graphs: Notes and practical considerations for researchers. *ReCALL*, 31(HPL-2003-4):1–38, 2004.

U. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993.

E. Filiol. Malware pattern scanning schemes secure against black-box analysis. *Journal in Computer Virology*, 2(1):35–50, 2006.

V. Franc and S. Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *The Journal of Machine Learning Research*, 10:2157–2192, 2009.

Y. Freund and R.E. Schapire. A brief introduction to boosting. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 1401–1406. Lawrence Erlbaum Associates LTD, 1999.

T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531, 1999.

O. Henchiri and N. Japkowicz. A feature selection and evaluation scheme for computer virus detection. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 891–895. IEEE, 2006.

R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.

T. Joachims. Advances in kernel methods. chapter Making Large-Scale SVM Learning Practical, pages 169–184. MIT Press, 1999.

G.H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. San Mateo, 1995.

J. Z. Kolter and M. A. Maloof. Learning to detect malicious executables in the wild. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 470–478, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1.

J.Z. Kolter and M.A. Maloof. Learning to detect and classify malicious executables in the wild. *The Journal of Machine Learning Research*, 7:2721–2744, 2006.

E. Menahem, A. Shabtai, L. Rokach, and Y. Elovici. Improving malware detection by applying multi-inducer ensemble. *Computational Statistics & Data Analysis*, 53(4):1483–1494, 2009.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

J. Newsome and D. Song. Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

J. Newsome, B. Karp, and D. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *Security and Privacy, 2005 IEEE Symposium on*, pages 226–241. IEEE, 2005.

N. O'Farrell. Cybercrime costs society over one trillion, 2011. URL `http://www.idguardian.com/headlines-cybercrime-costs-trillion/`.

J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257, 1987.

J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

J.J. Rodriguez, L.I. Kuncheva, and C.J. Alonso. Rotation forest: A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1619–1630, 2006.

L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010.

L. Rokach, R. Romano, and O. Maimon. Negation recognition in medical narrative reports. *Information Retrieval*, 11(6):499–538, 2008.

M.G. Schultz, E. Eskin, F. Zadok, and S.J. Stolfo. Data mining methods for detection of new malicious executables. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 38–49. IEEE, 2001.

Symantec. Symantec global internet security threat report trends for 2010, 2010. URL `http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf`.

G. Tahan, C. Glezer, Y. Elovici, and L. Rokach. Auto-sign: an automatic signature generator for high-speed malware filtering devices. *Journal in Computer Virology*, 6(2):91–103, 2010.

C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, pages 133–145. IEEE, 1999.

I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang. An intelligent pe-malware detection system based on association mining. *Journal in Computer Virology*, 4(4):323–334, 2008.

Y. Ye, L. Chen, D. Wang, T. Li, Q. Jiang, and M. Zhao. Sbmds: an interpretable string based malware detection system using svm ensemble with bagging. *Journal in Computer Virology*, 5 (4):283–293, 2009.

Y. Ye, T. Li, K. Huang, Q. Jiang, and Y. Chen. Hierarchical associative classifier (hac) for malware detection from the large and imbalanced gray list. *Journal of Intelligent Information Systems*, 35 (1):1–20, 2010.

S. Yu, S. Zhou, L. Liu, R. Yang, and J. Luo. Detecting malware variants by byte frequency. *Journal of Networks*, 6(4):638–645, 2011.

B. Zhang, J. Yin, J. Hao, D. Zhang, and S. Wang. Malicious codes detection based on ensemble learning. *Autonomic and Trusted Computing*, pages 468–477, 2007.

Journal of Machine Learning Research 13 (2012) 981-1006 Submitted 3/11; Revised 12/11; Published 4/12

Sampling Methods for the Nyström Method

Sanjiv Kumar SANJIVK@GOOGLE.COM
Google Research
76 Ninth Avenue
New York, NY 10011

Mehryar Mohri MOHRI@CS.NYU.EDU
Courant Institute and Google Research
251 Mercer Street
New York, NY 10012

Ameet Talwalkar AMEET@CS.BERKELEY.EDU
University of California, Berkeley
Division of Computer Science
465 Soda Hall
Berkeley, CA 94720

Editor: Inderjit Dhillon

Abstract

The Nyström method is an efficient technique to generate low-rank matrix approximations and is used in several large-scale learning applications. A key aspect of this method is the procedure according to which columns are sampled from the original matrix. In this work, we explore the efficacy of a variety of *fixed* and *adaptive* sampling schemes. We also propose a family of *ensemble*-based sampling algorithms for the Nyström method. We report results of extensive experiments that provide a detailed comparison of various fixed and adaptive sampling techniques, and demonstrate the performance improvement associated with the ensemble Nyström method when used in conjunction with either fixed or adaptive sampling schemes. Corroborating these empirical findings, we present a theoretical analysis of the Nyström method, providing novel error bounds guaranteeing a better convergence rate of the ensemble Nyström method in comparison to the standard Nyström method.

Keywords: low-rank approximation, nyström method, ensemble methods, large-scale learning

1. Introduction

A common problem in many areas of large-scale machine learning involves deriving a useful and efficient approximation of a large matrix. This matrix may be a kernel matrix used with support vector machines (Cortes and Vapnik, 1995; Boser et al., 1992), kernel principal component analysis (Schölkopf et al., 1998) or manifold learning (Platt, 2004; Talwalkar et al., 2008). Large matrices also naturally arise in other applications, for example, clustering, collaborative filtering, matrix completion, robust PCA, etc. For these large-scale problems, the number of matrix entries can be in the order of tens of thousands to millions, leading to difficulty in operating on, or even storing the matrix. An attractive solution to this problem involves using the Nyström method to generate a low-rank approximation of the original matrix from a subset of its columns (Williams and Seeger, 2000). A key aspect of the Nyström method is the procedure according to which the columns are sampled. This paper presents an analysis of different sampling techniques for the Nyström method both empirically and theoretically.[1]

In the first part of this work, we focus on various *fixed* sampling methods. The Nyström method was first introduced to the machine learning community (Williams and Seeger, 2000) using uniform sampling without replacement, and this remains the sampling method most commonly used in practice (Talwalkar et al., 2008; Fowlkes et al., 2004; de Silva and Tenenbaum, 2003; Platt, 2004). More recently, the Nyström method has been theoretically analyzed assuming sampling from fixed, non-uniform distributions over the columns (Drineas and Mahoney, 2005; Belabbas and Wolfe, 2009; Mahoney and Drineas, 2009). In this work, we present novel experiments with several real-world data sets comparing the performance of the Nyström method when used with uniform versus non-uniform sampling distributions. Although previous studies have compared uniform and non-uniform distributions in a more restrictive setting (Drineas et al., 2001; Zhang et al., 2008), our results are the first to compare uniform sampling with the sampling technique for which the Nyström method has theoretical guarantees. Our results suggest that uniform sampling, in addition to being more efficient both in time and space, produces more effective approximations. We further show the benefits of sampling without replacement. These empirical findings help motivate subsequent theoretical analyses.

The Nyström method has also been studied empirically and theoretically assuming more sophisticated iterative selection techniques (Smola and Schölkopf, 2000; Fine and Scheinberg, 2002; Bach and Jordan, 2002). In the second part of this work, we provide a survey of adaptive techniques that have been suggested for use with the Nyström method, and present an empirical comparison across these algorithms. As part of this work, we build upon ideas of Deshpande et al. (2006), in which an adaptive, error-driven sampling technique with relative error bounds was introduced for the related problem of matrix projection (see Kumar et al. 2009b for details). However, this technique requires the full matrix to be available at each step, and is impractical for large matrices. Hence, we propose a simple and efficient algorithm that extends the ideas of Deshpande et al. (2006) for adaptive sampling and uses only a small submatrix at each step. Our empirical results suggest a trade-off between time and space requirements, as adaptive techniques spend more time to find a concise subset of informative columns but provide improved approximation accuracy.

Next, we show that a new family of algorithms based on mixtures of Nyström approximations, *ensemble Nyström algorithms*, yields more accurate low-rank approximations than the standard Nyström method. Moreover, these ensemble algorithms naturally fit within distributed computing environments, where their computational costs are roughly the same as that of the standard Nyström method. This issue is of great practical significance given the prevalence of distributed computing frameworks to handle large-scale learning problems. We describe several variants of these algorithms, including one based on simple averaging of p Nyström solutions, an exponential weighting method, and a regression method which consists of estimating the mixture parameters of the ensemble using a few columns sampled from the matrix. We also report the results of extensive experiments with these algorithms on several data sets comparing different variants of the ensemble Nyström algorithms and demonstrating the performance improvements gained over the standard Nyström method.

[1] Portions of this work have previously appeared in the Conference on Artificial Intelligence and Statistics (Kumar et al., 2009a), the International Conference on Machine Learning (Kumar et al., 2009b) and Advances in Neural Information Processing Systems (Kumar et al., 2009c).

Finally, we present a theoretical analysis of the Nyström method, namely bounds on the reconstruction error for both the Frobenius norm and the spectral norm. We first present a novel bound for the Nyström method as it is often used in practice, that is, using uniform sampling without replacement. We next extend this bound to the ensemble Nyström algorithms, and show these novel generalization bounds guarantee a better convergence rate for these algorithms in comparison to the standard Nyström method.

The remainder of the paper is organized as follows. Section 2 introduces basic definitions, provides a short survey on related work and gives a brief presentation of the Nyström method. In Section 3, we study various fixed sampling schemes used with the Nyström method. In Section 4, we provide a survey of various adaptive techniques used for sampling-based low-rank approximation and introduce a novel adaptive sampling algorithm. Section 5 describes a family of ensemble Nyström algorithms and presents extensive experimental results. We present novel theoretical analysis in Section 6.

2. Preliminaries

Let $\mathbf{T} \in \mathbb{R}^{a\times b}$ be an arbitrary matrix. We define $\mathbf{T}^{(j)}$, $j = 1 \ldots b$, as the jth column vector of $\mathbf{T}$, $\mathbf{T}_{(i)}$, $i = 1 \ldots a$, as the ith row vector of $\mathbf{T}$ and $\|\cdot\|$ the l_2 norm of a vector. Furthermore, $\mathbf{T}^{(i:j)}$ refers to the ith through jth columns of $\mathbf{T}$ and $\mathbf{T}_{(i:j)}$ refers to the ith through jth rows of $\mathbf{T}$. If $\text{rank}(\mathbf{T}) = r$, we can write the thin Singular Value Decomposition (SVD) of this matrix as $\mathbf{T} = \mathbf{U}_T \mathbf{\Sigma}_T \mathbf{V}_T^\top$ where $\mathbf{\Sigma}_T$ is diagonal and contains the singular values of $\mathbf{T}$ sorted in decreasing order and $\mathbf{U}_T \in \mathbb{R}^{a\times r}$ and $\mathbf{V}_T \in \mathbb{R}^{b\times r}$ have orthogonal columns that contain the left and right singular vectors of $\mathbf{T}$ corresponding to its singular values. We denote by $\mathbf{T}_k$ the 'best' rank-k approximation to $\mathbf{T}$, that is, $\mathbf{T}_k = \text{argmin}_{\mathbf{V}\in\mathbb{R}^{a\times b}, \text{rank}(\mathbf{V})=k} \|\mathbf{T}-\mathbf{V}\|_\xi$, where $\xi \in \{2, F\}$ and $\|\cdot\|_2$ denotes the spectral norm and $\|\cdot\|_F$ the Frobenius norm of a matrix. We can describe this matrix in terms of its SVD as $\mathbf{T}_k = \mathbf{U}_{T,k}\mathbf{\Sigma}_{T,k}\mathbf{V}_{T,k}^\top$ where $\mathbf{\Sigma}_{T,k}$ is a diagonal matrix of the top k singular values of $\mathbf{T}$ and $\mathbf{U}_{T,k}$ and $\mathbf{V}_{T,k}$ are the matrices formed by the associated left and right singular vectors.

Now let $\mathbf{K} \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite (SPSD) kernel or Gram matrix with $\text{rank}(\mathbf{K}) = r \le n$, that is, a symmetric matrix for which there exists an $\mathbf{X} \in \mathbb{R}^{N\times n}$ such that $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$. We will write the SVD of $\mathbf{K}$ as $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, where the columns of $\mathbf{U}$ are orthogonal and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_r)$ is diagonal. The pseudo-inverse of $\mathbf{K}$ is defined as $\mathbf{K}^+ = \sum_{t=1}^r \sigma_t^{-1}\mathbf{U}^{(t)}\mathbf{U}^{(t)\top}$, and $\mathbf{K}^+ = \mathbf{K}^{-1}$ when $\mathbf{K}$ is full rank. For $k < r$, $\mathbf{K}_k = \sum_{t=1}^k \sigma_t \mathbf{U}^{(t)}\mathbf{U}^{(t)\top} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{U}_k^\top$ is the 'best' rank-k approximation to $\mathbf{K}$, that is, $\mathbf{K}_k = \text{argmin}_{\mathbf{K}'\in\mathbb{R}^{n\times n}, \text{rank}(\mathbf{K}')=k} \|\mathbf{K}-\mathbf{K}'\|_{\xi\in\{2,F\}}$, with $\|\mathbf{K}-\mathbf{K}_k\|_2 = \sigma_{k+1}$ and $\|\mathbf{K}-\mathbf{K}_k\|_F = \sqrt{\sum_{t=k+1}^r \sigma_t^2}$ (Golub and Loan, 1983).

We will be focusing on generating an approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$ based on a sample of $l \ll n$ of its columns. For now, we assume that the sample of l columns is given to us, though the focus of this paper will be on various methods for selecting columns. Let $\mathbf{C}$ denote the $n \times l$ matrix formed by these columns and $\mathbf{W}$ the $l \times l$ matrix consisting of the intersection of these l columns with the corresponding l rows of $\mathbf{K}$. Note that $\mathbf{W}$ is SPSD since $\mathbf{K}$ is SPSD. Without loss of generality, the columns and rows of $\mathbf{K}$ can be rearranged based on this sampling so that $\mathbf{K}$ and $\mathbf{C}$ be written as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}. \tag{1}$$

2.1 Nyström Method

The Nyström method uses $\mathbf{W}$ and $\mathbf{C}$ from (1) to approximate $\mathbf{K}$. Assuming a uniform sampling of the columns, the Nyström method generates a rank-k approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$ for $k < n$ defined by:

$$\widetilde{\mathbf{K}}_k^{nys} = \mathbf{C}\mathbf{W}_k^+\mathbf{C}^\top \approx \mathbf{K},$$

where $\mathbf{W}_k$ is the best k-rank approximation of $\mathbf{W}$ with respect to the spectral or Frobenius norm and $\mathbf{W}_k^+$ denotes the pseudo-inverse of $\mathbf{W}_k$. The Nyström method thus approximates the top k singular values ($\mathbf{\Sigma}_k$) and singular vectors ($\mathbf{U}_k$) of $\mathbf{K}$ as:

$$\widetilde{\mathbf{\Sigma}}_k^{nys} = \left(\frac{n}{l}\right)\mathbf{\Sigma}_{W,k} \quad \text{and} \quad \widetilde{\mathbf{U}}_k^{nys} = \sqrt{\frac{l}{n}}\mathbf{C}\mathbf{U}_{W,k}\mathbf{\Sigma}_{W,k}^+. \tag{2}$$

When $k = l$ (or more generally, whenever $k \geq \text{rank}(\mathbf{C})$), this approximation perfectly reconstructs three blocks of $\mathbf{K}$, and $\mathbf{K}_{22}$ is approximated by the Schur Complement of $\mathbf{W}$ in $\mathbf{K}$:

$$\widetilde{\mathbf{K}}_l^{nys} = \mathbf{C}\mathbf{W}^+\mathbf{C}^\top = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{21}\mathbf{W}^+\mathbf{K}_{21} \end{bmatrix}. \tag{3}$$

Since the running time complexity of SVD on $\mathbf{W}$ is in $O(kl^2)$ and matrix multiplication with $\mathbf{C}$ takes $O(kln)$, the total complexity of the Nyström approximation computation is in $O(kln)$.

2.2 Related Work

There has been a wide array of work on low-rank matrix approximation within the numerical linear algebra and computer science communities, much of which has been inspired by the celebrated result of Johnson and Lindenstrauss (1984), which showed that random low-dimensional embeddings preserve Euclidean geometry. This result has led to a family of random projection algorithms, which involves projecting the original matrix onto a random low-dimensional subspace (Papadimitriou et al., 1998; Indyk, 2006; Liberty, 2009). Alternatively, SVD can be used to generate 'optimal' low-rank matrix approximations, as mentioned earlier. However, both the random projection and the SVD algorithms involve storage and operating on the entire input matrix. SVD is more computationally expensive than random projection methods, though neither are linear in n in terms of time and space complexity. When dealing with sparse matrices, there exist less computationally intensive techniques such as Jacobi, Arnoldi, Hebbian and more recent randomized methods (Golub and Loan, 1983; Gorrell, 2006; Rokhlin et al., 2009; Halko et al., 2009) for generating low-rank approximations. These methods require computation of matrix-vector products and thus require operating on every non-zero entry of the matrix, which may not be suitable for large, dense matrices. Matrix sparsification algorithms (Achlioptas and Mcsherry, 2007; Arora et al., 2006), as the name suggests, attempt to sparsify dense matrices to speed up future storage and computational burdens, though they too require storage of the input matrix and exhibit superlinear processing time.

Alternatively, sampling-based approaches can be used to generate low-rank approximations. Research in this area dates back to classical theoretical results that show, for any arbitrary matrix, the existence of a subset of k columns for which the error in matrix projection (as defined in Kumar et al., 2009b) can be bounded relative to the optimal rank-k approximation of the matrix (Ruston, 1962). Deterministic algorithms such as rank-revealing QR (Gu and Eisenstat, 1996) can achieve nearly optimal matrix projection errors. More recently, research in the theoretical computer science community has been aimed at deriving bounds on matrix projection error using sampling-based approximations, including additive error bounds using sampling distributions based on the squared L_2 norms of the columns (Frieze et al., 1998; Drineas et al., 2006; Rudelson and Vershynin, 2007); relative error bounds using adaptive sampling techniques (Deshpande et al., 2006; Har-peled, 2006); and, relative error bounds based on distributions derived from the singular vectors of the input matrix, in work related to the column-subset selection problem (Drineas et al., 2008; Boutsidis et al., 2009). These sampling-based approximations all require visiting every entry of the matrix in order to get good performance guarantees for any matrix. However, as discussed in Kumar et al. (2009b), the task of matrix projection involves projecting the input matrix onto a low-rank subspace, which requires superlinear time and space with respect to n and is not always feasible for large-scale matrices.

There does exist, however, another class of sampling-based approximation algorithms that only store and operate on a subset of the original matrix. For arbitrary rectangular matrices, these algorithms are known as 'CUR' approximations (the name 'CUR' corresponds to the three low-rank matrices whose product is an approximation to the original matrix). The theoretical performance of CUR approximations has been analyzed using a variety of sampling schemes, although the column-selection processes associated with these analyses often require operating on the entire input matrix (Goreinov et al., 1997; Stewart, 1999; Drineas et al., 2008; Mahoney and Drineas, 2009).

In the context of symmetric positive semidefinite matrices, the Nyström method is a commonly used algorithm to efficiently generate low-rank approximations. The Nyström method was initially introduced as a quadrature method for numerical integration, used to approximate eigenfunction solutions (Nyström, 1928; Baker, 1977). More recently, it was presented in Williams and Seeger (2000) to speed up kernel algorithms and has been studied theoretically using a variety of sampling schemes (Smola and Schölkopf, 2000; Drineas and Mahoney, 2005; Zhang et al., 2008; Zhang and Kwok, 2009; Kumar et al., 2009a,b,c; Belabbas and Wolfe, 2009; Belabbas and Wolfe, 2009; Cortes et al., 2010; Talwalkar and Rostamizadeh, 2010). It has also been used for a variety of machine learning tasks ranging from manifold learning to image segmentation (Platt, 2004; Fowlkes et al., 2004; Talwalkar et al., 2008). A closely related algorithm, known as the Incomplete Cholesky Decomposition (Fine and Scheinberg, 2002; Bach and Jordan, 2002, 2005), can also be viewed as a specific sampling technique associated with the Nyström method (Bach and Jordan, 2005). As noted by Candès and Recht (2009) and Talwalkar and Rostamizadeh (2010), the Nyström approximation is related to the problem of matrix completion (Candès and Recht, 2009; Candès and Tao, 2009), which attempts to complete a low-rank matrix from a random sample of its entries. However, the matrix completion attempts to impute a low-rank matrix from a subset of (possibly perturbed) matrix entries, rather than a subset of matrix columns. This problem is related to, yet distinct from the Nyström method and sampling-based low-rank approximation algorithms in general, that deal with full-rank matrices that are amenable to low-rank approximation. Furthermore, when we have access to the underlying kernel function that generates the kernel matrix of interest, we can generate matrix entries on-the-fly as desired, providing us with more flexibility accessing the original matrix.

3. Fixed Sampling

Since the Nyström method operates on a small subset of $\mathbf{K}$, that is, $\mathbf{C}$, the selection of columns can significantly influence the accuracy of the approximation. In the remainder of the paper, we will discuss various sampling options that aim to select informative columns from $\mathbf{K}$. We begin with the most common class of sampling techniques that select columns using a fixed probability distribution. The most basic sampling technique involves *uniform* sampling of the columns. Alternatively, the *i*th column can be sampled non-uniformly with weight proportional to either its corresponding diagonal element $\mathbf{K}_{ii}$ (*diagonal sampling*) or the L_2 norm of the column (*column-norm sampling*) (Drineas et al., 2006; Drineas and Mahoney, 2005). There are additional computational costs associated with these non-uniform sampling methods: $O(n)$ time and space requirements for diagonal sampling and $O(n^2)$ time and space for column-norm sampling. These non-uniform sampling techniques are often presented using sampling with replacement to simplify theoretical analysis. Column-norm sampling has been used to analyze a general SVD approximation algorithm. Further, diagonal sampling with replacement was used by Drineas and Mahoney (2005) and Belabbas and Wolfe (2009) to bound the reconstruction error of the Nyström method.[2] In Drineas and Mahoney (2005) however, the authors suggest that column-norm sampling would be a better sampling assumption for the analysis of the Nyström method. We also note that Belabbas and Wolfe (2009) proposed a family of 'annealed determinantal' distributions for which multiplicative bounds on reconstruction error were derived. However, in practice, these distributions cannot be efficiently computed except for special cases coinciding with uniform and column-norm sampling. Similarly, although Mahoney and Drineas (2009) present multiplicative bounds for the CUR decomposition (which is quite similar to the Nyström method) when sampling from a distribution over the columns based on 'leverage scores,' these scores cannot be efficiently computed in practice for large-scale applications.

In the remainder of this section we present novel experimental results comparing the performance of these fixed sampling methods on several data sets. Previous studies have compared uniform and non-uniform in a more restrictive setting, using fewer types of kernels and focusing only on column-norm sampling (Drineas et al., 2001; Zhang et al., 2008). However, in this work, we provide the first comparison that includes diagonal sampling, which is the non-uniform distribution that is most scalable for large-scale applications and which has been used in some theoretical analyses of the Nyström method.

3.1 Data Sets

We used 5 data sets from a variety of applications, for example, computer vision and biology, as described in Table 1. SPSD kernel matrices were generated by mean centering the data sets and applying either a linear kernel or RBF kernel. The diagonals (respectively column norms) of these kernel matrices were used to calculate diagonal (respectively column-norm) distributions. Note that the diagonal distribution equals the uniform distribution for RBF kernels since diagonal entries of RBF kernel matrices always equal one.

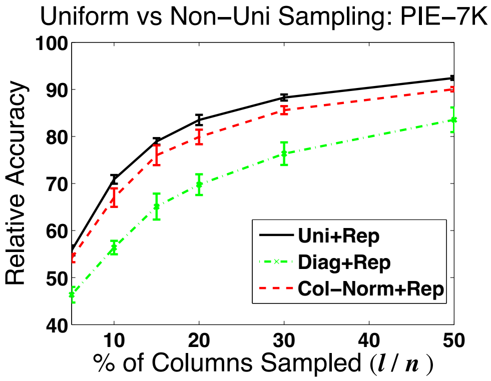
3.2 Experiments

We used the data sets described in the previous section to test the approximation accuracy for each sampling method. Low-rank approximations of $\mathbf{K}$ were generated using the Nyström method along with these sampling methods, and we measured the accuracy of reconstruction relative to the optimal

[2] Although Drineas and Mahoney (2005) claim to weight each column proportionally to $\mathbf{K}_{ii}^2$, they in fact use the diagonal sampling we present in this work, that is, weights proportional to $\mathbf{K}_{ii}$ (Drineas, 2008).

Name	Type	n	d	Kernel
PIE-2.7K	faces (profile)	2731	2304	linear
PIE-7K	faces (front)	7412	2304	linear
MNIST	digit images	4000	784	linear
ESS	proteins	4728	16	RBF
ABN	abalones	4177	8	RBF

Table 1: Description of the data sets and kernels used in fixed and adaptive sampling experiments (Sim et al., 2002; LeCun and Cortes, 1998; Gustafson et al., 2006; Asuncion and Newman, 2007). 'd' denotes the number of features in input space.



(a)

l/n	Data Set	Uniform+Rep	Diag+Rep	Col-Norm+Rep
5%	PIE-2.7K	**38.8** (**±1.5**)	38.3 (±0.9)	37.0 (±0.9)
	PIE-7K	**55.8** (**±1.1**)	46.4 (±1.7)	54.2 (±0.9)
	MNIST	**47.4** (**±0.8**)	46.9 (±0.7)	45.6 (±1.0)
	ESS	**45.1** (**±2.3**)	-	41.0 (±2.2)
	ABN	**47.3** (**±3.9**)	-	44.2 (±1.2)
20%	PIE-2.7K	**72.3** (**±0.9**)	65.0 (±0.9)	63.4 (±1.4)
	PIE-7K	**83.5** (**±1.1**)	69.8 (±2.2)	79.9 (±1.6)
	MNIST	**80.8** (**±0.5**)	79.4 (±0.5)	78.1 (±0.5)
	ESS	**80.1** (**±0.7**)	-	75.5 (±1.1)
	ABN	**77.1** (**±3.0**)	-	66.3 (±4.0)

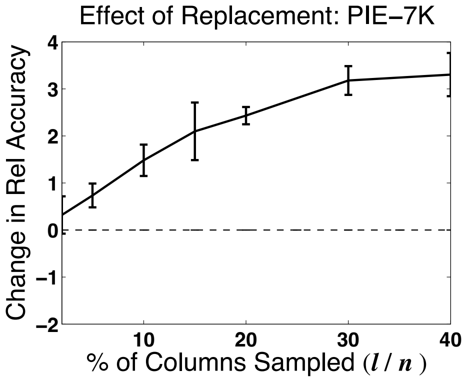
(b)

Figure 1: (a) Nyström relative accuracy for various sampling techniques on PIE-7K. (b) Nyström relative accuracy for various sampling methods for two values of l/n with $k = 100$. Values in parentheses show standard deviations for 10 different runs for a fixed l. '+Rep' denotes sampling with replacement. No error ('-') is reported for diagonal sampling with RBF kernels since diagonal sampling is equivalent to uniform sampling in this case.

rank-k approximation, $\mathbf{K}_k$, as:

$$\text{relative accuracy} = \frac{\|\mathbf{K}-\mathbf{K}_k\|_F}{\|\mathbf{K}-\widetilde{\mathbf{K}}_k\|_F} \times 100. \tag{4}$$

Note that the relative accuracy is lower bounded by zero and will approach one for good approximations. We fixed $k=100$ for all experiments, a value that captures more than 90% of the spectral energy for each data set. We first compared the effectiveness of the three sampling techniques using sampling with replacement. The results for PIE-7K are presented in Figure 1(a) and summarized for all data sets in Figure 1(b). The results across all data sets show that uniform sampling outperforms all other methods, while being much cheaper computationally and space-wise. Thus, while non-uniform sampling techniques may be effective in extreme cases where a few columns of $\mathbf{K}$ dominate in terms of $\|\cdot\|_2$, this situation does not tend to arise with real-world data, where uniform sampling is most effective.



(a)

Data Set	5%	10%	15%	30%
PIE-2.7K	0.8 (±.6)	1.7 (±.3)	2.3 (±.9)	4.4 (±.4)
PIE-7K	0.7 (±.3)	1.5 (±.3)	2.1 (±.6)	3.2 (±.3)
MNIST	1.0 (±.5)	1.9 (±.6)	2.3 (±.4)	3.4 (±.4)
ESS	0.9 (±.9)	1.8 (±.9)	2.2 (±.6)	3.7 (±.7)
ABN	0.7 (±1.2)	1.3 (±1.8)	2.6 (±1.4)	4.5 (±1.1)

(b)

Figure 2: Comparison of uniform sampling with and without replacement measured by the difference in relative accuracy. (a) Improvement in relative accuracy for PIE-7K when sampling without replacement. (b) Improvement in relative accuracy when sampling without replacement across all data sets for various l/n percentages.

Next, we compared the performance of uniform sampling with and without replacement. Figure 2(a) illustrates the effect of replacement for the PIE-7K data set for different l/n ratios. Similar results for the remaining data sets are summarized in Figure 2(b). The results show that uniform sampling without replacement improves the accuracy of the Nyström method over sampling with replacement, even when sampling less than 5% of the total columns. In summary, these experimental show that uniform sampling without replacement is the cheapest and most efficient sampling technique across several data sets (it is also the most commonly used method in practice). In Section 6, we present a theoretical analysis of the Nyström method using precisely this type of sampling.

4. Adaptive Sampling

In Section 3, we focused on fixed sampling schemes to create low-rank approximations. In this section, we discuss various sampling options that aim to select more informative columns from $\mathbf{K}$, while storing and operating on only $\mathrm{O}(ln)$ entries of $\mathbf{K}$. The Sparse Matrix Greedy Approximation (SMGA) (Smola and Schölkopf, 2000) and the Incomplete Cholesky Decomposition (ICL) (Fine and Scheinberg, 2002; Bach and Jordan, 2002) were the first such adaptive schemes suggested for the Nyström method. SMGA is a matching-pursuit algorithm that randomly selects a new sample at each round from a random subset of $s \ll n$ samples, with $s = 59$ in practice as per the suggestion of Smola and Schölkopf (2000). The runtime to select l columns is $\mathrm{O}(sl^2n)$, which is of the same order as the Nyström method itself when s is a constant and $k = l$ (see Section 2.1 for details).

Whereas SMGA was proposed as a sampling scheme to be used in conjunction with the Nyström method, ICL generates a low-rank factorization of $\mathbf{K}$ on-the-fly as it adaptively selects columns based on potential pivots of the Incomplete Cholesky Decomposition. ICL is a greedy, deterministic selection process that generates an approximation of the form $\widetilde{\mathbf{K}}^{icl} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top$ where $\widetilde{\mathbf{X}} \in \mathbb{R}^{n \times l}$ is low-rank. The runtime of ICL is $\mathrm{O}(l^2n)$. Although ICL does not generate an approximate SVD of $\mathbf{K}$, it does yield a low-rank approximation of $\mathbf{K}$ that can be used with the Woodbury approximation. Moreover, when $k = l$, the Nyström approximation generated from the l columns of $\mathbf{K}$ associated with the pivots selected by ICL is identical to $\widetilde{\mathbf{K}}^{icl}$ (Bach and Jordan, 2005). Related greedy adaptive sampling techniques were proposed by Ouimet and Bengio (2005) and Liu et al. (2006) in the contexts of spectral embedding and spectral mesh processing, respectively.

More recently, Zhang et al. (2008) and Zhang and Kwok (2009) proposed a technique to generate informative columns using centroids resulting from K-means clustering, with $K = l$. This algorithm, which uses out-of-sample extensions to generate a set of l representative columns of $\mathbf{K}$, has been shown to give good empirical accuracy (Zhang et al., 2008). Finally, an adaptive sampling technique with strong theoretical foundations (*adaptive-full*) was proposed in Deshpande et al. (2006). It requires a full pass through $\mathbf{K}$ in each iteration and is thus inefficient for large $\mathbf{K}$. In the remainder of this section, we first propose a novel adaptive technique that extends the ideas of Deshpande et al. (2006) and then present empirical results comparing the performance of this new algorithm with uniform sampling as well as SMGA, ICL, K-means and the *adaptive-full* techniques.

4.1 Adaptive Nyström Sampling

Instead of sampling all l columns from a fixed distribution, adaptive sampling alternates between selecting a set of columns and updating the distribution over all the columns. Starting with an initial distribution over the columns, $s < l$ columns are chosen to form a submatrix $\mathbf{C}'$. The probabilities are then updated as a function of previously chosen columns and s new columns are sampled and incorporated in $\mathbf{C}'$. This process is repeated until l columns have been selected. The adaptive sampling scheme in Deshpande et al. (2006) is detailed in Figure 3. Note that the sampling step, UPDATE-PROBABILITY-FULL, requires a full pass over $\mathbf{K}$ at each step, and hence $\mathrm{O}(n^2)$ time and space.

Input: $n \times n$ SPSD matrix ($\mathbf{K}$), number columns to be chosen (l), initial distribution over columns (P_0), number columns selected at each iteration (s)
Output: l indices corresponding to columns of $\mathbf{K}$

```
SAMPLE-ADAPTIVE($\mathbf{K}, n, l, P_0, s$)
$R \leftarrow$ set of $s$ indices sampled according to $P_0$
$t \leftarrow \frac{l}{s} - 1$ $\triangleright$ number of iterations
for $i \in [1 \ldots t]$ do
    $P_i \leftarrow$ UPDATE-PROBABILITY-FULL($R$)
    $R_i \leftarrow$ set of $s$ indices sampled according to $P_i$
    $R \leftarrow R \cup R_i$
return $R$
```

```
UPDATE-PROBABILITY-FULL($R$)
$\mathbf{C}' \leftarrow$ columns of $\mathbf{K}$ corresponding to indices in $R$
$\mathbf{U}_{C'} \leftarrow$ left singular vectors of $\mathbf{C}'$
$\mathbf{E} \leftarrow \mathbf{K} - \mathbf{U}_{C'}\mathbf{U}_{C'}^{\top}\mathbf{K}$
for $j \in [1 \ldots n]$ do
    if $j \in R$ then
        $P_j \leftarrow 0$
    else $P_j \leftarrow ||E_j||_2^2$
$P \leftarrow \frac{P}{||P||_2}$
return $P$
```

Figure 3: The adaptive sampling technique (Deshpande et al., 2006) that operates on the entire matrix $\mathbf{K}$ to compute the probability distribution over columns at each adaptive step.

We propose a simple sampling technique (*adaptive-partial*) that incorporates the advantages of adaptive sampling while avoiding the computational and storage burdens of the *adaptive-full* technique. At each iterative step, we measure the reconstruction error for each *row* of $\mathbf{C}'$ and the distribution over corresponding *columns* of $\mathbf{K}$ is updated proportional to this error. We compute the error for $\mathbf{C}'$, which is much smaller than $\mathbf{K}$, thus avoiding the $O(n^2)$ computation. As described in (3), if k' is fixed to be the number of columns in $\mathbf{C}'$, it will lead to $\mathbf{C}'_{nys} = \mathbf{C}'$ resulting in perfect reconstruction of $\mathbf{C}'$. So, one must choose a smaller k' to generate non-zero reconstruction errors from which probabilities can be updated (we used $k' = (\#\text{ columns in }\mathbf{C}')/2$ in our experiments). One artifact of using a k' smaller than the rank of $\mathbf{C}'$ is that all the columns of $\mathbf{K}$ will have a non-zero probability of being selected, which could lead to the selection of previously selected columns in the next iteration. However, sampling *without* replacement strategy alleviates this problem. Working with $\mathbf{C}'$ instead of $\mathbf{K}$ to iteratively compute errors makes this algorithm significantly more efficient than that of Deshpande et al. (2006), as each iteration takes $O(nlk' + l^3)$ time and requires at most the storage of l columns of $\mathbf{K}$. The details of the proposed sampling technique are outlined in Figure 4.

```
UPDATE-PROBABILITY-PARTIAL($R$)
    $\mathbf{C}' \leftarrow$ columns of $\mathbf{K}$ corresponding to indices in $R$
    $k' \leftarrow$ CHOOSE-RANK() $\triangleright$ low-rank ($k$) or $\frac{|R|}{2}$
    $\widetilde{\mathbf{\Sigma}}^{nys}_{k'}, \widetilde{\mathbf{U}}^{nys}_{k'} \leftarrow$ DO-NYSTRÖM $(\mathbf{C}', k')$ $\triangleright$ see Equation (2)
    $\mathbf{C}'_{nys} \leftarrow$ Spectral reconstruction using $\widetilde{\mathbf{\Sigma}}^{nys}_{k'}, \widetilde{\mathbf{U}}^{nys}_{k'}$
    $\mathbf{E} \leftarrow \mathbf{C}' - \mathbf{C}'_{nys}$
    for $j \in [1 \ldots n]$ do
        if $j \in R$ then
            $P_j \leftarrow 0$ $\triangleright$ sample without replacement
        else $P_j \leftarrow ||\mathbf{E}_{(j)}||_2^2$
    $P \leftarrow \frac{P}{||P||_2}$
    return $P$
```

Figure 4: The proposed adaptive sampling technique that uses a small subset of the original matrix **K** to adaptively choose columns. It does not need to store or operate on **K**.

$l/n\%$	Data Set	Uniform	ICL	SMGA	Adapt-Part	K-means	Adapt-Full
5%	PIE-2.7K	39.7 (0.7)	41.6 (0.0)	54.4 (0.6)	42.6 (0.8)	**61.3** (0.5)	44.2 (0.9)
	PIE-7K	58.6 (1.0)	50.1 (0.0)	68.1 (0.9)	61.4 (1.1)	**71.0** (0.7)	-
	MNIST	47.5 (0.9)	41.5 (0.0)	59.2 (0.5)	49.7 (0.9)	**72.9** (0.9)	50.3 (0.7)
	ESS	45.7 (2.6)	25.2 (0.0)	61.9 (0.5)	49.3 (1.5)	**64.2** (1.6)	-
	ABN	47.4 (5.5)	15.6 (0.0)	64.9 (1.8)	23.0 (2.8)	**65.7** (5.8)	50.7 (2.4)
10%	PIE-2.7K	58.2 (1.0)	61.1 (0.0)	72.7 (0.2)	60.8 (1.0)	**73.0** (1.1)	63.0 (0.3)
	PIE-7K	72.4 (0.7)	60.8 (0.0)	74.5 (0.6)	77.0 (0.6)	**82.8** (0.7)	-
	MNIST	66.8 (1.4)	58.3 (0.0)	72.2 (0.8)	69.3 (0.6)	**81.6** (0.6)	68.5 (0.5)
	ESS	66.8 (2.0)	39.1 (0.0)	74.7 (0.5)	70.0 (1.0)	**81.6** (1.0)	-
	ABN	61.0 (1.1)	25.8 (0.0)	67.1 (0.9)	33.6 (6.7)	**79.8** (0.9)	57.9 (3.9)
20%	PIE-2.7K	75.2 (1.0)	80.5 (0.0)	**86.1** (0.2)	78.7 (0.5)	85.5 (0.5)	80.6 (0.4)
	PIE-7K	85.6 (0.9)	69.5 (0.0)	79.4 (0.5)	86.2 (0.3)	**91.9** (0.3)	-
	MNIST	83.6 (0.4)	77.9 (0.0)	78.7 (0.2)	84.0 (0.6)	**88.4** (0.5)	80.4 (0.5)
	ESS	81.4 (2.1)	55.3 (0.0)	79.4 (0.7)	83.4 (0.3)	**90.0** (0.6)	-
	ABN	80.8 (1.7)	41.2 (0.0)	67.2 (2.2)	44.4 (6.7)	**85.1** (1.6)	62.4 (3.6)

Table 2: Nyström spectral reconstruction accuracy for various sampling methods for all data sets for $k = 100$ and three l/n percentages. Numbers in parenthesis indicate the standard deviations for 10 different runs for each l. Numbers in bold indicate the best performance on each data set, that is, each row of the table. Dashes ('-') indicate experiments that were too costly to run on the larger data sets (ESS, PIE-7K).

4.2 Experiments

We used the data sets in Table 1, and compared the effect of different sampling techniques on the relative accuracy of Nyström spectral reconstruction for $k = 100$. All experiments were conducted

$l/n\%$	Data Set	Uniform	ICL	SMGA	Adapt-Part	K-means	Adapt-Full
5%	PIE-2.7K	0.03	0.56	2.30	0.43	2.44	22.54
	PIE-7K	0.63	44.04	59.02	6.56	15.18	-
	MNIST	0.04	1.71	7.57	0.71	1.26	20.56
	ESS	0.07	2.87	62.42	0.85	3.48	-
	ABN	0.06	3.28	9.26	0.66	2.44	28.49
10%	PIE-2.7K	0.08	2.81	8.44	0.97	3.25	23.13
	PIE-7K	0.63	44.04	244.33	6.56	15.18	-
	MNIST	0.20	7.38	28.79	1.51	1.82	21.77
	ESS	0.29	11.01	152.30	2.04	7.16	-
	ABN	0.23	10.92	33.30	1.74	4.94	35.91
20%	PIE-2.7K	0.28	8.36	38.19	2.63	5.91	27.72
	PIE-7K	0.81	141.13	1107.32	13.80	12.08	-
	MNIST	0.46	16.99	51.96	4.03	2.91	26.53
	ESS	0.52	34.28	458.23	5.90	14.68	-
	ABN	1.01	38.36	199.43	8.54	12.56	97.39

Table 3: Run times (in seconds) corresponding to Nyström spectral reconstruction results in Table 2. Dashes ('-') indicate experiments that were too costly to run on the larger data sets (ESS, PIE-7K).

in Matlab on an x86 – 64 architecture using a single 2.4 Ghz core and 30GB of main memory. We used an implementation of ICL from Cawley and Talbot (2004) and an implementation of SMGA code from Smola (2000), using default parameters as set by these implementations. We wrote our own implementation of the K-means method using 5 iterations of K-means and employing an efficient (vectorized) function to compute L_2 distances between points and centroids at each iteration (Bunschoten, 1999).[3] Moreover, we used a random projection SVD solver to compute truncated SVD, using code by Tygert (2009).

The relative accuracy results across data sets for varying values of l are presented in Table 2, while the corresponding timing results are detailed in Table 3. The K-means algorithm was clearly the best performing adaptive algorithm, generating the most accurate approximations in almost all settings in roughly the same amount of time (or less) as other adaptive algorithms. Moreover, the proposed Nyström adaptive technique, which is a natural extension of an important algorithm introduced in the theory community, has performance similar to this original algorithm at a fraction of the cost, but it is nonetheless outperformed by the K-means algorithm. We further note that ICL performs the worst of all the adaptive techniques, and it is often worse than random sampling (this observation is also noted by Zhang et al. 2008).

The empirical results also suggest that the performance gain due to adaptive sampling is inversely proportional to the percentage of sampled columns—random sampling actually outperforms many of the adaptive approaches when sampling 20% of the columns. These empirical results suggest a trade-off between time and space requirements, as noted by Schölkopf and Smola (2002)[Chapter 10.2]. Adaptive techniques spend more time to find a concise subset of informative columns, but as in the case of the K-means algorithm, can provide improved approximation accuracy.

[3] Note that Matlab's built-in K-means function is quite inefficient.

5. Ensemble Sampling

In this section, we slightly shift focus, and discuss a meta algorithm called the *ensemble Nyström algorithm*. We treat each approximation generated by the Nyström method for a sample of l columns as an *expert* and combine $p \geq 1$ such experts to derive an improved hypothesis, typically more accurate than any of the original experts.

The learning set-up is defined as follows. We assume a fixed kernel function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that can be used to generate the entries of a kernel matrix $\mathbf{K}$. The learner receives a set S of lp columns randomly selected from matrix $\mathbf{K}$ uniformly without replacement. S is decomposed into p subsets $S_1, \ldots, S_p$. Each subset S_r, $r \in [1, p]$, contains l columns and is used to define a rank-k Nyström approximation $\widetilde{\mathbf{K}}_r$.[4] Dropping the rank subscript k in favor of the sample index r, $\widetilde{\mathbf{K}}_r$ can be written as $\widetilde{\mathbf{K}}_r = \mathbf{C}_r \mathbf{W}_r^+ \mathbf{C}_r^\top$, where $\mathbf{C}_r$ and $\mathbf{W}_r$ denote the matrices formed from the columns of S_r and $\mathbf{W}_r^+$ is the pseudo-inverse of the rank-k approximation of $\mathbf{W}_r$. The learner further receives a sample V of s columns used to determine the weight $\mu_r \in \mathbb{R}$ attributed to each expert $\widetilde{\mathbf{K}}_r$. Thus, the general form of the approximation, $\mathbf{K}^{ens}$, generated by the ensemble Nyström algorithm, with $k \leq \text{rank}(\mathbf{K}^{ens}) \leq pk$, is

$$\begin{aligned}
\widetilde{\mathbf{K}}^{ens} &= \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r \\
&= \begin{bmatrix} \mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_p \end{bmatrix} \begin{bmatrix} \mu_1 \mathbf{W}_1^+ & & \\ & \ddots & \\ & & \mu_p \mathbf{W}_p^+ \end{bmatrix} \begin{bmatrix} \mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_p \end{bmatrix}^\top .
\end{aligned} \tag{5}$$

As noted by Li et al. (2010), (5) provides an alternative description of the ensemble Nyström method as a block diagonal approximation of $\mathbf{W}_{ens}^+$, where $\mathbf{W}_{ens}$ is the $lp \times lp$ SPSD matrix associated with the lp sampled columns. Moreover, Li et al. (2010) further argues that computing $\mathbf{W}_{ens}^+$ would be preferable to making this block diagonal approximation and subsequently uses a random projection SVD solver to speed up computation of $\mathbf{W}_{ens}^+$ (Halko et al., 2009). However, this analysis is misleading as these two orthogonal approaches should not be viewed as competing methods. Rather, one can always use the ensemble based approach *along with* fast SVD solvers. This approach is most natural to improve performance on large-scale problems, and is precisely the approach we adopt in our experiments.

The mixture weights μ_r can be defined in many ways. The most straightforward choice consists of assigning equal weight to each expert, $\mu_r = 1/p$, $r \in [1, p]$. This choice does not require the additional sample V, but it ignores the relative quality of each Nyström approximation. Nevertheless, this simple *uniform method* already generates a solution superior to any one of the approximations $\widetilde{\mathbf{K}}_r$ used in the combination, as we shall see in the experimental section.

Another method, the *exponential weight method*, consists of measuring the reconstruction error $\hat{\varepsilon}_r$ of each expert $\widetilde{\mathbf{K}}_r$ over the validation sample V and defining the mixture weight as $\mu_r = \exp(-\eta \hat{\varepsilon}_r)/Z$, where $\eta > 0$ is a parameter of the algorithm and Z a normalization factor ensuring that the vector $\mu = (\mu_1, \ldots, \mu_p)$ belongs to the unit simplex Δ of $\mathbb{R}^p$: $\Delta = \{\mu \in \mathbb{R}^p : \mu \geq 0 \wedge \sum_{r=1}^{p} \mu_r = 1\}$. The choice of the mixture weights here is similar to those used in the Weighted Majority algorithm

[4] In this study, we focus on the class of base learners generated from Nyström approximation with uniform sampling of columns or from the adaptive K-means method. Alternatively, these base learners could be generated using other (or a combination of) sampling schemes discussed in Sections 3 and 4.

(Littlestone and Warmuth, 1994). Let $\mathbf{K}_V$ denote the matrix formed by using the samples from V as its columns and let $\widetilde{\mathbf{K}}_r^V$ denote the submatrix of $\widetilde{\mathbf{K}}_r$ containing the columns corresponding to the columns in V. The reconstruction error $\hat{\varepsilon}_r = \|\widetilde{\mathbf{K}}_r^V - \mathbf{K}_V\|$ can be directly computed from these matrices.

A more general class of methods consists of using the sample V to train the mixture weights μ_r to optimize a regression objective function such as the following:

$$\min_{\mu} \lambda \|\mu\|_2^2 + \| \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r^V - \mathbf{K}_V \|_F^2,$$

where $\lambda > 0$. This can be viewed as a ridge regression objective function and admits a closed form solution. We will refer to this method as the *ridge regression method*. Note that to ensure that the resulting matrix is SPSD for use in subsequent kernel-based algorithms, the optimization problem must be augmented with standard non-negativity constraints. This is not necessary however for reducing the reconstruction error, as in our experiments. Also, clearly, a variety of other regression algorithms such as Lasso can be used here instead.

The total complexity of the ensemble Nyström algorithm is $O(pl^3 + plkn + C_\mu)$, where C_μ is the cost of computing the mixture weights, μ, used to combine the p Nyström approximations. The mixture weights can be computed in constant time for the uniform method, in $O(psn)$ for the exponential weight method, or in $O(p^3 + p^2 ns)$ for the ridge regression method where $O(p^2 ns)$ time is required to compute a $p \times p$ matrix and $O(p^3)$ time is required for inverting this matrix. Furthermore, although the ensemble Nyström algorithm requires p times more space and CPU cycles than the standard Nyström method, these additional requirements are quite reasonable in practice. The space requirement is still manageable for even large-scale applications given that p is typically O(1) and l is usually a very small percentage of n (see Section 5.2 for further details). In terms of CPU requirements, we note that the algorithm can be easily parallelized, as all p experts can be computed simultaneously. Thus, with a cluster of p machines, the running time complexity of this algorithm is nearly equal to that of the standard Nyström algorithm with l samples.

5.1 Ensemble Woodbury Approximation

The Woodbury approximation is a useful tool to use alongside low-rank approximations to efficiently (and approximately) invert kernel matrices. We are able to apply the Woodbury approximation since the Nyström method represents $\widetilde{\mathbf{K}}$ as the product of low-rank matrices. This is clear from the definition of the Woodbury approximation:

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}, \tag{6}$$

where $\mathbf{A} = \lambda \mathbf{I}$ and $\widetilde{\mathbf{K}} = \mathbf{BCD}$ in the context of the Nyström method. In contrast, the ensemble Nyström method represents $\widetilde{\mathbf{K}}$ as the sum of products of low-rank matrices, where each of the p terms corresponds to a base learner. Hence, we cannot directly apply the Woodbury approximation as presented above. There is however, a natural extension of the Woodbury approximation in this setting, which at the simplest level involves running the approximation p times. Starting with p base learners with their associated weights, that is, $\widetilde{\mathbf{K}}_r$ and μ_r for $r \in [1, p]$, and defining $\mathbf{T}_0 = \lambda \mathbf{I}$, we perform the following series of calculations:

$$\begin{aligned}
\mathbf{T}_1^{-1} &= (\mathbf{T}_0 + \mu_1 \widetilde{\mathbf{K}}_1)^{-1}, \\
\mathbf{T}_2^{-1} &= (\mathbf{T}_1 + \mu_2 \widetilde{\mathbf{K}}_2)^{-1}, \\
&\cdots \\
\mathbf{T}_p^{-1} &= (\mathbf{T}_{p-1} + \mu_p \widetilde{\mathbf{K}}_p)^{-1}.
\end{aligned}$$

To compute $\mathbf{T}_1^{-1}$, notice that we can use Woodbury approximation as stated in (6) since we can express $\mu_1 \widetilde{\mathbf{K}}_1$ as the product of low-rank matrices and we know that $T_0^{-1} = \frac{1}{\lambda}\mathbf{I}$. More generally, for $1 \le i \le p$, given an expression of T_{i-1}^{-1} as a product of low-rank matrices, we can efficiently compute T_i^{-1} using the Woodbury approximation (we use the low-rank structure to avoid ever computing or storing a full $n \times n$ matrix). Hence, after performing this series of p calculations, we are left with the inverse of $\mathbf{T}_p$, which is exactly the quantity of interest since $\mathbf{T}_p = \lambda \mathbf{I} + \sum_{r=1}^{p} \mu_r \widetilde{\mathbf{K}}_r$. Although this algorithm requires p iterations of the Woodbury approximation, these iterations can be parallelized in a tree-like fashion. Hence, when working on a cluster, using an ensemble Nyström approximation along with the Woodbury approximation requires only a $\log_2(p)$ factor more time than using the standard Nyström method.[5]

5.2 Experiments

In this section, we present experimental results that illustrate the performance of the ensemble Nyström method. We again work with the data sets listed in Table 1, and compare the performance of various methods for calculating the mixture weights (μ_r). Throughout our experiments, we measure performance via relative accuracy (defined in (4)). For all experiments, we fixed the reduced rank to $k = 100$, and set the number of sampled columns to $l = 3\% \times n$.[6]

5.2.1 ENSEMBLE NYSTRÖM WITH VARIOUS MIXTURE WEIGHTS

We first show results for the ensemble Nyström method using different techniques to choose the mixture weights, as previously discussed. In these experiments, we focused on base learners generated via the Nyström method with uniform sampling of columns. Furthermore, for the exponential and the ridge regression variants, we sampled a set of $s = 20$ columns and used an additional 20 columns (s') as a hold-out set for selecting the optimal values of η and λ. The number of approximations, p, was varied from 2 to 25. As a baseline, we also measured the maximum relative accuracy across the p Nyström approximations used to construct $\widetilde{\mathbf{K}}^{ens}$. We also calculated the performance when using the optimal μ, that is, we used least-square regression to find the best possible choice of combination weights for a fixed set of p approximations by setting $s = n$. The results of these experiments are presented in Figure 5.[7] These results clearly show that the ensemble Nyström performance is significantly better than any of the individual Nyström approximations. We further note that the ensemble Nyström method tends to converge very quickly, and the most significant gain in performance occurs as p increases from 2 to 10.

[5] Note that we can also efficiently obtain singular values and singular vectors of the low-rank matrix $\mathbf{K}^{ens}$ using coherence-based arguments, as in Talwalkar and Rostamizadeh (2010).

[6] Similar results (not reported here) were observed for other values of k and l as well.

[7] Similar results (not reported here) were observed when measuring relative accuracy using the spectral norm instead of the Frobenium norm.

Base Learner	Method	PIE-2.7K	PIE-7K	MNIST	ESS	ABN
Uniform	Average Base Learner	26.9	46.3	34.2	30.0	38.1
	Best Base Learner	29.2	48.3	36.1	34.5	43.6
	Ensemble Uniform	33.0	57.5	47.3	43.9	49.8
	Ensemble Exponential	33.0	57.5	47.4	43.9	49.8
	Ensemble Ridge	35.0	58.5	54.0	44.5	53.6
K-means	Average Base Learner	47.6	62.9	62.5	42.2	60.6
	Best Base Learner	48.4	66.4	63.9	47.1	72.0
	Ensemble Uniform	**54.9**	71.3	76.9	52.2	76.4
	Ensemble Exponential	**54.9**	71.4	77.0	52.2	78.3
	Ensemble Ridge	**54.9**	**71.6**	**77.2**	**52.7**	**79.0**

Table 4: Relative accuracy for ensemble Nyström method with Nyström base learners generated with uniform sampling of columns or via the K-means algorithm.

5.2.2 EFFECT OF RANK

As mentioned earlier, the rank of the ensemble approximations can be p times greater than the rank of each of the base learners. Hence, to validate the results in Figure 5, we performed a simple experiment in which we compared the performance of the best base learner to the best rank k approximation of the uniform ensemble approximation (obtained via SVD of the uniform ensemble approximation). We again used base learners generated via the Nyström method with uniform sampling of columns. The results of this experiment, presented in Figure 6, suggest that the performance gain of the ensemble methods is not due to this increased rank.

5.2.3 EFFECT OF RIDGE

Figure 5 also shows that the ridge regression technique is the best of the proposed techniques, and generates nearly the optimal solution in terms of relative accuracy using the Frobenius norm. We also observed that when s is increased to approximately 5% to 10% of n, linear regression without any regularization performs about as well as ridge regression for both the Frobenius and spectral norm. Figure 7 shows this comparison between linear regression and ridge regression for varying values of s using a fixed number of experts ($p = 10$). In these experiments, we again used base learners generated via the Nyström method with uniform sampling of columns.

5.2.4 ENSEMBLE K-MEANS NYSTRÖM

In the previous experiments, we focused on base learners generated via the Nyström method with uniform sampling of columns. In light of the performance of the K-means algorithm in Section 4, we next explored the performance of this algorithm when used in conjunction with the ensemble Nyström method. We fixed the number of base learners to $p = 10$ and when using ridge regression to learn weights, we set $s = s' = 20$. As shown in Table 4, similar performance gains in comparison to the average or best base learner can be seen when using an ensemble of base learners derived from the K-means algorithm. Consistent with the experimental results of Section 4, the accuracy values are higher for K-means relative to uniform sampling, though as noted in the previous section, this increased performance comes with an added cost, as the K-means step is more expensive than random sampling.

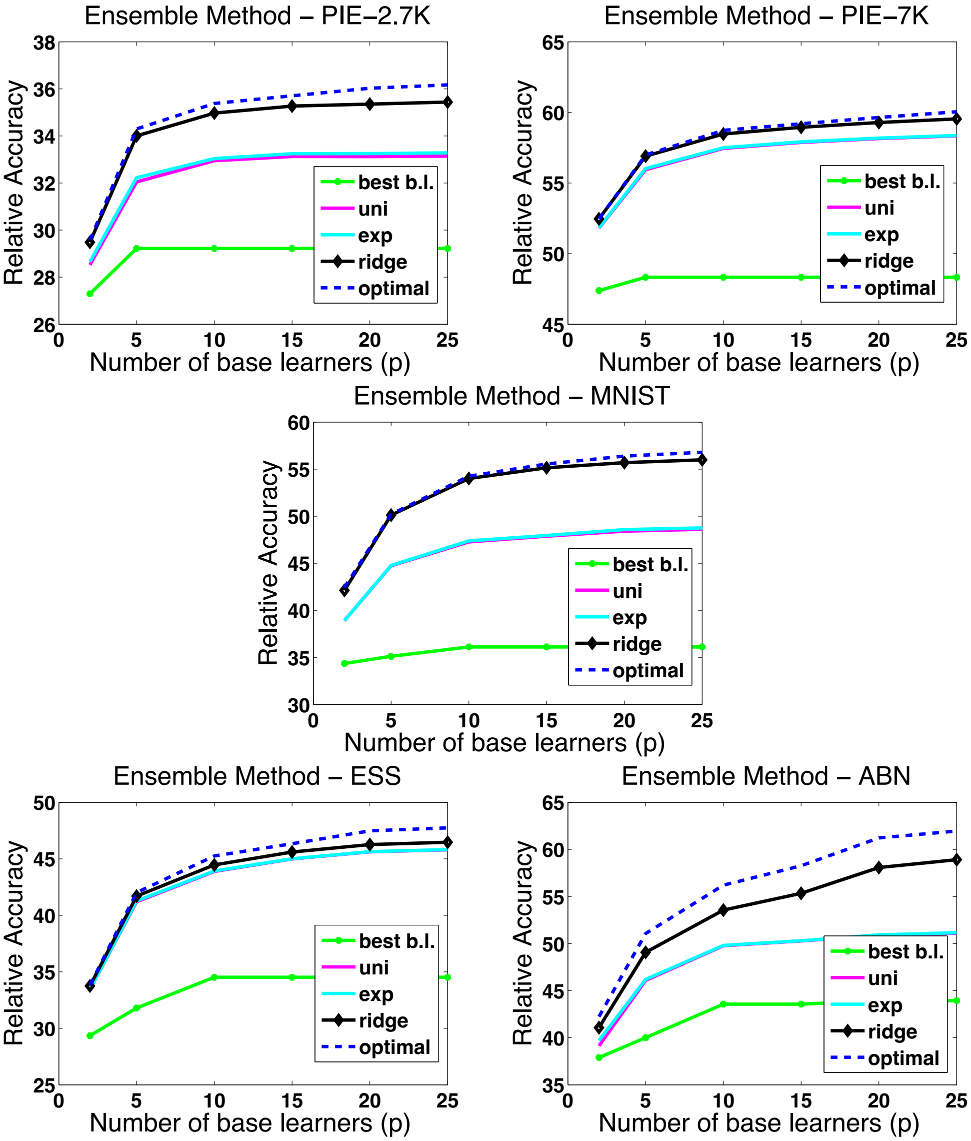


Figure 5: Relative accuracy for ensemble Nyström method using uniform ('uni'), exponential ('exp'), ridge ('ridge') and optimal ('optimal') mixture weights as well as the best ('best b.l.') of the p base learners used to create the ensemble approximations.

6. Theoretical Analysis

We now present theoretical results that compare the quality of the Nyström approximation to the 'best' low-rank approximation, that is, the approximation constructed from the top singular values and singular vectors of $\mathbf{K}$. This work, related to work by Drineas and Mahoney (2005), provides performance bounds for the Nyström method as it is often used in practice, that is, using uniform

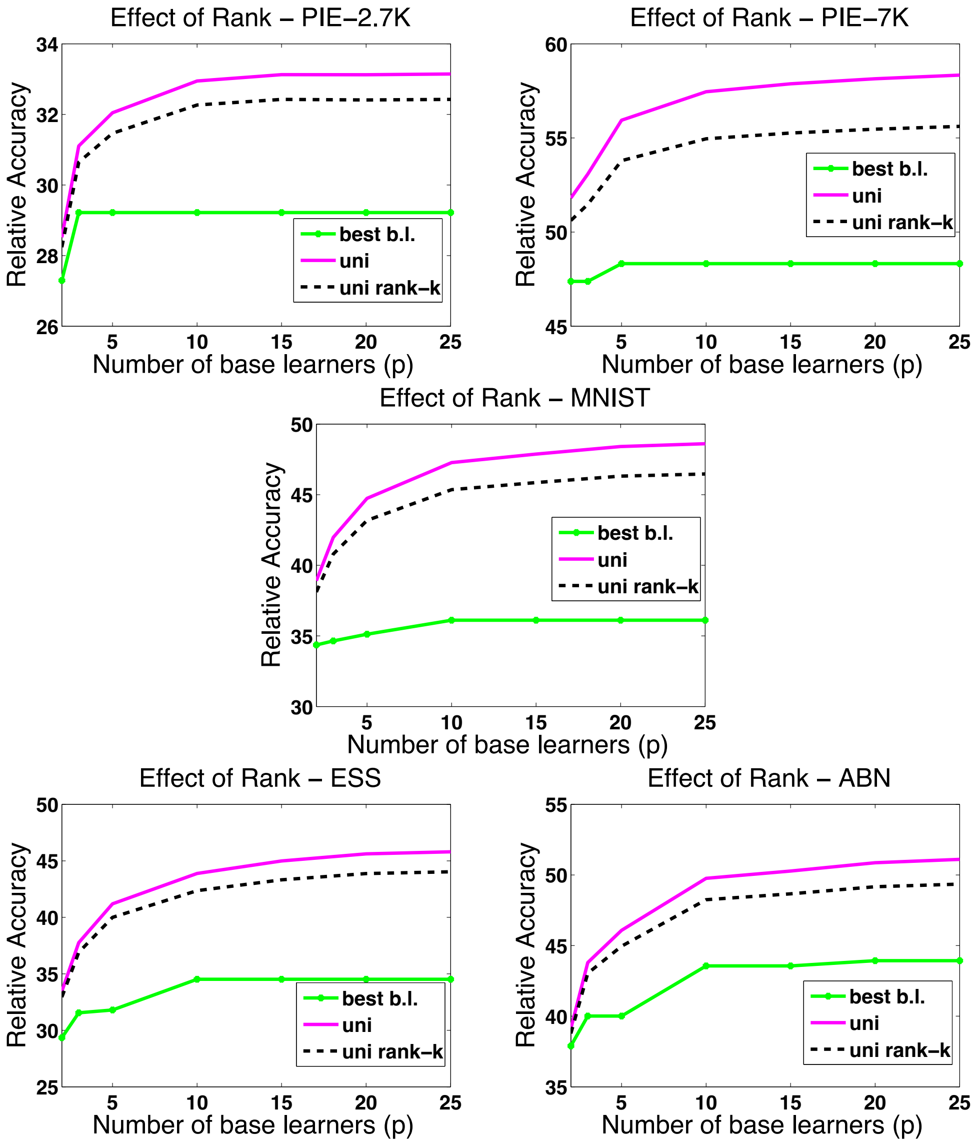
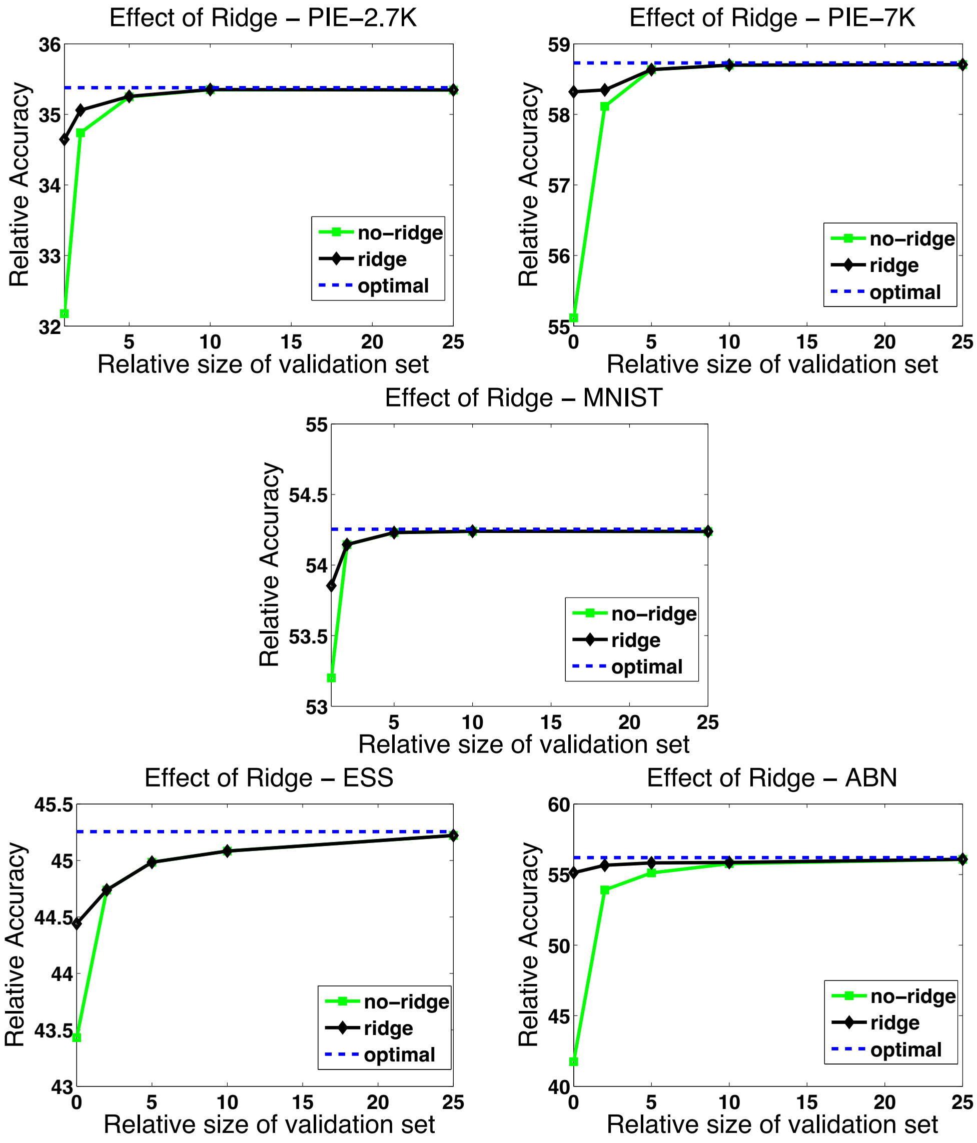


Figure 6: Relative accuracy for ensemble Nyström method using uniform ('uni') mixture weights, the optimal rank-k approximation of the uniform ensemble result ('uni rank-k') as well as the best ('best b.l.') of the p base learners used to create the ensemble approximations.

sampling without replacement, and holds for both the standard Nyström method as well as the ensemble Nyström method discussed in Section 5.

Our theoretical analysis of the Nyström method uses some results previously shown by Drineas and Mahoney (2005) as well as the following generalization of McDiarmid's concentration bound to sampling without replacement (Cortes et al., 2008).

Theorem 1 *Let* $Z_1, \ldots, Z_l$ *be a sequence of random variables sampled uniformly without replacement from a fixed set of* $l+u$ *elements* Z, *and let* $\phi\colon Z^l \to \mathbb{R}$ *be a symmetric function such that for all*



Effect of Ridge – PIE−2.7K

Effect of Ridge – PIE−7K

Effect of Ridge – MNIST

Effect of Ridge – ESS

Effect of Ridge – ABN

Figure 7: Comparison of relative accuracy for the ensemble Nyström method with $p = 10$ experts with weights derived from linear ('no-ridge') and ridge ('ridge') regression. The dotted line indicates the optimal combination. The relative size of the validation set equals $s/n \times 100$.

$i \in [1,l]$ *and for all* $z_1,\ldots,z_l \in Z$ *and* $z'_1,\ldots,z'_l \in Z$, $|\phi(z_1,\ldots,z_l) - \phi(z_1,\ldots,z_{i-1},z'_i,z_{i+1},\ldots,z_l)| \le c$. *Then, for all* $\epsilon > 0$*, the following inequality holds:*

$$\Pr\left[\phi - \mathbf{E}[\phi] \ge \epsilon\right] \le \exp\left[\frac{-2\epsilon^2}{\alpha(l,u)c^2}\right],$$

where $\alpha(l,u) = \frac{lu}{l+u-1/2}\,\frac{1}{1-1/(2\max\{l,u\})}$.

We define the *selection matrix* corresponding to a sample of l columns as the matrix $\mathbf{S} \in \mathbb{R}^{n \times l}$ defined by $\mathbf{S}_{ii} = 1$ if the ith column of $\mathbf{K}$ is among those sampled, $\mathbf{S}_{ij} = 0$ otherwise. Thus, $\mathbf{C} = \mathbf{KS}$ is the matrix formed by the columns sampled. Since $\mathbf{K}$ is SPSD, there exists $\mathbf{X} \in \mathbb{R}^{N \times n}$ such that $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$. We shall denote by $\mathbf{K}_{\max}$ the maximum diagonal entry of $\mathbf{K}$, $\mathbf{K}_{\max} = \max_i \mathbf{K}_{ii}$, and by $d^{\mathbf{K}}_{\max}$ the distance $\max_{ij} \sqrt{\mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}}$.

6.1 Standard Nyström Method

The following theorem gives an upper bound on the norm-2 error of the Nyström approximation of the form $\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 / \|\mathbf{K}\|_2 \le \|\mathbf{K} - \mathbf{K}_k\|_2 / \|\mathbf{K}\|_2 + O(1/\sqrt{l})$ and an upper bound on the Frobenius error of the Nyström approximation of the form $\|\mathbf{K} - \widetilde{\mathbf{K}}\|_F / \|\mathbf{K}\|_F \le \|\mathbf{K} - \mathbf{K}_k\|_F / \|\mathbf{K}\|_F + O(1/l^{\frac{1}{4}})$.

Theorem 2 *Let $\widetilde{\mathbf{K}}$ denote the rank-k Nyström approximation of $\mathbf{K}$ based on l columns sampled uniformly at random without replacement from $\mathbf{K}$, and $\mathbf{K}_k$ the best rank-k approximation of $\mathbf{K}$. Then, with probability at least $1 - \delta$, the following inequalities hold for any sample of size l:*

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 \le \|\mathbf{K} - \mathbf{K}_k\|_2 + \tfrac{2n}{\sqrt{l}} \mathbf{K}_{\max} \left[1 + \sqrt{\tfrac{n-l}{n-1/2} \tfrac{1}{\beta(l,n)} \log \tfrac{1}{\delta}}\, d^{\mathbf{K}}_{\max} / \mathbf{K}^{\frac{1}{2}}_{\max} \right],$$

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_F \le \|\mathbf{K} - \mathbf{K}_k\|_F + \left[\tfrac{64k}{l}\right]^{\frac{1}{4}} n \mathbf{K}_{\max} \left[1 + \sqrt{\tfrac{n-l}{n-1/2} \tfrac{1}{\beta(l,n)} \log \tfrac{1}{\delta}}\, d^{\mathbf{K}}_{\max} / \mathbf{K}^{\frac{1}{2}}_{\max} \right]^{\frac{1}{2}},$$

where $\beta(l,n) = 1 - \frac{1}{2\max\{l, n-l\}}$.

Proof To bound the norm-2 error of the Nyström method in the scenario of sampling without replacement, we start with the following general inequality given by Drineas and Mahoney (2005)[Proof of Lemma 4]:

$$\|\mathbf{K} - \widetilde{\mathbf{K}}\|_2 \le \|\mathbf{K} - \mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2,$$

where $\mathbf{Z} = \sqrt{\frac{n}{l}} \mathbf{XS}$. We then apply the McDiarmid-type inequality of Theorem 1 to $\phi(\mathbf{S}) = \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2$. Let $\mathbf{S}'$ be a sampling matrix selecting the same columns as $\mathbf{S}$ except for one, and let $\mathbf{Z}'$ denote $\sqrt{\frac{n}{l}} \mathbf{XS}'$. Let $\mathbf{z}$ and $\mathbf{z}'$ denote the only differing columns of $\mathbf{Z}$ and $\mathbf{Z}'$, then

$$\begin{aligned} |\phi(\mathbf{S}') - \phi(\mathbf{S})| &\le \|\mathbf{z}'\mathbf{z}'^\top - \mathbf{z}\mathbf{z}^\top\|_2 = \|(\mathbf{z}' - \mathbf{z})\mathbf{z}'^\top + \mathbf{z}(\mathbf{z}' - \mathbf{z})^\top\|_2 \\ &\le 2\|\mathbf{z}' - \mathbf{z}\|_2 \max\{\|\mathbf{z}\|_2, \|\mathbf{z}'\|_2\}. \end{aligned}$$

Columns of $\mathbf{Z}$ are those of $\mathbf{X}$ scaled by $\sqrt{n/l}$. The norm of the difference of two columns of $\mathbf{X}$ can be viewed as the norm of the difference of two feature vectors associated to $\mathbf{K}$ and thus can be bounded by $d_{\mathbf{K}}$. Similarly, the norm of a single column of $\mathbf{X}$ is bounded by $\mathbf{K}^{\frac{1}{2}}_{\max}$. This leads to the following inequality:

$$|\phi(\mathbf{S}') - \phi(\mathbf{S})| \le \frac{2n}{l} d^{\mathbf{K}}_{\max} \mathbf{K}^{\frac{1}{2}}_{\max}. \tag{7}$$

The expectation of ϕ can be bounded as follows:

$$\mathbf{E}[\Phi] = \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2] \le \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F] \le \frac{n}{\sqrt{l}} \mathbf{K}_{\max}, \tag{8}$$

where the last inequality follows Corollary 2 of Kumar et al. (2009a). The inequalities (7) and (8) combined with Theorem 1 give a bound on $\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_2$ and yield the statement of the theorem.

The following general inequality holds for the Frobenius error of the Nyström method (Drineas and Mahoney, 2005):

$$\|\mathbf{K}-\widetilde{\mathbf{K}}\|_F^2 \le \|\mathbf{K}-\mathbf{K}_k\|_F^2 + \sqrt{64k}\,\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F^2\, n\mathbf{K}_{ii}^{\max}. \tag{9}$$

Bounding the term $\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}\mathbf{Z}^\top\|_F^2$ as in the norm-2 case and using the concentration bound of Theorem 1 yields the result of the theorem. ■

6.2 Ensemble Nyström Method

The following error bounds hold for ensemble Nyström methods based on a convex combination of Nyström approximations.

Theorem 3 *Let S be a sample of pl columns drawn uniformly at random without replacement from $\mathbf{K}$, decomposed into p subsamples of size l, $S_1,\ldots,S_p$. For $r\in[1,p]$, let $\widetilde{\mathbf{K}}_r$ denote the rank-k Nyström approximation of $\mathbf{K}$ based on the sample S_r, and let $\mathbf{K}_k$ denote the best rank-k approximation of $\mathbf{K}$. Then, with probability at least $1-\delta$, the following inequalities hold for any sample S of size pl and for any μ in the unit simplex Δ and $\widetilde{\mathbf{K}}^{ens}=\sum_{r=1}^p \mu_r\widetilde{\mathbf{K}}_r$:*

$$\begin{aligned}
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_2 &\le \|\mathbf{K}-\mathbf{K}_k\|_2 + \\
&\qquad \tfrac{2n}{\sqrt{l}}\mathbf{K}_{\max}\left[1+\mu_{\max}p^{\frac{1}{2}}\sqrt{\tfrac{n-pl}{n-1/2}\tfrac{1}{\beta(pl,n)}\log\tfrac{1}{\delta}}\; d_{\max}^{\mathbf{K}}/\mathbf{K}_{\max}^{\frac{1}{2}}\right],\\
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_F &\le \|\mathbf{K}-\mathbf{K}_k\|_F + \\
&\qquad \left[\tfrac{64k}{l}\right]^{\frac{1}{4}} n\mathbf{K}_{\max}\left[1+\mu_{\max}p^{\frac{1}{2}}\sqrt{\tfrac{n-pl}{n-1/2}\tfrac{1}{\beta(pl,n)}\log\tfrac{1}{\delta}}\; d_{\max}^{\mathbf{K}}/\mathbf{K}_{\max}^{\frac{1}{2}}\right]^{\frac{1}{2}},
\end{aligned}$$

where $\beta(pl,n)=1-\frac{1}{2\max\{pl,n-pl\}}$ and $\mu_{\max}=\max_{r=1}^p \mu_r$.

Proof For $r\in[1,p]$, let $\mathbf{Z}_r=\sqrt{n/l}\,\mathbf{X}\mathbf{S}_r$, where $\mathbf{S}_r$ denotes the selection matrix corresponding to the sample S_r. By definition of $\widetilde{\mathbf{K}}^{ens}$ and the upper bound on $\|\mathbf{K}-\widetilde{\mathbf{K}}_r\|_2$ already used in the proof of theorem 2, the following holds:

$$\begin{aligned}
\|\mathbf{K}-\widetilde{\mathbf{K}}^{ens}\|_2 &= \Big\|\sum_{r=1}^p \mu_r(\mathbf{K}-\widetilde{\mathbf{K}}_r)\Big\|_2 \le \sum_{r=1}^p \mu_r\|\mathbf{K}-\widetilde{\mathbf{K}}_r\|_2\\
&\le \sum_{r=1}^p \mu_r\big(\|\mathbf{K}-\mathbf{K}_k\|_2 + 2\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2\big)\\
&= \|\mathbf{K}-\mathbf{K}_k\|_2 + 2\sum_{r=1}^p \mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2.
\end{aligned}$$

We apply Theorem 1 to $\phi(S)=\sum_{r=1}^p \mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$. Let S' be a sample differing from S by only one column. Observe that changing one column of the full sample S changes only one subsample S_r and thus only one term $\mu_r\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$. Thus, in view of the bound (7) on the change to $\|\mathbf{X}\mathbf{X}^\top-\mathbf{Z}_r\mathbf{Z}_r^\top\|_2$, the following holds:

$$|\phi(S')-\phi(S)| \le \frac{2n}{l}\mu_{\max} d_{\max}^{\mathbf{K}}\mathbf{K}_{\max}^{\frac{1}{2}}, \tag{10}$$

The expectation of Φ can be straightforwardly bounded by:

$$\mathbf{E}[\Phi(S)] = \sum_{r=1}^{p} \mu_r \mathbf{E}[\|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_2] \le \sum_{r=1}^{p} \mu_r \frac{n}{\sqrt{l}} \mathbf{K}_{\max} = \frac{n}{\sqrt{l}} \mathbf{K}_{\max}$$

using the bound (8) for a single expert. Plugging in this upper bound and the Lipschitz bound (10) in Theorem 1 yields the norm-2 bound for the ensemble Nyström method.

For the Frobenius error bound, using the convexity of the Frobenius norm square $\|\cdot\|_F^2$ and the general inequality (9), we can write

$$\begin{aligned}
\|\mathbf{K} - \widetilde{\mathbf{K}}^{ens}\|_F^2 &= \Big\| \sum_{r=1}^{p} \mu_r (\mathbf{K} - \widetilde{\mathbf{K}}_r) \Big\|_F^2 \le \sum_{r=1}^{p} \mu_r \|\mathbf{K} - \widetilde{\mathbf{K}}_r\|_F^2 \\
&\le \sum_{r=1}^{p} \mu_r \Big[\|\mathbf{K} - \mathbf{K}_k\|_F^2 + \sqrt{64k}\, \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F \, n\mathbf{K}_{ii}^{\max} \Big]. \\
&= \|\mathbf{K} - \mathbf{K}_k\|_F^2 + \sqrt{64k} \sum_{r=1}^{p} \mu_r \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F \, n\mathbf{K}_{ii}^{\max}.
\end{aligned}$$

The result follows by the application of Theorem 1 to $\psi(S) = \sum_{r=1}^{p} \mu_r \|\mathbf{X}\mathbf{X}^\top - \mathbf{Z}_r\mathbf{Z}_r^\top\|_F$ in a way similar to the norm-2 case. ■

The bounds of Theorem 3 are similar in form to those of Theorem 2. However, the bounds for the ensemble Nyström are tighter than those for any Nyström expert based on a single sample of size l even for a uniform weighting. In particular, for $\mu_i = 1/p$ for all i, the last term of the ensemble bound for norm-2 is smaller by a factor larger than $\mu_{\max} p^{\frac{1}{2}} = 1/\sqrt{p}$.

7. Conclusion

A key aspect of sampling-based matrix approximations is the method for the selection of representative columns. We discussed both fixed and adaptive methods for sampling the columns of a matrix. We saw that the approximation performance is significantly affected by the choice of the sampling algorithm and also that there is a tradeoff between choosing a more informative set of columns and the efficiency of the sampling algorithm. Furthermore, we introduced and discussed a new meta-algorithm based on an ensemble of several matrix approximations that generates favorable matrix reconstructions using base learners derived from either fixed or adaptive sampling schemes, and naturally fits within a distributed computing environment, thus making it quite efficient even in large-scale settings. We concluded with a theoretical analysis of the Nyström method (both the standard approach and the ensemble method) as it is often used in practice, namely using uniform sampling without replacement.

Acknowledgments

AT was supported by NSF award No. 1122732. We thank the editor and the reviewers for several insightful comments that helped improve the original version of this paper.

References

Dimitris Achlioptas and Frank Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2), 2007.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In *Approx-Random*, 2006.

Arthur Asuncion and David Newman. UCI machine learning repository. `http://www.ics.uci.edu/ mlearn/MLRepository.html`, 2007.

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

Francis R. Bach and Michael I. Jordan. Predictive low-rank decomposition for kernel methods. In *International Conference on Machine Learning*, 2005.

Christopher T. Baker. *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, 1977.

Mohamed A. Belabbas and Patrick J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences of the United States of America*, 106(2):369–374, January 2009. ISSN 1091-6490.

Mohamed A. Belabbas and Patrick J. Wolfe. On landmark selection and sampling in high-dimensional data analysis. `arXiv:0906.4582v1 [stat.ML]`, 2009.

Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Conference on Learning Theory*, 1992.

Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Symposium on Discrete Algorithms*, 2009.

Roland Bunschoten. `http://www.mathworks.com/matlabcentral/fileexchange/71-distance-m/`, 1999.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. `arXiv:0903.1476v1 [cs.IT]`, 2009.

Gavin Cawley and Nicola Talbot. Miscellaneous matlab software. `http://theoval.cmp.uea.ac.uk/matlab/default.html#cholinc`, 2004.

Corinna Cortes and Vladimir N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Corinna Cortes, Mehryar Mohri, Dmitry Pechyony, and Ashish Rastogi. Stability of transductive regression algorithms. In *International Conference on Machine Learning*, 2008.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *Conference on Artificial Intelligence and Statistics*, 2010.

Vin de Silva and Joshua Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Neural Information Processing Systems*, 2003.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Symposium on Discrete Algorithms*, 2006.

Petros Drineas. Personal communication, 2008.

Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Petros Drineas, Eleni Drinea, and Patrick S. Huggins. An experimental evaluation of a Monte-Carlo algorithm for svd. In *Panhellenic Conference on Informatics*, 2001.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices ii: computing a low-rank approximation to a matrix. *SIAM Journal of Computing*, 36(1), 2006.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Shai Fine and Katya Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Foundation of Computer Science*, 1998.

Gene Golub and Charles Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 2nd edition, 1983. ISBN 0-8018-3772-3 (hardcover), 0-8018-3739-1 (paperback).

Sergei A. Goreinov, Eugene E. Tyrtyshnikov, and Nickolai L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, 261:1–21, 1997.

Genevieve Gorrell. Generalized Hebbian algorithm for incremental singular value decomposition in natural language processing. In *European Chapter of the Association for Computational Linguistics*, 2006.

Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM Journal of Scientific Computing*, 17(4):848–869, 1996.

Adam Gustafson, Evan Snitkin, Stephen Parker, Charles DeLisi, and Simon Kasif. Towards the identification of essential genes using targeted genome sequencing and comparative analysis. *BMC:Genomics*, 7:265, 2006.

Nathan Halko, Per Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. `arXiv:0909.4061v1 [math.NA]`, 2009.

Sariel Har-peled. Low-rank matrix approximation in linear time, manuscript, 2006.

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM*, 53(3):307–323, 2006.

William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling techniques for the Nyström method. In *Conference on Artificial Intelligence and Statistics*, 2009a.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. On sampling-based approximate spectral decomposition. In *International Conference on Machine Learning*, 2009b.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Ensemble Nyström method. In *Neural Information Processing Systems*, 2009c.

Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`, 1998.

Mu Li, James T. Kwok, and Bao-Liang Lu. Making large-scale Nyström approximation possible. In *International Conference on Machine Learning*, 2010.

Edo Liberty. *Accelerated Dense Random Projections*. Ph.D. thesis, computer science department, Yale University, New Haven, CT, 2009.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Rong Liu, Varun Jain, and Hao Zhang. Subsampling for efficient spectral mesh processing. In *Computer Graphics International Conference*, 2006.

Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

Evert J. Nyström. Über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematicae*, 4(15):1–52, 1928.

Marie Ouimet and Yoshua Bengio. Greedy spectral embedding. In *Artificial Intelligence and Statistics*, 2005.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. In *Principles of Database Systems*, 1998.

John C. Platt. Fast embedding of sparse similarity graphs. In *Neural Information Processing Systems*, 2004.

Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2009.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: an approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

Anthony F. Ruston. Auerbach's theorem and tensor products of banach spaces. *Mathematical Proceedings of the Cambridge Philosophical Society*, 58:476–480, 1962.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press: Cambridge, MA, 2002.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. In *Conference on Automatic Face and Gesture Recognition*, 2002.

Alex J. Smola. SVLab. `http://alex.smola.org/data/svlab.tgz`, 2000.

Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *International Conference on Machine Learning*, 2000.

G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted qr approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.

Ameet Talwalkar and Afshin Rostamizadeh. Matrix coherence and the Nyström method. In *Conference on Uncertainty in Artificial Intelligence*, 2010.

Ameet Talwalkar, Sanjiv Kumar, and Henry Rowley. Large-scale manifold learning. In *Conference on Vision and Pattern Recognition*, 2008.

Mark Tygert. `http://www.mathworks.com/matlabcentral/fileexchange/21524-principal-component-analysis`, 2009.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Neural Information Processing Systems*, 2000.

Kai Zhang and James T. Kwok. Density-weighted Nyström method for computing large kernel eigensystems. *Neural Computation*, 21(1):121–146, 2009.

Kai Zhang, Ivor Tsang, and James Kwok. Improved Nyström low-rank approximation and error analysis. In *International Conference on Machine Learning*, 2008.

Journal of Machine Learning Research 13 (2012) 1007-1036 Submitted 2/11; Revised 1/12; Published 4/12

Positive Semidefinite Metric Learning Using Boosting-like Algorithms

Chunhua Shen CHUNHUA.SHEN@ADELAIDE.EDU.AU
The University of Adelaide
Adelaide, SA 5005, Australia

Junae Kim JUNAE.KIM@NICTA.COM.AU
NICTA, Canberra Research Laboratory
Locked Bag 8001
Canberra, ACT 2601, Australia

Lei Wang LEIW@UOW.EDU.AU
University of Wollongong
Wollongong, NSW 2522, Australia

Anton van den Hengel ANTON.VANDENHENGEL@ADELAIDE.EDU.AU
The University of Adelaide
Adelaide, SA 5005, Australia

Editors: Sören Sonnenburg, Francis Bach, Cheng Soon Ong

Abstract

The success of many machine learning and pattern recognition methods relies heavily upon the identification of an appropriate distance metric on the input data. It is often beneficial to learn such a metric from the input training data, instead of using a default one such as the Euclidean distance. In this work, we propose a boosting-based technique, termed BOOSTMETRIC, for learning a quadratic Mahalanobis distance metric. Learning a valid Mahalanobis distance metric requires enforcing the constraint that the matrix parameter to the metric remains positive semidefinite. Semidefinite programming is often used to enforce this constraint, but does not scale well and is not easy to implement. BOOSTMETRIC is instead based on the observation that any positive semidefinite matrix can be decomposed into a linear combination of trace-one rank-one matrices. BOOSTMETRIC thus uses rank-one positive semidefinite matrices as weak learners within an efficient and scalable boosting-based learning process. The resulting methods are easy to implement, efficient, and can accommodate various types of constraints. We extend traditional boosting algorithms in that its weak learner is a positive semidefinite matrix with trace and rank being one rather than a classifier or regressor. Experiments on various data sets demonstrate that the proposed algorithms compare favorably to those state-of-the-art methods in terms of classification accuracy and running time.

Keywords: Mahalanobis distance, semidefinite programming, column generation, boosting, Lagrange duality, large margin nearest neighbor

1. Introduction

The identification of an effective metric by which to measure distances between data points is an essential component of many machine learning algorithms including k-nearest neighbor (kNN), k-means clustering, and kernel regression. These methods have been applied to a range of problems, including image classification and retrieval (Hastie and Tibshirani, 1996; Yu et al., 2008; Jian and

Vemuri, 2007; Xing et al., 2002; Bar-Hillel et al., 2005; Boiman et al., 2008; Frome et al., 2007) amongst a host of others.

The Euclidean distance has been shown to be effective in a wide variety of circumstances. Boiman et al. (2008), for instance, showed that in generic object recognition with local features, *k*NN with a Euclidean metric can achieve comparable or better accuracy than more sophisticated classifiers such as support vector machines (SVMs). The Mahalanobis distance represents a generalization of the Euclidean distance, and offers the opportunity to learn a distance metric directly from the data. This learned Mahalanobis distance approach has been shown to offer improved performance over Euclidean distance-based approaches, and was particularly shown by Wang et al. (2010b) to represent an improvement upon the method of Boiman et al. (2008). It is the prospect of a significant performance improvement from fundamental machine learning algorithms which inspires the approach presented here.

If we let $\mathbf{a}_i, i = 1, 2 \cdots$, represent a set of points in $\mathbb{R}^D$, then the Mahalanobis distance, or Gaussian quadratic distance, between two points is

$$\|\mathbf{a}_i - \mathbf{a}_j\|_{\mathbf{X}} = \sqrt{(\mathbf{a}_i - \mathbf{a}_j)^\top \mathbf{X} (\mathbf{a}_i - \mathbf{a}_j)},$$

where $\mathbf{X} \succcurlyeq 0$ is a positive semidefinite (p.s.d.) matrix. The Mahalanobis distance is thus parameterized by a p.s.d. matrix, and methods for learning Mahalanobis distances are therefore often framed as constrained semidefinite programs. The approach we propose here, however, is based on boosting, which is more typically used for learning classifiers. The primary motivation for the boosting-based approach is that it scales well, but its efficiency in dealing with large data sets is also advantageous. The learning of Mahalanobis distance metrics represents a specific application of a more general method for matrix learning which we present below.

We are interested here in the case where the training data consist of a set of constraints upon the relative distances between data points,

$$\mathcal{I} = \{(\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k) \,|\, \mathbf{dist}_{ij} < \mathbf{dist}_{ik}\}, \tag{1}$$

where $\mathbf{dist}_{ij}$ measures the distance between $\mathbf{a}_i$ and $\mathbf{a}_j$. Each such constraint implies that "$\mathbf{a}_i$ is closer to $\mathbf{a}_j$ than $\mathbf{a}_i$ is to $\mathbf{a}_k$". Constraints such as these often arise when it is known that $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to the same class of data points while $\mathbf{a}_i, \mathbf{a}_k$ belong to different classes. These comparison constraints are thus often much easier to obtain than either the class labels or distances between data elements (Schultz and Joachims, 2003). For example, in video content retrieval, faces extracted from successive frames at close locations can be safely assumed to belong to the same person, without requiring the individual to be identified. In web search, the results returned by a search engine are ranked according to the relevance, an ordering which allows a natural conversion into a set of constraints.

The problem of learning a p.s.d. matrix such as $\mathbf{X}$ can be formulated in terms of estimating a projection matrix $\mathbf{L}$ where $\mathbf{X} = \mathbf{L}\mathbf{L}^\top$. This approach has the advantage that the p.s.d. constraint is enforced through the parameterization, but the disadvantage is that the relationship between the distance measure and the parameter matrix is less direct. In practice this approach has lead to local, rather than globally optimal solutions, however (see Goldberger et al., 2004 for example).

Methods such as Xing et al. (2002), Weinberger et al. (2005), Weinberger and Saul (2006) and Globerson and Roweis (2005) which seek $\mathbf{X}$ directly are able to guarantee global optimality, but at the cost of a heavy computational burden and poor scalability as it is not trivial to preserve the semidefiniteness of $\mathbf{X}$ during the course of learning. Standard approaches such as interior-point (IP) Newton methods need to calculate the Hessian. This typically requires $O(D^4)$ storage and has worst-case computational complexity of approximately $O(D^{6.5})$ where D is the size of the p.s.d. matrix. This is prohibitive for many real-world problems. An alternating projected (sub-)gradient approach is adopted in Weinberger et al. (2005), Xing et al. (2002) and Globerson and Roweis (2005). The disadvantages of this algorithm, however, are: 1) it is not easy to implement; 2) many parameters are involved; 3) usually it converges slowly.

We propose here a method for learning a p.s.d. matrix labeled BOOSTMETRIC. The method is based on the observation that any positive semidefinite matrix can be decomposed into a linear positive combination of trace-one rank-one matrices. The weak learner in BOOSTMETRIC is thus a trace-one rank-one p.s.d. matrix. The proposed BOOSTMETRIC algorithm has the following desirable properties:

1. BOOSTMETRIC is efficient and scalable. Unlike most existing methods, no semidefinite programming is required. At each iteration, only the largest eigenvalue and its corresponding eigenvector are needed.

2. BOOSTMETRIC can accommodate various types of constraints. We demonstrate the use of the method to learn a Mahalanobis distance on the basis of a set of proximity comparison constraints.

3. Like AdaBoost, BOOSTMETRIC does not have any parameter to tune. The user only needs to know when to stop. Also like AdaBoost it is easy to implement. No sophisticated optimization techniques are involved. The efficacy and efficiency of the proposed BOOSTMETRIC is demonstrated on various data sets.

4. We also propose a totally-corrective version of BOOSTMETRIC. As in TotalBoost (Warmuth et al., 2006) the weights of all the selected weak learners (rank-one matrices) are updated at each iteration.

   Both the stage-wise BOOSTMETRIC and totally-corrective BOOSTMETRIC methods are very easy to implement.

The primary contributions of this work are therefore as follows: 1) We extend traditional boosting algorithms such that each weak learner is a matrix with the trace and rank of one—which must be positive semidefinite—rather than a classifier or regressor; 2) The proposed algorithm can be used to solve many semidefinite optimization problems in machine learning and computer vision. We demonstrate the scalability and effectiveness of our algorithms on metric learning. Part of this work appeared in Shen et al. (2008, 2009). More theoretical analysis and experiments are included in this version. Next, we review some relevant work before we present our algorithms.

1.1 Related Work

Distance metric learning is closely related to subspace methods. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two classical dimensionality reduction techniques. PCA finds the subspace that captures the maximum variance within the input data while LDA tries to identify the projection which maximizes the between-class distance and minimizes the within-class variance. Locality preserving projection (LPP) finds a linear projection that preserves the neighborhood structure of the data set (He et al., 2005). Essentially, LPP linearly approximates the eigenfunctions of the Laplace Beltrami operator on the underlying manifold. The connection between LPP and LDA is also revealed in He et al. (2005). Wang et al. (2010a) extended LPP to supervised multi-label classification. Relevant component analysis (RCA) (Bar-Hillel et al., 2005) learns a metric from *equivalence* constraints. RCA can be viewed as extending LDA by incorporating must-link constraints and cannot-link constraints into the learning procedure. Each of these methods may be seen as devising a linear projection from the input space to a lower-dimensional output space. If this projection is characterized by the matrix $\mathbf{L}$, then note that these methods may be related to the problem of interest here by observing $\mathbf{X} = \mathbf{L}\mathbf{L}^\top$. This typically implies that $\mathbf{X}$ is rank-deficient.

Recently, there has been significant research interest in supervised distance metric learning using side information that is typically presented in a set of pairwise constraints. Most of these methods, although appearing in different formats, share a similar essential idea: to learn an optimal distance metric by keeping training examples in equivalence constraints close, and at the same time, examples in in-equivalence constraints well separated. Previous work of Xing et al. (2002), Weinberger et al. (2005), Jian and Vemuri (2007), Goldberger et al. (2004), Bar-Hillel et al. (2005) and Schultz and Joachims (2003) fall into this category. The requirement that $\mathbf{X}$ must be p.s.d. has led to the development of a number of methods for learning a Mahalanobis distance which rely upon constrained semidefinite programing. This approach has a number of limitations, however, which we now discuss with reference to the problem of learning a p.s.d. matrix from a set of constraints upon pairwise-distance comparisons. Relevant work on this topic includes Bar-Hillel et al. (2005), Xing et al. (2002), Jian and Vemuri (2007), Goldberger et al. (2004), Weinberger et al. (2005) and Globerson and Roweis (2005) amongst others.

Xing et al. (2002) first proposed the idea of learning a Mahalanobis metric for clustering using convex optimization. The inputs are two sets: a similarity set and a dis-similarity set. The algorithm maximizes the distance between points in the dis-similarity set under the constraint that the distance between points in the similarity set is upper-bounded. Neighborhood component analysis (NCA) (Goldberger et al., 2004) and large margin nearest neighbor (LMNN) (Weinberger et al., 2005) learn a metric by maintaining consistency in data's neighborhood and keep a large margin at the boundaries of different classes. It has been shown in Weinberger and Saul (2009); Weinberger et al. (2005) that LMNN delivers the state-of-the-art performance among most distance metric learning algorithms. Information theoretic metric learning (ITML) learns a suitable metric based on information theoretics (Davis et al., 2007). To partially alleviate the heavy computation of standard IP Newton methods, Bregman's cyclic projection is used in Davis et al. (2007). This idea is extended in Wang and Jin (2009), which has a closed-form solution and is computationally efficient.

There have been a number of approaches developed which aim to improve the scalability of the process of learning a metric parameterized by a p.s.d. metric $\mathbf{X}$. For example, Rosales and Fung (2006) approximate the p.s.d. cone using a set of linear constraints based on the diagonal dominance theorem. The approximation is not accurate, however, in the sense that it imposes too strong a condition on the learned matrix—one may not want to learn a diagonally dominant matrix. Alternative optimization is used in Xing et al. (2002) and Weinberger et al. (2005) to solve the semidefinite problem iteratively. At each iteration, a full eigen-decomposition is applied to project the solution back onto the p.s.d. cone. BOOSTMETRIC is conceptually very different to this approach, and additionally only requires the calculation of the first eigenvector. Tsuda et al. (2005) proposed to use matrix logarithms and exponentials to preserve positive definiteness. For the application of semidefinite kernel learning, they designed a matrix exponentiated gradient method to optimize von Neumann divergence based objective functions. At each iteration of matrix exponentiated gradient, a full eigen-decomposition is needed. In contrast, we only need to find the leading eigenvector.

The approach proposed here is directly inspired by the LMNN proposed in Weinberger and Saul (2009); Weinberger et al. (2005). Instead of using the hinge loss, however, we use the exponential loss and logistic loss functions in order to derive an AdaBoost-like (or LogitBoost-like) optimization procedure. In theory, any differentiable convex loss function can be applied here. Hence, despite similar purposes, our algorithm differs essentially in the optimization. While the formulation of LMNN looks more similar to SVMs, our algorithm, termed BOOSTMETRIC, largely draws upon AdaBoost (Schapire, 1999).

Column generation was first proposed by Dantzig and Wolfe (1960) for solving a particular form of structured linear program with an extremely large number of variables. The general idea of column generation is that, instead of solving the original large-scale problem (master problem), one works on a restricted master problem with a reasonably small subset of the variables at each step. The dual of the restricted master problem is solved by the simplex method, and the optimal dual solution is used to find the new column to be included into the restricted master problem. LPBoost (Demiriz et al., 2002) is a direct application of column generation in boosting. Significantly, LPBoost showed that in an LP framework, unknown weak hypotheses can be learned from the dual although the space of all weak hypotheses is infinitely large. Shen and Li (2010) applied column generation to boosting with general loss functions. It is these results that underpin BOOSTMETRIC.

The remaining content is organized as follows. In Section 2 we present some preliminary mathematics. In Section 3, we show the main results. Experimental results are provided in Section 4.

2. Preliminaries

We introduce some fundamental concepts that are necessary for setting up our problem. First, the notation used in this paper is as follows.

2.1 Notation

Throughout this paper, a matrix is denoted by a bold upper-case letter ($\mathbf{X}$); a column vector is denoted by a bold lower-case letter ($\boldsymbol{x}$). The ith row of $\mathbf{X}$ is denoted by $\mathbf{X}_{i:}$ and the ith column $\mathbf{X}_{:i}$. $\mathbf{1}$ and $\mathbf{0}$ are column vectors of 1's and 0's, respectively. Their size should be clear from the context. We denote the space of $D \times D$ symmetric matrices by $\mathbb{S}^D$, and positive semidefinite matrices by $\mathbb{S}^D_+$. $\mathbf{Tr}(\cdot)$ is the trace of a symmetric matrix and $\langle \mathbf{X}, \mathbf{Z} \rangle = \mathbf{Tr}(\mathbf{X}\mathbf{Z}^\top) = \sum_{ij} \mathbf{X}_{ij}\mathbf{Z}_{ij}$ calculates the inner product of two matrices. An element-wise inequality between two vectors like $\boldsymbol{u} \leq \boldsymbol{v}$ means $u_i \leq v_i$ for all i. We use $\mathbf{X} \succcurlyeq 0$ to indicate that matrix $\mathbf{X}$ is positive semidefinite. For a matrix $\mathbf{X} \in \mathbb{S}^D$, the following statements are equivalent: 1) $\mathbf{X} \succcurlyeq 0$ ($\mathbf{X} \in \mathbb{S}^D_+$); 2) All eigenvalues of $\mathbf{X}$ are nonnegative ($\lambda_i(\mathbf{X}) \geq 0, i = 1, \cdots, D$); and 3) $\forall \boldsymbol{u} \in \mathbb{R}^D, \boldsymbol{u}^\top \mathbf{X} \boldsymbol{u} \geq 0$.

2.2 A Theorem on Trace-one Semidefinite Matrices

Before we present our main results, we introduce an important theorem that serves the theoretical basis of BOOSTMETRIC.

Definition 1 *For any positive integer m, given a set of points* $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ *in a real vector or matrix space* Sp*, the* convex hull *of* Sp *spanned by m elements in* Sp *is defined as:*

$$\mathbf{Conv}_m(\mathrm{Sp}) = \left\{ \textstyle\sum_{i=1}^m w_i \boldsymbol{x}_i \middle| w_i \geq 0, \sum_{i=1}^m w_i = 1, \boldsymbol{x}_i \in \mathrm{Sp} \right\}.$$

Define the linear convex span of Sp *as:*[1]

$$\mathbf{Conv}(\mathrm{Sp}) = \bigcup_m \mathbf{Conv}_m(\mathrm{Sp}) = \left\{ \textstyle\sum_{i=1}^m w_i \boldsymbol{x}_i \middle| w_i \geq 0, \sum_{i=1}^m w_i = 1, \boldsymbol{x}_i \in \mathrm{Sp}, m \in \mathbb{Z}_+ \right\}.$$

Here $\mathbb{Z}_+$ *denotes the set of all positive integers.*

Definition 2 *Let us define* Γ_1 *to be the space of all positive semidefinite matrices* $\mathbf{X} \in \mathbb{S}_+^D$ *with trace equaling one:*

$$\Gamma_1 = \{\mathbf{X} \,|\, \mathbf{X} \succcurlyeq 0, \mathbf{Tr}(\mathbf{X}) = 1\};$$

and Ψ_1 *to be the space of all positive semidefinite matrices with both trace and rank equaling one:*

$$\Psi_1 = \{\mathbf{Z} \,|\, \mathbf{Z} \succcurlyeq 0, \mathbf{Tr}(\mathbf{Z}) = 1, \mathbf{Rank}(\mathbf{Z}) = 1\}.$$

We also define Γ_2 *as the convex hull of* Ψ_1*, that is,*

$$\Gamma_2 = \mathbf{Conv}(\Psi_1).$$

Lemma 3 *Let* Ψ_2 *be a convex polytope defined as* $\Psi_2 = \{\boldsymbol{\lambda} \in \mathbb{R}^D | \lambda_k \geq 0, \forall k = 0, \cdots, D, \sum_{k=1}^D \lambda_k = 1\}$*, then the points with only one element equaling one and all the others being zeros are the extreme points (vertexes) of* Ψ_2*. All the other points can not be extreme points.*

Proof Without loss of generality, let us consider such a point $\boldsymbol{\lambda}' = \{1, 0, \cdots, 0\}$. If $\boldsymbol{\lambda}'$ is not an extreme point of Ψ_2, then it must be possible to express it as a convex combination of a set of *other* points in Ψ_2: $\boldsymbol{\lambda}' = \sum_{i=1}^m w_i \boldsymbol{\lambda}^i$, $w_i > 0$, $\sum_{i=1}^m w_i = 1$ and $\boldsymbol{\lambda}^i \neq \boldsymbol{\lambda}'$. Then we have equations: $\sum_{i=1}^m w_i \lambda_k^i = 0, \forall k = 2, \cdots, D$. It follows that $\lambda_k^i = 0, \forall i$ and $k = 2, \cdots, D$. That means, $\lambda_1^i = 1 \; \forall i$. This is inconsistent with $\boldsymbol{\lambda}^i \neq \boldsymbol{\lambda}'$. Therefore such a convex combination does not exist and $\boldsymbol{\lambda}'$ must be an extreme point. It is trivial to see that any $\boldsymbol{\lambda}$ that has more than one active element is an convex combination of the above-defined extreme points. So they can not be extreme points. ■

Theorem 4 Γ_1 *equals to* Γ_2*; that is,* Γ_1 *is also the convex hull of* Ψ_1*. In other words, all* $\mathbf{Z} \in \Psi_1$*, form the set of extreme points of* Γ_1*.*

Proof It is easy to check that any convex combination $\sum_i w_i \mathbf{Z}_i$, such that $\mathbf{Z}_i \in \Psi_1$, resides in Γ_1, with the following two facts: 1) a convex combination of p.s.d. matrices is still a p.s.d. matrix; 2) $\mathbf{Tr}\big(\sum_i w_i \mathbf{Z}_i\big) = \sum_i w_i \, \mathbf{Tr}(\mathbf{Z}_i) = 1$.

By denoting $\lambda_1 \geq \cdots \geq \lambda_D \geq 0$ the eigenvalues of a $\mathbf{Z} \in \Gamma_1$, we know that $\lambda_1 \leq 1$ because $\sum_{i=1}^D \lambda_i = \mathbf{Tr}(\mathbf{Z}) = 1$. Therefore, all eigenvalues of $\mathbf{Z}$ must satisfy: $\lambda_i \in [0, 1]$, $\forall i = 1, \cdots, D$ and

[1] With slight abuse of notation, we also use the symbol $\mathbf{Conv}(\cdot)$ to denote convex span. In general it is not a convex hull.

$\sum_i^D \lambda_i = 1$. By looking at the eigenvalues of $\mathbf{Z}$ and using Lemma 3, it is immediate to see that a matrix $\mathbf{Z}$ such that $\mathbf{Z} \succcurlyeq 0, \mathbf{Tr}(\mathbf{Z}) = 1$ and $\mathbf{Rank}(\mathbf{Z}) > 1$ can not be an extreme point of Γ_1. The only candidates for extreme points are those rank-one matrices ($\lambda_1 = 1$ and $\lambda_{2,\cdots,D} = 0$). Moreover, it is not possible that some rank-one matrices are extreme points and others are not because the other two constraints $\mathbf{Z} \succcurlyeq 0$ and $\mathbf{Tr}(\mathbf{Z}) = 1$ do not distinguish between different rank-one matrices.

Hence, all $\mathbf{Z} \in \Psi_1$ form the set of extreme points of Γ_1. Furthermore, Γ_1 is a convex and compact set, which must have extreme points. The Krein-Milman Theorem (Krein and Milman, 1940) tells us that a convex and compact set is equal to the convex hull of its extreme points. ■

This theorem is a special case of the results from Overton and Womersley (1992) in the context of eigenvalue optimization. A different proof for the above theorem's general version can also be found in Fillmore and Williams (1971).

In the context of semidefinite optimization, what is of interest about Theorem 4 is as follows: it tells us that a bounded p.s.d. matrix constraint $\mathbf{X} \in \Gamma_1$ can be equivalently replaced with a set of constrains which belong to Γ_2. At the first glance, this is a highly counterintuitive proposition because Γ_2 involves many more complicated constraints. Both w_i and $\mathbf{Z}_i$ ($\forall i = 1, \cdots, m$) are unknown variables. Even worse, m could be extremely (or even infinitely) large. Nevertheless, this is the type of problems that *boosting* algorithms are designed to solve. Let us give a brief overview of boosting algorithms.

2.3 Boosting

Boosting is an example of ensemble learning, where multiple learners are trained to solve the same problem. Typically a boosting algorithm (Schapire, 1999) creates a single strong learner by incrementally adding base (weak) learners to the final strong learner. The base learner has an important impact on the strong learner. In general, a boosting algorithm builds on a user-specified base learning procedure and runs it repeatedly on modified data that are outputs from the previous iterations.

The general form of the boosting algorithm is sketched in Algorithm 1. The inputs to a boosting algorithm are a set of training example $\boldsymbol{x}$, and their corresponding class labels y. The final output is a strong classifier which takes the form

$$F_{\boldsymbol{w}}(\boldsymbol{x}) = \sum_{j=1}^J w_j h_j(\boldsymbol{x}). \tag{2}$$

Here $h_j(\cdot)$ is a base learner. From Theorem 4, we know that a matrix $\mathbf{X} \in \Gamma_1$ can be decomposed as

$$\mathbf{X} = \sum_{j=1}^J w_j \mathbf{Z}_j, \mathbf{Z}_j \in \Gamma_2. \tag{3}$$

By observing the similarity between Equations (2) and (3), we may view $\mathbf{Z}_j$ as a weak classifier and the matrix $\mathbf{X}$ as the strong classifier that we want to learn. This is exactly the problem that boosting methods have been designed to solve. This observation inspires us to solve a special type of semidefinite optimization problem using boosting techniques.

The sparse greedy approximation algorithm proposed by Zhang (2003) is an efficient method for solving a class of convex problems, and achieves fast convergence rates. It has also been shown that boosting algorithms can be interpreted within the general framework of Zhang (2003). The main idea of sequential greedy approximation, therefore, is as follows. Given an initialization $\boldsymbol{u}_0$, which is in a convex subset of a linear vector space, a matrix space or a functional space, the algorithm finds $\boldsymbol{u}_i$ and $\lambda \in (0,1)$ such that the objective function $F((1-\lambda)\boldsymbol{u}_{i-1} + \lambda \boldsymbol{u}_i)$ is minimized. Then the

Algorithm 1 The general framework of boosting.

```
Input: Training data.
Initialize a weight set $\boldsymbol{u}$ on the training examples;
for $j = 1, 2, \cdots,$ do
    · Receive a weak hypothesis $h_j(\cdot)$;
    · Calculate $w_j > 0$;
    · Update $\boldsymbol{u}$.
Output: A convex combination of the weak hypotheses: $F_{\boldsymbol{w}}(\boldsymbol{x}) = \sum_{j=1}^{J} w_j h_j(\boldsymbol{x})$.
```

solution $\boldsymbol{u}_i$ is updated as $\boldsymbol{u}_i = (1-\lambda)\boldsymbol{u}_{i-1} + \lambda \boldsymbol{u}_i$ and the iteration goes on. Clearly, $\boldsymbol{u}_i$ must remain in the original space. As shown next, our first case, which learns a metric using the hinge loss, greatly resembles this idea.

2.4 Distance Metric Learning Using Proximity Comparison

The process of measuring distance using a Mahalanobis metric is equivalent to linearly transforming the data by a projection matrix $\mathbf{L} \in \mathbb{R}^{D \times d}$ (usually $D \geq d$) before calculating the standard Euclidean distance:

$$\mathbf{dist}_{ij}^2 = \|\mathbf{L}^\top \mathbf{a}_i - \mathbf{L}^\top \mathbf{a}_j\|_2^2 = (\mathbf{a}_i - \mathbf{a}_j)^\top \mathbf{L}\mathbf{L}^\top (\mathbf{a}_i - \mathbf{a}_j) = (\mathbf{a}_i - \mathbf{a}_j)^\top \mathbf{X} (\mathbf{a}_i - \mathbf{a}_j).$$

As described above, the problem of learning a Mahalanobis metric can be approached in terms of learning the matrix $\mathbf{L}$, or the p.s.d. matrix $\mathbf{X}$. If $\mathbf{X} = \mathbf{I}$, the Mahalanobis distance reduces to the Euclidean distance. If $\mathbf{X}$ is diagonal, the problem corresponds to learning a metric in which different features are given different weights, *a.k.a.,* feature weighting. Our approach is to learn a full p.s.d. matrix $\mathbf{X}$, however, using BOOSTMETRIC.

In the framework of large-margin learning, we want to maximize the distance between $\mathbf{dist}_{ij}$ and $\mathbf{dist}_{ik}$. That is, we wish to make $\mathbf{dist}_{ik}^2 - \mathbf{dist}_{ij}^2 = (\mathbf{a}_i - \mathbf{a}_k)^\top \mathbf{X} (\mathbf{a}_i - \mathbf{a}_k) - (\mathbf{a}_i - \mathbf{a}_j)^\top \mathbf{X} (\mathbf{a}_i - \mathbf{a}_j)$ as large as possible under some regularization. To simplify notation, we rewrite the distance between $\mathbf{dist}_{ij}^2$ and $\mathbf{dist}_{ik}^2$ as $\mathbf{dist}_{ik}^2 - \mathbf{dist}_{ij}^2 = \langle \mathbf{A}_r, \mathbf{X} \rangle$, where

$$\mathbf{A}_r = (\mathbf{a}_i - \mathbf{a}_k)(\mathbf{a}_i - \mathbf{a}_k)^\top - (\mathbf{a}_i - \mathbf{a}_j)(\mathbf{a}_i - \mathbf{a}_j)^\top, \tag{4}$$

for $r = 1, \cdots, |\mathcal{I}|$ and $|\mathcal{I}|$ is the size of the set of constraints $\mathcal{I}$ defined in Equation (1).

3. Algorithms

In this section, we define the optimization problems for metric learning. We mainly investigate the cases using the hinge loss, exponential loss and logistic loss functions. In order to derive an efficient optimization strategy, we look at their Lagrange dual problems and design boosting-like approaches for efficiency.

3.1 Learning with the Hinge Loss

Our goal is to derive a general algorithm for p.s.d. matrix learning with the hinge loss function. Assume that we want to find a p.s.d. matrix $\mathbf{X} \succcurlyeq 0$ such that a set of constraints

$$\langle \mathbf{A}_r, \mathbf{X} \rangle > 0, r = 1, 2, \cdots,$$

are satisfied as *well* as possible. Here $\mathbf{A}_r$ is as defined in (4). These constraints need not all be strictly satisfied and thus we define the margin $\rho_r = \langle \mathbf{A}_r, \mathbf{X} \rangle, \forall r$.

Putting it into the maximum margin learning framework, we want to minimize the following trace norm regularized objective function: $\sum_r F(\langle \mathbf{A}_r, \mathbf{X} \rangle) + v\,\mathbf{Tr}(\mathbf{X})$, with $F(\cdot)$ a convex loss function and v a regularization constant. Here we have used the trace norm regularization. Of course a Frobenius norm regularization term can also be used here. Minimizing the Frobenius norm $||\mathbf{X}||_{\mathrm{F}}^2$, which is equivalent to minimize the ℓ_2 norm of the eigenvalues of $\mathbf{X}$, penalizes a solution that is far away from the identity matrix. With the hinge loss, we can write the optimization problem as:

$$\max_{\rho, \mathbf{X}, \boldsymbol{\xi}} \; \rho - v \textstyle\sum_{r=1}^{|\mathcal{I}|} \xi_r, \text{ s.t.: } \langle \mathbf{A}_r, \mathbf{X} \rangle \geq \rho - \xi_r, \forall r; \mathbf{X} \succcurlyeq 0, \mathbf{Tr}(\mathbf{X}) = 1; \; \boldsymbol{\xi} \geq \mathbf{0}. \tag{5}$$

Here $\mathbf{Tr}(\mathbf{X}) = 1$ removes the scale ambiguity because the distance inequalities are scale invariant.

We can decompose $\mathbf{X}$ into: $\mathbf{X} = \sum_{j=1}^{J} w_j \mathbf{Z}_j$, with $w_j > 0$, $\mathbf{Rank}(\mathbf{Z}_j) = 1$ and $\mathbf{Tr}(\mathbf{Z}_j) = 1$, $\forall j$. So we have

$$\langle \mathbf{A}_r, \mathbf{X} \rangle = \big\langle \mathbf{A}_r, \textstyle\sum_{j=1}^{J} w_j \mathbf{Z}_j \big\rangle = \textstyle\sum_{j=1}^{J} w_j \big\langle \mathbf{A}_r, \mathbf{Z}_j \big\rangle = \textstyle\sum_{j=1}^{J} w_j \mathbf{H}_{rj} = \mathbf{H}_{r:} \boldsymbol{w}, \forall r. \tag{6}$$

Here $\mathbf{H}_{rj}$ is a shorthand for $\mathbf{H}_{rj} = \big\langle \mathbf{A}_r, \mathbf{Z}_j \big\rangle$. Clearly, $\mathbf{Tr}(\mathbf{X}) = \mathbf{1}^\top \boldsymbol{w}$. Using Theorem 4, we replace the p.s.d. conic constraint in the primal (5) with a linear convex combination of rank-one unitary matrices: $\mathbf{X} = \sum_j w_j \mathbf{Z}_j$, and $\mathbf{1}^\top \boldsymbol{w} = 1$. Substituting $\mathbf{X}$ in (5), we have

$$\max_{\rho, \boldsymbol{w}, \boldsymbol{\xi}} \; \rho - v \textstyle\sum_{r=1}^{|\mathcal{I}|} \xi_r, \text{ s.t.: } \mathbf{H}_{r:} \boldsymbol{w} \geq \rho - \xi_r, (r = 1, \ldots, |\mathcal{I}|); \boldsymbol{w} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{w} = 1; \; \boldsymbol{\xi} \geq \mathbf{0}. \tag{7}$$

The Lagrange dual problem of the above linear programming problem (7) is easily derived:

$$\min_{\pi, \boldsymbol{u}} \; \pi \text{ s.t.: } \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{H}_{r:} \leq \pi \mathbf{1}^\top; \mathbf{1}^\top \boldsymbol{u} = 1, \mathbf{0} \leq \boldsymbol{u} \leq v\mathbf{1}.$$

We can then use column generation to solve the original problem iteratively by looking at both the primal and dual problems. See Shen et al. (2008) for the algorithmic details. In this work we are more interested in smooth loss functions such as the exponential loss and logistic loss, as presented in the sequel.

3.2 Learning with the Exponential Loss

By employing the exponential loss, we want to optimize

$$\begin{aligned} &\min_{\mathbf{X}, \boldsymbol{\rho}} \log\big(\textstyle\sum_{r=1}^{|\mathcal{I}|} \exp(-\rho_r)\big) + v\,\mathbf{Tr}(\mathbf{X}) \\ &\text{s.t.: } \rho_r = \langle \mathbf{A}_r, \mathbf{X} \rangle, r = 1, \cdots, |\mathcal{I}|, \; \mathbf{X} \succcurlyeq 0. \end{aligned} \tag{8}$$

Note that: 1) We are proposing a logarithmic version of the sum of exponential loss. This transform does not change the original optimization problem of sum of exponential loss because the logarithmic function is strictly monotonically increasing. 2) A regularization term $\mathbf{Tr}(\mathbf{X})$ has been applied. Without this regularization, one can always multiply $\mathbf{X}$ by an arbitrarily large scale factor in order to make the exponential loss approach zero in the case of all constraints being satisfied. This trace-norm regularization may also lead to low-rank solutions. 3) An auxiliary variable $\rho_r, r = 1, \ldots$ must be introduced for deriving a meaningful dual problem, as we show later.

We now derive the Lagrange dual of the problem that we are interested in. The original problem (8) now becomes

$$\begin{aligned}&\min_{\boldsymbol{\rho},\boldsymbol{w}} \log\left(\textstyle\sum_{r=1}^{|\mathcal{I}|}\exp(-\rho_r)\right) + v\mathbf{1}^\top \boldsymbol{w} \\ &\text{s.t.:} \, \rho_r = \mathbf{H}_{r:}\boldsymbol{w}, r = 1, \cdots, |\mathcal{I}|; \boldsymbol{w} \geq \mathbf{0}.\end{aligned} \tag{9}$$

We have used the Equation (6). In order to derive its dual, we write its Lagrangian

$$L(\boldsymbol{w},\boldsymbol{\rho},\boldsymbol{u}) = \log\left(\textstyle\sum_{r=1}^{|\mathcal{I}|}\exp(-\rho_r)\right) + v\mathbf{1}^\top\boldsymbol{w} + \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r(\rho_r - \mathbf{H}_{r:}\boldsymbol{w}) - \boldsymbol{p}^\top\boldsymbol{w},$$

with $\boldsymbol{p} \geq 0$. The dual problem is obtained by finding the saddle point of L; that is, $\sup_{\boldsymbol{u}} \inf_{\boldsymbol{w},\boldsymbol{\rho}} L$.

$$\begin{aligned}\inf_{\boldsymbol{w},\boldsymbol{\rho}} L &= \inf_{\boldsymbol{\rho}} \overbrace{\log\left(\textstyle\sum_{r=1}^{|\mathcal{I}|}\exp(-\rho_r)\right) + \boldsymbol{u}^\top\boldsymbol{\rho}}^{L_1} + \inf_{\boldsymbol{w}} \overbrace{(v\mathbf{1}^\top - \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r\mathbf{H}_{r:} - \boldsymbol{p}^\top)\boldsymbol{w}}^{L_2} \\ &= -\textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \log u_r.\end{aligned} \tag{10}$$

The infimum of L_1 is found by setting its first derivative to zero and we have:

$$\inf_{\boldsymbol{\rho}} L_1 = \begin{cases} -\sum_r u_r \log u_r & \text{if } \boldsymbol{u} \geq \mathbf{0}, \mathbf{1}^\top\boldsymbol{u} = 1, \\ -\infty & \text{otherwise.}\end{cases}$$

The infimum is Shannon entropy. L_2 is linear in $\boldsymbol{w}$, hence it must be $\mathbf{0}$. It leads to

$$\textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{H}_{r:} \leq v\mathbf{1}^\top. \tag{11}$$

The Lagrange dual problem of (9) is an entropy maximization problem, which writes

$$\max_{\boldsymbol{u}} - \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \log u_r, \text{ s.t.: } \boldsymbol{u} \geq \mathbf{0}, \mathbf{1}^\top\boldsymbol{u} = 1, \text{and (11)}. \tag{12}$$

Weak and strong duality hold under mild conditions (Boyd and Vandenberghe, 2004). That means, one can usually solve one problem from the other. The KKT conditions link the optimal between these two problems. In our case, it is

$$u_r^\star = \frac{\exp(-\rho_r^\star)}{\sum_{k=1}^{|\mathcal{I}|}\exp(-\rho_k^\star)}, \forall r. \tag{13}$$

While it is possible to devise a totally-corrective column generation based optimization procedure for solving our problem as the case of LPBoost (Demiriz et al., 2002), we are more interested in considering *one-at-a-time* coordinate-wise descent algorithms, as the case of AdaBoost (Schapire, 1999). Let us start from some basic knowledge of column generation because our coordinate descent strategy is inspired by column generation.

If we know all the bases $\mathbf{Z}_j$ $(j = 1 \ldots J)$ and hence the entire matrix $\mathbf{H}$ is known. Then either the primal (9) or the dual (12) can be trivially solved (at least in theory) because both are convex optimization problems. We can solve them in polynomial time. Especially the primal problem is convex minimization with simple nonnegativeness constraints. Off-the-shelf software like LBFGS-B (Zhu et al., 1997) can be used for this purpose. Unfortunately, in practice, we do not access all the bases: the possibility of $\mathbf{Z}$ is infinite. In convex optimization, column generation is a technique that is designed for solving this difficulty.

Column generation was originally advocated for solving large scale linear programs (Lübbecke and Desrosiers, 2005). Column generation is based on the fact that for a linear program, the number of non-zero variables of the optimal solution is equal to the number of constraints. Therefore, although the number of possible variables may be large, we only need a small subset of these in the optimal solution. For a general convex problem, we can use column generation to obtain an *approximate* solution. It works by only considering a small subset of the entire variable set. Once it is solved, we ask the question:"Are there any other variables that can be included to improve the solution?". So we must be able to solve the subproblem: given a set of dual values, one either identifies a variable that has a favorable reduced cost, or indicates that such a variable does not exist. Essentially, column generation finds the variables with negative reduced costs without explicitly enumerating all variables.

Instead of directly solving the primal problem (9), we find the most violated constraint in the dual (12) iteratively for the current solution and adds this constraint to the optimization problem. For this purpose, we need to solve

$$\hat{\mathbf{Z}} = \operatorname{argmax}_{\mathbf{Z}} \left\{ \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \left\langle \mathbf{A}_r, \mathbf{Z} \right\rangle, \text{ s.t.: } \mathbf{Z} \in \Psi_1 \right\}. \tag{14}$$

We discuss how to efficiently solve (14) later. Now we move on to derive a coordinate descent optimization procedure.

3.3 Coordinate Descent Optimization

We show how an AdaBoost-like optimization procedure can be derived.

3.3.1 OPTIMIZING FOR w_j

Since we are interested in the *one-at-a-time* coordinate-wise optimization, we keep $w_1, w_2, \ldots, w_{j-1}$ fixed when solving for w_j. The cost function of the primal problem is (in the following derivation, we drop those terms irrelevant to the variable w_j)

$$C_p(w_j) = \log \left[\textstyle\sum_{r=1}^{|\mathcal{I}|} \exp(-\rho_r^{j-1}) \cdot \exp(-\mathbf{H}_{rj} w_j) \right] + v w_j.$$

Clearly, C_p is convex in w_j and hence there is only one minimum that is also globally optimal. The first derivative of C_p w.r.t. w_j vanishes at optimality, which results in

$$\textstyle\sum_{r=1}^{|\mathcal{I}|} (\mathbf{H}_{rj} - v) u_r^{j-1} \exp(-w_j \mathbf{H}_{rj}) = 0. \tag{15}$$

If $\mathbf{H}_{rj}$ is discrete, such as $\{+1, -1\}$ in standard AdaBoost, we can obtain a closed-form solution similar to AdaBoost. Unfortunately in our case, $\mathbf{H}_{rj}$ can be any real value. We instead use bisection to search for the optimal w_j. The bisection method is one of the root-finding algorithms. It repeatedly divides an interval in half and then selects the subinterval in which a root exists. Bisection is a simple and robust, although it is not the fastest algorithm for root-finding. Algorithm 2 gives the bisection procedure. We have used the fact that the l.h.s. of (15) must be positive at w_l. Otherwise no solution can be found. When $w_j = 0$, clearly the l.h.s. of (15) is positive.

Algorithm 2 Bisection search for w_j.

```
Input: An interval $[w_l, w_u]$ known to contain the optimal value of $w_j$ and convergence
       tolerance $\varepsilon > 0$.
repeat
    $w_j = 0.5(w_l + w_u)$;
    if l.h.s. of (15) $> 0$ then
        $w_l = w_j$;
    else
        $w_u = w_j$.
until $w_u - w_l < \varepsilon$ ;
Output: $w_j$.
```

3.3.2 UPDATING $\boldsymbol{u}$

The rule for updating $\boldsymbol{u}$ can be easily obtained from (13). At iteration j, we have

$$u_r^j \propto \exp(-\rho_r^j) \propto u_r^{j-1} \exp(-\mathbf{H}_{rj} w_j), \text{ and } \sum_{r=1}^{|\mathcal{I}|} u_r^j = 1,$$

derived from (13). So once w_j is calculated, we can update $\boldsymbol{u}$ as

$$u_r^j = \frac{u_r^{j-1} \exp(-\mathbf{H}_{rj} w_j)}{z}, r = 1, \dots, |\mathcal{I}|, \tag{16}$$

where z is a normalization factor so that $\sum_{r=1}^{|\mathcal{I}|} u_r^j = 1$. This is exactly the same as AdaBoost.

3.4 The Base Learning Algorithm

In this section, we show that the optimization problem (14) can be exactly and efficiently solved using eigenvalue-decomposition (EVD).

From $\mathbf{Z} \succcurlyeq 0$ and $\mathbf{Rank}(\mathbf{Z}) = 1$, we know that $\mathbf{Z}$ has the format: $\mathbf{Z} = \boldsymbol{v}\boldsymbol{v}^\top$, $\boldsymbol{v} \in \mathbb{R}^D$; and $\mathbf{Tr}(\mathbf{Z}) = 1$ means $\|\boldsymbol{v}\|_2 = 1$. We have

$$\left\langle \sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{A}_r, \mathbf{Z} \right\rangle = \boldsymbol{v} \left(\sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{A}_r \right) \boldsymbol{v}^\top.$$

By denoting

$$\hat{\mathbf{A}} = \sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{A}_r, \tag{17}$$

the base learning optimization equals:

$$\max_{\boldsymbol{v}} \boldsymbol{v}^\top \hat{\mathbf{A}} \boldsymbol{v}, \text{ s.t.: } \|\boldsymbol{v}\|_2 = 1. \tag{18}$$

It is clear that the largest eigenvalue of $\hat{\mathbf{A}}$, $\lambda_{\max}(\hat{\mathbf{A}})$, and its corresponding eigenvector $\boldsymbol{v}_1$ gives the solution to the above problem. Note that $\hat{\mathbf{A}}$ is symmetric.

$\lambda_{\max}(\hat{\mathbf{A}})$ is also used as one of the stopping criteria of the algorithm. Form the condition (11), $\lambda_{\max}(\hat{\mathbf{A}}) < v$ means that we are not able to find a new base matrix $\hat{\mathbf{Z}}$ that violates (11)—the algorithm converges.

Algorithm 3 Positive semidefinite matrix learning with stage-wise boosting.

```
Input:
  • Training set triplets $(\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k) \in \mathcal{I}$; Compute $\mathbf{A}_r, r = 1, 2, \cdots$, using (4).
  • $J$: maximum number of iterations;
  • (optional) regularization parameter $v$; We may simply set $v$ to a very small value, for
    example, $10^{-7}$.
Initialize: $u_r^0 = \frac{1}{|\mathcal{I}|}, r = 1 \cdots |\mathcal{I}|$;
for $j = 1, 2, \cdots, J$ do
    · Find a new base $\mathbf{Z}_j$ by finding the largest eigenvalue ($\lambda_{\max}(\hat{\mathbf{A}})$) and its eigenvector of
      $\hat{\mathbf{A}}$ in (17);
    · if $\lambda_{\max}(\hat{\mathbf{A}}) < v$ then
        break (converged);
    · Compute $w_j$ using Algorithm 2;
    · Update $\boldsymbol{u}$ to obtain $u_r^j, r = 1, \cdots |\mathcal{I}|$ using (16);
Output: The final p.s.d. matrix $\mathbf{X} \in \mathbb{R}^{D \times D}$, $\mathbf{X} = \sum_{j=1}^{J} w_j \mathbf{Z}_j$.
```

Eigenvalue decompositions is one of the main computational costs in our algorithm. There are approximate eigenvalue solvers, which guarantee that for a symmetric matrix $\mathbf{U}$ and any $\varepsilon > 0$, a vector $\boldsymbol{v}$ is found such that $\boldsymbol{v}^\top \mathbf{U} \boldsymbol{v} \geq \lambda_{\max} - \varepsilon$. To approximately find the largest eigenvalue and eigenvector can be very efficient using Lanczos or power method. We can use the MATLAB function eigs to calculate the largest eigenvector, which calls mex files of ARPACK. ARPACK is a collection of Fortran subroutines designed to solve large scale eigenvalue problems. When the input matrix is symmetric, this software uses a variant of the Lanczos process called the implicitly restarted Lanczos method.

Another way to reduce the time for computing the leading eigenvector is to compute an approximate EVD by a fast Monte Carlo algorithm such as the linear time SVD algorithm developed in Drineas et al. (2004).

We summarize our main algorithmic results in Algorithm 3.

3.5 Learning with the Logistic Loss

We have considered the exponential loss in the last content. The proposed framework is so general that it can also accommodate other convex loss functions. Here we consider the logistic loss, which penalizes mis-classifications with more moderate penalties than the exponential loss. It is believed on noisy data, the logistic loss may achieve better classification performance.

With the same settings as in the case of the exponential loss, we can write our optimization problem as

$$\begin{aligned} & \min_{\boldsymbol{\rho}, \boldsymbol{w}} \textstyle\sum_{r=1}^{|\mathcal{I}|} \mathrm{logit}(\rho_r) + v \mathbf{1}^\top \boldsymbol{w} \\ & \text{s.t.:} \rho_r = \mathbf{H}_{r:} \boldsymbol{w}, r = 1, \cdots, |\mathcal{I}|, \boldsymbol{w} \geq 0. \end{aligned} \tag{19}$$

Here $\text{logit}(\cdot)$ is the logistic loss defined as $\text{logit}(z) = \log(1+\exp(-z))$. Similarly, we derive its Lagrange dual as

$$\min_{\boldsymbol{u}} \textstyle\sum_{r=1}^{|\mathcal{I}|} \text{logit}^*(-u_r)$$
$$\text{s.t.:} \textstyle\sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{H}_{r:} \leq v\mathbf{1}^\top,$$

where $\text{logit}^*(\cdot)$ is the Fenchel conjugate function of $\text{logit}(\cdot)$, defined as

$$\text{logit}^*(-u) = u\log(u) + (1-u)\log(1-u),$$

when $0 \leq u \leq 1$, and ∞ otherwise. So the Fenchel conjugate of $\text{logit}(\cdot)$ is the binary entropy function. We have reversed the sign of $\boldsymbol{u}$ when deriving the dual.

Again, according to the KKT conditions, we have

$$u_r^\star = \frac{\exp(-\rho_r^\star)}{1+\exp(-\rho_r^\star)}, \ \forall r, \tag{20}$$

at optimality. From (20) we can also see that u must be in $(0,1)$.

Similarly, we want to optimize the primal cost function in a coordinate descent way. First, let us find the relationship between u_r^j and u_r^{j-1}. Here j is the iteration index. From (20), it is trivial to obtain

$$u_r^j = \frac{1}{(1/u_r^{j-1} - 1)\exp(\mathbf{H}_{rj}w_j) + 1}, \ \forall r. \tag{21}$$

The optimization of w_j can be solved by looking for the root of

$$\textstyle\sum_{r=1}^{|\mathcal{I}|} \mathbf{H}_{rj} u_r^j - v = 0, \tag{22}$$

where u_r^j is a function of w_j as defined in (21).

Therefore, in the case of the logistic loss, to find w_j, we modify the bisection search of Algorithm 2:

- Line 3: **if** l.h.s. of (22) > 0 **then** ...

and Line 7 of Algorithm 3:

- Line 7: Update $\boldsymbol{u}$ using (21).

3.6 Totally Corrective Optimization

In this section, we derive a totally-corrective version of BOOSTMETRIC, similar to the case of TotalBoost (Warmuth et al., 2006; Shen and Li, 2010) for classification, in the sense that the coefficients of all weak learners are updated at each iteration.

Unlike the stage-wise optimization, here we do not need to keep previous weights of weak learners $w_1, w_2, \ldots, w_{j-1}$. Instead, the weights of all the selected weak learners $w_1, w_2, \ldots, w_j$ are updated at each iteration j. As discussed, our learning procedure is able to employ various loss functions such as the hinge loss, exponential loss or logistic loss. To devise a totally-corrective optimization procedure for solving our problem efficiently, we need to ensure the object function

Algorithm 4 Positive semidefinite matrix learning with totally corrective boosting.

```
Input:
    • Training set triplets $(\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k) \in \mathcal{I}$; Compute $\mathbf{A}_r, r = 1, 2, \cdots$, using (4).
    • $J$: maximum number of iterations;
    • Regularization parameter $v$.
Initialize: $u_r^0 = \frac{1}{|\mathcal{I}|}, r = 1 \cdots |\mathcal{I}|$;
for $j = 1, 2, \cdots, J$ do
    · Find a new base $\mathbf{Z}_j$ by finding the largest eigenvalue ($\lambda_{\max}(\hat{\mathbf{A}})$) and its eigenvector of
      $\hat{\mathbf{A}}$ in (17);
    · if $\lambda_{\max}(\hat{\mathbf{A}}) < v$ then
        break (converged);
    · Optimize for $w_1, w_2, \cdots, w_j$ by solving the primal problem (9) when the exponential
      loss is used or (19) when the logistic loss is used;
    · Update $\boldsymbol{u}$ to obtain $u_r^j, r = 1, \cdots |\mathcal{I}|$ using (13) (exponential loss) or (20) (logistic loss);
Output: The final p.s.d. matrix $\mathbf{X} \in \mathbb{R}^{D \times D}$, $\mathbf{X} = \sum_{j=1}^{J} w_j \mathbf{Z}_j$.
```

to be differentiable with respect to the variables $w_1, w_2, \ldots, w_j$. Here, we use the exponential loss function and the logistic loss function. It is possible to use sub-gradient descent methods when a non-smooth loss function like the hinge loss is used.

It is clear that solving for $\boldsymbol{w}$ is a typical convex optimization problem since it has a differentiable and convex function (9) when the exponential loss is used, or (19) when the logistic loss is used. Hence it can be solved using off-the-shelf gradient-descent solvers like L-BFGS-B (Zhu et al., 1997).

Since all the weights $w_1, w_2, \ldots, w_j$ are updated, u_r^j on $r = 1 \ldots |\mathcal{I}|$ need not to be updated but re-calculated at each iteration j. To calculate u_r^j, we use (13) (exponential loss) or (20) (logistic loss) instead of (16) or (21) respectively. Totally-corrective BOOSTMETRIC methods are very simple to implement. Algorithm 4 gives the summary of this algorithm. Next, we show the convergence property of Algorithm 4. Formally, we want to show the following theorem.

Theorem 5 *Algorithm 4 makes progress at each iteration. In other words, the objective value is decreased at each iteration. Therefore, in the limit, Algorithm 4 solves the optimization problem* (9) *(or* (19)*) globally to a desired accuracy.*

Proof Let us consider the exponential loss case of problem (9). The proof follows the same discussion for the logistic loss, or any other smooth convex loss function. Assume that the current solution is a finite subset of base learners (rank-one trace-one matrices) and their corresponding linear coefficients $\boldsymbol{w}$. If we add a base matrix $\hat{\mathbf{Z}}$ that is not in the current subset, and the corresponding $\hat{w} = 0$, then the objective value and the solution must remain unchanged. We can conclude that the current learned base learners and $\boldsymbol{w}$ are the optimal solution already.

Consider the case that this optimality condition is violated. We need to show that we can find a base learner $\hat{\mathbf{Z}}$, which is not in the current set of all the selected base learners, such that $\hat{w} > 0$

holds. Now assume that $\hat{\mathbf{Z}}$ is the base learner found by solving (18), and the convergence condition $\lambda_{\max}(\hat{\mathbf{A}}) \leq v$ is not satisfied. So, we have $\lambda_{\max}(\hat{\mathbf{A}}) = \left\langle \sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{A}_r, \hat{\mathbf{Z}} \right\rangle > v$.

If, after this weak learner $\hat{\mathbf{Z}}$ is added into the primal problem, the primal solution remains unchanged, that is, the corresponding $\hat{w} = 0$, then from the optimality condition that L_2 in (10) must be zero, we know that $\hat{p} = v - \left\langle \sum_{r=1}^{|\mathcal{I}|} u_r \mathbf{A}_r, \hat{\mathbf{Z}} \right\rangle < 0$. This contradicts the fact the Lagrange multiplier $\hat{p} \geq 0$.

We can conclude that after the base learner $\hat{\mathbf{Z}}$ is added into the primal problem, its corresponding $\hat{w}$ must admit a positive value. It means that one more free variable is added into the problem and re-solving the primal problem would reduce the objective value. Hence a strict decrease in the objective is guaranteed. So Algorithm 4 makes progress at each iteration.

Furthermore, as the optimization problems involved are all convex, there are no local optimal solutions. Therefore Algorithm 4 is guaranteed to converge to the global solution.

Note that the above proof establishes the convergence of Algorithm 4 but it remains unclear about the convergence rate. ■

3.7 Multi-pass BoostMetric

In this section, we show that BoostMetric can use multi-pass learning to enhance the performance.

Our BoostMetric uses training set triplets $(\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k) \in \mathcal{I}$ as input for training. The Mahalanobis distance metric $\mathbf{X}$ can be viewed as a linear transformation in the Euclidean space by projecting the data using matrix $\mathbf{L}$ $(\mathbf{X} = \mathbf{L}\mathbf{L}^\top)$. That is, nearest neighbors of samples using Mahalanobis distance metric $\mathbf{X}$ are the same as nearest neighbors using Euclidean distance in the transformed space. BoostMetric assumes that the triplets of input training set approximately represent the actual nearest neighbors of samples in the transformed space defined by the Mahalanobis metric. However, even though the triplets of BoostMetric consist of nearest neighbors of the original training samples, generated triplets are not exactly the same as the actual nearest neighbors of training samples in the transformed space by $\mathbf{L}$.

We can refine the results of BoostMetric iteratively, as in the multiple-pass LMNN (Weinberger and Saul, 2009): BoostMetric can estimate the triplets in the transformed space under a multiple-pass procedure as close to actual triplets as possible. The rule for multi-pass BoostMetric is simple. At each pass p $(p = 1, 2, \cdots)$, we decompose the learned Mahalanobis distance metric $\mathbf{X}_{p-1}$ of previous pass into transformation matrix $\mathbf{L}_p$. The initial matrix $\mathbf{L}_1$ is an identity matrix. Then we generate the training set triplets from the set of points $\{\mathbf{L}^\top \mathbf{a}_1, \ldots, \mathbf{L}^\top \mathbf{a}_m\}$ where $\mathbf{L} = \mathbf{L}_1 \cdot \mathbf{L}_2 \cdot \cdots \cdot \mathbf{L}_p$. The final Mahalanobis distance metric $\mathbf{X}$ becomes $\mathbf{L}\mathbf{L}^\top$ in Multi-pass BoostMetric.

4. Experiments

In this section, we present experiments on data visualization, classification and image retrieval tasks.

		MNIST	USPS	Letters	yFaces	bal	wine	iris
# of samples		70,000	11,000	20,000	2,414	625	178	150
# of triplets		450,000	69,300	94,500	15,210	3,942	1,125	945
dimension		784	256	16	1,024	4	13	4
dimension after PCA		164	60		300			
# of samples for training		50,000	7,700	10,500	1,690	438	125	105
# cross validation samples		10,000	1,650	4,500	362	94	27	23
# test samples		10,000	1,650	5,000	362	93	26	22
# of classes		10	10	26	38	3	3	3
# of runs		1	10	1	10	10	10	10
Error Rates	Euclidean	3.19	4.78 (0.40)	5.42	28.07 (2.07)	18.60 (3.96)	28.08 (7.49)	3.64 (4.18)
	PCA	3.10	3.49 (0.62)		28.65 (2.18)			
	LDA	8.76	6.96 (0.68)	4.44	**5.08 (1.15)**	12.58 (2.38)	0.77 (1.62)	3.18 (3.07)
	RCA	7.85	5.35 (0.52)	4.64	7.65 (1.08)	17.42 (3.58)	**0.38 (1.22)**	3.18 (3.07)
	NCA					18.28 (3.58)	28.08 (7.49)	3.18 (3.74)
	LMNN	2.30	3.49 (0.62)	3.82	14.75 (12.11)	12.04 (5.59)	3.46 (3.82)	3.64 (2.87)
	ITML	2.80	3.85 (1.13)	7.20	19.39 (2.11)	10.11 (4.06)	28.46 (8.35)	3.64 (3.59)
	BoostMetric-E	2.65	2.53 (0.47)	3.06	6.91 (1.90)	10.11 (3.45)	3.08 (3.53)	3.18 (3.74)
	BoostMetric-E, MP	2.62	2.24 (0.40)	2.80	6.77 (1.77)	10.22 (4.43)	1.92 (2.03)	3.18 (4.31)
	BoostMetric-E, TC	2.20	2.25 (0.51)	2.82	7.13 (1.40)	10.22 (2.39)	4.23 (3.82)	3.18 (3.07)
	BoostMetric-E, MP, TC	2.34	2.23 (0.34)	3.74	7.29 (1.58)	10.32 (3.09)	2.69 (3.17)	3.18 (4.31)
	BoostMetric-L	2.66	2.38 (0.31)	2.80	6.93 (1.59)	9.89 (3.12)	3.08 (3.03)	3.18 (3.74)
	BoostMetric-L, MP	2.72	2.22 (0.31)	2.70	6.66 (1.35)	10.22 (4.25)	1.15 (1.86)	3.18 (4.31)
	BoostMetric-L, TC	**2.10**	**2.13 (0.41)**	2.48	7.71 (1.68)	9.57 (3.18)	3.85 (4.05)	3.64 (2.87)
	BoostMetric-L, MP, TC	2.11	2.10 (0.42)	**2.36**	7.15 (1.32)	**8.49 (3.71)**	3.08 (3.03)	**2.73 (2.35)**
Comp. Time	LMNN	10.98h	20s	1249s	896s	5s	2s	2s
	ITML	0.41h	72s	55s	5970s	8s	4s	4s
	BoostMetric-E	2.83h	144s	3s	628s	less than 1s	2s	less than 1s
	BoostMetric-L	0.89h	65s	34s	256s	less than 1s	2s	less than 1s

Table 1: Comparison of test classification error rates (%) of a 3-nearest neighbor classifier on benchmark data sets. Results of NCA are not available either because the algorithm does not converge or due to the out-of-memory problem. BoostMetric-E indicates BOOSTMETRIC with the exponential loss and BoostMetric-L is BOOSTMETRIC with the logistic loss; both use stage-wise optimization. "MP" means Multiple-Pass BOOSTMETRIC and "TC" is BOOSTMETRIC with totally corrective optimization. We report computational time as well.

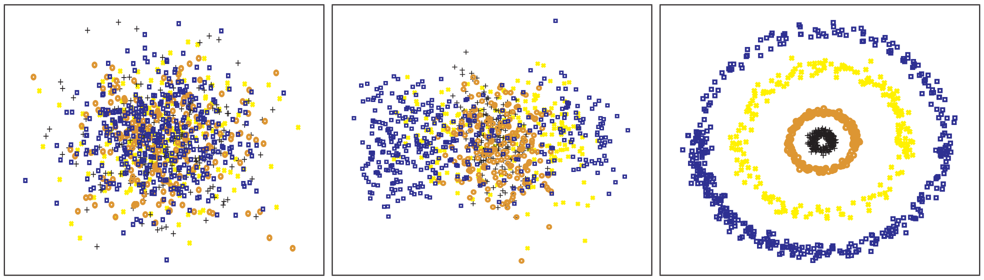


Figure 1: The data are projected into 2D with PCA (left), LDA (middle) and BOOSTMETRIC (right). Both PCA and LDA fail to recover the data structure. The local structure of the data is preserved after projection by BOOSTMETRIC.

4.1 An Illustrative Example

We demonstrate a data visualization problem on an artificial toy data set (concentric circles) in Figure 1. The data set has four classes. The first two dimensions follow concentric circles while the left eight dimensions are all random Gaussian noise. In this experiment, 9000 triplets are generated for training. When the scale of the noise is large, PCA fails find the first two informative dimensions. LDA fails too because clearly each class does not follow a Gaussian distraction and their centers overlap at the same point. The proposed BOOSTMETRIC algorithm find the informative features. The eigenvalues of $\mathbf{X}$ learned by BOOSTMETRIC are $\{0.542, 0.414, 0.007, 0, \cdots, 0\}$, which indicates that BOOSTMETRIC successfully reveals the data's underlying 2D structure. We have used the exponential loss in this experiment.

4.2 Classification on Benchmark Data Sets

We evaluate BOOSTMETRIC on 7 data sets of different sizes. Some of the data sets have very high dimensional inputs. We use PCA to decrease the dimensionality before training on these data sets (MNIST, USPS and yFaces). PCA pre-processing helps to eliminate noises and speed up computation. Table 1 summarizes the data sets in detail. We have used USPS and MNIST handwritten digits, Yale face recognition data sets, and a few UCI machine learning data sets.[2]

Experimental results are obtained by averaging over 10 runs (except for large data sets MNIST and Letter). We randomly split the data sets for each run. We have used the same mechanism to generate training triplets as described in Weinberger et al. (2005). Briefly, for each training point $\mathbf{a}_i$, k nearest neighbors that have same labels as y_i (targets), as well as k nearest neighbors that have different labels from y_i (imposers) are found. We then construct triplets from $\mathbf{a}_i$ and its corresponding targets and imposers. For all the data sets, we have set $k = 3$ (3-nearest-neighbor). We have compared our method against a few methods: RCA (Bar-Hillel et al., 2005), NCA (Goldberger et al., 2004), ITML (Davis et al., 2007) and LMNN (Weinberger et al., 2005). Also in Table 1, "Euclidean" is the baseline algorithm that uses the standard Euclidean distance. The codes for these compared algorithms are downloaded from the corresponding author's website. Experiment setting for LMNN follows Weinberger et al. (2005). The slack variable parameter for ITML is tuned using

[2] UCI data sets can be found at `http://archive.ics.uci.edu/ml/`.

v	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}
Bal	8.98 (2.59)	8.88 (2.52)	8.88 (2.52)	8.88 (2.52)	8.93 (2.52)
B-Cancer	2.11 (0.69)	2.11 (0.69)	2.11 (0.69)	2.11 (0.69)	2.11 (0.69)
Diabetes	26.0 (1.33)	26.0 (1.33)	26.0 (1.33)	26.0 (1.34)	26.0 (1.46)

Table 2: Test error (%) of a 3-nearest neighbor classifier with different values of the parameter v. Each experiment is run 10 times. We report the mean and variance. As expected, as long as v is sufficiently small, in a wide range it almost does not affect the final classification performance.

cross validation over the values $0.01, 0.1, 1, 10$ as in Davis et al. (2007). For BOOSTMETRIC, we have set $v = 10^{-7}$, the maximum number of iterations $J = 500$.

BOOSTMETRIC has different variants which use 1) the exponential loss (BOOSTMETRIC-E), 2) the logistic loss (BOOSTMETRIC-L), 3) multiple pass evaluation (MP) for updating triplets with the exponential and logistic loss, and 4) two optimization strategies, namely, stage-wise optimization and totally corrective optimization. The experiments are conducted by using Matlab and a C-mex implementation of the L-BFGS-B algorithm.

As reported in Table 1, we can conclude: 1) BOOSTMETRIC consistently improves the accuracy of kNN classification using Euclidean distance on most data sets. So learning a Mahalanobis metric based upon the large margin concept indeed leads to improvements in kNN classification. 2) BOOSTMETRIC outperforms other state-of-the-art algorithms in most cases (on 5 out of 7 data sets). LMNN is the second best algorithm on these 7 data sets statistically. LMNN's results are consistent with those given in Weinberger et al. (2005). ITML is faster than BOOSTMETRIC on most large data sets such as MNIST. However it has higher error rates than BOOSTMETRIC in our experiment. 3) NCA can only be run on a few small data sets. In general NCA does not perform well. Initialization is important for NCA because NCA's objective function is highly non-convex and can only find a local optimum.

In this experiment, LMNN solves for the global optimum (learning $\mathbf{X}$) except for the Wine data set. When the LMNN solver solves for $\mathbf{X}$ on the Wine data set, the error rate is large ($20.77\% \pm 14.18\%$). So instead we have solved for the projection matrix $\mathbf{L}$ on Wine. Also note that the number of training data on Iris, Wine and Bal in Weinberger et al. (2005) are different from our experiment. We have used these data sets from UCI. For the experiment on MNIST, if we deskew the handwritten digits data first as in Weinberger and Saul (2009), the final accuracy can be slightly improved. Here we have not deskewed the data.

4.2.1 INFLUENCE OF v

Previously, we claim that the stage-wise version of BOOSTMETRIC is parameter-free like AdaBoost. However, we do have a parameter v. Actually, AdaBoost simply set $v = 0$. The coordinate-wise gradient descent optimization strategy of AdaBoost leads to an ℓ_1-norm regularized maximum margin classifier (Rosset et al., 2004). It is shown that AdaBoost minimizes its loss criterion with an ℓ_1 constraint on the coefficient vector. Given the similarity of the optimization of BOOSTMETRIC with AdaBoost, we conjecture that BOOSTMETRIC has the same property. Here we empirically prove that *as long as v is sufficiently small, the final performance is not affected by the value of v*. We have set v from 10^{-8} to 10^{-4} and run BOOSTMETRIC on 3 UCI data sets. Table 2 reports the final 3NN classification error with different v. The results are nearly identical.

For the totally corrective version of BOOSTMETRIC, similar results are observed. Actually for LMNN, it was also reported that the regularization parameter does not have a significant impact on the final results in a wide range (Weinberger and Saul, 2009).

4.2.2 COMPUTATIONAL TIME

As we discussed, one major issue in learning a Mahalanobis distance is heavy computational cost because of the semidefiniteness constraint.

We have shown the running time of the proposed algorithm in Table 1 for the classification tasks.[3] Our algorithm is generally fast. Our algorithm involves matrix operations and an EVD for finding its largest eigenvalue and its corresponding eigenvector. The time complexity of this EVD is $O(D^2)$ with D the input dimensions. We compare our algorithm's running time with LMNN in Figure 2 on the artificial data set (concentric circles). Our algorithm is stage-wise BOOSTMETRIC with the exponential loss. We vary the input dimensions from 50 to 1000 and keep the number of triplets fixed to 250. LMNN does not use standard interior-point SDP solvers, which do not scale well. Instead LMNN heuristically combines sub-gradient descent in both the matrices $\mathbf{L}$ and $\mathbf{X}$. At each iteration, $\mathbf{X}$ is projected back onto the p.s.d. cone using EVD. So a full EVD with time complexity $O(D^3)$ is needed. Note that LMNN is much faster than SDP solvers like CSDP (Borchers, 1999). As seen from Figure 2, when the input dimensions are low, BOOSTMETRIC is comparable to LMNN. As expected, when the input dimensions become large, BOOSTMETRIC is significantly faster than LMNN. Note that our implementation is in Matlab. Improvements are expected if implemented in C/C++.

4.3 Visual Object Categorization

In the following experiments, unless otherwise specified, BOOSTMETRIC means the stage-wise BOOSTMETRIC with the exponential loss.

The proposed BOOSTMETRIC and the LMNN are further compared on visual object categorization tasks. The first experiment uses four classes of the Caltech-101 object recognition database (Fei-Fei et al., 2006), including Motorbikes (798 images), Airplanes (800), Faces (435), and Background-Google (520). The task is to label each image according to the presence of a particular object. This experiment involves both object categorization (Motorbikes versus Airplanes) and object retrieval (Faces versus Background-Google) problems. In the second experiment, we compare the two methods on the MSRC data set including 240 images.[4] The objects in the images can be categorized into nine classes, including *building, grass, tree, cow, sky, airplane, face, car and bicycle*. Different from the first experiment, each image in this database often contains multiple objects. The regions corresponding to each object have been manually pre-segmented, and the task is to label each region according to the presence of a particular object. Some examples are shown in Figure 3.

[3] We have run all the experiments on a desktop with an Intel Core$^{\text{TM}}$2 Duo CPU, 4G RAM and Matlab 7.7 (64-bit version).

[4] See `http://research.microsoft.com/en-us/projects/objectclassrecognition/`.

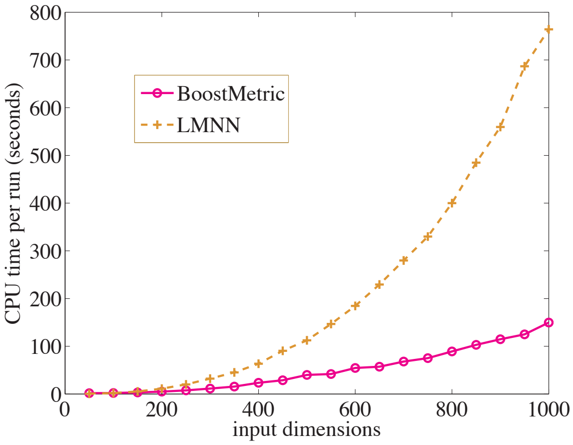


Figure 2: Computation time of the proposed BOOSTMETRIC (stage-wise, exponential loss) and the LMNN method versus the input data's dimensions on an artificial data set. BOOSTMETRIC is faster than LMNN with large input dimensions because at each iteration BOOSTMETRIC only needs to calculate the largest eigenvector and LMNN needs a full eigen-decomposition.

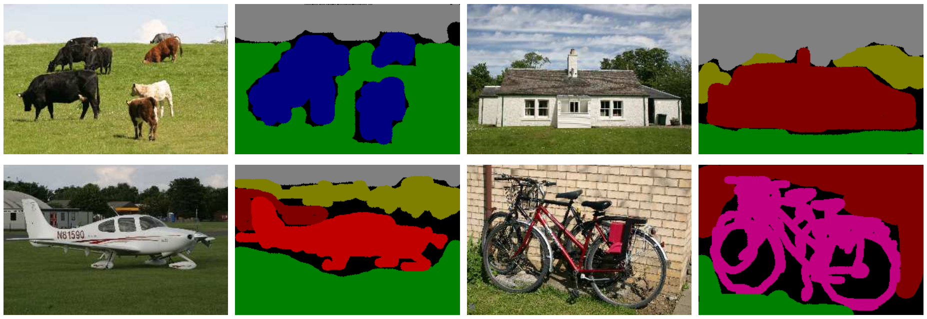


Figure 3: Examples of the images in the MSRC data set and the pre-segmented regions labeled using different colors.

4.3.1 EXPERIMENT ON THE CALTECH-101 DATA SET

For each image of the four classes, a number of interest regions are identified by the Harris-affine detector (Mikolajczyk and Schmid, 2004) and each region is characterized by the SIFT descriptor (Lowe, 2004). The total number of interest regions extracted from the four classes are about 134,000, 84,000, 57,000, and 293,000, respectively. To accumulate statistics, the images of two involved object classes are randomly split as 10 pairs of training/test subsets. Restricted to the images in a training subset (those in a test subset are only used for test), their local descriptors are clustered to form visual words by using k-means clustering. Each image is then represented by a histogram containing the number of occurrences of each visual word.

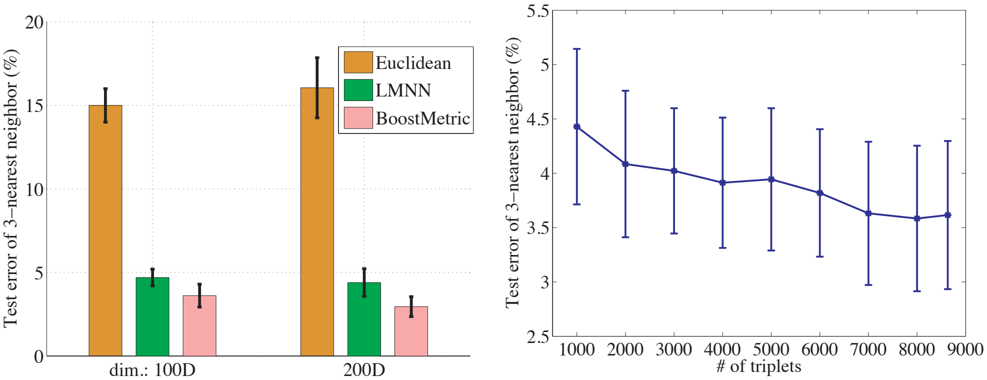


Figure 4: Test error (3-nearest neighbor) of BOOSTMETRIC on the Motorbikes versus Airplanes data sets. The second plot shows the test error against the number of training triplets with a 100-word codebook.

Motorbikes versus Airplanes This experiment discriminates the images of a motorbike from those of an airplane. In each of the 10 pairs of training/test subsets, there are 959 training images and 639 test images. Two visual codebooks of size 100 and 200 are used, respectively. With the resulting histograms, the proposed BOOSTMETRIC and the LMNN are learned on a training subset and evaluated on the corresponding test subset. Their averaged classification error rates are compared in Figure 4 (left). For both visual codebooks, the proposed BOOSTMETRIC achieves lower error rates than the LMNN and the Euclidean distance, demonstrating its superior performance. We also apply a linear SVM classifier with its regularization parameter carefully tuned by 5-fold cross-validation. Its error rates are $3.87\% \pm 0.69\%$ and $3.00\% \pm 0.72\%$ on the two visual codebooks, respectively. In contrast, a 3NN with BOOSTMETRIC has error rates $3.63\% \pm 0.68\%$ and $2.96\% \pm 0.59\%$. Hence, the performance of the proposed BOOSTMETRIC is comparable to the state-of-the-art SVM classifier. Also, Figure 4 (right) plots the test error of the BOOSTMETRIC against the number of triplets for training. The general trend is that more triplets lead to smaller errors.

Faces versus Background-Google This experiment uses the two object classes as a retrieval problem. The target of retrieval is face images. The images in the class of Background-Google are randomly collected from the Internet and they represent the non-target class. BOOSTMETRIC is first learned from a training subset and retrieval is conducted on the corresponding test subset. In each of the 10 training/test subsets, there are 573 training images and 382 test images. Again, two visual codebooks of size 100 and 200 are used. Each face image in a test subset is used as a query, and its distances from other test images are calculated by the proposed BoostMetric, LMNN and the Euclidean distance, respectively. For each metric, the *Precision* of the retrieved top 5, 10, 15 and 20 images are computed. The *Precision* values from each query are averaged on this test subset and then averaged over the 10 test subsets. The retrieval precision of these metrics is shown in Figure 5 (with a codebook size 100). As we can see that the BOOSTMETRIC consistently attains the highest values on both visual codebooks, which again verifies its advantages over LMNN and Euclidean distance. With a codebook size 200, very similar results are obtained.

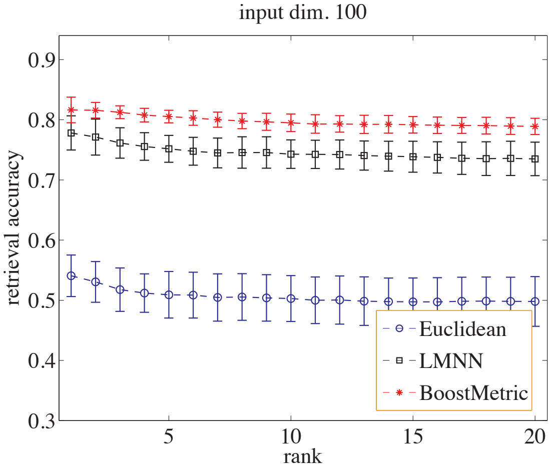


Figure 5: Retrieval accuracy of distance metric learning algorithms on the Faces versus Background-Google data set. Error bars show the standard deviation.

4.3.2 EXPERIMENT ON THE MSRC DATA SET

The 240 images of the MSRC database are randomly halved into 10 groups of training and test sets. Given a set of training images, the task is to predict the class label for each of the pre-segmented regions in a test image. We follow the work in Winn et al. (2005) to extract features and conduct experiments. Specifically, each image is converted from the RGB color space to the CIE Lab color space. First, three Gaussian low-pass filters are applied to the L, a, and b channels, respectively. The standard deviation σ of the filters are set to $1, 2$, and 4, respectively, and the filter size is defined as 4σ. This step produces 9 filter responses for each pixel in an image. Second, three Laplacian of Gaussian (LoG) filters are applied to the L channel only, with $\sigma = 1, 2, 4, 8$ and the filter size of 4σ. This step gives rise to 4 filter responses for each pixel. Lastly, the first derivatives of the Gaussian filter with $\sigma = 2, 4$ are computed from the L channel along the row and column directions, respectively. This results in 4 more filter responses. After applying this set of filter banks, each pixel is represented by a 17-dimensional feature vectors. All the feature vectors from a training set are clustered using the k-means clustering with a Mahalanobis distance.[5] By setting k to 2000, a visual codebook of 2000 visual words is obtained. We implement the word-merging approach in Winn et al. (2005) and obtain a compact and discriminative codebook of 300 visual words. Each pre-segmented object region is then represented as a 300-dimensional histogram.

The proposed BOOSTMETRIC is compared with the LMNN algorithm as follows. With 10 nearest neighbors information, about $20,000$ triplets are constructed and used to train the BOOSTMETRIC. To ensure convergence, the maximum number of iterations is set as 5000 in the optimization of training BOOSTMETRIC. The training of LMNN follows the default setting. kNN classifiers with the two learned Mahalanobis distances and the Euclidean distance are applied to each training and test group to categorize an object region. The categorization error rate on each test group is summarized in Table 3. As expected, both learned Mahalanobis distances achieve superior categorization

[5] Note that this Mahalanobis distance is different from the one that we are going to learn with the BOOSTMETRIC.

group index	Euclidean	LMNN	BOOSTMETRIC
1	9.19	6.71	4.59
2	5.78	3.97	3.25
3	6.69	2.97	2.60
4	5.54	3.69	4.43
5	6.52	5.80	4.35
6	7.30	4.01	3.28
7	7.75	2.21	2.58
8	7.20	4.17	4.55
9	6.13	3.07	4.21
10	8.42	5.13	5.86
average:	7.05	4.17	**3.97**
standard devision:	1.16	1.37	**1.03**

Table 3: Comparison of the categorization performance.

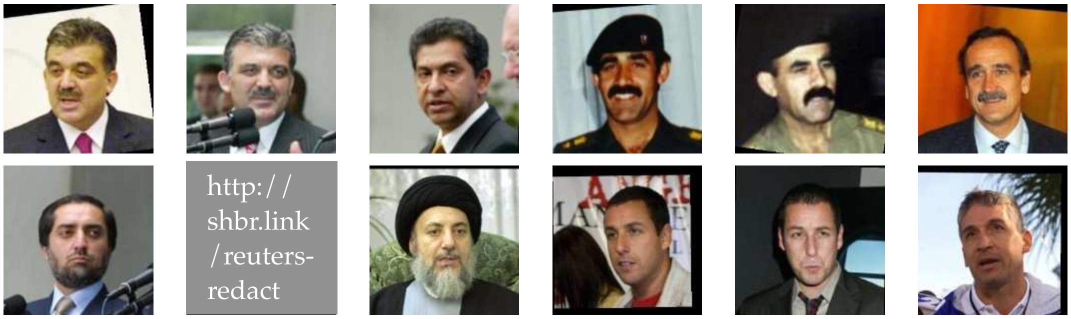


Figure 6: Four generated triplets based on the pairwise information provided by the LFW data set. For the three images in each triplet, the first two belong to the same individual and the third one is a different individual.

performance to the Euclidean distance. Moreover, the proposed BOOSTMETRIC achieves better performance than the LMNN, as indicated by its lower average categorization error rate and the smaller standard deviation. Also, the *k*NN classifier using the proposed BOOSTMETRIC achieves comparable or even higher categorization performance than those reported in Winn et al. (2005). Besides the categorization performance, we compare the computational efficiency of the BOOSTMETRIC and the LMNN in learning a Mahalanobis distance. The computational time result is based on the Matlab codes for both methods. In this experiment, the average time cost by the BOOSTMETRIC for learning the Mahalanobis distance is 3.98 hours, whereas the LMNN takes about 8.06 hours to complete this process. Hence, the proposed BOOSTMETRIC has a shorter training process than the LMNN method. This again demonstrates the computational advantage of the BOOSTMETRIC over the LMNN method.

4.4 Unconstrained Face Recognition

We use the "labeled faces in the wild" (LFW) data set (Huang et al., 2007) for face recognition in this experiment.

number of triplets	100D	200D	300D	400D
3,000	80.91 (1.76)	82.39 (1.73)	83.40 (1.46)	83.64 (1.66)
6,000	81.13 (1.76)	82.59 (1.84)	83.58 (1.25)	83.70 (1.73)
9,000	81.01 (1.69)	82.63 (1.68)	83.65 (1.70)	83.72 (1.47)
12,000	81.06 (1.63)	83.00 (1.38)	83.60 (1.89)	83.57 (1.47)
15,000	81.10 (1.71)	82.78 (1.83)	83.69 (1.62)	83.80 (1.85)
18,000	81.37 (2.15)	83.19 (1.76)	83.60 (1.66)	83.81 (1.55)

Table 4: Comparison of the face recognition accuracy (%) of our proposed BOOSTMETRIC on the LFW data set by varying the PCA dimensionality and the number of triplets for each fold.

This is a data set of unconstrained face images, which has a large range of variations seen in real world, including $13,233$ images of $5,749$ people collected from news articles on Internet. The face recognition task here is *pair matching*—given two face images, to determine if these two images are of the same individual. So we classify unseen pairs to determine whether each image in the pair indicates the same individual or not, by applying M*k*NN of Guillaumin et al. (2009) instead of *k*NN.

Features of face images are extracted by computing 3-scale, 128-dimensional SIFT descriptors (Lowe, 2004), which center on 9 points of facial features extracted by a facial feature descriptor, same as described in Guillaumin et al. (2009). PCA is then performed on the SIFT vectors to reduce the dimension to between 100 and 400.

Simple recognition systems with a single descriptor Table 4 shows our BOOSTMETRIC's performance by varying PCA dimensionality and the number of triplets. Increasing the number of training triplets gives slight improvement of recognition accuracy. The dimension after PCA has more impact on the final accuracy for this task.

In Figure 7, we have drawn ROC curves of other algorithms for face recognition. To obtain our ROC curve, M*k*NN has moved the threshold value across the distributions of match and mismatch similarity scores. Figure 7 (a) shows methods that use a single descriptor and a single classifier only. As can be seen, our system using BOOSTMETRIC outperforms all the others in the literature with a very small computational cost.

Complex recognition systems with one or more descriptors Figure 7 (b) plots the performance of more complicated recognition systems that use hybrid descriptors or combination of classifiers. See Table 5 for details. We can see that the performance of our BOOSTMETRIC is close to the state-of-the-art.

In particular, BOOSTMETRIC outperforms the method of Guillaumin et al. (2009), which has a similar pipeline but uses LMNN for learning a metric. This comparison also confirms the importance of learning an appropriate metric for vision problems.

5. Conclusion

We have presented a new algorithm, BOOSTMETRIC, to learn a positive semidefinite metric using boosting techniques. We have generalized AdaBoost in the sense that the weak learner of BOOSTMETRIC is a matrix, rather than a classifier. Our algorithm is simple and efficient. Experiments show its better performance over a few state-of-the-art existing metric learning methods. We are currently combining the idea of on-line learning into BOOSTMETRIC to make it handle even larger data sets.

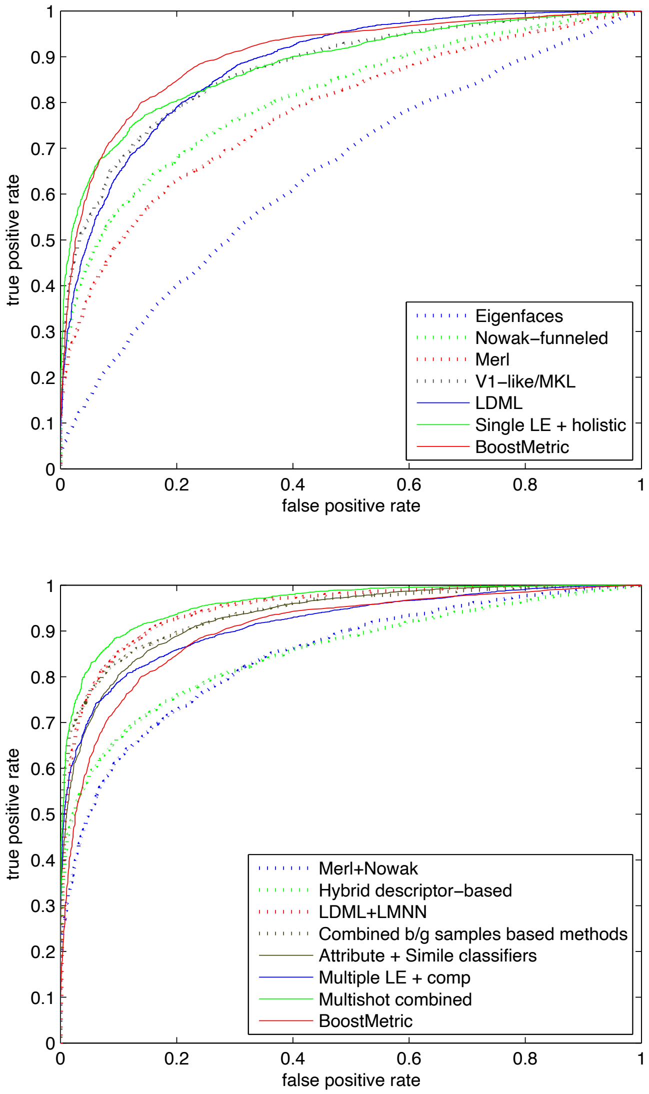


Figure 7: (top) ROC Curves that use a single descriptor and a single classifier, (bottom) ROC curves that use hybrid descriptors are plotted. Our BOOSTMETRIC with a single classifier is also plotted. Each point on the curves is the average over the 10 folds of rates for a fixed threshold.

We also want to learn a metric using BOOSTMETRIC in the semi-supervised, and multi-task learning setting. It has been shown in Weinberger and Saul (2009) that the classification performance can be improved by learning multiple local metrics. We will extend BOOSTMETRIC to learn multiple metrics. Finally, we will explore to generalize BOOSTMETRIC for solving more general semidefinite matrix learning problems in machine learning.

	single descriptor + single classifier	multiple descriptors/classifiers
Turk and Pentland (1991)	60.02 (0.79) 'Eigenfaces'	-
Nowak and Jurie (2007)	73.93 (0.49) 'Nowak-funneled'	-
Huang et al. (2008)	70.52 (0.60) 'Merl'	76.18 (0.58) 'Merl+Nowak'
Wolf et al. (2008)	-	78.47 (0.51) 'Hybrid descriptor-based'
Wolf et al. (2009)	72.02 -	86.83 (0.34) 'Combined b/g samples based'
Pinto et al. (2009)	79.35 (0.55) 'V1-like/MKL'	-
Taigman et al. (2009)	83.20 (0.77) -	**89.50 (0.40)** 'Multishot combined'
Kumar et al. (2009)	-	85.29 (1.23) 'attribute + simile classifiers'
Cao et al. (2010)	81.22 (0.53) 'single LE + holistic'	84.45 (0.46) 'multiple LE + comp'
Guillaumin et al. (2009)	83.2 (0.4) 'LDML'	87.5 (0.4) 'LMNN + LDML'
BOOSTMETRIC	**83.81 (1.55)** 'BOOSTMETRIC' on SIFT	-

Table 5: Test accuracy in percentage (mean and standard deviation) on the LFW data set. ROC curve labels in Figure 7 are described here with details.

References

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *J. Machine Learning Research*, 6:937–965, 2005.

O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.

B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2010.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operation Research*, 8 (1):101–111, 1960.

J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Int'l Conf. Machine Learning*, pages 209–216, Corvalis, Oregon, 2007. ACM Press.

A. Demiriz, K.P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.

P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a compressed approximate matrix decomposition. *SIAM J. Computing*, 36:2006, 2004.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):594–611, April 2006.

P. A. Fillmore and J. P. Williams. Some convexity theorems for matrices. *Glasgow Math. Journal*, 12:110–117, 1971.

A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Proc. Advances in Neural Information Processing Systems*, 2005.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Proc. Advances in Neural Information Processing Systems*. MIT Press, 2004.

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Proc. IEEE Int'l Conf. Computer Vision*, 2009.

T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996.

X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using Laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

G. B. Huang, M. J. Jones, and E. Learned-Miller. LFW results using a combined nowak plus merl recognizer. In *Faces in Real-Life Images Workshop in Euro. Conf. Computer Vision*, 2008.

B. Jian and B. C. Vemuri. Metric learning using Iwasawa decomposition. In *Proc. IEEE Int'l Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007. IEEE.

M. Krein and D. Milman. On extreme points of regular convex sets. *Studia Mathematica*, 9:133–138, 1940.

N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proc. IEEE Int'l Conf. Computer Vision*, 2009.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Computer Vision*, 60 (2):91–110, 2004.

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operation Research*, 53 (6):1007–1023, 2005.

K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int'l J. Computer Vision*, 60(1):63–86, 2004.

E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

M. L. Overton and R. S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. Matrix Analysis and Application*, 13(1):41–45, 1992.

N. Pinto, J. J. DiCarlo, and D. D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2009.

R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proc. ACM SIGKDD Int'l Conf. Knowledge discovery and Data Mining*, pages 367–373. ACM, 2006.

S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *J. Machine Learning Research*, 5:941–973, 2004.

R. E. Schapire. Theoretical views of boosting and applications. In *Proc. Int'l Conf. Algorithmic Learning Theory*, pages 13–25, London, UK, 1999. Springer-Verlag.

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Proc. Advances in Neural Information Processing Systems*. MIT Press, 2003.

C. Shen and H. Li. On the dual formulation of boosting algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(12):2216–2231, 2010.

C. Shen, A. Welsh, and L. Wang. PSDBoost: Matrix-generation linear programming for positive semidefinite matrices learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Proc. Advances in Neural Information Processing Systems*, pages 1473–1480, Vancouver, B.C., Canada, December 2008. MIT Press.

C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning with boosting. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Proc. Advances in Neural Information Processing Systems*, pages 1651–1659, Vancouver, B.C., Canada, December 2009. MIT Press.

Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proc. British Machine Vision Conf.*, 2009.

K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. *J. Machine Learning Research*, 6:995–1018, December 2005.

M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 1991.

H. Wang, H. Huang, and C. Ding. Discriminant Laplacian embedding. In *Proc. AAAI Conf. Artificial Intelligence*, pages 618–623, 2010a.

S. Wang and R. Jin. An information geometry approach for distance metric learning. In *Proc. Int'l Conf. Artificial Intelligence and Statistics*, pages 591–598, 2009.

Z. Wang, Y. Hu, and L.-T. Chia. Image-to-class distance metric learning for image classification. In *Proc. Euro. Conf. Computer Vision*, volume Lecture Notes in Computer Science 6311/2010, pages 706–719, 2010b.

M. K. Warmuth, J. Liao, and G. Rätsch. Totally corrective boosting algorithms that maximize the margin. In *Int'l Conf. Machine Learning*, pages 1001–1008, Pittsburgh, Pennsylvania, 2006.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int'l J. Computer Vision*, 70(1):77–90, 2006.

K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Machine Learning Research*, 10:207–244, 2009.

K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proc. Advances in Neural Information Processing Systems*, pages 1473–1480, 2005.

J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1800–1807, 2005.

L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in Euro. Conf. Computer Vision*, 2008.

L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Proc. Asian Conf. Computer Vision*, 2009.

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Proc. Advances in Neural Information Processing Systems*. MIT Press, 2002.

J. Yu, J. Amores, N. Sebe, P. Radeva, and Q. Tian. Distance learning for similarity estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(3):451–462, 2008.

T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. Information Theory*, 49(3):682–691, 2003.

C. Zhu, R. H. Byrd, and J. Nocedal. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. *ACM Trans. Mathematical Software*, 23(4):550–560, 1997.

Journal of Machine Learning Research 13 (2012) 1037-1057 Submitted 6/11; Revised 1/12; Published 4/12

Consistent Model Selection Criteria on High Dimensions

Yongdai Kim YDKIM@STATS.SNU.AC.KR
Department of Statistics
Seoul National University
Seoul 151-742, Korea

Sunghoon Kwon SHKWON0522@GMAIL.COM
School of Statistics
University of Minnesota
Minneapolis, MN 55455, USA

Hosik Choi CHOI.HOSIK@GMAIL.COM
Department of Informational Statistics
Hoseo University
Chungnam 336-795, Korea

Editor: Xiaotong Shen

Abstract

Asymptotic properties of model selection criteria for high-dimensional regression models are studied where the dimension of covariates is much larger than the sample size. Several sufficient conditions for model selection consistency are provided. Non-Gaussian error distributions are considered and it is shown that the maximal number of covariates for model selection consistency depends on the tail behavior of the error distribution. Also, sufficient conditions for model selection consistency are given when the variance of the noise is neither known nor estimated consistently. Results of simulation studies as well as real data analysis are given to illustrate that finite sample performances of consistent model selection criteria can be quite different.

Keywords: model selection consistency, general information criteria, high dimension, regression

1. Introduction

Model selection is a fundamental task for high-dimensional statistical modeling where the number of covariates can be much larger than the sample size. In such cases, classical model selection criteria such as the Akaike information criterion or AIC (Akaike, 1973), the Bayesian information criterion or BIC (Schwarz, 1978) and cross validations or generalized cross validation (Craven and Wahba, 1979; Stone, 1974) tend to select more variables than necessary. See, for example, Broman and Speed (2002) and Casella et al. (2009). Also, Yang and Barron (1998) discussed severe selection bias of AIC which damages predictive performance for high-dimensional models.

Recently, various model selection criteria for high-dimensional models have been introduced. Wang et al. (2009) proposed a modified BIC which is consistent when the dimension of covariates is diverging slower than the sample size. Here, the consistency of a model selection criterion means that the probability of the selected model being equal to the true model converges to 1. See Section 2 for a rigorous definition. The extended BIC by Chen and Chen (2008) and corrected RIC by Zhang and Shen (2010) are shown to be consistent even when the dimension of covariates is larger than the sample size. Some sparse penalized approaches including the LASSO (Least Absolute Shrinkage and Selection Operator) (Tibshirani, 1996) and SCAD (Smoothly Clipped Absolute Deviation) (Fan and Li, 2001) are proven to be consistent for high-dimensional models. See Zhao and Yu (2006) for the LASSO and Kim et al. (2008) for the SCAD.

In this paper, we study asymptotic properties of a large class of model selection criteria based on the generalized information criterion (GIC) considered by Shao (1997). The class of GICs is large enough to include many well known model selection criteria such as the AIC, BIC, modified BIC by Wang et al. (2009), risk inflation criterion (RIC) by Foster and George (1994), modified risk inflation criterion (MRIC) by Foster and George (1994), corrected RIC by Zhang and Shen (2010). Also, as we will show, the extended BIC by Chen and Chen (2008) is asymptotically equivalent to a GIC.

We give sufficient conditions for a given GIC to be consistent. Our sufficient conditions are general enough to include cases where the error distribution can be other than Gaussian and the variance of the error distribution is not consistently estimated. For a case of the Gaussian error distribution with consistent estimator of the variance, our sufficient conditions include most of the previously proposed consistent model selection criteria such as the modified BIC (Wang et al., 2009), extended BIC (Chen and Chen, 2008) and corrected RIC (Zhang and Shen, 2010).

For high-dimensional models, it is not practically feasible to find the best model among all possible submodels since the number of submodels are too large. A simple remedy is to find a sequence of submodels with increasing complexities (e.g., increasing number of covariates) and find the best model among them using a given model selection criterion. Examples of constructing a sequence of submodels are the forward selection procedure and solution paths of penalized regression approaches. Our sufficient conditions are still valid as long as the sequence of submodels includes the true model with probability converging to 1. We discuss more on these issues in Section 4.1.

The paper is organized as follows. In Section 2, the GIC is introduced. In Section 3, sufficient conditions for the consistency of GICs are given. Various remarks about application of GICs to real data analysis are given in Section 4. In Section 5, results of simulations as well as a real data analysis are presented, and concluding remarks follow in Section 6.

2. Generalized Information Criterion

Let $\mathcal{L} = \{(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)\}$ be a given data set of independent pairs of response and covariates, where $y_i \in R$ and $\mathbf{x}_i \in R^{p_n}$. Suppose the true regression model for $(y, \mathbf{x})$ is given as

$$y = \mathbf{x}'\beta^* + \varepsilon,$$

where $\beta^* \in R^{p_n}, \mathrm{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$. For simplicity, we assume that σ^2 is known. For unknown σ^2, see Section 4.2.

Let $Y_n = (y_1, \dots, y_n)'$ and $\mathbf{X}_n$ be the $n \times p_n$ dimensional design matrix whose jth column is $X_n^j = (x_{1j}, \dots, x_{nj})'$. For given $\beta \in R^{p_n}$, let

$$R_n(\beta) = \|Y_n - \mathbf{X}_n\beta\|^2,$$

where $\|\cdot\|$ is the Euclidean norm. For a given subset $\pi \subset \{1, \dots, p_n\}$, let

$$\hat{\beta}_\pi = \mathrm{argmin}_{\beta:\beta_j=0, j\in\pi^c} R_n(\beta).$$

For a given sequence of positive numbers $\{\lambda_n\}$, the GIC indexed by $\{\lambda_n\}$, denoted by GIC_{λ_n}, gives a sequence of random subsets $\hat{\pi}_{\lambda_n}$ of $\{1,\ldots,p_n\}$ defined as

$$\hat{\pi}_{\lambda_n} = \mathrm{argmin}_{\pi\subset\{1,\ldots,p_n\}} R_n(\hat{\beta}_\pi) + \lambda_n|\pi|\sigma^2,$$

where $|\pi|$ is the cardinality of π. The AIC corresponds to $\lambda_n = 2$, the BIC to $\lambda_n = \log n$, the RIC of Foster and George (1994) to $\lambda_n = 2\log p_n$, the RIC of Zhang and Shen (2010) to $\lambda_n = 2(\log p_n + \log\log p_n)$. Shao (1997) studied the asymptotic properties of the GIC focusing on the AIC and BIC.

When p_n is large, it would not be wise to search all possible subsets of $\{1,\ldots,p_n\}$. Instead, we set an upper bound on the cardinality of π, say s_n and search the optimal model among submodels whose cardinalities are smaller than s_n. Chen and Chen (2008) considered a similar model selection procedure. Let $\mathcal{M}^{s_n} = \{\pi \subset \{1,\ldots,p_n\} : |\pi| \le s_n\}$. We define the restricted GIC_{λ_n} as

$$\hat{\pi}_{\lambda_n} = \mathrm{argmin}_{\pi\in\mathcal{M}^{s_n}} R_n(\hat{\beta}_\pi) + \lambda_n|\pi|\sigma^2. \tag{1}$$

The restricted GIC is the same as the GIC if $s_n = p_n$. In the following, we will only consider the restricted GIC and suppress the term "restricted" unless there is any confusion.

3. Consistency of GIC on High Dimensions

Let $\pi_n^* = \{j : |\beta_j^*| \neq 0\}$. We say that the GIC_{λ_n} is consistent if

$$\Pr(\hat{\pi}_{\lambda_n} = \pi_n^*) \to 1$$

as $n \to \infty$. In this section, we prove the consistency of the GIC_{λ_n} under regularity conditions.

For a given subset π of $\{1,\ldots,p_n\}$, let $\mathbf{X}_\pi = (X_n^j, j \in \pi)$ be the $n \times |\pi|$ matrix whose columns consist of $X_n^j, j \in \pi$. For a given symmetric matrix $\mathbf{A}$, let $\xi(\mathbf{A})$ be the smallest eigenvalue of $\mathbf{A}$.

3.1 Regularity Conditions

We assume the following regularity conditions.

- A1 : There exists a positive constant M_1 such that $X_n^{j'} X_n^j/n \le M_1$ for all $j = 1,\ldots,p_n$ and all n.
- A2 : There is a positive constant M_2 such that $\xi(\mathbf{X}_{\pi_n^*}'\mathbf{X}_{\pi_n^*}/n) \ge M_2$ for all n.
- A3 : There exist positive constants c_1 and M_3 such that $0 \le c_1 < 1/2$ and $\rho_n \ge M_3 n^{-c_1}$, where

$$\rho_n = \inf_{\pi:|\pi|\le s_n} \xi(\mathbf{X}_\pi'\mathbf{X}_\pi/n).$$

- A4 : There exist positive constants c_2 and M_4 such that $2c_1 < c_2 \le 1$ and

$$n^{(1-c_2)/2} \min_{j\in\pi_n^*} |\beta_j^*| \ge M_4.$$

- A5 : $q_n = O(n^{c_3})$ for some $0 \le c_3 < c_2$, and $q_n \le s_n$, where $q_n = |\pi_n^*|$.

Condition A1 assumes that the covariates are bounded. Condition A2 means that the design matrix of the true model is well posed. Condition A3 is called the sparse Riesz condition and used in Chen and Chen (2008), Zhang (2010) and Kim and Kwon (2012). Condition A4 and A5 allow the nonzero regression coefficients to converge to 0 and the number of signal variables to diverge, respectively.

Remark 1 *Condition A3 implies that* $s_n \leq n$.

3.2 The Main Theorem

The following theorem proves consistency of the GIC_{λ_n}. The proofs are deferred to Appendix.

Theorem 2 *Suppose* $\text{E}(\varepsilon^{2k}) < \infty$ *for some integer* $k > 0$. *If* $\lambda_n = o(n^{c_2-c_1})$ *and* $p_n/(\lambda_n\rho_n)^k \to 0$, *then the* GIC_{λ_n} *is consistent.*

In Theorem 2, p_n can diverge only polynomially fast in n since $p_n = o(\lambda_n^k) = o(n^{kc_2})$. Since k can be considered as a degree of tail lightness of the error distribution, we can conclude that the lighter the tail of the error distribution is, the more covariates the GIC_{λ_n} is consistent with. When ε is Gaussian, the following theorem proves that the GIC_{λ_n} can be consistent when p_n diverges exponentially fast.

Theorem 3 *Suppose* $\varepsilon \sim N(0,\sigma^2)$. *If* $\lambda_n = o(n^{c_2-c_1}), s_n \log p_n = o(n^{c_2-c_1})$ *and* $\lambda_n - 2\log p_n - \log\log p_n \to \infty$, *then the* GIC_{λ_n} *is consistent.*

In the following, we give three examples for (i) fixed p_n, (ii) polynomially diverging p_n and (iii) exponentially diverging p_n. For simplicity, we let $c_1 = 0$ (i.e., $\rho_n \geq M_3 > 0$), $c_2 = 1$ (i.e., $\min_{j\in\pi_n^*} |\beta_j^*| > 0$) and $c_3 = 0$ (i.e., q_n is fixed). In addition, we let s_n be fixed.

Example 1 *Consider a standard case where* p_n *is fixed and* n *goes to infinity. Theorem 2 implies that the* GIC_{λ_n} *is consistent if* $\lambda_n/n \to 0$ *and* $\lambda_n \to \infty$ *regardless of the tail lightness (i.e.,* k*) of the error distribution, provided the variance exists. The BIC, which is the GIC with* $\lambda_n = \log n$, *satisfies these conditions and hence is consistent. Note that the AIC does not satisfy the conditions in Theorem 2. Any GIC with* $\lambda_n = n^c, 0 < c < 1$ *is consistent, which suggests that the class of consistent model selection criteria is quite large. See Shao (1997) for more discussions.*

Example 2 *Consider a case of* $p_n = n^\gamma, \gamma > 0$. *The GIC with* $\lambda_n = n^\xi, 0 < \xi < 1$ *and* $\gamma < k\xi$ *is consistent. That is, for larger* p_n, *we need larger* λ_n *for consistency, which is reasonable because we need to be more careful not to overfit when* p_n *is large. When the error distribution is Gaussian, Theorem 3 can be compared with other previous results of consistency. First, the BIC (i.e., the GIC with* $\lambda_n = \log n$*) is consistent when* $\gamma < 1/2$. *For* $0 < \gamma < 1$, *Theorem 3 implies that the modified BIC of Wang et al. (2009), which is a GIC with* $\lambda_n = \log\log p_n \log n$, *is consistent. Chen and Chen (2008) proposed a model selection criterion called the extended BIC given by*

$$\hat{\pi}^{eBIC} = \text{argmin}_{\pi\subset\{1,\ldots,p_n\},|\pi|\leq K} R_n(\hat{\beta}_\pi) + |\pi|\sigma^2 \log n + 2\kappa\sigma^2 \log \binom{p_n}{|\pi|}$$

for some $K > 0$ *and* $0 \leq \kappa \leq 1$, *and proved that the extended BIC is consistent when* $\kappa > 1 - 1/(2\gamma)$. *Since* $\log\binom{p_n}{|\pi|} \asymp |\pi| \log p_n$ *for* $|\pi| \leq K$, *we have*

$$|\pi|\sigma^2 \log n + 2\gamma\sigma^2 \log \binom{p_n}{|\pi|} \asymp (\log n + 2\kappa \log p_n)|\pi|\sigma^2.$$

Hence, Theorem 3 confirms the result of Chen and Chen (2008).

Example 3 *When the error distribution is Gaussian, the GIC can be consistent for exponentially increasing p_n (i.e., ultra-high dimensional cases). The GIC with $\lambda_n = n^{\xi}, 0 < \xi < 1$ is consistent when $p_n = O(\exp(\alpha n^{\gamma}))$ for $0 < \gamma < \xi$ and $\alpha > 0$. Also, it can be shown by Theorem 3 that the extended BIC with $\gamma = 1$ is consistent with $p_n = O(\exp(\alpha n^{\gamma}))$ for $0 < \gamma < 1/2$. The consistency of the corrected RIC of Zhang and Shen (2010) can be confirmed by Theorem 3, but the regularity conditions for Theorem 3 are more general than those of Zhang and Shen (2010).*

4. Remarks

Remarks regarding to applications of the GIC to real data analysis are given.

4.1 Construction of Sub-Models

For high-dimensional models, it is computationally infeasible to search the optimal model among all possible submodels. A simple remedy is to construct a sequence of submodels and select the optimal model among the sequence of submodels. Examples of constructing a sequence of submodels are the forward selection (Wang, 2009) and the solution path of a sparse penalized estimator obtained by, for example, the Lars algorithm (Efron et al., 2004) or the PLUS algorithm (Zhang, 2010). The following algorithm exemplifies the model selection procedure with the GIC and a sparse penalized regression approach.

- For a given sparse penalty $J_\eta(t)$ indexed by $\eta \geq 0$, find the solution path of a penalized estimator $\{\hat{\beta}(\eta) : \eta > 0\}$, where

$$\hat{\beta}(\eta) = \text{argmin}_{\beta} \left(R_n(\beta) + \sum_{j=1}^{p} J_\eta(|\beta_j|) \right).$$

  The LASSO corresponds to $J_\eta(t) = \eta t$ and the SCAD penalty corresponds to

$$\begin{aligned} J_\eta(t) &= \eta t I(0 \leq 0 < \eta) \\ &+ \left\{ \frac{a\eta(t-\eta) - (t^2 - \eta^2)/2}{a-1} + \eta^2 \right\} I(\eta \leq t < a\eta) \\ &+ \left\{ \frac{(a-1)\eta^2}{2} + \eta^2 \right\} I(t \geq a\eta) \end{aligned}$$

  for some $a > 2$.

- Let $S(\eta) = \{j : \hat{\beta}(\eta)_j \neq 0\}$ and $\Upsilon = \{\eta : S(\eta) \neq S(\eta-), |S(\eta)| \leq s_n\}$.
- Apply the GIC$_{\lambda_n}$ to $S(\eta), \eta \in \Upsilon$ to select the optimal model. That is, let $\hat{\pi}_{\lambda_n} = S(\eta^*)$ where

$$\eta^* = \text{argmin}_{\eta \in \Upsilon} \left(R_n(\hat{\beta}_\eta) + \lambda_n |S(\eta)| \right)$$

  and

$$\hat{\beta}_\eta = \text{argmin}_{\beta : \beta_j = 0, j \in S(\eta)^c} R_n(\beta).$$

It is easy to see that a consistent GIC is still consistent with a sequence of sub-models as long as the sequence of submodels includes the true model with probability converging to 1. For the LASSO solution path, Zhao and Yu (2006) proved the selection consistency under the irrepresentable condition, which is almost necessary (Zou, 2006). However, the irrepresentable condition is hardly satisfied for high-dimensional models. The consistency of the solution path of a nonconvex penalized estimator with either the SCAD penalty or minimax concave penalty is proved by Zhang (2010) and Kim and Kwon (2012). By combining Theorem 4 of Kim and Kwon (2012) and Theorem 2 of the current paper, we can prove the consistency of the GIC with the solution path of the SCAD penalty or minimax concave penalty, which is formally stated in the following theorem.

Theorem 4 *Condition A3 is replaced by A3', where*

- *A3': There exist positive constants c_1 and M_3 such that $0 \leq c_1 < 1/2$ and $\rho_n \geq M_3 n^{c_1/2}$.*

Suppose $\mathrm{E}(\varepsilon^{2k}) < \infty$ for some integer $k > 0$. If $p_n = o(n^{k(c_2/2-c_1)})$, the under the regularity conditions A1 to A5 with A3 being replaced by A3', the solution path of the SCAD or minimax concave penalty included the true model with probability converging to 1, and hence the GIC$_{\lambda_n}$ with $\lambda_n = o(n^{c_2-c_1})$ is consistent with the solution path of the SCAD or minimax concave penalty.

Remark 5 *Condition A3' is a technical modification needed for Theorem 4 of Kim and Kwon (2012). Note that A3 is weaker than A3', which is an advantage of using the l_0 penalty rather than nonconvex penalties which are linear around 0.*

Remark 6 *Theorem 3 can be modified similarly for the GIC with the solution path of the SCAD or minimax concave penalty, since Theorem 4 of Kim and Kwon (2012) can be modified accordingly for the Gaussian error distribution.*

4.2 Estimation of the Variance

To use the GIC in practice, we need to know σ^2. If σ^2 is unknown, we can replace it by its estimate. Theorems 2 and 3 are still valid as long as σ^2 is estimated consistently. When p_n is fixed, we can estimate σ^2 consistently by the mean squared error of the full model. For high-dimensional data, it is not obvious how to estimate σ^2. However, a weaker condition can be put on an estimator $\hat{\sigma}^2$ of σ^2 for the GIC to be consistent. Suppose that

$$0 < r_{inf} = \liminf \frac{\hat{\sigma}^2}{\sigma^2} \leq \limsup \frac{\hat{\sigma}^2}{\sigma^2} = r_{sup} < \infty \tag{2}$$

with probability 1. This condition essentially assumes that $\hat{\sigma}^2$ is neither too small nor too large. It is not difficult to show that Theorem 2 is still valid with $\hat{\sigma}^2$ satisfying (2). This, however, is not true for Theorem 3. A slightly weak version of Theorem 3 which only requires (2) is given in the following theorem.

Theorem 7 *Suppose $\varepsilon \sim N(0, \sigma^2)$. Let $\hat{\sigma}^2$ be an estimator of σ^2 satisfying (2). If $\lambda_n = o(n^{c_2-c_1})$ and $\lambda_n - 2M_1 \log p_n/\rho_n r_{inf} \to \infty$, then the GIC$_{\lambda_n}$ with the estimated variance is consistent.*

The corrected RIC, the GIC with $\lambda_n = 2(\log p_n + \log\log p_n)$, does not satisfy the condition in Theorem 7, and hence may not be consistent with an estimated variance. On the other hand, the GIC with $\lambda_n = \alpha_n \log p_n$ is consistent as long as $\alpha_n \to \infty$.

4.3 The Size of s_n

For condition A5, s_n should be large enough so that $q_n \leq s_n$. In many cases, s_n can be sufficiently large for practical purposes. For example, suppose $\{\mathbf{x}_i, i \leq n\}$ are independent and identically distributed p_n dimensional random vectors such that $\mathrm{E}(\mathbf{x}_1) = \mathbf{0}$ and $\mathrm{Var}(\mathbf{x}_1) = \Sigma = [\sigma_{jk}]$. For a given $\rho > 0$, let s^* be the largest integer such that the smallest eigenvalue of $\Sigma_\eta = [\sigma_{jk}, j,k \in \eta]$ is greater than ρ for any $\eta \subset \{1,\ldots,p_n\}$ with $|\eta| \leq s^*$. For example, when Σ is compound symmetry, that is $\sigma_{jj} = 1$ and $\sigma_{jk} = \nu$ for $j \neq k$ and $\nu \in [0,1)$, the smallest eigenvalue of Σ_η is $1-\nu$ for all $\eta \subset \{1,\ldots,p_n\}$ and hence $s^* = p_n$ if $1-\nu > \rho$. Let $\mathbf{A} = \Sigma_\eta - \mathbf{X}_\eta^{'}\mathbf{X}_\eta/n$. By the inequality (2) in Greenshtein and Ritov (2004), we have

$$\sup_{j,k}\left|\sum_{i=1}^{n} x_{ij}x_{ik}/n - \sigma_{jk}\right| = O_p\left(\sqrt{\frac{\log n}{n}}\right),$$

and hence $\sup_{jk}|a_{jk}| = O_p(\sqrt{\log n/n})$, where a_{jk} is the (j,k) entry of $\mathbf{A}$. Since the largest eigenvalue of $\mathbf{A}$ is bounded by $|\eta|O_p(\sqrt{\log n/n})$, the smallest eigenvalue of $\mathbf{X}_\eta{}'\mathbf{X}_\eta/n$ is greater than $\rho - |\eta|O_p(\sqrt{\log n/n})$ if $|\eta| \leq s^*$. So, we can let $s_n = \min\{n^c, s^*\}$ for $c < 1/2$.

5. Numerical Analysis

In this section, we investigate finite sample performance of various GICs by simulation experiments as well as real data analysis. We consider the five GICs whose corresponding λ_ns are given as

- $\mathrm{GIC}_1(=\mathrm{BIC}) : \lambda_n^{(1)} = \log n$,
- $\mathrm{GIC}_2 : \lambda_n^{(2)} = p_n^{1/3}$,
- $\mathrm{GIC}_3 : \lambda_n^{(3)} = 2\log p_n$,
- $\mathrm{GIC}_4 : \lambda_n^{(4)} = 2(\log p_n + \log\log p_n)$,
- $\mathrm{GIC}_5 : \lambda_n^{(5)} = \log\log n \log p_n$,
- $\mathrm{GIC}_6 : \lambda_n^{(6)} = \log n \log p_n$.

The GIC_1 is the BIC. By Theorem 2, the GIC_2 can be consistent when $\mathrm{E}(\varepsilon^8) < \infty$. That is, the GIC_2 can be consistent when the tail of the error distribution is heavier than that of the Gaussian distribution. The GIC_3 and GIC_4 are the RIC of Foster and George (1994) and the corrected RIC of Zhang and Shen (2010). The GIC_5 and GIC_6 are consistent when the error distribution is Gaussian.

5.1 Simulation 1

The first simulation model is

$$y = \mathbf{x}^{'}\beta^* + \varepsilon$$

where $\mathbf{x} = (x_1,\ldots,x_p)^{'}$ is a multivariate Gaussian random vector with mean 0 and covariances of x_k and x_l being $0.5^{|k-l|}$. The ε is a random variable with mean 0 and $\sigma^2 = 4$. For $\beta^* = (3,1.5,0,0,2,0_{p-5}^{'})^{'}$ with 0_k denoting a $k-$dimensional vector of zeros. This simulation setup was considered in Fan and

Li (2001). We consider two distributions for ε : the Gaussian distribution and the t-distribution with 3 degrees of freedom multiplied by a positive constant to make the variance be 4.

First, we compare performances of the GICs applied to all possible submodels with those applied to submodels constructed by the solution path of a sparse penalized approach. For a sparse penalized approach, we use the SCAD penalty with the PLUS algorithm (Zhang, 2010). Table 1 summarizes the results when $p = 10$ and $n = 100$ based on 300 repetitions of the simulation. In the table, 'Signal', 'Noise', 'PTM' and 'Error (s.e.)' represent the average number of variables included in the selected model among the signal variables, the average number of variables included in the selected model among noisy variables, the proportion of the true model being exactly identified, and the average of the squared Euclidean distance of $\hat{\beta}_{\hat{\pi}_{\lambda_n}}$ form β^* with the standard error in the parenthesis, respectively. From Table 1, we can see that the results based on the SCAD solution path are almost identical to those based on the all possible search, which suggests that the model selection with the SCAD solution path is a promising alternative to all possible search.

Submodels	Criterion	Signal	Noise	PTM	Error (s.e.)
All	GIC$_1$	3	0.22	0.80	0.220 (0.013)
	GIC$_2$	3	0.92	0.39	0.371 (0.018)
	GIC$_3$	3	0.22	0.80	0.220 (0.013)
	GIC$_4$	3	0.09	0.91	0.190 (0.016)
	GIC$_5$	3	0.39	0.67	0.267 (0.016)
	GIC$_6$	3	0.02	0.98	0.158 (0.015)
SCAD	GIC$_1$	3	0.21	0.80	0.218 (0.013)
	GIC$_2$	3	0.93	0.40	0.367 (0.018)
	GIC$_3$	3	0.21	0.80	0.218 (0.013)
	GIC$_4$	3	0.10	0.90	0.191 (0.016)
	GIC$_5$	3	0.39	0.67	0.266 (0.016)
	GIC$_6$	3	0.03	0.97	0.163 (0.015)

Table 1: Comparison of the 6 GICs with the all possible search and SCAD solution path when $p = 10$ and $n = 100$.

For simulation with high-dimensional models, we consider $p = 500$ and $p = 3000$. The results of prediction accuracy and variable selectivity for $n = 100$ and $n = 300$ with the error distribution being the Gaussian and t-distributions are presented in Tables 2 and 3, respectively. We use the SCAD solution path to construct a sequence of submodels. The values are the averages based on 300 repetitions of the simulation.

First of all, the GIC$_1$ (the BIC) is the worst in terms of prediction accuracy for $p = 500$ and $p = 3000$. This is mainly because the GIC$_1$ selects too many noisy variables compared to the other selection criteria even though it detects signal variables well. The GIC$_4$ is the best in terms of both the prediction accuracy and variable selectivity for $n = 100$, and the GIC$_6$ is the best for $n = 300$. The GIC$_2$, GIC$_3$ and GIC$_5$ perform reasonably well but tend to select variables more necessary. By comparing the results of the Gaussian and t distributions, we have found that less signal and more noisy variables are selected when the tail of the error distribution is heavier. However, the relative performances of the model selection criteria are similar. That is, the GIC$_1$ is the worst, the GIC$_4$ and GIC$_6$ are the best and so on. Based on these observations, we conclude that (i) model

n	p	Criterion	Signal	Noise	PTM	Error (s.e.)
100	500	GIC$_1$	2.99	4.35	0.00	1.369 (0.039)
		GIC$_2$	2.98	1.25	0.26	0.706 (0.037)
		GIC$_3$	2.96	0.20	0.80	0.351 (0.036)
		GIC$_4$	2.95	0.05	0.90	0.289 (0.036)
		GIC$_5$	2.98	0.67	0.52	0.509 (0.035)
		GIC$_6$	2.81	0.00	0.81	0.620 (0.061)
	3000	GIC$_1$	2.99	5.69	0.00	1.667 (0.036)
		GIC$_2$	2.94	0.26	0.76	0.444 (0.047)
		GIC$_3$	2.92	0.14	0.82	0.431 (0.049)
		GIC$_4$	2.89	0.05	0.87	0.445 (0.053)
		GIC$_5$	2.95	0.58	0.55	0.569 (0.046)
		GIC$_6$	2.63	0.00	0.63	1.092 (0.075)
300	500	GIC$_1$	3	4.89	0.00	0.561 (0.015)
		GIC$_2$	3	1.69	0.15	0.280 (0.010)
		GIC$_3$	3	0.17	0.84	0.083 (0.005)
		GIC$_4$	3	0.03	0.97	0.057 (0.004)
		GIC$_5$	3	0.40	0.66	0.119 (0.007)
		GIC$_6$	3	0.00	1.00	0.049 (0.002)
	3000	GIC$_1$	3	9.80	0.00	1.045 (0.018)
		GIC$_2$	3	0.38	0.67	0.136 (0.008)
		GIC$_3$	3	0.20	0.83	0.099 (0.007)
		GIC$_4$	3	0.02	0.98	0.057 (0.004)
		GIC$_5$	3	0.47	0.60	0.154 (0.009)
		GIC$_6$	3	0.00	1.00	0.050 (0.002)

Table 2: Comparison of the 6 GICs with Simulation 1 when the error follows the Gaussian distribution.

selection criteria specialized for high-dimensional models are necessary for optimal prediction and variable selection, (ii) finite sample performances of consistent GICs are quite different, and (iii) the tail lightness of the error distribution does not affect seriously to relative performances of model selection criteria.

5.2 Simulation 2

We consider a more challenging case by modifying the model for Simulation 1. We divide the p components of β^* into continuous blocks of size 20. We randomly select 5 blocks and assign the value $(3,1.5,0,0,2,0_{15}')/1.5$ to each block. The entries in other blocks are set to be zero.

The results are summarized in Tables 4 and 5. We observe similar phenomena as in Simulation 1: the GIC$_1$ is the worst, the GIC$_4$ and GIC$_6$ are the best and etc. However, when $n = 100$, the GIC$_1$ is better in terms of prediction accuracy than some other GICs which are selection consistent, which is an example of the conflict between selection consistency and prediction optimality.

n	p	Criterion	Signal	Noise	PTM	Error (s.e.)
100	500	GIC$_1$	2.98	4.27	0.09	2.236 (0.702)
		GIC$_2$	2.97	1.24	0.51	1.478 (0.696)
		GIC$_3$	2.96	0.48	0.81	1.224 (0.696)
		GIC$_4$	2.94	0.35	0.86	1.198 (0.695)
		GIC$_5$	2.97	0.82	0.68	1.348 (0.696)
		GIC$_6$	2.84	0.12	0.83	1.271 (0.692)
	3000	GIC$_1$	2.96	5.45	0.01	1.683 (0.106)
		GIC$_2$	2.92	0.51	0.74	0.701 (0.094)
		GIC$_3$	2.91	0.40	0.78	0.673 (0.088)
		GIC$_4$	2.88	0.22	0.82	0.619 (0.086)
		GIC$_5$	2.94	0.69	0.68	0.729 (0.093)
		GIC$_6$	2.59	0.03	0.59	1.273 (0.086)
300	500	GIC$_1$	3	4.26	0.06	0.501 (0.034)
		GIC$_2$	3	1.52	0.38	0.261 (0.022)
		GIC$_3$	3	0.28	0.84	0.100 (0.013)
		GIC$_4$	3	0.08	0.95	0.063 (0.008)
		GIC$_5$	3	0.49	0.75	0.133 (0.016)
		GIC$_6$	3	0.00	1.00	0.044 (0.003)
	3000	GIC$_1$	3	9.58	0.00	1.057 (0.061)
		GIC$_2$	3	0.83	0.71	0.248 (0.043)
		GIC$_3$	3	0.59	0.81	0.205 (0.042)
		GIC$_4$	3	0.24	0.91	0.131 (0.029)
		GIC$_5$	3	0.90	0.68	0.262 (0.044)
		GIC$_6$	3	0.02	0.99	0.062 (0.019)

Table 3: Comparison of the 6 GICs with Simulation 1 when the error follows the t-distribution.

5.3 Real Data Analysis

We analyze the data set used in Scheetz et al. (2006), which consists of gene expression levels of 18,975 genes obtained from 120 rats. The main objective of the analysis is to find genes that are correlated with gene TRIM32 known to cause Bardet-Biedl syndromes. As was done by Huang et al. (2008), we first select 3000 genes with the largest variance in expression level, and then choose the top p genes that have the largest absolute correlation with gene TRIM32 among the selected 3000 genes.

We compare prediction accuracies of the 6 GICs with the submodels obtained from the SCAD solution path. Each data set was divided into two parts, training and test data sets, by randomly selecting 2/3 observations and 1/3 observations, respectively. We use the training data set to select the model and estimate the regression coefficients, and use the test data set to evaluate the prediction performance.

For estimation of the error variance, Zou et al. (2007) used the mean squared error of the full model when $p < n$. This approach, however, is not applicable to our data set since $p > n$. A heuristic method is to set p_{max} first, and to select a model among the SCAD solution path whose number of

n	p	Criterion	Signal	Noise	PTM	Error (s.e.)
100	500	GIC$_1$	14.82	5.11	0.00	3.553 (0.225)
		GIC$_2$	14.67	2.39	0.14	3.211 (0.242)
		GIC$_3$	14.40	1.47	0.24	3.654 (0.285)
		GIC$_4$	14.17	1.16	0.25	4.212 (0.302)
		GIC$_5$	14.57	1.86	0.21	3.242 (0.254)
		GIC$_6$	13.04	0.72	0.16	7.758 (0.398)
	3000	GIC$_1$	12.08	12.19	0.00	20.192 (1.186)
		GIC$_2$	11.51	5.78	0.01	19.783 (1.061)
		GIC$_3$	11.36	5.34	0.01	20.051 (1.055)
		GIC$_4$	11.06	4.37	0.01	20.649 (1.021)
		GIC$_5$	11.68	6.62	0.01	19.616 (1.103)
		GIC$_6$	10.11	2.47	0.01	22.755 (0.894)
300	500	GIC$_1$	15	4.56	0.00	0.795 (0.015)
		GIC$_2$	15	1.63	0.17	0.516 (0.013)
		GIC$_3$	15	0.19	0.82	0.311 (0.009)
		GIC$_4$	15	0.03	0.97	0.278 (0.007)
		GIC$_5$	15	0.39	0.68	0.345 (0.011)
		GIC$_6$	15	0.00	1.00	0.270 (0.006)
	3000	GIC$_1$	15	9.60	0.00	1.322 (0.020)
		GIC$_2$	15	0.32	0.72	0.340 (0.010)
		GIC$_3$	15	0.14	0.88	0.300 (0.008)
		GIC$_4$	15	0.01	0.99	0.267 (0.006)
		GIC$_5$	15	0.40	0.66	0.358 (0.010)
		GIC$_6$	15	0.00	1.00	0.264 (0.006)

Table 4: Comparison of the 6 GICs with Simulation 2 when the error follows the Gaussian distribution.

nonzero coefficients is equal to p_{max}, and to estimate the error variance by the mean squared error of the selected model. Following the results of Scheetz et al. (2006), Chiang et al. (2006), Huang et al. (2008), and Kim et al. (2008), we guess that a reasonable model size would be in between 20 and 40. Table 6 compares the 6 GICs with the number of pre-screened genes being $p = 500$ and $p = 3000$, when the error variance is estimated with p_{max} being 20, 30 and 40, respectively. All values are the arithmetic means of the results from 100 replicated random partitions. In the table, ‘Nonzero’ denotes the number of nonzero coefficients in the selected model and ‘Error (s.e.)’ is the prediction error on the test data set and the standard error in the parenthesis obtained on the test data. For $p = 500$, the lowest prediction error is achieved by the GIC$_2$ and the GIC$_3$, GIC$_4$ and GIC$_5$ perform reasonably well with $p_{max} = 20$. For $p = 3000$, the lowest prediction error is achieved by the GIC$_5$ with $p_{max} = 20$. So, we choose $p_{max} = 20$ for estimation of the error variance.

As argued by Yang (2005), the standard error obtained by random partition could be misleading. As a supplement, we draw the box plots of the 100 prediction errors of the 6 GICs with $p_{max} = 20$ obtained from 100 random partitions in Figure 1. The relative performances of the GICs with the real data are different from those of the simulation studies in the previous subsections. The GIC$_2$,

n	p	Criterion	Signal	Noise	PTM	Error (s.e.)
100	500	GIC$_1$	14.65	3.89	0.07	3.974 (0.401)
		GIC$_2$	14.55	2.07	0.35	3.686 (0.411)
		GIC$_3$	14.40	1.45	0.41	3.870 (0.421)
		GIC$_4$	14.17	1.10	0.41	4.378 (0.424)
		GIC$_5$	14.53	1.85	0.38	3.649 (0.412)
		GIC$_6$	13.00	0.59	0.25	7.848 (0.471)
	3000	GIC$_1$	11.99	9.41	0.02	19.768 (1.154)
		GIC$_2$	11.53	5.23	0.08	19.806 (1.066)
		GIC$_3$	11.47	4.78	0.08	19.641 (1.029)
		GIC$_4$	11.19	3.89	0.08	19.968 (0.959)
		GIC$_5$	11.61	5.92	0.08	19.96 (1.101)
		GIC$_6$	10.35	2.28	0.03	21.96 (0.899)
300	500	GIC$_1$	14.99	4.81	0.05	0.990 (0.098)
		GIC$_2$	14.99	2.33	0.32	0.748 (0.098)
		GIC$_3$	14.99	0.75	0.78	0.519 (0.094)
		GIC$_4$	14.99	0.40	0.89	0.451 (0.090)
		GIC$_5$	14.99	1.00	0.66	0.565 (0.094)
		GIC$_6$	14.99	0.06	0.98	0.339 (0.053)
	3000	GIC$_1$	15	8.18	0.00	1.226 (0.051)
		GIC$_2$	15	0.58	0.73	0.420 (0.040)
		GIC$_3$	15	0.31	0.86	0.358 (0.037)
		GIC$_4$	15	0.12	0.95	0.314 (0.032)
		GIC$_5$	15	0.63	0.70	0.430 (0.041)
		GIC$_6$	15	0.01	0.99	0.272 (0.015)

Table 5: Comparison of the 6 GICs with Simulation 2 when the error follows the t-distribution.

GIC$_3$ and GIC$_5$ have lower prediction errors than the GIC$_4$ and GIC$_6$ while the formers tend to select more variables than necessary in the simulation studies. This observation suggests that there might be many signal genes whose impacts on the response variable are relatively small.

6. Concluding Remarks

The range of consistent model selection criteria is rather large, and it is not clear which one is better with finite samples. It would be interesting to rewrite the class of GICs as $\{\lambda_n = \alpha_n \log p_n : \alpha_n > 0\}$. The GIC$_3$, GIC$_5$ and GIC$_6$ correspond to $\alpha_n = 2$, $\alpha_n = \log\log n$ and $\alpha_n = \log n$, respectively. When the rue model is expected to be very sparse, it would be better to let α_n be rather large (e.g., $\alpha_n = \log n$), while a smaller α_n (e.g., $\alpha_n = 2$ or $\alpha_n = \log\log n$) would be better when many signal covariates with small regression coefficients are expected to exist. The relation of the GICs with larger α_n with those with smaller α_n would be similar to the relation between the AIC and BIC for standard fixed dimensional models.

$p_{max} = 20$

	p			
	500		3000	
	Error (s.e.)	Nonzero	Error (s.e.)	Nonzero
GIC$_1$	0.742 (0.038)	15.91	0.766 (0.036)	18.62
GIC$_2$	**0.649** (0.028)	10.95	0.686 (0.035)	3.91
GIC$_3$	0.656 (0.031)	6.99	0.697 (0.035)	3.69
GIC$_4$	0.677 (0.034)	5.57	0.719 (0.037)	2.78
GIC$_5$	0.664 (0.030)	9.76	**0.667** (0.032)	4.92
GIC$_6$	0.732 (0.038)	3.03	0.792 (0.039)	1.82

$p_{max} = 30$

	p			
	500		3000	
	Error (s.e.)	Nonzero	Error (s.e.)	Nonzero
GIC$_1$	0.890 (0.035)	27.26	0.868 (0.039)	26.07
GIC$_2$	0.825 (0.038)	21.77	0.698 (0.031)	14.04
GIC$_3$	0.752 (0.029)	17.53	0.696 (0.031)	13.25
GIC$_4$	**0.722** (0.029)	15.19	0.691 (0.034)	10.76
GIC$_5$	0.800 (0.030)	20.29	0.729 (0.032)	15.99
GIC$_6$	0.688 (0.030)	11.31	**0.683** (0.034)	5.53

$p_{max} = 40$

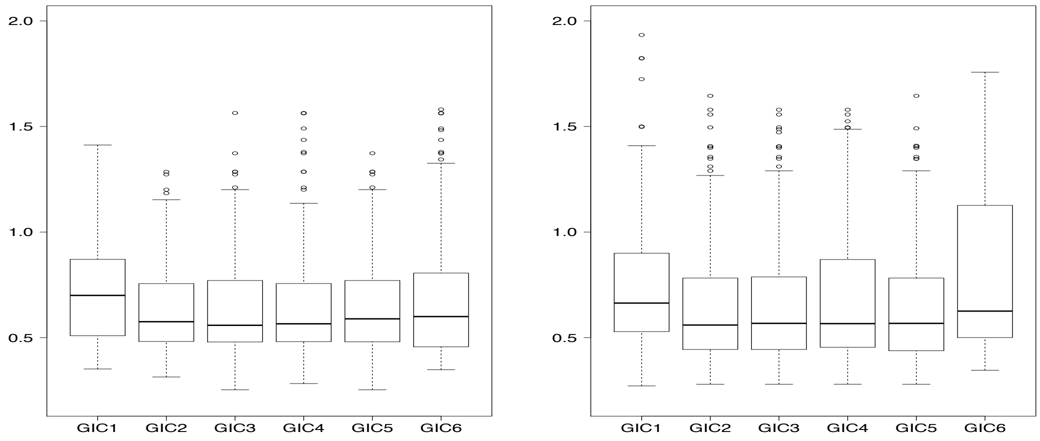
	p			
	500		3000	
	Error (s.e.)	Nonzero	Error (s.e.)	Nonzero
GIC$_1$	1.040 (0.077)	34.54	0.936 (0.041)	33.80
GIC$_2$	0.916 (0.036)	29.59	0.892 (0.041)	27.27
GIC$_3$	0.859 (0.035)	25.10	0.878 (0.040)	26.37
GIC$_4$	**0.846** (0.039)	23.02	0.846 (0.038)	25.00
GIC$_5$	0.890 (0.035)	28.20	0.910 (0.040)	28.60
GIC$_6$	0.763 (0.029)	18.69	**0.800** (0.037)	21.02

Table 6: Comparison of the 6 GICs with the gene expression data. The bold face numbers represent the lowest prediction errors among the 6GICs.

Estimation of σ^2 is an open question. We may use the BIC-like criterion by assuming the Gaussian distribution:

$$\hat{\pi}_{\lambda_n} = \text{argmin}_{\pi \subset \{1,\ldots,p_n\}} \log(R_n(\hat{\beta}_\pi)/n) + \lambda_n|\pi|.$$

If $R_n(\hat{\beta}_\pi)/n$ is bounded above from ∞ and below from 0 in probability (uniformly in π and n), we could derive similar asymptotic properties for the BIC-like criteria as the GICs. We leave this problem as future work.



(a) $p = 500$

(b) $p = 3000$

Figure 1: The boxplot of the prediction errors when (a) $p = 500$ and (b) $p = 3000$ with $p_{max} = 20$.

For consistency, the smallest eigenvalue of the design matrix of the true model is assumed to be sufficiently large (i.e., condition A2). However, it is frequently observed for large dimensional data that some covariates are highly correlated and they affect the output similarly. In this case, selecting some covariates and ignoring the others, which is done by a standard model selection method, is not optimal. See Zou and Hastie (2005) for an example. It would be interesting to develop consistent model selection methods for such cases.

Acknowledgments

This research was supported by the National Research Foundation of Korea grant number 20100012671 funded by the Korea government.

Appendix A. Proof of Theorem 2

Without loss of generality, we let $\pi_n^* = \{1,\ldots,q_n\}$. Let $\hat{\beta}^* = \hat{\beta}_{\pi_n^*}$. Let $\hat{Y}_\pi = \mathbf{X}_n\hat{\beta}_\pi$ and $\hat{Y}_n^* = \mathbf{X}_n\hat{\beta}_{\pi_n^*}$. We let $\beta^* = (\beta^{(1)*},\beta^{(2)*})$, where $\beta^{(1)*} \in R^{q_n}$ and $\beta^{(2)*} \in R^{p_n-q_n}$. Let $\mathbf{C}_n = \mathbf{X}_n'\mathbf{X}_n/n$ and $\mathbf{C}_n^{(i,j)} = \mathbf{X}_n^{(i)'}\mathbf{X}_n^{(j)}/n$ for $i, j = 1,2$. We need the following two lemmas.

Lemma 8

$$\max_{j\leq q_n} |\hat{\beta}_j^* - \beta_j^*| = o_p(n^{-(1-c_2)/2}).$$

Proof. Let $z_j = \sqrt{n}(\hat{\beta}_j^* - \beta_j^*)$. For proving Lemma 8, we will show

$$\max_{j\leq q_n} |z_j| = o_p(n^{c_2/2}).$$

Write

$$\mathbf{z} = (\mathbf{C}_n^{(1,1)})^{-1} \frac{\mathbf{X}_n^{(1)'} \varepsilon_n}{\sqrt{n}} = \mathbf{H}^{(1)'} \varepsilon_n,$$

where $\mathbf{z} = (z_1, \dots, z_{q_n})'$, $\varepsilon_n = (\varepsilon_1, \dots, \varepsilon_n)'$ and $\mathbf{H}^{(1)'} = (\mathbf{h}_1^{(1)}, \dots, \mathbf{h}_{q_n}^{(1)})' = (\mathbf{C}_n^{(1,1)})^{-1}\mathbf{X}_n^{(1)'}/\sqrt{n}$. Since $\mathbf{H}^{(1)'}\mathbf{H}^{(1)} = (\mathbf{C}_n^{(1,1)})^{-1}$, A2 of the regularity conditions implies $\|\mathbf{h}_j^{(1)}\|_2^2 \leq 1/M_2$ for all $j \leq q_n$. Hence, $\mathrm{E}(z_j)^{2k} < \infty$ for all $j \leq q_n$ since $\mathrm{E}(\varepsilon_i)^{2k} < \infty$. Thus

$$\Pr(|z_j| > t) = O(t^{-2k}).$$

For any $\eta > 0$, we can write

$$\begin{aligned}
\Pr(|z_j| > \eta n^{c_2/2} \text{ for some } j = 1, \dots, q_n) &\leq \sum_{j=1}^{q_n} \Pr(|z_j| > \eta n^{c_2/2}) \\
&\leq \sum_{j=1}^{q_n} \frac{1}{\eta} n^{-c_2 k} \\
&= \frac{1}{\eta} q_n n^{-c_2 k} \leq \frac{1}{\eta} n^{-(c_2 - c_3)k} \to 0,
\end{aligned}$$

which completes the proof. ■

Lemma 9

$$\max_{q_n < j \leq p_n} | < Y_n - \hat{Y}_n^*, X_n^j > | = o_p(\sqrt{n\lambda_n \rho_n}).$$

Proof. Note that

$$\begin{aligned}
& (< Y_n - \hat{Y}_n^*, X_n^j >, j = q_n + 1, \dots, p_n) \\
= \; & \mathbf{X}_n^{(2)'} \left(Y_n - \mathbf{X}_n^{(1)} \hat{\beta}^{*(1)} \right) \\
= \; & \mathbf{X}_n^{(2)'} \left(Y_n - \mathbf{X}_n^{(1)} \frac{1}{n} (\mathbf{C}_n^{(1,1)})^{-1} \mathbf{X}_n^{(1)'} Y_n \right) \\
= \; & \mathbf{X}_n^{(2)'} \left(\mathbf{X}_n^{(1)} \beta^{*(1)} + \varepsilon_n - \mathbf{X}_n^{(1)} \frac{1}{n} (\mathbf{C}_n^{(1,1)})^{-1} \mathbf{X}_n^{(1)'} (\mathbf{X}_n^{(1)} \beta^{*(1)} + \varepsilon_n) \right) \\
= \; & \mathbf{X}_n^{(2)'} \left(\mathbf{I} - \frac{1}{n} \mathbf{X}_n^{(1)} (\mathbf{C}_n^{(1,1)})^{-1} \mathbf{X}_n^{(1)'} \right) \varepsilon_n.
\end{aligned}$$

Hence, we have

$$< Y_n - \hat{Y}_n^*, X_n^j > /\sqrt{n} = \mathbf{h}_j^{(2)'} \varepsilon_n \quad \text{for } j = q_n + 1, \dots, p_n, \tag{3}$$

where $\mathbf{h}_j^{(2)}$ is the $j - q_n$ column vector of $\mathbf{H}^{(2)}$ and

$$\mathbf{H}^{(2)'} = \mathbf{C}_n^{(2,1)} (\mathbf{C}_n^{(1,1)})^{-1} \frac{1}{\sqrt{n}} \mathbf{X}_n^{(1)'} - \frac{1}{\sqrt{n}} \mathbf{X}_n^{(2)'}.$$

Note that

$$\mathbf{H}^{(2)'}\mathbf{H}^{(2)} = \frac{1}{n} \mathbf{X}_n^{(2)'} \left(\mathbf{I} - \mathbf{X}_n^{(1)} (\mathbf{X}_n^{(1)'} \mathbf{X}_n^{(1)})^{-1} \mathbf{X}_n^{(1)'} \right) \mathbf{X}_n^{(2)}.$$

Since the all eigenvalues of $\mathbf{I}-\mathbf{X}_n^{(1)}(\mathbf{X}_n^{(1)'}\mathbf{X}_n^{(1)})^{-1}\mathbf{X}_n^{(1)'}$ are between 0 and 1, we have $\|\mathbf{h}_j^{(2)}\|_2^2 \leq M_1$ for all $j = q_n+1,\ldots,p_n$. Hence, $\mathrm{E}(\xi_j)^{2k} < \infty$, where $\xi_j = < Y_n - \hat{Y}_n^*, X_n^j > /\sqrt{n}$, and so

$$Pr\,(|\xi_j| > t) = O(t^{-2k}).$$

Finally, for any $\eta > 0$,

$$\begin{aligned}
& \Pr\Big(| < Y_n - \hat{Y}_n^*, X_n^j > | > \eta\sqrt{n\lambda_n\rho_n} \text{ for some } j = q_n+1,\ldots,p_n\Big) \\
= \;& \Pr\Big(|\xi_j| > \eta\sqrt{\lambda_n\rho_n} \text{ for some } j = q_n+1,\ldots,p_n\Big) \\
\leq \;& \sum_{j=q_n+1}^{p_n} \Pr\Big(|\xi_j| > \eta\sqrt{\lambda_n\rho_n}\Big) \\
= \;& (p_n - q_n)O\left(\frac{1}{(\lambda_n\rho_n)^k}\right) = O\left(\frac{p_n}{(\lambda_n\rho_n)^k}\right) \to 0,
\end{aligned}$$

which completes the proof. ■

Proof of Theorem 2. For any π, we can write

$$\begin{gathered}
R_n(\hat{\beta}_\pi) + \lambda_n|\pi|\sigma^2 - R_n(\hat{\beta}^*) - \lambda_n|\pi_n^*|\sigma^2 \\
= -2\textstyle\sum_{j=q_n+1}^{p_n}\hat{\beta}_{\pi,j} < Y_n - \hat{Y}_n^*, X^j > + (\hat{\beta}_\pi - \hat{\beta}^*)'(\mathbf{X}_n'\mathbf{X}_n)(\hat{\beta}_\pi - \hat{\beta}^*) + \lambda_n(|\pi| - |\pi_n^*|)\sigma^2.
\end{gathered}$$

By Condition A3,

$$(\hat{\beta}_\pi - \hat{\beta}^*)'(\mathbf{X}_n'\mathbf{X}_n)(\hat{\beta}_\pi - \hat{\beta}^*) \geq \sum_{j\in\pi\cup\pi^*} n\rho_n(\hat{\beta}_{\pi,j} - \hat{\beta}_j^*)^2.$$

Hence, we have for any $\pi \in \mathcal{M}^{s_n}$,

$$R_n(\hat{\beta}_\pi) + \lambda_n|\pi|\sigma^2 - R_n(\hat{\beta}^*) - \lambda_n|\pi_n^*|\sigma^2 \geq \sum_{j\in\pi\cup\pi_n^*} w_j,$$

where

$$w_j = -2\hat{\beta}_{\pi,j} < Y_n - \hat{Y}_n^*, X_n^j > I(j \notin \pi_n^*) + n\rho_n(\hat{\beta}_{\pi,j} - \hat{\beta}_j^*)^2 + \lambda_n(I(j \in \pi - \pi_n^*) - I(j \in \pi_n^* - \pi))\sigma^2.$$

For $j \in \pi_n^* - \pi$, we have $w_j = n\rho_n\hat{\beta}_j^{*2} - \lambda_n\sigma^2$. Let

$$A_n = \{n\rho_n\hat{\beta}_j^{*2} - \lambda_n\sigma^2 > 0, j = 1,\ldots,q_n\}. \tag{4}$$

Then, $\Pr(A_n) \to 1$ by Lemma 8 and Conditions A3 and A4.

For $j \in \pi - \pi_n^*$

$$\begin{aligned}
w_j &= -2\hat{\beta}_{\pi,j} < Y_n - \hat{Y}_n^*, X_n^j > + n\rho_n\hat{\beta}_{\pi,j}^2 + \lambda_n\sigma^2 \\
&\geq - < Y_n - \hat{Y}_n^*, X_n^j >^2 /(n\rho_n) + \lambda_n\sigma^2.
\end{aligned}$$

Let

$$B_n = \{- < Y_n - \hat{Y}_n^*, X_n^j >^2 /(n\rho_n) + \lambda_n\sigma^2 > 0, j = q_n+1,\ldots,p_n\}. \tag{5}$$

Then, $\Pr(B_n) \to 1$ by Lemma 9.

For $j \in \pi \cap \pi_n^*$,

$$w_j = n\rho_n(\hat{\beta}_{\pi,j} - \hat{\beta}_j^*)^2 \geq 0.$$

To sum up, on $A_n \cap B_n$,

$$R_n(\hat{\beta}_\pi) + \lambda_n|\pi|\sigma^2 - R_n(\hat{\beta}^*) - \lambda_n|\pi_n^*|\sigma^2 > 0$$

for all $\pi \neq \pi_n^*$. Since $\Pr(A_n \cap B_n) \to 1$, the proof is done. ■

Appendix B. Proof of Theorem 3

For given $\pi \subset \{1,\ldots,p_n\}$, let $\mathbf{M}_\pi$ be the projection operator onto the space spanned by $(X^{(j)}, j \in \pi)$. That is, $\mathbf{M}_\pi = \mathbf{X}_\pi(\mathbf{X}_\pi^{'}\mathbf{X}_\pi)^{-1}\mathbf{X}_\pi^{'}$ provided $\mathbf{X}_\pi$ is of full rank. Let $\mathbf{X}_n\beta_n^* = \mu_n$ and $\mathbf{I}$ be the $n \times n$ identity matrix. Without loss of generality, we assume $\sigma^2 = 1$.

Lemma 10 *There exists $\eta > 0$ such that for any $\pi \in \mathcal{M}^{s_n}$ with $\pi_n^* \nsubseteq \pi$,*

$$\mu_n^{'}(\mathbf{I} - \mathbf{M}_\pi)\mu_n \geq \eta|\pi^-|n^{c_2 - c_1},$$

where $\pi^- = \pi_n^* - \pi$.

Proof. For given $\pi \in \mathcal{M}^{s_n}$ with $\pi_n^* \nsubseteq \pi$, we have

$$\begin{aligned} & \mu_n^{'}(\mathbf{I} - \mathbf{M}_\pi)\mu_n \\ = & \inf_{\alpha \in R^{|\pi|}} \|\mathbf{X}_{\pi^-}\beta_{\pi^-}^* - \mathbf{X}_\pi\alpha\|^2 \\ = & \inf_{\alpha \in R^{|\pi|}} (\beta_{\pi^-}^{*'}, \alpha^{'})(\mathbf{X}_{\pi^-}, \mathbf{X}_\pi)^{'}(\mathbf{X}_{\pi^-}, \mathbf{X}_\pi)(\beta_{\pi^-}^{*'}, \alpha^{'})^{'} \\ \geq & \; n\|\beta_{\pi^-}^*\|^2\rho_n \\ \geq & \; M_3M_4|\pi^-|n^{c_2 - c_1}, \end{aligned}$$

where $\beta_{\pi^-}^* = (\beta_j^*, j \in \pi^-)$ and the last inequality is due to Condition A4. ■

Lemma 11 *For given $\pi \subset \{1,\ldots,p_n\}$, let*

$$Z_\pi = \frac{\mu_n^{'}(\mathbf{I} - \mathbf{M}_\pi)\varepsilon_n}{\sqrt{\mu_n^{'}(\mathbf{I} - \mathbf{M}_\pi)\mu_n}}.$$

Then

$$\max_{\pi \in \mathcal{M}^{s_n}} |Z_\pi| = O_p(\sqrt{s_n \log p_n}).$$

Proof. Note that $Z_\pi \sim N(0,1)$ for all $\pi \in \mathcal{M}^{s_n}$. Since

$$\Pr(|Z_\pi| > t) \leq C\exp(-t^2/2) \tag{6}$$

for some $C > 0$, we have

$$\begin{aligned}\Pr\left(\max_{\pi\in\mathcal{M}^{s_n}} |Z_\pi| > t\right) &\leq \sum_{\pi\in\mathcal{M}^{s_n}} C\exp(-t^2/2)\\ &\leq Cp_n^{s_n}\exp(-t^2/2).\end{aligned}$$

Hence, if we let $t = \sqrt{ws_n \log p_n}$,

$$\Pr\left(\max_{\pi\in\mathcal{M}^{s_n}} |Z_\pi| > t\right) \leq C\exp((-w/2+1)s_n\log p_n) \to 0$$

as $w \to \infty$. ■

Lemma 12

$$\max_{\pi\in\mathcal{M}^{s_n}} \varepsilon_n^{'}\mathbf{M}_\pi\varepsilon_n = O_p(s_n\log p_n).$$

Proof. For given $\pi \subset \{1,\ldots,p_n\}$, let $r(p)$ be the rank of $\mathbf{X}_\pi$. Note that $\varepsilon_n^{'}\mathbf{M}_\pi\varepsilon_n \sim \chi^2(r(\pi))$ where $\chi^2(k)$ is the chi-square distribution with degree of freedom k. It is easy to see that (see, for example, Yang 1999)

$$\Pr(\varepsilon_n^{'}\mathbf{M}_\pi\varepsilon_n \geq t) \leq \exp\left(-\frac{t-r(\pi)}{2}\right)\left(\frac{t}{r(\pi)}\right)^{r(\pi)/2}. \tag{7}$$

Hence

$$\Pr\left(\max_{\pi\in\mathcal{M}^{s_n}} \varepsilon_n^{'}\mathbf{M}_\pi\varepsilon_n \geq t\right) \leq \sum_{k=1}^{s_n}\binom{p_n}{k}\Pr(W_k \leq t),$$

where $W_k \sim \chi^2(k)$. Since $\Pr(W_k \geq t) \leq \Pr(W_{s_n} \geq t)$, we have

$$\begin{aligned}\Pr\left(\max_{\pi\in\mathcal{M}^{s_n}} \mathbf{e}_n^{'}\mathbf{M}_\pi\mathbf{e}_n \geq t\right) &\leq \Pr(W_{s_n} \geq t)\sum_{k=1}^{s_n}\binom{p_n}{k}\\ &\leq \Pr(W_{s_n} \geq t)p_n^{s_n}. \end{aligned}\tag{8}$$

The proof is done by applying (7) to (8). ■

Proof of Theorem 3. First, we will show that $\Pr(\pi_n^* \not\subseteq \hat{\pi}_{\lambda_n}) \to 0$. For given $\pi \subset \{1,\ldots,p_n\}$, let $R_n(\pi) = R_n(\hat{\beta}_\pi)$. Note that $R_n(\pi) = Y_n^{'}(\mathbf{I}-\mathbf{M}_\pi)Y_n$. For $\pi \not\supseteq \pi_n^*$, Lemmas 10, 11 and 12 imply

$$\begin{aligned}&R_n(\pi) - R_n(\pi_n^*) + \lambda_n(|\pi| - |\pi_n^*|)\sigma^2\\ =\ &\mu_n^{'}(\mathbf{I}-\mathbf{M}_\pi)\mu_n + 2\mu_n^{'}(\mathbf{I}-\mathbf{M}_\pi)\varepsilon_n + \varepsilon_n^{'}(\mathbf{M}_{\pi^*}-\mathbf{M}_\pi)\varepsilon_n + \lambda_n(|\pi| - |\pi_n^*|)\sigma^2\\ \geq\ &\eta|\pi^-|n^{c_2-c_1} - 2\sqrt{\eta|\pi^-|n^{c_2-c_1}}O_p(\sqrt{s_n\log p_n}) - O_p(s_n\log p_n) - |\pi^-|\lambda_n,\end{aligned}$$

where $\pi^- = \pi_n^* - \pi$. Since $s_n\log p_n \leq o(n^{c_2-c_1})$ and $\lambda_n = o(n^{c_2-c_1})$, the proof is done.

It remains to show that the probability of

$$\inf_{\pi\in\mathcal{M}^{s_n},\pi\supsetneq\pi_n^*} R_n(\pi) - R_n(\pi_n^*) + \lambda_n(|\pi| - |\pi_n^*|)\sigma^2 > 0 \tag{9}$$

converges to 1. By Theorem 1 of Zhang and Shen (2010), the probability of (9) is larger than

$$2-\left(1+e^{1/2}\exp\left(-\frac{\lambda_n-\log\lambda_n}{2}\right)\right)^{p_n-q_n},$$

which converges to 1 when $2\log p_n-\lambda_n+\log\lambda_n\to-\infty$. The equivalent condition with $2\log p_n-\lambda_n+\log\lambda_n\to-\infty$ is $\lambda_n-2\log p_n-\log\log p_n\to\infty$. ■

Appendix C. Proof of Theorem 4

By Theorem 4 of Kim and Kwon (2012), the solution path of the SCAD or minimax concave penalty include the true model with probability converging to 1. Since condition A3' is stronger than condition A3, the GIC_{λ_n} with $\lambda_n=o(n^{c_2-c_1})$ is consistent, and so is with the solution path of the SCAD or minimax concave penalty.

Appendix D. Proof of Theorem 7

Let $\tilde{A}_n$ and $\tilde{B}_n$ be the sets defined in (4) and (5) except that σ^2 is replaced by $\hat{\sigma}^2$. It suffices to show that $\Pr(\tilde{A}_n\cap\tilde{B}_n)\to 1$. It is not difficult to prove $\Pr(\tilde{A}_n)\to 1$ by Lemma 8 and (2).

For $\tilde{B}_n$, since $\varepsilon_i\sim N(0,\sigma^2)$, (3) implies

$$<Y_n-\hat{Y}_n^*,X_n^j>/\sqrt{n}\sim N(0,\sigma_j^2)$$

where $\sigma_j^2\le\sigma^2M_1$. By (6), we have

$$\begin{aligned}\Pr(\tilde{B}_n^c) &\le \Pr(<Y_n-\hat{Y}_n^*,X_n^j>^2>n\rho_n\lambda_n\hat{\sigma}^2 \text{ for some } j=q_n+1,\ldots,p_n)\\ &\le Cp_n\exp(-\rho_n r_{inf}\lambda_n/2M_1).\end{aligned}$$

Hence, as long as $2M_1\log p_n/(\rho_n r_{inf})-\lambda_n\to-\infty$, $\Pr(\tilde{B}_n^c)\to 0$ and the proof is done. ■

References

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrox and F. Caski, editors, *Second International Symposium on Information Theory*, volume 1, pages 267–281. Budapest: Akademiai Kiado, 1973.

K. W. Broman and T. P. Speed. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society, Ser. B*, 64:641–656, 2002.

G. Casella, F. J. Giron, M. L. Martinez, and E. Moreno. Consistency of bayesian procedure for variable selection. *The Annals of Statistics*, 37:1207–1228, 2009.

J. Chen and Z. Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 24:759–771, 2008.

A. P. Chiang, J. S. Beck, H.-J. Yen, M. K. Tayeh, T. E. Scheetz, R. Swiderski, D. Nishimura, T. A. Braun, K.-Y. Kim, J. Huang, K. Elbedour, R. Carmi, D. C. Slusarski, T. L. Casavant, E. M. Stone, and V. C. Sheffield. Homozygosity mapping with snp arrays identifies a novel gene for bardet-biedl syndrome (bbs10). *Proc. Nat. Acad. Sci.*, 103:6287–6292, 2006.

P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross validation. *Numer. Math.*, 31:377–403, 1979.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The 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.

D. P. Foster and E. I. George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, 22:1947–1975, 1994.

E. Greenshtein and Y. Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10:971–988, 2004.

J. Huang, S. Ma, and C-H. Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18:1603–1618, 2008.

Y. Kim and S. Kwon. The global optimality of the smoothly clipped absolute deviation penalized estimator. *Biometrika*, forthcoming, 2012.

Y. Kim, H. Choi, and H. Oh. Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, 103:1665–1673, 2008.

T. E. Scheetz, K.-Y. A. Kim, R. E. Swiderski, A. R. Philp1, T. A. Braun, K. L. Knudtson, A. M. Dorrance, G. F. DiBona, J. Huang, T. L. Casavant, V. C. Sheffield, and E. M. Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103:14429–14434, 2006.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.

M. Stone. Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society, Ser. B*, 39:111–147, 1974.

R. J. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Ser. B*, 58:267–288, 1996.

H. Wang. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104:1512–1524, 2009.

H. Wang, B. Li, and C. Leng. Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society, Ser. B*, 71:671–683, 2009.

Y. Yang. Model selection for nonparametric regression. *Statistica Sinica*, 9:475–499, 1999.

Y. Yang. Can the strengths of aic and bic be shared? a conflict between model identification and regression estimation. *Biometrika*, 92:937–950, 2005.

Y. Yang and A. R. Barron. An asymptotic property of model selection criteria. *IEEE Transanctions in Information Theory*, 44:95–116, 1998.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38:894–942, 2010.

Y. Zhang and X. Shen. Model selection procedure for high-dimensional data. *Statistical Analysis and Data Mining*, 3:350–358, 2010.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Reserach*, 7:2541–2563, 2006.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Ser. B*, 67:301–320, 2005.

H. Zou, T. Hastie, and R. Tibshirani. On the "degree of freedom" of lasso. *The Annals of Statsitics*, 35:2173–2192, 2007.

Submitted 8/11; Revised 2/12; Published 4/12

The huge Package for High-dimensional Undirected Graph Estimation in R

Tuo Zhao TOURZHAO@JHU.EDU
Han Liu[*] HANLIU@CS.JHU.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA

Kathryn Roeder ROEDER@STAT.CMU.EDU
John Lafferty[†] LAFFERTY@CS.CMU.EDU
Larry Wasserman[†] LARRY@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA, 15213

Editor: Mikio Braun

Abstract

We describe an R package named huge which provides easy-to-use functions for estimating high dimensional undirected graphs from data. This package implements recent results in the literature, including Friedman et al. (2007), Liu et al. (2009, 2012) and Liu et al. (2010). Compared with the existing graph estimation package glasso, the huge package provides extra features: (1) instead of using Fortan, it is written in C, which makes the code more portable and easier to modify; (2) besides fitting Gaussian graphical models, it also provides functions for fitting high dimensional semiparametric Gaussian copula models; (3) more functions like data-dependent model selection, data generation and graph visualization; (4) a minor convergence problem of the graphical lasso algorithm is corrected; (5) the package allows the user to apply both lossless and lossy screening rules to scale up large-scale problems, making a tradeoff between computational and statistical efficiency.

Keywords: high-dimensional undirected graph estimation, glasso, huge, semiparametric graph estimation, data-dependent model selection, lossless screening, lossy screening

1. Overview

Undirected graphs is a natural approach to describe the conditional independence among many variables. Each node of the graph represents a single variable and no edge between two variables implies that they are conditional independent given all other variables. In the past decade, significant progress has been made on designing efficient algorithms to learn undirected graphs from high-dimensional observational data sets. Most of these methods are based on either the penalized maximum-likelihood estimation (Friedman et al., 2007) or penalized regression methods (Meinshausen and Bühlmann, 2006). Existing packages include glasso, Covpath and CLIME. In particu-

[*] Also in the Department of Biostatistics.
[†] Also in the Department of Machine Learning.

lar, the glasso package has been widely adopted by statisticians and computer scientists due to its friendly user-inference and efficiency.

In this paper[1] we describe a newly developed R package named huge (High-dimensional Undirected Graph Estimation) coded in C. The package includes a wide range of functional modules and addresses some drawbacks of the graphical lasso algorithm. To gain more scalability, the package supports two modes of screening, lossless (Witten et al., 2011) and lossy screening. When using lossy screening, the user can select the desired screening level to scale up for high-dimensional problems, but this introduces some estimation bias.

2. Software Design and Implementation

The package huge aims to provide a general framework for high-dimensional undirected graph estimation. The package includes Six functional modules (M1-M6) facilitate a flexible pipeline for analysis (Figure 1).

M1. Data Generator: The function huge.generator() can simulate multivariate Gaussian data with different undirected graphs, including hub, cluster, band, scale-free, and Erdös-Rényi random graphs. The sparsity level of the obtained graph and signal-to-noise ratio can also be set up by users.

M2. Semiparametric Transformation: The function huge.npn() implements the nonparanormal method (Liu et al., 2009, 2012) for estimating a semiparametric Gaussian copula model.The nonparanormal family extends the Gaussian distribution by marginally transforming the variables. Computationally, the nonparanormal transformation only requires one pass through the data matrix.

M3. Graph Screening: The scr argument in the main function huge() controls the use of large-scale correlation screening before graph estimation. The function supports the lossless screening (Witten et al., 2011) and the lossy screening. Such screening procedures can greatly reduce the computational cost and achieve equal or even better estimation by reducing the variance at the expense of increased bias.

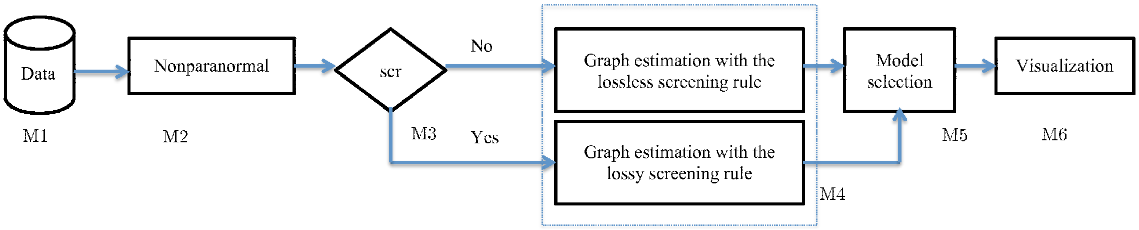

Data
Nonparanormal
scr
No
Yes
Graph estimation with the lossless screening rule
Graph estimation with the lossy screening rule
Model selection
Visualization
M1
M2
M3
M4
M5
M6

Figure 1: The graph estimation pipeline.

M4. Graph Estimation: Similar to the glasso package, the method argument in the huge() function supports two estimation methods: (i) the neighborhood pursuit algorithm (Meinshausen and Bühlmann, 2006) and (ii) the graphical lasso algorithm (Friedman et al., 2007). We apply the coordinate descent with active set and covariance update, as well as other tricks suggested in Friedman et al. (2010). We modified the warm start trick to address the potential divergence problem of the graphical lasso algorithm (Mazumder and Hastie, 2011). The code is also memory-optimized using the sparse matrix data structure when estimating and storing full regularization paths for large

[1] This paper is only a summary of the package huge. For more details please refer to the online vignette.

data sets. we also provide a complementary graph estimation method based on thresholding the sample correlation matrix, which is computationally efficient and widely applied in biomedical research.

M5. Model Selection: The function huge.select() provides two regularization parameter selection methods: the stability approach for regularization selection (StARS) (Liu et al., 2010); and rotation information criterion (RIC). We also provide a likelihood-based extended Bayesian information criterion.

M6. Graph Visualization: The plotting functions huge.plot() and plot() provide visualizations of the simulated data sets, estimated graphs and paths. The implementation is based on the igraph package.

3. User Interface by Example

We illustrate the user interface by analyzing a stock market data which we contribute to the huge package. We acquired closing prices from all stocks in the S&P 500 for all the days that the market was open between Jan 1, 2003 and Jan 1, 2008. This gave us 1258 samples for the 452 stocks that remained in the S&P 500 during the entire time period.

```
> library(huge)
> data(stockdata)                                          # Load the data
> x = log(stockdata$data[2:1258,]/stockdata$data[1:1257,])  # Preprocessing
> x.npn = huge.npn(x, npn.func="truncation")               # Nonparanormal
> out.npn = huge(x.npn,method = "glasso", nlambda=40,lambda.min.ratio = 0.4)
```

Here the data have been transformed by calculating the log-ratio of the price at time t to price at time $t-1$. The nonparanormal transformation is applied to the data, and a graph is estimated using the graphical lasso (the default is the Meinshausen-Bühlmann estimator). The program automatically sets up a sequence of 40 regularization parameters and estimates the graph path. The lossless screening method is applied by default.

4. Performance Benchmark

To compare huge with glasso (ver 1.4), we consider four scenarios with varying sample sizes n and dimensionality d, as shown in Table 1. We simulate the data from a normal distribution with the Erdös-Rényi random graph structure (sparsity 1%). Timings (in seconds) are computed over 10 values of the corresponding regularization parameter, and the range of regularization parameters is chosen so that each method produced approximately the same number of non-zero estimates. The convergence threshold of both glasso and huge is chosen to be 10^{-4}. For these simulations, CLIME (ver 1.0) and Covpath (ver 0.2) were unable to obtain timing results due to their numerical instability.

For the neighborhood pursuit, we can see that huge achieves the best performance. In particular, when the lossy screening rule is applied, huge automatically reduces each individual lasso problem from the original dimension d to the sample size n, therefore a better efficiency can be achieved when d is much larger than n. Based on our experiments, the speed up due to the lossy screening rule can be up to more than 500%.

Method	$d = 1000$ $n = 100$	$d = 2000$ $n = 150$	$d = 3000$ $n = 200$	$d = 4000$ $n = 300$
huge-neighborhood pursuit (lossy)	3.246 (0.147)	13.47 (0.665)	35.87 (0.97)	247.2 (14.26)
huge-neighborhood pursuit	4.240 (0.288)	42.41 (2.338)	147.9 (4.102)	357.8 (28.00)
glasso-neighborhood pursuit	37.23 (0.516)	296.9 (4.533)	850.7 (8.180)	3095 (150.5)
huge-graphical lasso (lossy)	39.61 (2.391)	289.9 (17.54)	905.6 (25.84)	2370 (168.9)
huge-graphical lasso (lossless)	47.86 (3.583)	328.2 (30.09)	1276 (43.61)	2758 (326.2)
glasso-graphical lasso	131.9 (5.816)	1054 (47.52)	3463 (107.6)	8041 (316.9)

Table 1: Experimental Results

Unlike the neighborhood pursuit, the graphical lasso estimates the inverse covariance matrix. The screening rule (Witten et al., 2011) greatly reduces the computation required by the graphical lasso algorithm and gains an extra performance boost.

5. Summary and Acknowledgement

We developed a new package named huge for high dimensional undirected graph estimation. The package is complementary to the existing glasso package by providing extra features and functional modules. We plan to maintain and support this package in the future. Tuo Zhao is partially supported by the Google Summer of Code program 2011. Han Liu, John Lafferty, and Larry Wasserman are supported by NSF grant IIS-1116730 and AFOSR contract FA9550-09-1-0373. Kathryn Roeder is supported by National Institute of Mental Health grant MH057881.

References

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009.

H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection for high dimensional graphical models. *Advances in Neural Information Processing Systems*, 2010.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High dimensional semiparametric gaussian copula graphical models. Technical report, Johns Hopkins University, 2012.

R. Mazumder and T. Hastie. The graphical lasso: New insights and alternatives. Technical report, Stanford University, 2011.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

D. Witten, J. Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, to appear, 2011.

Journal of Machine Learning Research 13 (2012) 1063-1095 Submitted 10/10; Revised 10/11; Published 4/12

Analysis of a Random Forests Model

Gérard Biau[*] GERARD.BIAU@UPMC.FR
LSTA & LPMA
Université Pierre et Marie Curie – Paris VI
Boîte 158, Tour 15-25, 2ème étage
4 place Jussieu, 75252 Paris Cedex 05, France

Editor: Bin Yu

Abstract

Random forests are a scheme proposed by Leo Breiman in the 2000's for building a predictor ensemble with a set of decision trees that grow in randomly selected subspaces of data. Despite growing interest and practical use, there has been little exploration of the statistical properties of random forests, and little is known about the mathematical forces driving the algorithm. In this paper, we offer an in-depth analysis of a random forests model suggested by Breiman (2004), which is very close to the original algorithm. We show in particular that the procedure is consistent and adapts to sparsity, in the sense that its rate of convergence depends only on the number of strong features and not on how many noise variables are present.

Keywords: random forests, randomization, sparsity, dimension reduction, consistency, rate of convergence

1. Introduction

In a series of papers and technical reports, Breiman (1996, 2000, 2001, 2004) demonstrated that substantial gains in classification and regression accuracy can be achieved by using ensembles of trees, where each tree in the ensemble is grown in accordance with a random parameter. Final predictions are obtained by aggregating over the ensemble. As the base constituents of the ensemble are tree-structured predictors, and since each of these trees is constructed using an injection of randomness, these procedures are called "random forests."

1.1 Random Forests

Breiman's ideas were decisively influenced by the early work of Amit and Geman (1997) on geometric feature selection, the random subspace method of Ho (1998) and the random split selection approach of Dietterich (2000). As highlighted by various empirical studies (see for instance Breiman, 2001; Svetnik et al., 2003; Diaz-Uriarte and de Andrés, 2006; Genuer et al., 2008, 2010), random forests have emerged as serious competitors to state-of-the-art methods such as boosting (Freund and Shapire, 1996) and support vector machines (Shawe-Taylor and Cristianini, 2004). They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting. In fact, they are considered to be one of the most accurate general-purpose learning techniques available. The survey by Genuer et al. (2008) may provide the reader with practical guidelines and a good starting point for understanding the method.

[*] Also at DMA, Ecole Normale Supérieure, 45 rue d'Ulm, 75230 Paris Cedex 05, France.

In Breiman's approach, each tree in the collection is formed by first selecting at random, at each node, a small group of input coordinates (also called features or variables hereafter) to split on and, secondly, by calculating the best split based on these features in the training set. The tree is grown using CART methodology (Breiman et al., 1984) to maximum size, without pruning. This subspace randomization scheme is blended with bagging (Breiman, 1996; Bühlmann and Yu, 2002; Buja and Stuetzle, 2006; Biau et al., 2010) to resample, with replacement, the training data set each time a new individual tree is grown.

Although the mechanism appears simple, it involves many different driving forces which make it difficult to analyse. In fact, its mathematical properties remain to date largely unknown and, up to now, most theoretical studies have concentrated on isolated parts or stylized versions of the algorithm. Interesting attempts in this direction are by Lin and Jeon (2006), who establish a connection between random forests and adaptive nearest neighbor methods (see also Biau and Devroye, 2010, for further results); Meinshausen (2006), who studies the consistency of random forests in the context of conditional quantile prediction; and Biau et al. (2008), who offer consistency theorems for various simplified versions of random forests and other randomized ensemble predictors. Nevertheless, the statistical mechanism of "true" random forests is not yet fully understood and is still under active investigation.

In the present paper, we go one step further into random forests by working out and solidifying the properties of a model suggested by Breiman (2004). Though this model is still simple compared to the "true" algorithm, it is nevertheless closer to reality than any other scheme we are aware of. The short draft of Breiman (2004) is essentially based on intuition and mathematical heuristics, some of them are questionable and make the document difficult to read and understand. However, the ideas presented by Breiman are worth clarifying and developing, and they will serve as a starting point for our study.

Before we formalize the model, some definitions are in order. Throughout the document, we suppose that we are given a training sample $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ of i.i.d. $[0,1]^d \times \mathbb{R}$-valued random variables ($d \geq 2$) with the same distribution as an independent generic pair $(\mathbf{X}, Y)$ satisfying $\mathbb{E}Y^2 < \infty$. The space $[0,1]^d$ is equipped with the standard Euclidean metric. For fixed $\mathbf{x} \in [0,1]^d$, our goal is to estimate the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ using the data $\mathcal{D}_n$. In this respect, we say that a regression function estimate r_n is consistent if $\mathbb{E}[r_n(\mathbf{X}) - r(\mathbf{X})]^2 \to 0$ as $n \to \infty$. The main message of this paper is that Breiman's procedure is consistent and adapts to sparsity, in the sense that its rate of convergence depends only on the number of strong features and not on how many noise variables are present.

1.2 The Model

Formally, a random forest is a predictor consisting of a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$, where $\Theta_1, \Theta_2, \ldots$ are i.i.d. outputs of a randomizing variable Θ. These random trees are combined to form the aggregated regression estimate

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_\Theta\left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n)\right],$$

where $\mathbb{E}_\Theta$ denotes expectation with respect to the random parameter, conditionally on $\mathbf{X}$ and the data set $\mathcal{D}_n$. In the following, to lighten notation a little, we will omit the dependency of the estimates in the sample, and write for example $\bar{r}_n(\mathbf{X})$ instead of $\bar{r}_n(\mathbf{X}, \mathcal{D}_n)$. Note that, in practice, the above expectation is evaluated by Monte Carlo, that is, by generating M (usually large) random trees, and taking the average of the individual outcomes (this procedure is justified by the law of large numbers, see the appendix in Breiman, 2001). The randomizing variable Θ is used to determine how the successive cuts are performed when building the individual trees, such as selection of the coordinate to split and position of the split.

In the model we have in mind, the variable Θ is assumed to be independent of $\mathbf{X}$ and the training sample $\mathcal{D}_n$. This excludes in particular any bootstrapping or resampling step in the training set. This also rules out any data-dependent strategy to build the trees, such as searching for optimal splits by optimizing some criterion on the actual observations. However, we allow Θ to be based on a second sample, independent of, but distributed as, $\mathcal{D}_n$. This important issue will be thoroughly discussed in Section 3.

With these warnings in mind, we will assume that each individual random tree is constructed in the following way. All nodes of the tree are associated with rectangular cells such that at each step of the construction of the tree, the collection of cells associated with the leaves of the tree (i.e., external nodes) forms a partition of $[0,1]^d$. The root of the tree is $[0,1]^d$ itself. The following procedure is then repeated $\lceil \log_2 k_n \rceil$ times, where $\log_2$ is the base-2 logarithm, $\lceil . \rceil$ the ceiling function and $k_n \geq 2$ a deterministic parameter, fixed beforehand by the user, and possibly depending on n.

1. At each node, a coordinate of $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ is selected, with the j-th feature having a probability $p_{nj} \in (0,1)$ of being selected.

2. At each node, once the coordinate is selected, the split is at the midpoint of the chosen side.

Each randomized tree $r_n(\mathbf{X}, \Theta)$ outputs the average over all Y_i for which the corresponding vectors $\mathbf{X}_i$ fall in the same cell of the random partition as $\mathbf{X}$. In other words, letting $A_n(\mathbf{X}, \Theta)$ be the rectangular cell of the random partition containing $\mathbf{X}$,

$$r_n(\mathbf{X}, \Theta) = \frac{\sum_{i=1}^n Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}}{\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X}, \Theta)},$$

where the event $\mathcal{E}_n(\mathbf{X}, \Theta)$ is defined by

$$\mathcal{E}_n(\mathbf{X}, \Theta) = \left[\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]} \neq 0 \right].$$

(Thus, by convention, the estimate is set to 0 on empty cells.) Taking finally expectation with respect to the parameter Θ, the random forests regression estimate takes the form

$$\bar{r}_n(\mathbf{X}) = \mathbb{E}_\Theta \left[r_n(\mathbf{X}, \Theta) \right] = \mathbb{E}_\Theta \left[\frac{\sum_{i=1}^n Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}}{\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X}, \Theta)} \right].$$

Let us now make some general remarks about this random forests model. First of all, we note that, by construction, each individual tree has exactly $2^{\lceil \log_2 k_n \rceil}$ ($\approx k_n$) terminal nodes, and each leaf has Lebesgue measure $2^{-\lceil \log_2 k_n \rceil}$ ($\approx 1/k_n$). Thus, if $\mathbf{X}$ has uniform distribution on $[0,1]^d$, there will be on average about n/k_n observations per terminal node. In particular, the choice $k_n = n$ induces a very small number of cases in the final leaves, in accordance with the idea that the single trees should not be pruned.

Next, we see that, during the construction of the tree, at each node, each candidate coordinate $X^{(j)}$ may be chosen with probability $p_{nj} \in (0,1)$. This implies in particular $\sum_{j=1}^d p_{nj} = 1$. Although we do not precise for the moment the way these probabilities are generated, we stress that they may be induced by a second sample. This includes the situation where, at each node, randomness is introduced by selecting at random (with or without replacement) a small group of input features to split on, and choosing to cut the cell along the coordinate—inside this group—which most decreases some empirical criterion evaluated on the extra sample. This scheme is close to what the original random forests algorithm does, the essential difference being that the latter algorithm uses the actual data set to calculate the best splits. This point will be properly discussed in Section 3.

Finally, the requirement that the splits are always achieved at the middle of the cell sides is mainly technical, and it could eventually be replaced by a more involved random mechanism—based on the second sample—at the price of a much more complicated analysis.

The document is organized as follows. In Section 2, we prove that the random forests regression estimate $\bar{r}_n$ is consistent and discuss its rate of convergence. As a striking result, we show under a sparsity framework that the rate of convergence depends only on the number of active (or strong) variables and not on the dimension of the ambient space. This feature is particularly desirable in high-dimensional regression, when the number of variables can be much larger than the sample size, and may explain why random forests are able to handle a very large number of input variables without overfitting. Section 3 is devoted to a discussion, and a small simulation study is presented in Section 4. For the sake of clarity, proofs are postponed to Section 5.

2. Asymptotic Analysis

Throughout the document, we denote by $N_n(\mathbf{X},\Theta)$ the number of data points falling in the same cell as $\mathbf{X}$, that is,

$$N_n(\mathbf{X},\Theta) = \sum_{i=1}^{n} \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}.$$

We start the analysis with the following simple theorem, which shows that the random forests estimate $\bar{r}_n$ is consistent.

Theorem 1 *Assume that the distribution of* $\mathbf{X}$ *has support on* $[0,1]^d$. *Then the random forests estimate* $\bar{r}_n$ *is consistent whenever* $p_{nj}\log k_n \to \infty$ *for all* $j = 1,\ldots,d$ *and* $k_n/n \to 0$ *as* $n \to \infty$.

Theorem 1 mainly serves as an illustration of how the consistency problem of random forests predictors may be attacked. It encompasses, in particular, the situation where, at each node, the coordinate to split is chosen uniformly at random over the d candidates. In this "purely random" model, $p_{nj} = 1/d$, independently of n and j, and consistency is ensured as long as $k_n \to \infty$ and $k_n/n \to 0$. This is however a radically simplified version of the random forests used in practice, which does not explain the good performance of the algorithm. To achieve this goal, a more in-depth analysis is needed.

There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation. As an example, wavelet coefficients of images often exhibit exponential decay, and a relatively small subset of all wavelet coefficients allows for a good approximation of the original image. Such signals have few non-zero coefficients and can therefore be described as sparse in the signal domain (see for instance Bruckstein et al., 2009). Similarly, recent advances in high-throughput technologies—such as array comparative genomic hybridization—indicate that, despite the huge dimensionality of problems, only a small number of genes may play a role in determining the outcome and be required to create good predictors (van't Veer et al., 2002, for instance). Sparse estimation is playing an increasingly important role in the statistics and machine learning communities, and several methods have recently been developed in both fields, which rely upon the notion of sparsity (e.g., penalty methods like the Lasso and Dantzig selector, see Tibshirani, 1996; Candès and Tao, 2005; Bunea et al., 2007; Bickel et al., 2009, and the references therein).

Following this idea, we will assume in our setting that the target regression function $r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$, which is initially a function of $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$, depends in fact only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the d features. In other words, letting $\mathbf{X}_{\mathcal{S}} = (X_j : j \in \mathcal{S})$ and $S = \text{Card}\, \mathcal{S}$, we have

$$r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}]$$

or equivalently, for any $\mathbf{x} \in [0,1]^d$,

$$r(\mathbf{x}) = r^\star(\mathbf{x}_{\mathcal{S}}) \quad \mu\text{-a.s.}, \tag{1}$$

where μ is the distribution of $\mathbf{X}$ and $r^\star : [0,1]^S \to \mathbb{R}$ is the section of r corresponding to $\mathcal{S}$. To avoid trivialities, we will assume throughout that $\mathcal{S}$ is nonempty, with $S \geq 2$. The variables in the set $\mathcal{W} = \{1, \ldots, d\} - \mathcal{S}$ (for $\mathcal{W}$eak) have thus no influence on the response and could be safely removed. In the dimension reduction scenario we have in mind, the ambient dimension d can be very large, much larger than the sample size n, but we believe that the representation is sparse, that is, that very few coordinates of r are non-zero, with indices corresponding to the set $\mathcal{S}$. Note however that representation (1) does not forbid the somehow undesirable case where $S = d$. As such, the value S characterizes the sparsity of the model: The smaller S, the sparser r.

Within this sparsity framework, it is intuitively clear that the coordinate-sampling probabilities should ideally satisfy the constraints $p_{nj} = 1/S$ for $j \in \mathcal{S}$ (and, consequently, $p_{nj} = 0$ otherwise). However, this is a too strong requirement, which has no chance to be satisfied in practice, except maybe in some special situations where we know beforehand which variables are important and which are not. Thus, to stick to reality, we will rather require in the following that $p_{nj} = (1/S)(1 + \xi_{nj})$ for $j \in \mathcal{S}$ (and $p_{nj} = \xi_{nj}$ otherwise), where $p_{nj} \in (0,1)$ and each ξ_{nj} tends to 0 as n tends to infinity. We will see in Section 3 how to design a randomization mechanism to obtain such probabilities, on the basis of a second sample independent of the training set $\mathcal{D}_n$. At this point, it is important to note that the dimensions d and S are held constant throughout the document. In particular, these dimensions are *not* functions of the sample size n, as it may be the case in other asymptotic studies.

We have now enough material for a deeper understanding of the random forests algorithm. To lighten notation a little, we will write

$$W_{ni}(\mathbf{X}, \Theta) = \frac{\mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}}{N_n(\mathbf{X}, \Theta)} \mathbf{1}_{\mathcal{E}_n(\mathbf{X}, \Theta)},$$

so that the estimate takes the form

$$\bar{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta \left[W_{ni}(\mathbf{X}, \Theta)\right] Y_i.$$

Let us start with the variance/bias decomposition

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 + \mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2, \tag{2}$$

where we set

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta \left[W_{ni}(\mathbf{X},\Theta)\right] r(\mathbf{X}_i).$$

The two terms of (2) will be examined separately, in Proposition 2 and Proposition 4, respectively. Throughout, the symbol $\mathbb{V}$ denotes variance.

Proposition 2 *Assume that* $\mathbf{X}$ *is uniformly distributed on* $[0,1]^d$ *and, for all* $\mathbf{x} \in \mathbb{R}^d$,

$$\sigma^2(\mathbf{x}) = \mathbb{V}[Y \,|\, \mathbf{X} = \mathbf{x}] \leq \sigma^2$$

for some positive constant σ^2*. Then, if* $p_{nj} = (1/S)(1+\xi_{nj})$ *for* $j \in \mathcal{S}$,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 \leq C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2d} (1+\xi_n) \frac{k_n}{n(\log k_n)^{S/2d}},$$

where

$$C = \frac{288}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2d}.$$

The sequence (ξ_n) *depends on the sequences* $\{(\xi_{nj}) : j \in \mathcal{S}\}$ *only and tends to* 0 *as n tends to infinity.*

Remark 3 *A close inspection of the end of the proof of Proposition 2 reveals that*

$$1+\xi_n = \prod_{j \in \mathcal{S}} \left[(1+\xi_{nj})^{-1}\left(1 - \frac{\xi_{nj}}{S-1}\right)^{-1}\right]^{1/2d}.$$

In particular, if $a < p_{nj} < b$ *for some constants* $a,b \in (0,1)$*, then*

$$1+\xi_n \leq \left(\frac{S-1}{S^2 a(1-b)}\right)^{S/2d}.$$

The main message of Proposition 2 is that the variance of the forests estimate is $O(k_n/(n(\log k_n)^{S/2d}))$. This result is interesting by itself since it shows the effect of aggregation on the variance of the forest. To understand this remark, recall that individual (random or not) trees are proved to be consistent by letting the number of cases in each terminal node become large (see Devroye et al., 1996, Chapter 20), with a typical variance of the order k_n/n. Thus, for such trees, the choice $k_n = n$ (i.e., about one observation on average in each terminal node) is clearly not suitable and leads to serious overfitting and variance explosion. On the other hand, the variance of the forest is of the order $k_n/(n(\log k_n)^{S/2d})$. Therefore, letting $k_n = n$, the variance is of the order $1/(\log n)^{S/2d}$, a quantity which still goes to 0 as n grows! Proof of Proposition 2 reveals that this log term is a by-product of the Θ-averaging process, which appears by taking into consideration the correlation between trees. We believe that it provides an interesting perspective on why random forests are still able to do a good job, despite the fact that individual trees are not pruned.

Note finally that the requirement that $\mathbf{X}$ is uniformly distributed on the hypercube could be safely replaced by the assumption that $\mathbf{X}$ has a density with respect to the Lebesgue measure on $[0,1]^d$ and the density is bounded from above and from below. The case where the density of $\mathbf{X}$ is not bounded from below necessitates a specific analysis, which we believe is beyond the scope of the present paper. We refer the reader to Biau and Devroye (2010) for results in this direction (see also Remark 10 in Section 5).

Let us now turn to the analysis of the bias term in equality (2). Recall that $r^\star$ denotes the section of r corresponding to $\mathcal{S}$.

Proposition 4 *Assume that* $\mathbf{X}$ *is uniformly distributed on* $[0,1]^d$ *and* $r^\star$ *is L-Lipschitz on* $[0,1]^S$. *Then, if* $p_{nj} = (1/S)(1+\xi_{nj})$ *for* $j \in \mathcal{S}$,

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \leq \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right] e^{-n/2k_n},$$

where $\gamma_n = \min_{j\in\mathcal{S}} \xi_{nj}$ *tends to* 0 *as n tends to infinity.*

This result essentially shows that the rate at which the bias decreases to 0 depends on the number of strong variables, not on d. In particular, the quantity ${k_n}^{-(0.75/(S\log 2))(1+\gamma_n)}$ should be compared with the ordinary partitioning estimate bias, which is of the order ${k_n}^{-2/d}$ under the smoothness conditions of Proposition 4 (see for instance Györfi et al., 2002). In this respect, it is easy to see that ${k_n}^{-(0.75/(S\log 2))(1+\gamma_n)} = \mathrm{o}({k_n}^{-2/d})$ as soon as $S \leq \lfloor 0.54d \rfloor$ ($\lfloor . \rfloor$ is the integer part function). In other words, when the number of active variables is less than (roughly) half of the ambient dimension, the bias of the random forests regression estimate decreases to 0 much faster than the usual rate. The restriction $S \leq \lfloor 0.54d \rfloor$ is not severe, since in all practical situations we have in mind, d is usually very large with respect to S (this is, for instance, typically the case in modern genome biology problems, where d may be of the order of billion, and in any case much larger than the actual number of active features). Note at last that, contrary to Proposition 2, the term $e^{-n/2k_n}$ prevents the extreme choice $k_n = n$ (about one observation on average in each terminal node). Indeed, an inspection of the proof of Proposition 4 reveals that this term accounts for the probability that $N_n(\mathbf{X},\Theta)$ is precisely 0, that is, $A_n(\mathbf{X},\Theta)$ is empty.

Recalling the elementary inequality $ze^{-nz} \leq e^{-1}/n$ for $z \in [0,1]$, we may finally join Proposition 2 and Proposition 4 and state our main theorem.

Theorem 5 *Assume that* $\mathbf{X}$ *is uniformly distributed on* $[0,1]^d$, $r^\star$ *is L-Lipschitz on* $[0,1]^S$ *and, for all* $\mathbf{x} \in \mathbb{R}^d$,

$$\sigma^2(\mathbf{x}) = \mathbb{V}[Y \,|\, \mathbf{X} = \mathbf{x}] \leq \sigma^2$$

for some positive constant σ^2. *Then, if* $p_{nj} = (1/S)(1+\xi_{nj})$ *for* $j \in \mathcal{S}$, *letting* $\gamma_n = \min_{j\in\mathcal{S}} \xi_{nj}$, *we have*

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \leq \Xi_n \frac{k_n}{n} + \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}},$$

where

$$\Xi_n = C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2d} (1+\xi_n) + 2e^{-1}\left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right]$$

and

$$C = \frac{288}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2d}.$$

The sequence (ξ_n) *depends on the sequences* $\{(\xi_{nj}) : j \in \mathcal{S}\}$ *only and tends to* 0 *as n tends to infinity.*

As we will see in Section 3, it may be safely assumed that the randomization process allows for $\xi_{nj} \log n \to 0$ as $n \to \infty$, for all $j \in \mathcal{S}$. Thus, under this condition, Theorem 5 shows that with the optimal choice

$$k_n \propto n^{1/(1+\frac{0.75}{S\log 2})},$$

we get

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = O\left(n^{\frac{-0.75}{S\log 2+0.75}}\right).$$

This result can be made more precise. Denote by $\mathcal{F}_S$ the class of (L,σ^2)-smooth distributions $(\mathbf{X},Y)$ such that $\mathbf{X}$ has uniform distribution on $[0,1]^d$, the regression function $r^\star$ is Lipschitz with constant L on $[0,1]^S$ and, for all $\mathbf{x} \in \mathbb{R}^d$, $\sigma^2(\mathbf{x}) = \mathbb{V}[Y \,|\, \mathbf{X} = \mathbf{x}] \leq \sigma^2$.

Corollary 6 *Let*

$$\Xi = C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2d} + 2e^{-1}\left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right]$$

and

$$C = \frac{288}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2d}.$$

Then, if $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$, with $\xi_{nj}\log n \to 0$ as $n \to \infty$, for the choice

$$k_n \propto \left(\frac{L^2}{\Xi}\right)^{1/(1+\frac{0.75}{S\log 2})} n^{1/(1+\frac{0.75}{S\log 2})},$$

we have

$$\limsup_{n\to\infty} \sup_{(\mathbf{X},Y)\in\mathcal{F}_S} \frac{\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2}{\left(\Xi L^{\frac{2S\log 2}{0.75}}\right)^{\frac{0.75}{S\log 2+0.75}} n^{\frac{-0.75}{S\log 2+0.75}}} \leq \Lambda,$$

where Λ is a positive constant independent of r, L and σ^2.

This result reveals the fact that the L_2-rate of convergence of $\bar{r}_n(\mathbf{X})$ to $r(\mathbf{X})$ depends only on the number S of strong variables, and not on the ambient dimension d. The main message of Corollary 6 is that if we are able to properly tune the probability sequences $(p_{nj})_{n\geq 1}$ and make them sufficiently fast to track the informative features, then the rate of convergence of the random forests estimate will be of the order $n^{\frac{-0.75}{S\log 2+0.75}}$. This rate is strictly faster than the usual rate $n^{-2/(d+2)}$ as soon as $S \leq \lfloor 0.54d \rfloor$. To understand this point, just recall that the rate $n^{-2/(d+2)}$ is minimax optimal for the class $\mathcal{F}_d$ (see, for example Ibragimov and Khasminskii, 1980, 1981, 1982), seen as a collection of regression functions over $[0,1]^d$, *not* $[0,1]^S$. However, in our setting, the intrinsic dimension of the regression problem is S, not d, and the random forests estimate cleverly adapts to the sparsity of the problem. As an illustration, Figure 1 shows the plot of the function $S \mapsto 0.75/(S\log 2 + 0.75)$ for S ranging from 2 to $d = 100$.

It is noteworthy that the rate of convergence of the ξ_{nj} to 0 (and, consequently, the rate at which the probabilities p_{nj} approach $1/S$ for $j \in \mathcal{S}$) will eventually depend on the ambient dimension d through the ratio S/d. The same is true for the Lipschitz constant L and the factor $\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})$ which both appear in Corollary 6. To figure out this remark, remember first that the support of r is

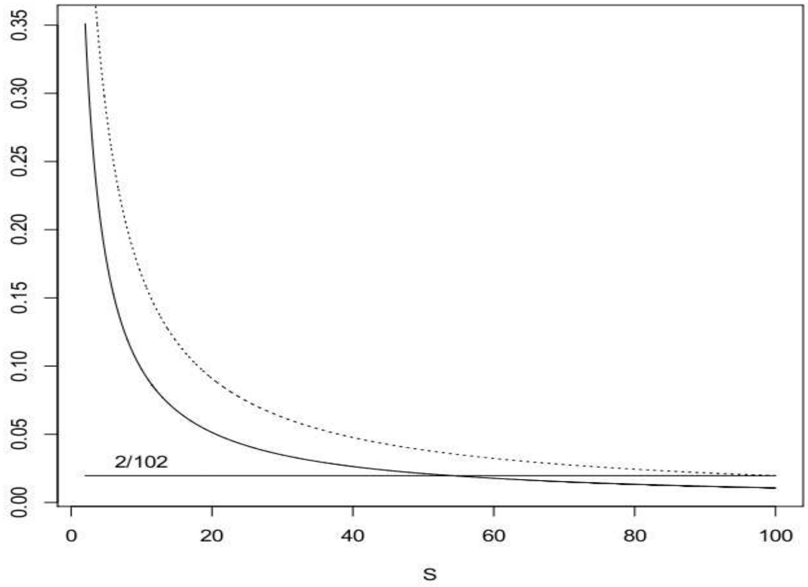


Figure 1: **Solid line**: Plot of the function $S \mapsto 0.75/(S\log 2+0.75)$ for S ranging from 2 to $d = 100$. **Dotted line**: Plot of the minimax rate power $S \mapsto 2/(S+2)$. The horizontal line shows the value of the d-dimensional rate power $2/(d+2) \approx 0.0196$.

contained in $\mathbb{R}^S$, so that the later supremum (respectively, the Lipschitz constant) is in fact a supremum (respectively, a Lipschitz constant) over $\mathbb{R}^S$, *not* over $\mathbb{R}^d$. Next, denote by $\mathcal{C}_p(s)$ the collection of functions $\eta : [0,1]^p \to [0,1]$ for which each derivative of order s satisfies a Lipschitz condition. It is well known that the ε-entropy $\log_2(\mathcal{N}_\varepsilon)$ of $\mathcal{C}_p(s)$ is $\Phi(\varepsilon^{-p/(s+1)})$ as $\varepsilon \downarrow 0$ (Kolmogorov and Tihomirov, 1961), where $a_n = \Phi(b_n)$ means that $a_n = O(b_n)$ and $b_n = O(a_n)$. Here we have an interesting interpretation of the dimension reduction phenomenon: Working with Lipschitz functions on $\mathbb{R}^S$ (that is, $s = 0$) is roughly equivalent to working with functions on $\mathbb{R}^d$ for which all $[(d/S)-1]$-th order derivatives are Lipschitz! For example, if $S = 1$ and $d = 25$, $(d/S)-1 = 24$ and, as there are 25^{24} such partial derivatives in $\mathbb{R}^{25}$, we note immediately the potential benefit of recovering the "true" dimension S.

Remark 7 *The reduced-dimensional rate $n^{\frac{-0.75}{S\log 2+0.75}}$ is strictly larger than the S-dimensional optimal rate $n^{-2/(S+2)}$, which is also shown in Figure 1 for S ranging from 2 to 100. We do not know whether the latter rate can be achieved by the algorithm.*

Remark 8 *The optimal parameter k_n of Corollary 6 depends on the unknown distribution of $(\mathbf{X}, Y)$, especially on the smoothness of the regression function and the effective dimension S. To correct this situation, adaptive (i.e., data-dependent) choices of k_n, such as data-splitting or cross-validation, should preserve the rate of convergence of the estimate. Another route we may follow is to analyse the effect of bootstrapping the sample before growing the individual trees (i.e., bagging). It is our* *belief that this procedure should also preserve the rate of convergence, even for overfitted trees ($k_n \approx n$), in the spirit of Biau et al. (2010). However, such a study is beyond the scope of the present paper.*

Remark 9 *For further references, it is interesting to note that Proposition 2 (variance term) is a consequence of aggregation, whereas Proposition 4 (bias term) is a consequence of randomization.*

It is also stimulating to keep in mind the following analysis, which has been suggested to us by a referee. Suppose, to simplify, that $Y = r(\mathbf{X})$ (no-noise regression) and that $\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta) = 1$ a.s. In this case, the variance term is 0 and we have

$$\bar{r}_n(\mathbf{X}) = \tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_{\Theta}\left[W_{ni}(\Theta,\mathbf{X})\right] Y_i.$$

Set $\mathbf{Z}_n = (Y, Y_1, \ldots, Y_n)$. Then

$$\begin{aligned}
\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 &= \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - Y\right]^2 \\
&= \mathbb{E}\left[\mathbb{E}\left[(\bar{r}_n(\mathbf{X}) - Y)^2 \,|\, \mathbf{Z}_n\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(\bar{r}_n(\mathbf{X}) - \mathbb{E}[\bar{r}_n(\mathbf{X}) \,|\, \mathbf{Z}_n])^2 \,|\, \mathbf{Z}_n\right]\right] + \mathbb{E}\left[\mathbb{E}[\bar{r}_n(\mathbf{X}) \,|\, \mathbf{Z}_n] - Y\right]^2.
\end{aligned}$$

The conditional expectation in the first of the two terms above may be rewritten under the form

$$\mathbb{E}\left[\mathrm{Cov}\left(\mathbb{E}_{\Theta}\left[r_n(\mathbf{X},\Theta)\right], \mathbb{E}_{\Theta'}\left[r_n(\mathbf{X},\Theta')\right] \,|\, \mathbf{Z}_n\right)\right],$$

where Θ' is distributed as, and independent of, Θ. Attention shows that this last term is indeed equal to

$$\mathbb{E}\left[\mathbb{E}_{\Theta,\Theta'}\mathrm{Cov}\left(r_n(\mathbf{X},\Theta), r_n(\mathbf{X},\Theta') \,|\, \mathbf{Z}_n\right)\right].$$

The key observation is that if trees have strong predictive power, then they can be unconditionally strongly correlated while being conditionally weakly correlated. This opens an interesting line of research for the statistical analysis of the bias term, in connection with Amit (2002) and Blanchard (2004) conditional covariance-analysis ideas.

3. Discussion

The results which have been obtained in Section 2 rely on appropriate behavior of the probability sequences $(p_{nj})_{n\geq 1}$, $j = 1, \ldots, d$. We recall that these sequences should be in $(0,1)$ and obey the constraints $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$ (and $p_{nj} = \xi_{nj}$ otherwise), where the $(\xi_{nj})_{n\geq 1}$ tend to 0 as n tends to infinity. In other words, at each step of the construction of the individual trees, the random procedure should track and preferentially cut the strong coordinates. In this more informal section, we briefly discuss a random mechanism for inducing such probability sequences.

Suppose, to start with an imaginary scenario, that we already know which coordinates are strong, and which are not. In this ideal case, the random selection procedure described in the introduction may be easily made more precise as follows. A positive integer M_n—possibly depending on n—is fixed beforehand and the following splitting scheme is iteratively repeated at each node of the tree:

1. Select at random, with replacement, M_n candidate coordinates to split on.

2. If the selection is all weak, then choose one at random to split on. If there is more than one strong variable elected, choose one at random and cut.

Within this framework, it is easy to see that each coordinate in $\mathcal{S}$ will be cut with the "ideal" probability

$$p_n^\star = \frac{1}{S}\left[1-\left(1-\frac{S}{d}\right)^{M_n}\right].$$

Though this is an idealized model, it already gives some information about the choice of the parameter M_n, which, in accordance with the results of Section 2 (Corollary 6), should satisfy

$$\left(1-\frac{S}{d}\right)^{M_n} \log n \to 0 \quad \text{as } n \to \infty.$$

This is true as soon as

$$M_n \to \infty \quad \text{and} \quad \frac{M_n}{\log n} \to \infty \quad \text{as } n \to \infty.$$

This result is consistent with the general empirical finding that M_n (called `mtry` in the R package `RandomForests`) does not need to be very large (see, for example, Breiman, 2001), but not with the widespread belief that M_n should not depend on n. Note also that if the M_n features are chosen at random *without* replacement, then things are even more simple since, in this case, $p_n^\star = 1/S$ for all n large enough.

In practice, we have only a vague idea about the size and content of the set $\mathcal{S}$. However, to circumvent this problem, we may use the observations of an independent second set $\mathcal{D}_n'$ (say, of the same size as $\mathcal{D}_n$) in order to mimic the ideal split probability $p_n^\star$. To illustrate this mechanism, suppose—to keep things simple—that the model is linear, that is,

$$Y = \sum_{j\in\mathcal{S}} a_j X^{(j)} + \varepsilon,$$

where $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ is uniformly distributed over $[0,1]^d$, the a_j are non-zero real numbers, and ε is a zero-mean random noise, which is assumed to be independent of $\mathbf{X}$ and with finite variance. Note that, in accordance with our sparsity assumption, $r(\mathbf{X}) = \sum_{j\in\mathcal{S}} a_j X^{(j)}$ depends on $\mathbf{X}_{\mathcal{S}}$ only.

Assume now that we have done some splitting and arrived at a current set of terminal nodes. Consider any of these nodes, say $A = \prod_{j=1}^d A_j$, fix a coordinate $j \in \{1, \ldots, d\}$, and look at the weighted conditional variance $\mathbb{V}[Y|X^{(j)} \in A_j]\,\mathbb{P}(X^{(j)} \in A_j)$. It is a simple exercise to prove that if $\mathbf{X}$ is uniform and $j \in \mathcal{S}$, then the split on the j-th side which most decreases the weighted conditional variance is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$. On the other hand, if $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

On the practical side, the conditional variances are of course unknown, but they may be estimated by replacing the theoretical quantities by their respective sample estimates (as in the CART procedure, see Breiman, 2001, Chapter 8, for a thorough discussion) evaluated on the second sample $\mathcal{D}_n'$. This suggests the following procedure, at each node of the tree:

1. Select at random, with replacement, M_n candidate coordinates to split on.

2. For each of the M_n elected coordinates, calculate the best split, that is, the split which most decreases the within-node sum of squares on the second sample $\mathcal{D}'_n$.

3. Select one variable at random among the coordinates which output the best within-node sum of squares decreases, and cut.

This procedure is indeed close to what the random forests algorithm does. The essential difference is that we suppose to have at hand a second sample $\mathcal{D}'_n$, whereas the original algorithm performs the search of the optimal cuts on the original observations $\mathcal{D}_n$. This point is important, since the use of an extra sample preserves the independence of Θ (the random mechanism) and $\mathcal{D}_n$ (the training sample). We do not know whether our results are still true if Θ depends on $\mathcal{D}_n$ (as in the CART algorithm), but the analysis does not appear to be simple. Note also that, at step 3, a threshold (or a test procedure, as suggested in Amaratunga et al., 2008) could be used to choose among the most significant variables, whereas the actual algorithm just selects the best one. In fact, depending on the context and the actual cut selection procedure, the informative probabilities p_{nj} ($j \in \mathcal{S}$) may obey the constraints $p_{nj} \to p_j$ as $n \to \infty$ (thus, p_j is not necessarily equal to $1/S$), where the p_j are positive and satisfy $\sum_{j\in\mathcal{S}} p_j = 1$. This should not affect the results of the article.

This empirical randomization scheme leads to complicate probabilities of cuts which, this time, vary at each node of each tree and are not easily amenable to analysis. Nevertheless, observing that the average number of cases per terminal node is about n/k_n, it may be inferred by the law of large numbers that each variable in $\mathcal{S}$ will be cut with probability

$$p_{nj} \approx \frac{1}{S}\left[1-\left(1-\frac{S}{d}\right)^{M_n}\right](1+\zeta_{nj}),$$

where ζ_{nj} is of the order $O(k_n/n)$, a quantity which anyway goes fast to 0 as n tends to infinity. Put differently, for $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}(1+\xi_{nj}),$$

where ξ_{nj} goes to 0 and satisfies the constraint $\xi_{nj}\log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$. This is coherent with the requirements of Corollary 6. We realize however that this is a rough approach, and that more theoretical work is needed here to fully understand the mechanisms involved in CART and Breiman's original randomization process.

It is also noteworthy that random forests use the so-called out-of-bag samples (i.e., the bootstrapped data which are not used to fit the trees) to construct a variable importance criterion, which measures the prediction strength of each feature (see, e.g., Genuer et al., 2010). As far as we are aware, there is to date no systematic mathematical study of this criterion. It is our belief that such a study would greatly benefit from the sparsity point of view developed in the present paper, but is unfortunately much beyond its scope. Lastly, it would also be interesting to work out and extend our results to the context of unsupervised learning of trees. A good route to follow with this respect is given by the strategies outlined in Section 5.5 of Amit and Geman (1997).

4. A Small Simulation Study

Even though the first vocation of the present paper is theoretical, we offer in this short section some experimental results on synthetic data. Our aim is not to provide a thorough practical study of the random forests method, but rather to illustrate the main ideas of the article. As for now, we let $\mathcal{U}([0,1]^d)$ (respectively, $\mathcal{N}(0,1)$) be the uniform distribution over $[0,1]^d$ (respectively, the standard Gaussian distribution). Specifically, three models were tested:

1. [**Sinus**] For $\mathbf{x} \in [0,1]^d$, the regression function takes the form

$$r(\mathbf{x}) = 10\sin(10\pi x^{(1)}).$$

   We let $Y = r(\mathbf{X}) + \varepsilon$ and $\mathbf{X} \sim \mathcal{U}([0,1]^d)$ $(d \geq 1)$, with $\varepsilon \sim \mathcal{N}(0,1)$.

2. [**Friedman #1**] This is a model proposed in Friedman (1991). Here,

$$r(\mathbf{x}) = 10\sin(\pi x^{(1)}x^{(2)}) + 20(x^{(3)} - .05)^2 + 10x^{(4)} + 5x^{(5)}$$

   and $Y = r(\mathbf{X}) + \varepsilon$, where $\mathbf{X} \sim \mathcal{U}([0,1]^d)$ $(d \geq 5)$ and $\varepsilon \sim \mathcal{N}(0,1)$.

3. [**Tree**] In this example, we let $Y = r(\mathbf{X}) + \varepsilon$, where $\mathbf{X} \sim \mathcal{U}([0,1]^d)$ $(d \geq 5)$, $\varepsilon \sim \mathcal{N}(0,1)$ and the function r has itself a tree structure. This tree-type function, which is shown in Figure 2, involves only five variables.

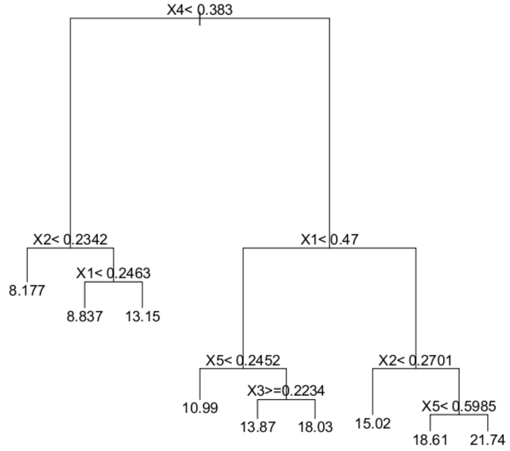


Figure 2: The tree used as regression function in the model **Tree**.

We note that, although the ambient dimension d may be large, the effective dimension of model 1 is $S = 1$, whereas model 2 and model 3 have $S = 5$. In other words, $\mathcal{S} = \{1\}$ for model 1, whereas $\mathcal{S} = \{1, \ldots, 5\}$ for model 2 and model 3. Observe also that, in our context, the model **Tree** should be considered as a "no-bias" model, on which the random forests algorithm is expected to perform well.

In a first series of experiments, we let $d = 100$ and, for each of the three models and different values of the sample size n, we generated a learning set of size n and fitted a forest (10000 trees) with $\mathtt{mtry} = d$. For $j = 1, \ldots, d$, the ratio (number of times the j-th coordinate is split)/(total number of splits over the forest) was evaluated, and the whole experiment was repeated 100 times. Figure 3, Figure 4 and Figure 5 report the resulting boxplots for each of the first twenty variables and different values of n. These figures clearly enlighten the fact that, as n grows, the probability of cuts does concentrate on the informative variables only and support the assumption that $\xi_{nj} \to 0$ as $n \to \infty$ for each $j \in \mathcal{S}$.

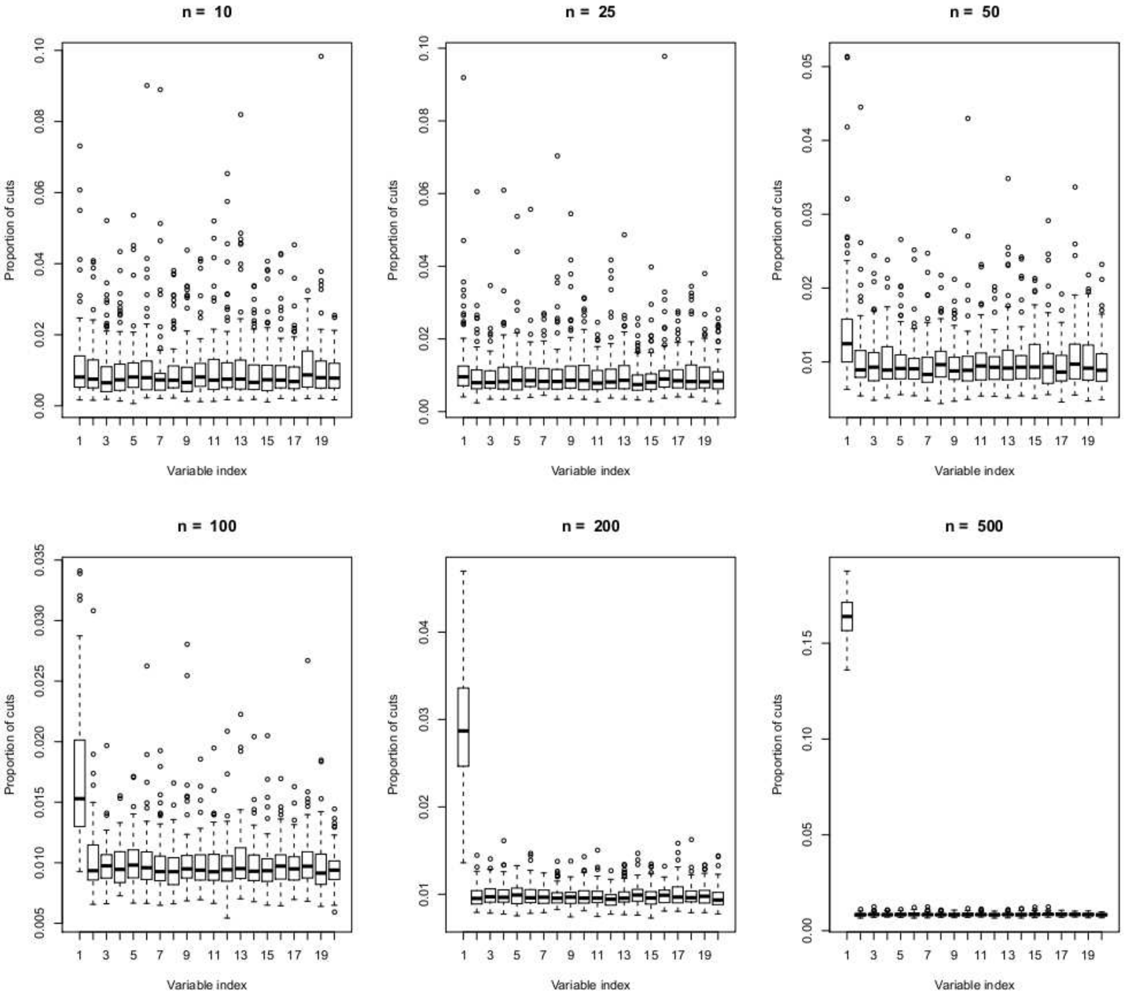


Figure 3: Boxplots of the empirical probabilities of cuts for model **Sinus** ($\mathcal{S} = \{1\}$).

Next, in a second series of experiments, for each model, for different values of d and for sample sizes n ranging from 10 to 1000, we generated a learning set of size n, a test set of size 50 000 and evaluated the mean squared error (MSE) of the random forests (RF) method via the Monte Carlo

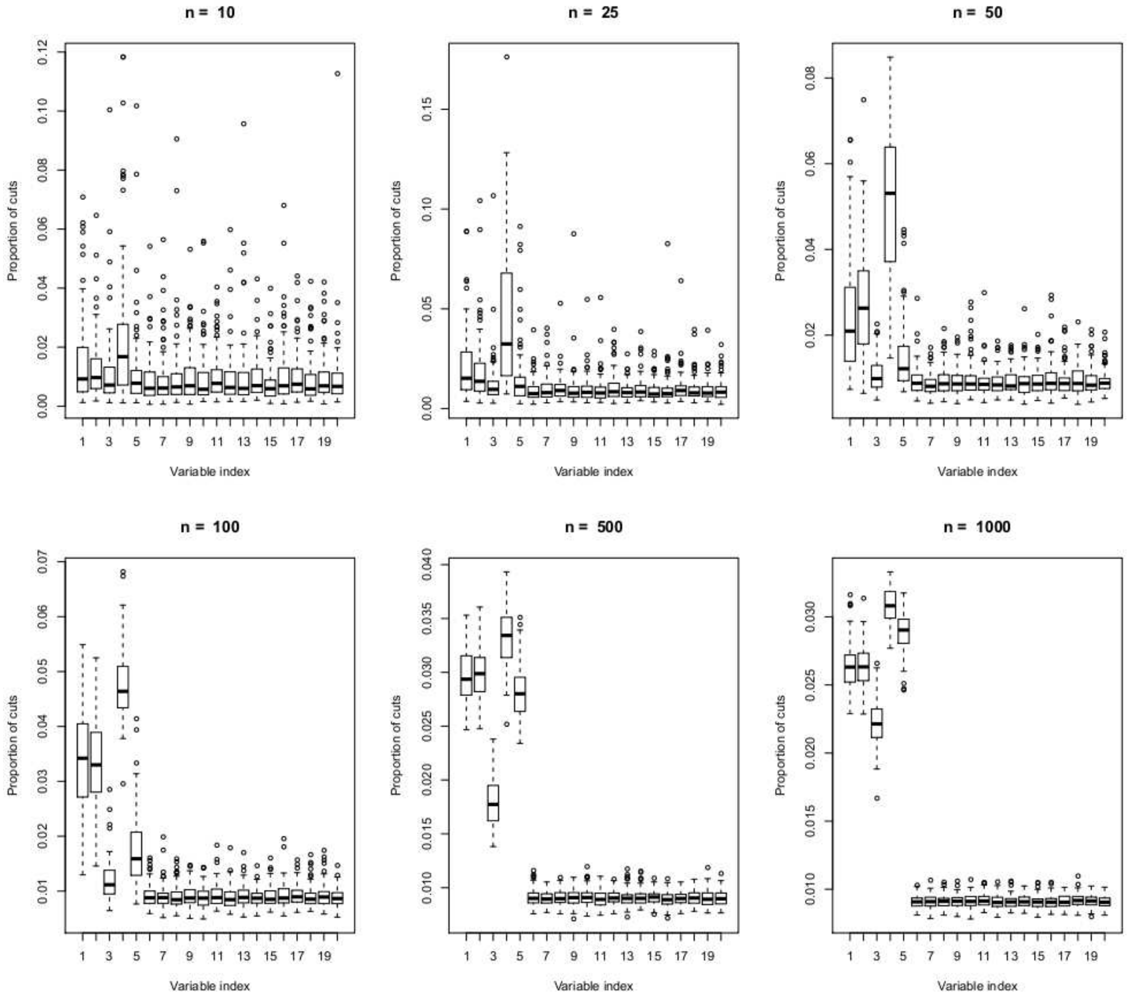


Figure 4: Boxplots of the empirical probabilities of cuts for model **Friedman #1** ($\mathcal{S} = \{1, \ldots, 5\}$).

approximation

$$\text{MSE} \approx \frac{1}{50\,000} \sum_{j=1}^{50\,000} \left[\text{RF}(\text{test data \#j}) - r(\text{test data \#j})\right]^2 .$$

All results were averaged over 100 data sets. The random forests algorithm was performed with the parameter `mtry` automatically tuned by the R package `RandomForests`, 1000 random trees and the minimum node size set to 5 (which is the default value for regression). Besides, in order to compare the "true" algorithm with the approximate model discussed in the present document, an alternative method was also tested. This auxiliary algorithm has characteristics which are identical

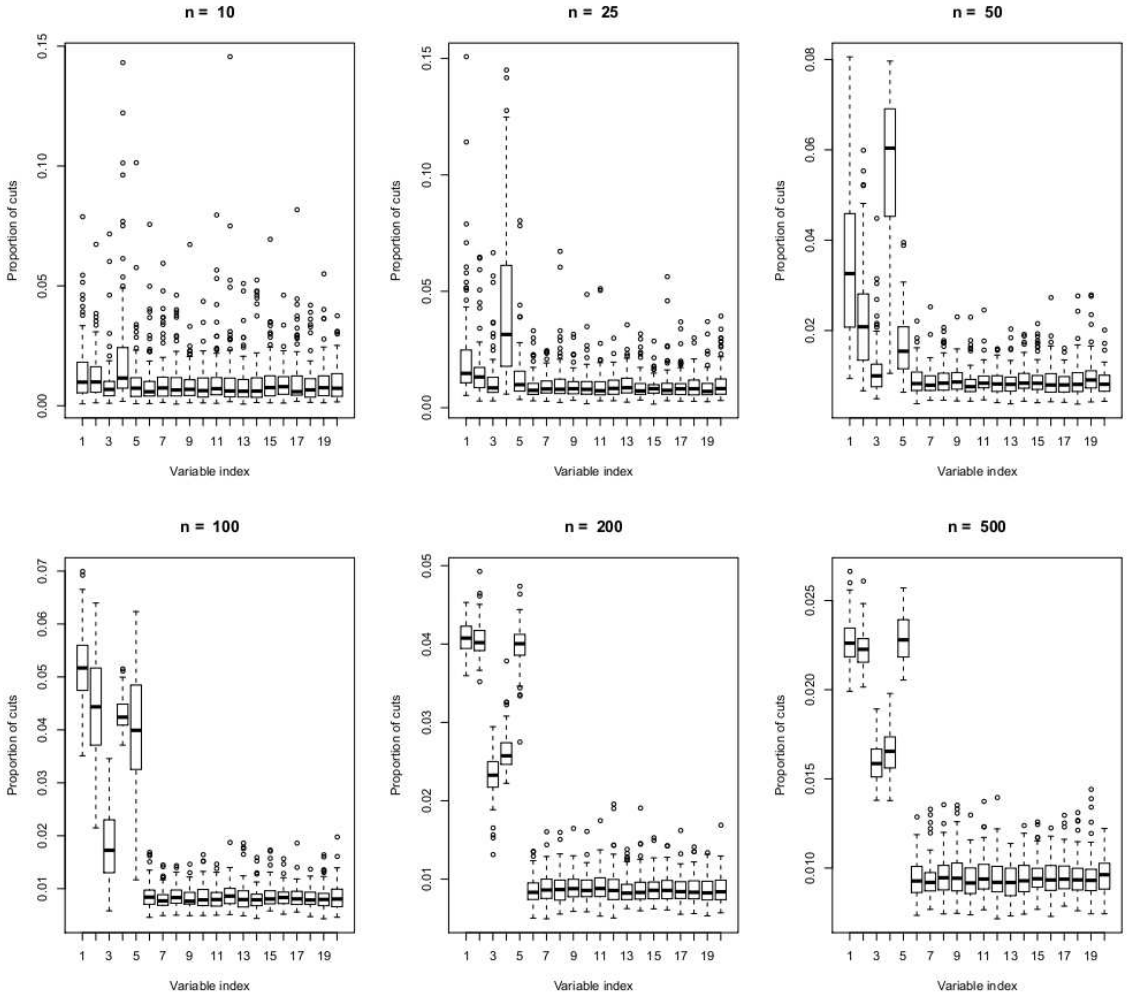


Figure 5: Boxplots of the empirical probabilities of cuts for model **Tree** ($\mathcal{S} = \{1, \ldots, 5\}$).

to the original ones (same `mtry`, same number of random trees), *with the notable difference that now the maximum number of nodes is fixed beforehand*. For the sake of coherence, since the minimum node size is set to 5 in the `RandomForests` package, the number of terminal nodes in the custom algorithm was calibrated to $\lceil n/5 \rceil$. It must be stressed that the essential difference between the standard random forests algorithm and the alternative one is that the number of cases in the final leaves is fixed in the former, whereas the latter assumes a fixed number of terminal nodes. In particular, in both algorithms, cuts are performed using the actual sample, just as CART does. To keep things simple, no data-splitting procedure has been incorporated in the modified version.

Figure 6, Figure 7 and Figure 8 illustrate the evolution of the MSE value with respect to n and d, for each model and the two tested procedures. First, we note that the overall performance of the alternative method is very similar to the one of the original algorithm. This confirms our idea that the model discussed in the present paper is a good approximation of the authentic Breiman's forests. Next, we see that for a sufficiently large n, the capabilities of the forests are nearly independent of d, in accordance with the idea that the (asymptotic) rate of convergence of the method should only depend on the "true" dimensionality S (Theorem 5). Finally, as expected, it is noteworthy that both algorithms perform well on the third model, which has been precisely designed for a tree-structured predictor.

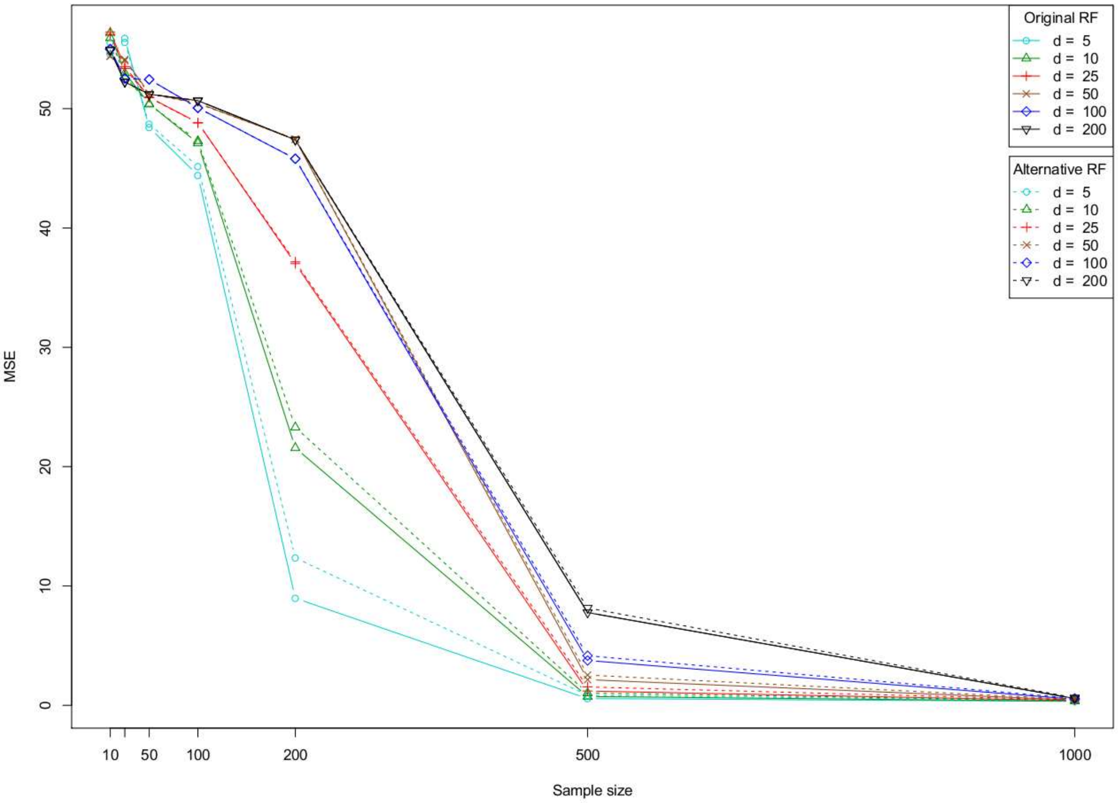


Figure 6: Evolution of the MSE for model **Sinus** ($S = 1$).

5. Proofs

Throughout this section, we will make repeated use of the following two facts.

Fact 1 *Let $K_{nj}(\mathbf{X}, \Theta)$ be the number of times the terminal node $A_n(\mathbf{X}, \Theta)$ is split on the j-th coordinate ($j = 1, \dots, d$). Then, conditionally on $\mathbf{X}$, $K_{nj}(\mathbf{X}, \Theta)$ has binomial distribution with parameters*

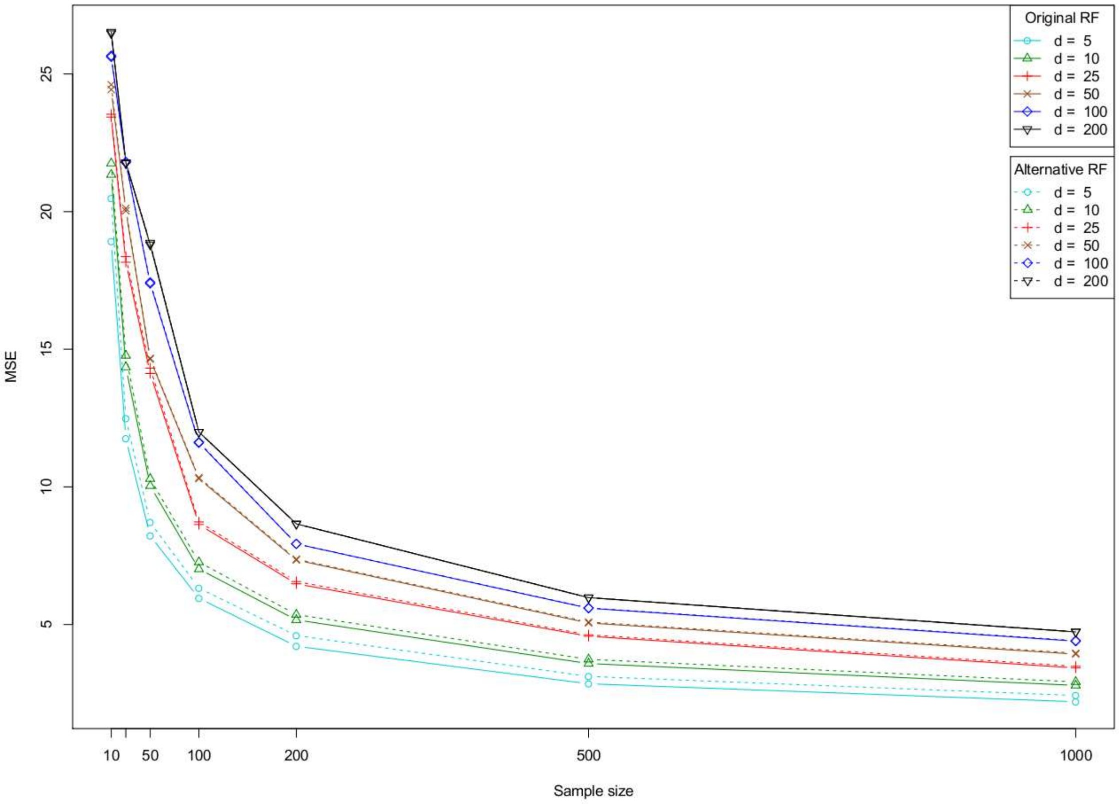


Figure 7: Evolution of the MSE for model **Friedman #1** ($S = 5$).

$\lceil \log_2 k_n \rceil$ *and* p_{nj} *(by independence of* $\mathbf{X}$ *and* Θ*). Moreover, by construction,*

$$\sum_{j=1}^{d} K_{nj}(\mathbf{X}, \Theta) = \lceil \log_2 k_n \rceil.$$

Recall that we denote by $N_n(\mathbf{X}, \Theta)$ the number of data points falling in the same cell as $\mathbf{X}$, that is,

$$N_n(\mathbf{X}, \Theta) = \sum_{i=1}^{n} \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}.$$

Let λ be the Lebesgue measure on $[0,1]^d$.

Fact 2 *By construction,*

$$\lambda(A_n(\mathbf{X}, \Theta)) = 2^{-\lceil \log_2 k_n \rceil}.$$

In particular, if $\mathbf{X}$ *is uniformly distributed on* $[0,1]^d$*, then the distribution of* $N_n(\mathbf{X}, \Theta)$ *conditionally on* $\mathbf{X}$ *and* Θ *is binomial with parameters n and* $2^{-\lceil \log_2 k_n \rceil}$ *(by independence of the random variables* $\mathbf{X}, \mathbf{X}_1, \ldots, \mathbf{X}_n, \Theta$*).*

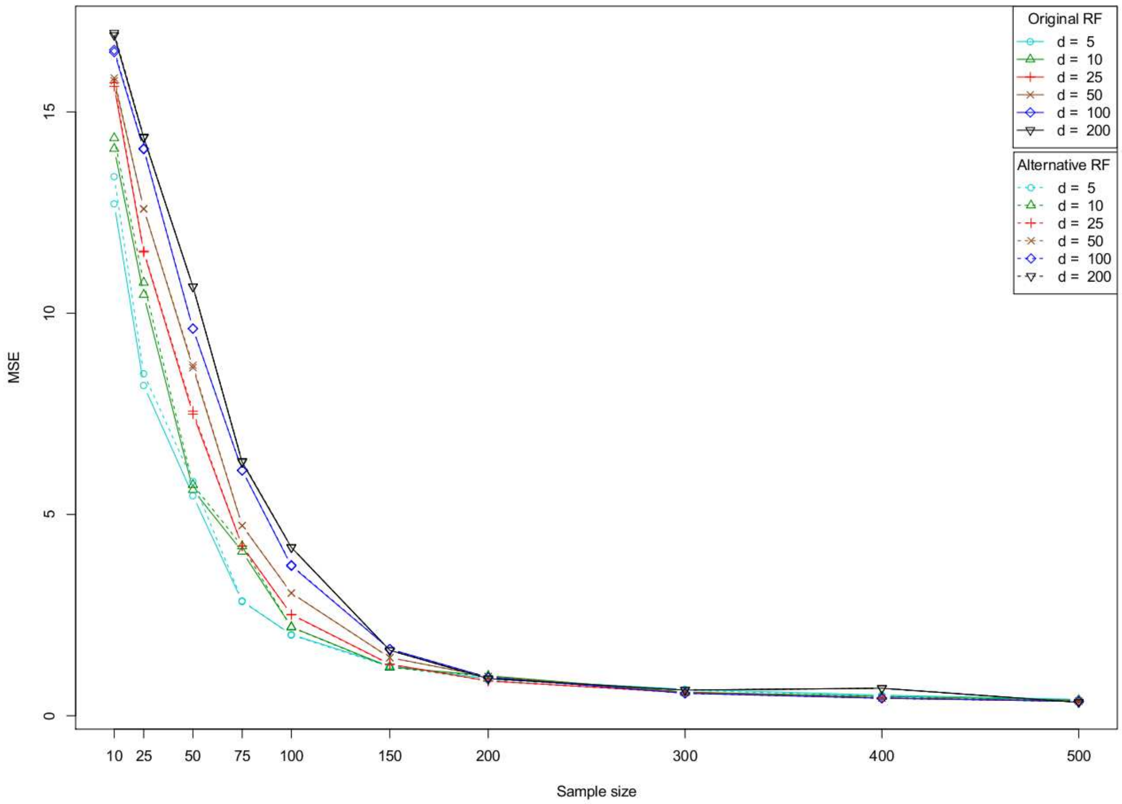


Figure 8: Evolution of the MSE for model **Tree** ($S = 5$).

Remark 10 *If $\mathbf{X}$ is not uniformly distributed but has a probability density f on $[0,1]^d$, then, conditionally on $\mathbf{X}$ and Θ, $N_n(\mathbf{X},\Theta)$ is binomial with parameters n and $\mathbb{P}(\mathbf{X}_1 \in A_n(\mathbf{X},\Theta)\,|\,\mathbf{X},\Theta)$. If f is bounded from above and from below, this probability is of the order $\lambda(A_n(\mathbf{X},\Theta)) = 2^{-\lceil \log_2 k_n \rceil}$, and the whole approach can be carried out without difficulty. On the other hand, for more general densities, the binomial probability depends on $\mathbf{X}$, and this makes the analysis significantly harder.*

5.1 Proof of Theorem 1

Observe first that, by Jensen's inequality,

$$\begin{aligned}\mathbb{E}\,[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})]^2 &= \mathbb{E}\,[\mathbb{E}_\Theta\,[r_n(\mathbf{X},\Theta) - r(\mathbf{X})]]^2\\ &\leq \mathbb{E}\,[r_n(\mathbf{X},\Theta) - r(\mathbf{X})]^2.\end{aligned}$$

A slight adaptation of Theorem 4.2 in Györfi et al. (2002) shows that $\bar{r}_n$ is consistent if both $\operatorname{diam}(A_n(\mathbf{X},\Theta)) \to 0$ in probability and $N_n(\mathbf{X},\Theta) \to \infty$ in probability.

Let us first prove that $N_n(\mathbf{X},\Theta) \to \infty$ in probability. To see this, consider the random tree partition defined by Θ, which has by construction exactly $2^{\lceil \log_2 k_n \rceil}$ rectangular cells, say $A_1, \ldots, A_{2^{\lceil \log_2 k_n \rceil}}$.

Let $N_1,\ldots,N_{2^{\lceil \log_2 k_n \rceil}}$ denote the number of observations among $\mathbf{X},\mathbf{X}_1,\ldots,\mathbf{X}_n$ falling in these $2^{\lceil \log_2 k_n \rceil}$ cells, and let $\mathcal{C}=\{\mathbf{X},\mathbf{X}_1,\ldots,\mathbf{X}_n\}$ denote the set of positions of these $n+1$ points. Since these points are independent and identically distributed, fixing the set $\mathcal{C}$ and Θ, the conditional probability that $\mathbf{X}$ falls in the ℓ-th cell equals $N_\ell/(n+1)$. Thus, for every fixed $M \geq 0$,

$$\begin{aligned}
\mathbb{P}\left(N_n(\mathbf{X},\Theta)<M\right) &= \mathbb{E}\left[\mathbb{P}\left(N_n(\mathbf{X},\Theta)<M \,|\, \mathcal{C},\Theta\right)\right] \\
&= \mathbb{E}\left[\sum_{\ell=1,\ldots,2^{\lceil \log_2 k_n \rceil}:N_\ell<M} \frac{N_\ell}{n+1}\right] \\
&\leq \frac{M2^{\lceil \log_2 k_n \rceil}}{n+1} \\
&\leq \frac{2Mk_n}{n+1},
\end{aligned}$$

which converges to 0 by our assumption on k_n.

It remains to show that $\operatorname{diam}(A_n(\mathbf{X},\Theta)) \to 0$ in probability. To this aim, let $V_{nj}(\mathbf{X},\Theta)$ be the size of the j-th dimension of the rectangle containing $\mathbf{X}$. Clearly, it suffices to show that $V_{nj}(\mathbf{X},\Theta) \to 0$ in probability for all $j=1,\ldots,d$. To this end, note that

$$V_{nj}(\mathbf{X},\Theta) \overset{\mathcal{D}}{=} 2^{-K_{nj}(\mathbf{X},\Theta)},$$

where, conditionally on $\mathbf{X}$, $K_{nj}(\mathbf{X},\Theta)$ has a binomial $\mathcal{B}(\lceil \log_2 k_n \rceil, p_{nj})$ distribution, representing the number of times the box containing $\mathbf{X}$ is split along the j-th coordinate (Fact 1). Thus

$$\begin{aligned}
\mathbb{E}\left[V_{nj}(\mathbf{X},\Theta)\right] &= \mathbb{E}\left[2^{-K_{nj}(\mathbf{X},\Theta)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[2^{-K_{nj}(\mathbf{X},\Theta)} \,|\, \mathbf{X}\right]\right] \\
&= (1-p_{nj}/2)^{\lceil \log_2 k_n \rceil},
\end{aligned}$$

which tends to 0 as $p_{nj}\log k_n \to \infty$.

5.2 Proof of Proposition 2

Recall that

$$\bar{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X},\Theta)\right] Y_i,$$

where

$$W_{ni}(\mathbf{X},\Theta) = \frac{\mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}}{N_n(\mathbf{X},\Theta)} \mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)}$$

and

$$\mathcal{E}_n = \left[N_n(\mathbf{X},\Theta) \neq 0\right].$$

Similarly,

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X},\Theta)\right] r(\mathbf{X}_i).$$

We have

$$\begin{aligned}
\mathbb{E}\left[\bar{r}_n(\mathbf{X})-\tilde{r}_n(\mathbf{X})\right]^2 &= \mathbb{E}\left[\sum_{i=1}^{n}\mathbb{E}_{\Theta}\left[W_{ni}(\mathbf{X},\Theta)\right](Y_i-r(\mathbf{X}_i))\right]^2\\
&= \mathbb{E}\left[\sum_{i=1}^{n}\mathbb{E}^2_{\Theta}\left[W_{ni}(\mathbf{X},\Theta)\right](Y_i-r(\mathbf{X}_i))^2\right]\\
&\quad \text{(the cross terms are 0 since } \mathbb{E}[Y_i|\mathbf{X}_i]=r(\mathbf{X}_i))\\
&= \mathbb{E}\left[\sum_{i=1}^{n}\mathbb{E}^2_{\Theta}\left[W_{ni}(\mathbf{X},\Theta)\right]\sigma^2(\mathbf{X}_i)\right]\\
&\le \sigma^2\mathbb{E}\left[\sum_{i=1}^{n}\mathbb{E}^2_{\Theta}\left[W_{ni}(\mathbf{X},\Theta)\right]\right]\\
&= n\sigma^2\mathbb{E}\left[\mathbb{E}^2_{\Theta}\left[W_{n1}(\mathbf{X},\Theta)\right]\right],
\end{aligned}$$

where we used a symmetry argument in the last equality. Observe now that

$$\begin{aligned}
\mathbb{E}^2_{\Theta}\left[W_{n1}(\mathbf{X},\Theta)\right] &= \mathbb{E}_{\Theta}\left[W_{n1}(\mathbf{X},\Theta)\right]\mathbb{E}_{\Theta'}\left[W_{n1}(\mathbf{X},\Theta')\right]\\
&\quad \text{(where } \Theta' \text{ is distributed as, and independent of, } \Theta)\\
&= \mathbb{E}_{\Theta,\Theta'}\left[W_{n1}(\mathbf{X},\Theta)W_{n1}(\mathbf{X},\Theta')\right]\\
&= \mathbb{E}_{\Theta,\Theta'}\left[\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta')]}}{N_n(\mathbf{X},\Theta)N_n(\mathbf{X},\Theta')}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta')}\right]\\
&= \mathbb{E}_{\Theta,\Theta'}\left[\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}}{N_n(\mathbf{X},\Theta)N_n(\mathbf{X},\Theta')}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta')}\right].
\end{aligned}$$

Consequently,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X})-\tilde{r}_n(\mathbf{X})\right]^2 \le n\sigma^2\mathbb{E}\left[\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}}{N_n(\mathbf{X},\Theta)N_n(\mathbf{X},\Theta')}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta')}\right].$$

Therefore

$$\begin{aligned}
&\mathbb{E}\left[\bar{r}_n(\mathbf{X})-\tilde{r}_n(\mathbf{X})\right]^2\\
&\quad\le n\sigma^2\mathbb{E}\left[\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}\right)\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta')]}\right)}\right]\\
&\quad= n\sigma^2\mathbb{E}\left[\mathbb{E}\left[\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}\right)}\right.\right.\\
&\qquad\qquad\left.\left.\times\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta')]}\right)}\,|\,\mathbf{X},\mathbf{X}_1,\Theta,\Theta'\right]\right]\\
&\quad= n\sigma^2\mathbb{E}\left[\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}\mathbb{E}\left[\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}\right)}\right.\right.\\
&\qquad\qquad\left.\left.\times\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta')]}\right)}\,|\,\mathbf{X},\mathbf{X}_1,\Theta,\Theta'\right]\right]\\
&\quad= n\sigma^2\mathbb{E}\left[\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}\mathbb{E}\left[\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}\right)}\right.\right.\\
&\qquad\qquad\left.\left.\times\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta')]}\right)}\,|\,\mathbf{X},\Theta,\Theta'\right]\right]
\end{aligned}$$

by the independence of the random variables $\mathbf{X},\mathbf{X}_1,\ldots,\mathbf{X}_n,\Theta,\Theta'$. Using the Cauchy-Schwarz inequality, the above conditional expectation can be upper bounded by

$$\begin{aligned}
&\mathbb{E}^{1/2}\left[\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}\right)^2}\,|\,\mathbf{X},\Theta\right]\times\mathbb{E}^{1/2}\left[\frac{1}{\left(1+\sum_{i=2}^n\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta')]}\right)^2}\,|\,\mathbf{X},\Theta'\right]\\
&\le\frac{3\times 2^{2\lceil\log_2 k_n\rceil}}{n^2}\\
&\quad\text{(by Fact 2 and technical Lemma 11)}\\
&\le\frac{12k_n^2}{n^2}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\mathbb{E}\left[\bar{r}_n(\mathbf{X})-\tilde{r}_n(\mathbf{X})\right]^2&\le\frac{12\sigma^2k_n^2}{n}\mathbb{E}\left[\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}\right]\\
&=\frac{12\sigma^2k_n^2}{n}\mathbb{E}\left[\mathbb{E}_{\mathbf{X}_1}\left[\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')]}\right]\right]\\
&=\frac{12\sigma^2k_n^2}{n}\mathbb{E}\left[\mathbb{P}_{\mathbf{X}_1}\left(\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')\right)\right].\qquad(3)
\end{aligned}$$

Next, using the fact that $\mathbf{X}_1$ is uniformly distributed over $[0,1]^d$, we may write

$$\begin{aligned}
\mathbb{P}_{\mathbf{X}_1}\left(\mathbf{X}_1\in A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')\right)&=\lambda\left(A_n(\mathbf{X},\Theta)\cap A_n(\mathbf{X},\Theta')\right)\\
&=\prod_{j=1}^d\lambda\left(A_{nj}(\mathbf{X},\Theta)\cap A_{nj}(\mathbf{X},\Theta')\right),
\end{aligned}$$

where

$$A_n(\mathbf{X},\Theta) = \prod_{j=1}^{d} A_{nj}(\mathbf{X},\Theta) \quad \text{and} \quad A_n(\mathbf{X},\Theta') = \prod_{j=1}^{d} A_{nj}(\mathbf{X},\Theta').$$

On the other hand, we know (Fact 1) that, for all $j = 1,\dots,d$,

$$\lambda\left(A_{nj}(\mathbf{X},\Theta)\right) \overset{\mathcal{D}}{=} 2^{-K_{nj}(\mathbf{X},\Theta)},$$

where, conditionally on $\mathbf{X}$, $K_{nj}(\mathbf{X},\Theta)$ has a binomial $\mathcal{B}(\lceil \log_2 k_n \rceil, p_{nj})$ distribution and, similarly,

$$\lambda\left(A_{nj}(\mathbf{X},\Theta')\right) \overset{\mathcal{D}}{=} 2^{-K'_{nj}(\mathbf{X},\Theta')},$$

where, conditionally on $\mathbf{X}$, $K'_{nj}(\mathbf{X},\Theta')$ is binomial $\mathcal{B}(\lceil \log_2 k_n \rceil, p_{nj})$ and independent of $K_{nj}(\mathbf{X},\Theta)$. In the rest of the proof, to lighten notation, we write K_{nj} and K'_{nj} instead of $K_{nj}(\mathbf{X},\Theta)$ and $K'_{nj}(\mathbf{X},\Theta')$, respectively. Clearly,

$$\begin{aligned}\lambda\left(A_{nj}(\mathbf{X},\Theta) \cap A_{nj}(\mathbf{X},\Theta')\right) &\le 2^{-\max(K_{nj},K'_{nj})} \\ &= 2^{-K'_{nj}} 2^{-(K_{nj}-K'_{nj})_+}\end{aligned}$$

and, consequently,

$$\prod_{j=1}^{d} \lambda\left(A_{nj}(\mathbf{X},\Theta) \cap A_{nj}(\mathbf{X},\Theta')\right) \le 2^{-\lceil \log_2 k_n \rceil} \prod_{j=1}^{d} 2^{-(K_{nj}-K'_{nj})_+}$$

(since, by Fact 1, $\sum_{j=1}^{d} K_{nj} = \lceil \log_2 k_n \rceil$). Plugging this inequality into (3) and applying Hölder's inequality, we obtain

$$\begin{aligned}\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 &\le \frac{12\sigma^2 k_n}{n} \mathbb{E}\left[\prod_{j=1}^{d} 2^{-(K_{nj}-K'_{nj})_+}\right] \\ &= \frac{12\sigma^2 k_n}{n} \mathbb{E}\left[\mathbb{E}\left[\prod_{j=1}^{d} 2^{-(K_{nj}-K'_{nj})_+} \,|\, \mathbf{X}\right]\right] \\ &\le \frac{12\sigma^2 k_n}{n} \mathbb{E}\left[\prod_{j=1}^{d} \mathbb{E}^{1/d}\left[2^{-d(K_{nj}-K'_{nj})_+} \,|\, \mathbf{X}\right]\right].\end{aligned}$$

Each term in the product may be bounded by technical Proposition 13, and this leads to

$$\begin{aligned}\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 &\le \frac{288\sigma^2 k_n}{\pi n} \prod_{j=1}^{d} \min\left(1, \left[\frac{\pi}{16\lceil \log_2 k_n \rceil p_{nj}(1-p_{nj})}\right]^{1/2d}\right) \\ &\le \frac{288\sigma^2 k_n}{\pi n} \prod_{j=1}^{d} \min\left(1, \left[\frac{\pi \log 2}{16(\log k_n) p_{nj}(1-p_{nj})}\right]^{1/2d}\right).\end{aligned}$$

Using the assumption on the form of the p_{nj}, we finally conclude that

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 \le C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2d} (1+\xi_n) \frac{k_n}{n(\log k_n)^{S/2d}},$$

where

$$C = \frac{288}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2d}$$

and

$$1+\xi_n = \prod_{j\in\mathcal{S}}\left[(1+\xi_{nj})^{-1}\left(1-\frac{\xi_{nj}}{S-1}\right)^{-1}\right]^{1/2d}.$$

Clearly, the sequence (ξ_n), which depends on the $\{(\xi_{nj}) : j \in \mathcal{S}\}$ only, tends to 0 as n tends to infinity.

5.3 Proof of Proposition 4

We start with the decomposition

$$\begin{aligned}
&\mathbb{E}\left[\tilde{r}_n(\mathbf{X})-r(\mathbf{X})\right]^2\\
&\quad= \mathbb{E}\left[\sum_{i=1}^{n}\mathbb{E}_\Theta\left[W_{ni}(\mathbf{X},\Theta)\right](r(\mathbf{X}_i)-r(\mathbf{X}))+\left(\sum_{i=1}^{n}\mathbb{E}_\Theta\left[W_{ni}(\mathbf{X},\Theta)\right]-1\right)r(\mathbf{X})\right]^2\\
&\quad=\mathbb{E}\left[\mathbb{E}_\Theta\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)(r(\mathbf{X}_i)-r(\mathbf{X}))+\left(\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)-1\right)r(\mathbf{X})\right]\right]^2\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)(r(\mathbf{X}_i)-r(\mathbf{X}))+\left(\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)-1\right)r(\mathbf{X})\right]^2,
\end{aligned}$$

where, in the last step, we used Jensen's inequality. Consequently,

$$\begin{aligned}
&\mathbb{E}\left[\tilde{r}_n(\mathbf{X})-r(\mathbf{X})\right]^2\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)(r(\mathbf{X}_i)-r(\mathbf{X}))\right]^2+\mathbb{E}\left[r(\mathbf{X})\mathbf{1}_{\mathcal{E}_n^c(\mathbf{X},\Theta)}\right]^2\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)(r(\mathbf{X}_i)-r(\mathbf{X}))\right]^2+\left[\sup_{\mathbf{x}\in[0,1]^d}r^2(\mathbf{x})\right]\mathbb{P}\left(\mathcal{E}_n^c(\mathbf{X},\Theta)\right).
\end{aligned}\tag{4}$$

Let us examine the first term on the right-hand side of (4). Observe that, by the Cauchy-Schwarz inequality,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n} \sqrt{W_{ni}(\mathbf{X},\Theta)}\sqrt{W_{ni}(\mathbf{X},\Theta)}\,|r(\mathbf{X}_i)-r(\mathbf{X})|\right]^2\\
&\quad\le \mathbb{E}\left[\left(\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\right)\left(\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)^2\right)\right]\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)^2\right]\\
&\qquad\text{(since the weights are subprobability weights).}
\end{aligned}$$

Thus, denoting by $\|\mathbf{X}\|_{\mathcal{S}}$ the norm of $\mathbf{X}$ evaluated over the components in $\mathcal{S}$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\\
&\quad\le \mathbb{E}\left[\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r^{\star}(\mathbf{X}_{i\mathcal{S}})-r^{\star}(\mathbf{X}_{\mathcal{S}})\right)^2\right]\\
&\quad\le L^2\sum_{i=1}^{n}\mathbb{E}\left[W_{ni}(\mathbf{X},\Theta)\|\mathbf{X}_i-\mathbf{X}\|_{\mathcal{S}}^2\right]\\
&\quad= nL^2\mathbb{E}\left[W_{n1}(\mathbf{X},\Theta)\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\right]\\
&\qquad\text{(by symmetry).}
\end{aligned}$$

But

$$\begin{aligned}
&\mathbb{E}\left[W_{n1}(\mathbf{X},\Theta)\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\right]\\
&\quad=\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}}{N_n(\mathbf{X},\Theta)}\mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)}\right]\\
&\quad=\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}}{1+\sum_{i=2}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}}\right]\\
&\quad=\mathbb{E}\left[\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\frac{\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}}{1+\sum_{i=2}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}}\,|\,\mathbf{X},\mathbf{X}_1,\Theta\right]\right].
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbb{E}\left[W_{n1}(\mathbf{X},\Theta)\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\right]\\
&\quad=\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}\mathbb{E}\left[\frac{1}{1+\sum_{i=2}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}}\,|\,\mathbf{X},\mathbf{X}_1,\Theta\right]\right]\\
&\quad=\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}\mathbb{E}\left[\frac{1}{1+\sum_{i=2}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}}\,|\,\mathbf{X},\Theta\right]\right]\\
&\qquad\text{(by the independence of the random variables }\mathbf{X},\mathbf{X}_1,\ldots,\mathbf{X}_n,\Theta\text{).}
\end{aligned}$$

By Fact 2 and technical Lemma 11,

$$\mathbb{E}\left[\frac{1}{1+\sum_{i=2}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}}\,|\,\mathbf{X},\Theta\right]\leq\frac{2^{\lceil\log_2 k_n\rceil}}{n}\leq\frac{2k_n}{n}.$$

Consequently,

$$\mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\leq 2L^2k_n\mathbb{E}\left[\|\mathbf{X}_1-\mathbf{X}\|_{\mathcal{S}}^2\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}\right].$$

Letting

$$A_n(\mathbf{X},\Theta)=\prod_{j=1}^{d}A_{nj}(\mathbf{X},\Theta),$$

we obtain

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\\&\quad\leq 2L^2k_n\sum_{j\in\mathcal{S}}\mathbb{E}\left[|\mathbf{X}_1^{(j)}-\mathbf{X}^{(j)}|^2\mathbf{1}_{[\mathbf{X}_1\in A_n(\mathbf{X},\Theta)]}\right]\\&\quad=2L^2k_n\sum_{j\in\mathcal{S}}\mathbb{E}\left[\rho_j(\mathbf{X},\mathbf{X}_1,\Theta)\mathbb{E}_{\mathbf{X}_1^{(j)}}\left[|\mathbf{X}_1^{(j)}-\mathbf{X}^{(j)}|^2\mathbf{1}_{[\mathbf{X}_1^{(j)}\in A_{nj}(\mathbf{X},\Theta)]}\right]\right]\end{aligned}$$

where, in the last equality, we set

$$\rho_j(\mathbf{X},\mathbf{X}_1,\Theta)=\prod_{t=1,\ldots,d,t\neq j}\mathbf{1}_{[\mathbf{X}_1^{(t)}\in A_{nt}(\mathbf{X},\Theta)]}.$$

Therefore, using the fact that $\mathbf{X}_1$ is uniformly distributed over $[0,1]^d$,

$$\mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\leq 2L^2k_n\sum_{j\in\mathcal{S}}\mathbb{E}\left[\rho_j(\mathbf{X},\mathbf{X}_1,\Theta)\lambda^3\left(A_{nj}(\mathbf{X},\Theta)\right)\right].$$

Observing that

$$\begin{aligned}&\lambda\left(A_{nj}(\mathbf{X},\Theta)\right)\times\mathbb{E}_{[\mathbf{X}_1^{(t)}:t=1,\ldots,d,t\neq j]}\left[\rho_j(\mathbf{X},\mathbf{X}_1,\Theta)\right]\\&\quad=\lambda\left(A_n(\mathbf{X},\Theta)\right)\\&\quad=2^{-\lceil\log_2 k_n\rceil}\\&\qquad\text{(Fact 2)},\end{aligned}$$

we are led to

$$\begin{aligned}&\mathbb{E}\left[\sum_{i=1}^{n}W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\\&\quad\leq 2L^2\sum_{j\in\mathcal{S}}\mathbb{E}\left[\lambda^2\left(A_{nj}(\mathbf{X},\Theta)\right)\right]\\&\quad=2L^2\sum_{j\in\mathcal{S}}\mathbb{E}\left[2^{-2K_{nj}(\mathbf{X},\Theta)}\right]\\&\quad=2L^2\sum_{j\in\mathcal{S}}\mathbb{E}\left[\mathbb{E}\left[2^{-2K_{nj}(\mathbf{X},\Theta)}\,|\,\mathbf{X}\right]\right],\end{aligned}$$

where, conditionally on $\mathbf{X}$, $K_{nj}(\mathbf{X},\Theta)$ has a binomial $\mathcal{B}(\lceil \log_2 k_n\rceil, p_{nj})$ distribution (Fact 1). Consequently,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n} W_{ni}(\mathbf{X},\Theta)\left(r(\mathbf{X}_i)-r(\mathbf{X})\right)\right]^2\\
&\quad\le 2L^2\sum_{j\in\mathcal{S}}\left(1-0.75p_{nj}\right)^{\lceil \log_2 k_n\rceil}\\
&\quad\le 2L^2\sum_{j\in\mathcal{S}}\exp\left(-\frac{0.75}{\log 2}p_{nj}\log k_n\right)\\
&\quad= 2L^2\sum_{j\in\mathcal{S}}\frac{1}{k_n^{\frac{0.75}{S\log 2}(1+\xi_{nj})}}\\
&\quad\le \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}},
\end{aligned}$$

with $\gamma_n=\min_{j\in\mathcal{S}}\xi_{nj}$.

To finish the proof, it remains to bound the second term on the right-hand side of (4), which is easier. Just note that

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_n^c(\mathbf{X},\Theta)\right) &= \mathbb{P}\left(\sum_{i=1}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}=0\right)\\
&= \mathbb{E}\left[\mathbb{P}\left(\sum_{i=1}^{n}\mathbf{1}_{[\mathbf{X}_i\in A_n(\mathbf{X},\Theta)]}=0\,|\,\mathbf{X},\Theta\right)\right]\\
&= \left(1-2^{-\lceil \log_2 k_n\rceil}\right)^n\\
&\quad\text{(by Fact 2)}\\
&\le e^{-n/2k_n}.
\end{aligned}$$

Putting all the pieces together, we finally conclude that

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X})-r(\mathbf{X})\right]^2 \le \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}}+\left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right]e^{-n/2k_n},$$

as desired.

5.4 Some Technical Results

The following result is an extension of Lemma 4.1 in Györfi et al. (2002). Its proof is given here for the sake of completeness.

Lemma 11 *Let Z be a binomial $\mathcal{B}(N,p)$ random variable, with $p\in(0,1]$. Then*

(i)

$$\mathbb{E}\left[\frac{1}{1+Z}\right]\le\frac{1}{(N+1)p}.$$

(ii)

$$\mathbb{E}\left[\frac{1}{Z}\mathbf{1}_{[Z\geq 1]}\right] \leq \frac{2}{(N+1)p}.$$

(iii)

$$\mathbb{E}\left[\frac{1}{1+Z^2}\right] \leq \frac{3}{(N+1)(N+2)p^2}.$$

Proof To prove statement (i), we write

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{1+Z}\right] &= \sum_{j=0}^{N} \frac{1}{1+j}\binom{N}{j} p^j (1-p)^{N-j} \\
&= \frac{1}{(N+1)p}\sum_{j=0}^{N}\binom{N+1}{j+1} p^{j+1}(1-p)^{N-j} \\
&\leq \frac{1}{(N+1)p}\sum_{j=0}^{N+1}\binom{N+1}{j} p^{j}(1-p)^{N+1-j} \\
&= \frac{1}{(N+1)p}.
\end{aligned}$$

The second statement follows from the inequality

$$\mathbb{E}\left[\frac{1}{Z}\mathbf{1}_{[Z\geq 1]}\right] \leq \mathbb{E}\left[\frac{2}{1+Z}\right]$$

and the third one by observing that

$$\mathbb{E}\left[\frac{1}{1+Z^2}\right] = \sum_{j=0}^{N}\frac{1}{1+j^2}\binom{N}{j} p^j (1-p)^{N-j}.$$

Therefore

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{1+Z^2}\right] &= \frac{1}{(N+1)p}\sum_{j=0}^{N}\frac{1+j}{1+j^2}\binom{N+1}{j+1} p^{j+1}(1-p)^{N-j} \\
&\leq \frac{3}{(N+1)p}\sum_{j=0}^{N}\frac{1}{2+j}\binom{N+1}{j+1} p^{j+1}(1-p)^{N-j} \\
&\leq \frac{3}{(N+1)p}\sum_{j=0}^{N+1}\frac{1}{1+j}\binom{N+1}{j} p^{j}(1-p)^{N+1-j} \\
&\leq \frac{3}{(N+1)(N+2)p^2} \\
&\quad \text{(by } (i)\text{)}.
\end{aligned}$$

■

Lemma 12 *Let Z_1 and Z_2 be two independent binomial $\mathcal{B}(N,p)$ random variables. Set, for all $z \in \mathbb{C}^\star$, $\varphi(z) = \mathbb{E}[z^{Z_1 - Z_2}]$. Then*

(i) *For all $z \in \mathbb{C}^\star$,*

$$\varphi(z) = \left[p(1-p)(z+z^{-1}) + 1 - 2p(1-p)\right]^N.$$

(ii) *For all $j \in \mathbb{N}$,*

$$\mathbb{P}(Z_1 - Z_2 = j) = \frac{1}{2\pi i}\int_\Gamma \frac{\varphi(z)}{z^{j+1}}\mathrm{d}z,$$

where Γ is the positively oriented unit circle.

(iii) *For all $d \geq 1$,*

$$\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right] \leq \frac{24}{\pi}\int_0^1 \exp\left(-4Np(1-p)t^2\right)\mathrm{d}t.$$

Proof Statement (i) is clear and (ii) is an immediate consequence of Cauchy's integral formula (Rudin, 1987). To prove statement (iii), write

$$\begin{aligned}
\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right] &= \sum_{j=0}^{N} 2^{-dj}\,\mathbb{P}\left((Z_1-Z_2)_+ = j\right) \\
&= \sum_{j=0}^{N} 2^{-dj}\,\mathbb{P}\left(Z_1-Z_2 = j\right) \\
&\leq \sum_{j=0}^{\infty} 2^{-dj}\,\mathbb{P}\left(Z_1-Z_2 = j\right) \\
&= \frac{1}{2\pi i}\int_\Gamma \frac{\varphi(z)}{z}\sum_{j=0}^{\infty}\left(\frac{2^{-d}}{z}\right)^j \mathrm{d}z \\
&\quad \text{(by statement } (ii)\text{)} \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\varphi(e^{i\theta})}{1-2^{-d}e^{-i\theta}}\mathrm{d}\theta \\
&\quad \text{(by setting } z = e^{i\theta}, \theta \in [-\pi,\pi]\text{)} \\
&= \frac{2^{d-1}}{\pi}\int_{-\pi}^{\pi} [1+2p(1-p)(\cos\theta - 1)]^N \frac{e^{i\theta}}{2^d e^{i\theta} - 1}\mathrm{d}\theta \\
&\quad \text{(by statement } (i)\text{)}.
\end{aligned}$$

Noting that

$$\frac{e^{i\theta}}{2^d e^{i\theta}-1} = \frac{2^d - e^{i\theta}}{2^{2d} - 2^{d+1}\cos\theta + 1},$$

we obtain

$$\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right] \leq \frac{2^{d-1}}{\pi}\int_{-\pi}^{\pi} [1+2p(1-p)(\cos\theta-1)]^N \frac{2^d - \cos\theta}{2^{2d}-2^{d+1}\cos\theta+1}\mathrm{d}\theta.$$

The bound

$$\frac{2^d-\cos\theta}{2^{2d}-2^{d+1}\cos\theta+1}\leq\frac{2^d+1}{(2^d-1)^2}$$

leads to

$$\begin{aligned}
&\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right]\\
&\leq\frac{2^{d-1}(2^d+1)}{\pi(2^d-1)^2}\int_{-\pi}^{\pi}[1+2p(1-p)(\cos\theta-1)]^N\,\mathrm{d}\theta\\
&=\frac{2^{d}(2^d+1)}{\pi(2^d-1)^2}\int_{0}^{\pi}[1+2p(1-p)(\cos\theta-1)]^N\,\mathrm{d}\theta\\
&=\frac{2^{d}(2^d+1)}{\pi(2^d-1)^2}\int_{0}^{\pi}\left[1-4p(1-p)\sin^2(\theta/2)\right]^N\,\mathrm{d}\theta\\
&(\cos\theta-1=-2\sin^2(\theta/2))\\
&=\frac{2^{d+1}(2^d+1)}{\pi(2^d-1)^2}\int_{0}^{\pi/2}\left[1-4p(1-p)\sin^2\theta\right]^N\,\mathrm{d}\theta.
\end{aligned}$$

Using the elementary inequality $(1-z)^N\leq e^{-Nz}$ for $z\in[0,1]$ and the change of variable

$$t=\tan(\theta/2),$$

we finally obtain

$$\begin{aligned}
\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right]&\leq\frac{2^{d+2}(2^d+1)}{\pi(2^d-1)^2}\int_0^1\exp\left(-\frac{16Np(1-p)t^2}{(1+t^2)^2}\right)\frac{1}{1+t^2}\mathrm{d}t\\
&\leq C_d\int_0^1\exp\left(-4Np(1-p)t^2\right)\mathrm{d}t,
\end{aligned}$$

with

$$C_d=\frac{2^{d+2}(2^d+1)}{\pi(2^d-1)^2}.$$

The conclusion follows by observing that $C_d\leq 24/\pi$ for all $d\geq 1$. ■

Evaluating the integral in statement (*iii*) of Lemma 12 leads to the following proposition:

Proposition 13 *Let Z_1 and Z_2 be two independent binomial $\mathcal{B}(N,p)$ random variables, with $p\in(0,1)$. Then, for all $d\geq 1$,*

$$\mathbb{E}\left[2^{-d(Z_1-Z_2)_+}\right]\leq\frac{24}{\pi}\min\left(1,\sqrt{\frac{\pi}{16Np(1-p)}}\right).$$

Acknowledgments

I would like to acknowledge support for this project from the French National Research Agency under grant ANR-09-BLAN-0051-02 “CLARA” and from the INRIA project “CLASSIC” hosted by Ecole Normale Supérieure and CNRS. I greatly thank the Action Editor and two referees for valuable comments and insightful suggestions, which lead to a substantial improvement of the paper. I would also like to thank my colleague Jean-Patrick Baudry for his precious help on the simulation section.

References

D. Amaratunga, J. Cabrera, and Y.S. Lee. Enriched random forests. *Bioinformatics*, 24:2010–2014, 2008.

Y. Amit. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. The MIT Press, Cambridge, 2002.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

G. Biau and L. Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101:2499–2518, 2010.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

G. Biau, F. Cérou, and A. Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research*, 11:687–712, 2010.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37:1705–1732, 2009.

G. Blanchard. Different paradigms for choosing sequential reweighting algorithms. *Neural Computation*, 16:811–836, 2004.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. *Some Infinity Theory for Predictor Ensembles*. Technical Report 577, UC Berkeley, 2000. URL `http://www.stat.berkeley.edu/~breiman`.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman. *Consistency For a Simple Model of Random Forests*. Technical Report 670, UC Berkeley, 2004. URL `http://www.stat.berkeley.edu/~breiman`.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51:34–81, 2009.

P. Bühlmann and B. Yu. Analyzing bagging. *The Annals of Statistics*, 30:927–961, 2002.

A. Buja and W. Stuetzle. Observations on bagging. *Statistica Sinica*, 16:323–352, 2006.

F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the Lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

E.J. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35:2313–2351, 2005.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

R. Diaz-Uriarte and S.A. de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:1471–2105, 2006.

T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000.

Y. Freund and R. Shapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Machine Learning: Proceedings of the 13th International Conference*, pages 148–156, San Francisco, 1996. Morgan Kaufmann.

J.H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19:1–67, 1991.

R. Genuer, J.-M. Poggi, and C. Tuleau. *Random Forests: Some Methodological Insights*. arXiv:0811.3619, 2008.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31:2225–2236, 2010.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, New York, 2002.

T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998.

I.A. Ibragimov and R.Z. Khasminskii. On nonparametric estimation of regression. *Doklady Akademii Nauk SSSR*, 252:780–784, 1980.

I.A. Ibragimov and R.Z. Khasminskii. *Statistical Estimation: Asymptotic Theory*. Springer-Verlag, New York, 1981.

I.A. Ibragimov and R.Z. Khasminskii. On the bounds for quality of nonparametric regression function estimation. *Theory of Probability and its Applications*, 27:81–94, 1982.

A.N. Kolmogorov and V.M. Tihomirov. ε-entropy and ε-capacity of sets in functional spaces. *American Mathematical Society Translations*, 17:277–364, 1961.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590, 2006.

N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

W. Rudin. *Real and Complex Analysis, 3rd Edition*. McGraw-Hill, New York, 1987.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan, and B. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43:1947–1958, 2003.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

L.J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

Journal of Machine Learning Research 13 (2012) 1097-1157 Submitted 3/11; Revised 11/11; Published 4/12

Towards Integrative Causal Analysis of Heterogeneous Data Sets and Studies

Ioannis Tsamardinos[*] TSAMARD@ICS.FORTH.GR
Sofia Triantafillou[*] STRIANT@ICS.FORTH.GR
Vincenzo Lagani VLAGANI@ICS.FORTH.GR
Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
N. Plastira 100 Vassilika Vouton
GR-700 13 Heraklion, Crete, Greece

Editor: Chris Meek

Abstract

We present methods able to predict the presence and strength of conditional and unconditional dependencies (correlations) between two variables Y and Z *never jointly measured* on the same samples, based on multiple data sets measuring a set of common variables. The algorithms are specializations of prior work on learning causal structures from overlapping variable sets. This problem has also been addressed in the field of *statistical matching*. The proposed methods are applied to a wide range of domains and are shown to accurately predict the presence of thousands of dependencies. Compared against prototypical statistical matching algorithms and within the scope of our experiments, the proposed algorithms make predictions that are better correlated with the sample estimates of the unknown parameters on test data ; this is particularly the case when the number of commonly measured variables is low.

The enabling idea behind the methods is to induce one or all *causal* models that are simultaneously consistent with (fit) all available data sets and prior knowledge and reason with them. This allows constraints stemming from causal assumptions (e.g., Causal Markov Condition, Faithfulness) to propagate. Several methods have been developed based on this idea, for which we propose the unifying name Integrative Causal Analysis (INCA). A contrived example is presented demonstrating the theoretical potential to develop more general methods for co-analyzing heterogeneous data sets. The computational experiments with the novel methods provide evidence that causally-inspired assumptions such as Faithfulness often hold to a good degree of approximation in many real systems and could be exploited for statistical inference. Code, scripts, and data are available at www.mensxmachina.org.

Keywords: integrative causal analysis, causal discovery, Bayesian networks, maximal ancestral graphs, structural equation models, causality, statistical matching, data fusion

1. Introduction

In several domains it is often the case that several data sets (studies) may be available related to a specific analysis question. Meta-analysis methods attempt to collect, evaluate and combine the results of several studies regarding a single hypothesis. However, studies may be heterogeneous in

[*] Also in Department of Computer Science, University of Crete.

several aspects, and thus not amenable to standard meta-analysis techniques. For example, different studies may be measuring different sets of variables or under different experimental conditions.

One approach to allow the co-analysis of heterogeneous data sets in the context of prior knowledge is to try to induce one or all *causal* models that are simultaneously consistent with all available data sets and pieces of knowledge. Subsequently, one can reason with this set of consistent models. We have named this approach *Integrative Causal Analysis* (INCA).

The use of *causal* models may allow additional inferences than what is possible with non-causal models. This is because the former employ additional assumptions connecting the concept of causality with observable and estimable quantities such as conditional independencies and dependencies. These assumptions further constrain the space of consistent models and may lead to new inferences. Two of the most common causal assumptions in the literature are the Causal Markov Condition and the Faithfulness Condition (Spirtes et al., 2001); intuitively, these conditions assume that the observed dependencies and independencies in the data are due to the causal structure of the observed system and not due to accidental properties of the distribution parameters (Spirtes et al., 2001). Another interpretation of these conditions is that the set of independencies is stable to small perturbations of the joint distribution (Pearl, 2000) of the data.

The idea of inducing causal models from several data sets has already appeared in several prior works. Methods for inducing causal models from samples measured under different experimental conditions are described in Cooper and Yoo (1999), Tian and Pearl (2001), Claassen and Heskes (2010), Eberhardt (2008); Eberhardt et al. (2010) and Hyttinen et al. (2011, 2010). Other methods deal with the co-analysis of data sets defined over different variable sets (Tillman et al., 2008; Triantafillou et al., 2010; Tillman and Spirtes, 2011). In Tillman (2009) and Tsamardinos and Borboudakis (2010) approaches that induce causal models from data sets defined over semantically similar variables (e.g., a dichotomous variable for Smoking in one data set and a continuous variable for Cigarettes-Per-Day in a second) are explored. Methods for inducing causal models in the context of prior knowledge also exist (Angelopoulos and Cussens, 2008; Borboudakis et al., 2011; Meek, 1995; Werhli and Husmeier, 2007; O'Donnell et al., 2006). INCA as a unifying common theme was first presented in Tsamardinos and Triantafillou (2009) where a mathematical formulation is given of the co-analysis of data sets that are heterogeneous in several of the above aspects. In Section 3, we present a contrived example demonstrating the theoretical potential to develop such general methods.

In this paper, we focus on the problem of analyzing data sets defined over different variable sets, as proof-of-concept of the main idea. We develop methods that could be seen as special cases of general algorithms that have appeared for this problem (Tillman et al., 2008; Triantafillou et al., 2010; Tillman and Spirtes, 2011). The methods are able to predict the presence and strength of conditional and unconditional dependencies (correlations) between two variables Y and Z *never jointly measured* on the same samples, based on multiple data sets measuring a set of common variables.

To evaluate the methods we simulate the above situation in a way that it becomes testable: a single data set is partitioned to three data sets that do not share samples. A different set of variables is excluded from each of the first two data sets, while the third is hold out for testing. Based on the first two data sets the algorithms predict certain pairs of the excluded variables should be dependent. These are then tested in the third test set containing all variables.

The proposed algorithms make numerous predictions that range in the thousands for large data sets; the predictions are highly accurate, significantly more accurate than predictions made at random. The methods also successfully predict certain conditional dependencies between pairs of variables Y,Z never measured together in a study. In addition, when linear causal relations and Gaussian error terms are assumed, the algorithms successfully predict the strength of the linear correlation between Y and Z. The latter observation is an example where the INCA approach can give rise to algorithms that provide quantitative inferences (strength of dependence), and are not limited to qualitative inferences (e.g., presence of dependencies).

Inferring the correlation between Y and Z in the above setting has also been addressed by *statistical matching* algorithms (D'Orazio et al., 2006), often found under the name of data fusion in Europe. Statistical matching algorithms make predictions based on parametric distributional assumptions, instead of causally-inspired assumptions. We have implemented two prototypical statistical matching algorithms and performed a comparative evaluation. Within the scope of our experiments, the proposed algorithms make predictions that are better correlated with the sample estimates of the unknown parameters on test data; this is particularly the case when the number of commonly measured variables is low. In addition, the proposed algorithms make predictions in cases where some statistical matching procedures fail to do so and vice versa, and thus, the two approaches can be considered complementary in this respect.

There are several philosophical and practical implications of the above results. First, the results provide ample statistical evidence that some of the typical assumptions employed in causal modeling hold abundantly (at least to a good level of approximation) in a wide range of domains and lead to accurate inferences. *To obtain the results the causal semantics are not employed per se*, that is, we do not predict the effects of experiments and manipulations. In other words, one could view the assumptions made by the causal models as constraints or priors on probability distributions encountered in Nature without any reference to causal semantics.

Second, the results point to the utility and potential impact of the approach: co-analysis provides novel inferences as a norm, not only in contrived toy problems or rare situations. Future INCA-based algorithms that are able to handle all sorts of heterogeneous data sets that vary in terms of experimental conditions, study design and sampling methodology (e.g., case-control vs. i.i.d. sampling, cross-sectional vs. temporal measurements) could potentially one day enable the automated large-scale integrative analysis of a large part of available data and knowledge to construct causal models.

The rest of this document is organized as follows: Section 2 briefly presents background on causal modeling with Maximal Ancestral Graphs. Section 3 discusses the scope and vision of the INCA approach. Section 4 presents the example scenario employed in all evaluations. Section 5 formalizes the problem of co-analysis of data sets measuring different quantities. Sections 6 and 7 present the algorithms and their comparative evaluation for predicting unconditional and conditional dependencies respectively, between variables not jointly measured. Section 8 extends the theory to devise an algorithm that can also predict the strength of the dependence. Section 9 presents the statistical matching theory and comparative evaluation. The paper concludes with Section 10 and 11 discussing the related work and the paper in general.

2. Modeling Causality with Maximal Ancestral Graphs

Maximal Ancestral Graphs (MAGs) is a type of graphical model that represents causal relations among a set of measured (observed) variables $\mathbf{O}$ as well as probabilistic properties, such as conditional independencies (independence model). *The probabilistic properties of MAGs can be de-*

veloped without any reference to their causal semantics; nevertheless, we also briefly discuss their causal interpretation.

MAGs can be viewed as a generalization of Causal Bayesian Networks. The causal semantics of an edge $A \to B$ imply that A is probabilistically causing B, that is, an (appropriate) manipulation of A results in a change of the distribution of B. Edges $A \leftrightarrow B$ imply that A and B are associated but neither A causes B nor vice-versa. Under certain conditions, the independencies implied by the model are given by a graphical criterion called m-separation, defined below. A desired property of MAGs is that they are closed under marginalization: the marginal of a MAG is a MAG. MAGs can also represent the presence of selection bias, but this is out of the scope of the present paper. We present the key theory of MAGs, introduced in Richardson and Spirtes (2002).

A path in a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a sequence of distinct vertices $\langle V_0, V_1, \dots, V_n \rangle$ all of them in $\mathbf{O}$ s.t for $0 \leq i < n$, V_i and V_{i+1} are adjacent in $\mathcal{G}$. A path from V_0 to V_n is *directed* if for $0 \leq i < n$, V_i is a parent V_{i+1}. X is called an *ancestor* of Y and Y a *descendent* of X if $X = Y$ or there is a directed path from X to Y in $\mathcal{G}$. $\mathbf{An}_{\mathcal{G}}(X)$ is used to denote the set of ancestors of node X in $\mathcal{G}$. A *directed cycle* in $\mathcal{G}$ occurs when $X \to Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. An *almost directed cycle* in $\mathcal{G}$ occurs when $X \leftrightarrow Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$.

Definition 1 (Mixed and Ancestral Graph) *A graph is mixed if all of its edges are either directed or bi-directed. A mixed graph is* **ancestral** *if the graph does not contain any directed or almost directed cycles.*

Given a path $p = \langle V_0, V_1, \dots, V_n \rangle$, node $V_i, i \in 1, 2, \dots, n$ is a *collider* on p if both edges incident to V_i have an arrowhead towards V_i. We also say that triple (V_{i-1}, V_i, V_{i+1}) forms a collider. Otherwise V_i is called a *non-collider* on p. The criterion of m-separation leads to a graphical way of determining the probabilistic properties stemming from the causal semantics of the graph:

Definition 2 (m-connection, m-separation) *In a mixed graph $\mathcal{G} = (\mathbf{E}, \mathbf{V})$, a path p between A and B is* ***m-connecting*** *relative to (condition to) a set of vertices $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \{A, B\}$ if*

1. *Every non-collider on p is not a member of $\mathbf{Z}$.*

2. *Every collider on the path is an ancestor of some member of $\mathbf{Z}$.*

*A and B are said to be m-**separated** by $\mathbf{Z}$ if there is no m-connecting path between A and B relative to $\mathbf{Z}$. Otherwise, we say they are m-**connected** given $\mathbf{Z}$. We denote the m-separation of A and B given $\mathbf{Z}$ as $MSep(A; B|\mathbf{Z})$. Non-empty sets $\mathbf{A}$ and $\mathbf{B}$ are m-separated given $\mathbf{Z}$ (symb. $MSep(\mathbf{A}; \mathbf{B}|\mathbf{Z})$) if for every $A \in \mathbf{A}$ and every $B \in \mathbf{B}$ A and B are m-separated given $\mathbf{Z}$. ($\mathbf{A}$, $\mathbf{B}$ and $\mathbf{Z}$ are disjoint). We also define the set of all m-separations as $\mathcal{J}_m(\mathcal{G})$:*

$$\mathcal{J}_m(\mathcal{G}) \equiv \{\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle, s.t.\ MSep(\mathbf{X}; \mathbf{Y} | \mathbf{Z}) \text{ and } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}\}.$$

We also define the set $\mathcal{J}$ of all conditional independencies $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$, where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are disjoint sets of variables, in the joint distribution of $\mathcal{P}$ of $\mathbf{O}$:

$$\mathcal{J}(\mathcal{P}) \equiv \{\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} | \rangle, \text{s.}t., \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \text{ and } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O}\}.$$

The set $\mathcal{J}(\mathcal{P})$ is also called the *independence model* of $\mathcal{P}$. The m-separation criterion is meant to connect the graph with the observed independencies in the distribution under the following assumption:

Definition 3 (Faithfulness) *We call a distribution $\mathcal{P}$ over a set of variables* **O** ***faithful*** *to a graph $\mathcal{G}$, and vice versa, iff:*

$$\mathcal{J}(\mathcal{P}) = \mathcal{J}_m(\mathcal{G}).$$

A graph is faithful iff there exists a distribution faithful to it. When the above equation holds, we say the Faithfulness Condition holds for the graph and the distribution.

When the faithfulness condition holds, every m-separation present in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{J}(\mathcal{P})$ and vice-versa. The following definition describes a subset of ancestral graphs in which every missing edge (non-adjacency) corresponds to at least one conditional independence:

Definition 4 (Maximal Ancestral Graph, MAG) *An ancestral graph $\mathcal{G}$ is called* maximal *if for every pair of non-adjacent vertices (X,Y), there is a (possibly empty) set $\mathbf{Z}$, $X,Y \notin \mathbf{Z}$ such that $\langle X,Y|\mathbf{Z}\rangle \in \mathcal{J}(\mathcal{G})$.*

Every ancestral graph can be transformed into a unique equivalent MAG (i.e., with the same independence model) with the possible addition of bi-directed edges. We denote the marginal of a distribution $\mathcal{P}$ over a set of variables $V \setminus L$ $\mathbf{L}$ as $\mathcal{P}[_{\mathbf{L}}$, and the independence model stemming from the marginalized distribution as $\mathcal{J}(\mathcal{P})[_{\mathbf{L}}$, that is,

$$\mathcal{J}(\mathcal{P}[_{\mathbf{L}}) = \mathcal{J}(\mathcal{P})[_{\mathbf{L}} \equiv \{\langle \mathbf{X},\mathbf{Y}|\mathbf{Z}\rangle \in \mathcal{J}(\mathcal{P}) : (\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z})\cap\mathbf{L} = \emptyset\}.$$

Equivalently, we define the set of m-separations of $\mathcal{G}$ restricted on the marginal variables as:

$$\mathcal{J}_m(\mathcal{G})[_{\mathbf{L}} \equiv \{\langle \mathbf{X},\mathbf{Y}|\mathbf{Z}\rangle \in \mathcal{J}_m(\mathcal{G}) : (\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z})\cap\mathbf{L} = \emptyset\}.$$

A simple graphical transformation for a MAG $\mathcal{G}$ faithful to a distribution $\mathcal{P}$ with independence model $\mathcal{J}(\mathcal{P})$ exists that provides a unique MAG $\mathcal{G}[_{\mathbf{L}}$ that represents the causal ancestral relations and the independence model $\mathcal{J}(\mathcal{P})[_{\mathbf{L}}$ after marginalizing out variables in $\mathbf{L}$. Formally,

Definition 5 (Marginalized Graph $\mathcal{G}[_L$) *Graph $\mathcal{G}[_L$ has vertex set $\mathbf{V}\setminus\mathbf{L}$, and edges defined as follows: If X,Y are s.t. , $\forall \mathbf{Z} \subseteq \mathbf{V}\setminus(\mathbf{L}\cup\{X,Y\})$, $\langle X,Y|\mathbf{Z}\rangle \notin \mathcal{J}(\mathcal{G})$ and*

$$\begin{array}{llll} X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) & & X \leftrightarrow Y & \\ X \in \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) & \textit{then} & X \rightarrow Y & \text{in } \mathcal{G}[_L. \\ X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \in \mathbf{An}_{\mathcal{G}}(X) & & X \leftarrow Y & \end{array}$$

We will call $\mathcal{G}[_L$ the marginalized graph $\mathcal{G}$ over **L**.

The following result has been proved in Richardson and Spirtes (2002):

Theorem 6 *If $\mathcal{G}$ is a MAG over* **V**, *and $\mathbf{L} \subseteq \mathbf{V}$, then $\mathcal{G}[_{\mathbf{L}}$ is also a MAG and*

$$\mathcal{J}_m(\mathcal{G})[_{\mathbf{L}} = \mathcal{J}_m(\mathcal{G}[_L).$$

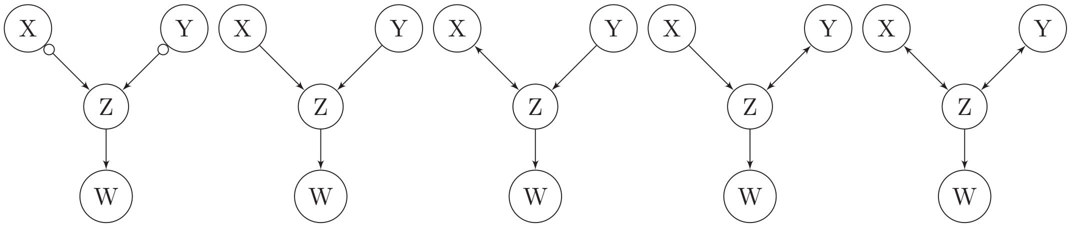


Figure 1: A PAG (left) and the MAGs of the respective equivalence class; all MAGs represent the same independence model over variables $\{X,Y,Z,W\}$.

If $\mathcal{G}$ is faithful to a distribution $\mathcal{P}$ over $\mathbf{V}$, then the above theorem implies that $\mathcal{J}(\mathcal{P})[_{\mathbf{L}}= \mathcal{J}(\mathcal{G})[_{\mathbf{L}}= \mathcal{J}(\mathcal{G}[_L)$; in other words the graph $\mathcal{G}[_{\mathbf{L}}$ constructed by the above process faithfully represents the marginal independence model $\mathcal{J}[_{\mathbf{L}}(\mathcal{P})$.

Different MAGs encode different causal information, but may share the same independence models and thus are statistically indistinguishable based on these models alone. Such MAGs define a Markov equivalence class based on the concepts of unshielded collider and discriminating path: A triple of nodes (X,Y,W) is called *unshielded* if X is adjacent to Y, Y is adjacent to W, and X is not adjacent to W. A path $p = \langle X,\dots,W,V,Y\rangle$ is called a *discriminating* path for V if X is not adjacent to Y, and every vertex between X and Y is a collider on p and an ancestor of Y. The following result has been proved in Spirtes and Richardson (1996):

Proposition 7 *Two MAGs over the same vertex set are Markov equivalent if and only if:*

1. *They share the same edges.*

2. *They share the same unshielded colliders.*

3. *If a path p is discriminating for a vertex V in both graphs, V is a collider on the path on one graph if and only if it is a collider on the path on the other.*

A *Partial Ancestral Graph* is a graph containing (up to) three kinds of endpoints: arrowhead ($>$), tail ($-$), and circle ($\circ$), and represents a MAG Markov equivalence class in the following manner: It has the same adjacencies as any member of the equivalence class, and every non-circle endpoint is invariant in any member of the equivalence class. Circle endpoints correspond to uncertainties; the definitions of paths are extended with the prefix *possible* to denote that there is a configuration of the uncertainties in the path rendering the path ancestral or *m*-connecting. For example if $X \circ\!-\!\circ Y \circ\!\rightarrow W$, $\langle X,Y,W\rangle$ is a possible ancestral path from X to W, but not a possible ancestral path from W to X. An example PAG, and some of the MAGs in the respective equivalence class are shown in Figure 1. FCI (Spirtes et al., 2001; Zhang, 2008) is a sound algorithm which outputs a PAG over a set of variables $\mathbf{V}$ when given access to an independence model over $\mathbf{V}$.

The MAG formulation is a generalization of the graph of a (Causal) Bayesian Network (CBN) intended to explicitly model and reason with latent variables and particularly, latent confounding variables. The absence of such confounding variables is (often unrealistically) assumed when learning Causal Bayesian Networks, named the *Causal Sufficiency* assumption. The presence of latent confounders can be modeled in MAGs with bidirectional edges. The graph of a CBN is a MAG without bidirectional edges. Similarly, the Faithfulness Condition we define for MAGs generalizes the Faithfulness for CBNs. This work is inspired by the following scenario: *there exists an unknown causal mechanism over variables* $\mathbf{V}$*, represented by a faithful CBN* $\langle \mathcal{P}, \mathcal{G} \rangle$. Based on the theory presented in this section (Theorem 6), each marginal distribution of $\mathbf{P}$ over a subset $\mathbf{O} = \mathbf{V} \setminus \mathbf{L}$ is *faithful* to the MAG $\mathcal{G}[_\mathbf{L}$ described in definition 5.

3. Scope and Motivation of Integrative Causal Analysis

A general objective is to develop algorithms that are able to co-analyze data sets that are heterogeneous in various aspects, including data sets defined over different variables sets, experimental conditions, sampling methodologies (e.g., observational vs. case-control sampling) and others. In addition, cross-sectional data sets could be eventually co-analyzed with temporal data sets measuring either time-series data or repeated measurements data. Finally, the integrative analysis should also include prior knowledge about the data and their semantics. Some of the tasks of the integrative analysis can be the identification of the causal structure of the data generating mechanism, the selection of the next most promising experiment, the construction of predictive models, the prediction of the effect of manipulations, or the selection of the manipulation that best achieves a desired effect.

The work in this paper however, focuses on providing a first step towards this direction. It addresses the problem of learning the structure of the data generating process from data sets defined over different variable sets. In addition, it focuses on providing proof-of-concept experiments of the main INCA idea on the simplest cases and comparing against current alternatives. Finally, it gives methods that predict the strength of dependence between Y and Z, which can be seen as constructing a simple predictive model without having access to the joint distribution of the data.

We now make concrete some of these ideas by presenting a motivating fictitious integrative analysis scenario:

- **Study 1** (i.i.d., observational sampling, variables A, B, C, D): A scientist is studying the "relation" between contraceptives and breast cancer. In a random sample of women, he measures variables $\{A, B, C, D\}$ corresponding to quantities Suffers from *Thrombosis (Yes/No)*, *Contraceptives (Yes/No)*, *Concentration of Protein C in the Blood (numerical)* and *Develops Breast Cancer by 60 Years Old (Yes/No)*. The researcher then develops predictive models for Breast Cancer and, given that he finds B associated with D (among other associations), announces taking contraceptives as a risk-factor for developing Breast Cancer.

- **Study 2** (randomized controlled trial, variables A, B, C, D): Another scientist checks whether (variable C) *Protein C* (causally) protects against cancer. In a randomized controlled experiment she randomly assigns women into two groups and measures the same variables $\{A, B, C, D\}$. The first group is injected with high levels of the protein in their blood, while the latter is injected with enzymes that dissolve only the specific protein, effectively removing it from the blood. If C and D are negatively correlated in her data, the scientist concludes that the protein is causally protecting against the development of breast cancer. Notice that, data from Study 2 cannot be merged with Study 1 because the joint distributions of the data may be different. For example, assuming that C is caused by the disease D (e.g., the disease changes the concentration of the protein in the blood) then C will be highly associated with D in Study 1; in contrast, in Study 2 where the levels of C exclusively depend on the group

Variables Study	A Thrombosis (Yes/No)	B Contraceptives (Yes/No)	C Protein C (numerical)	D Cancer (Yes/No)	E Protein Y (numerical)	F Protein Z (numerical)
1 (observational data)	Yes	No	10.5	Yes	-	-
	No	Yes	5.3	No	-	-
	No	Yes	0.01	No	-	-
2 (experimental data)	No	No	**0**(Control)	No	-	-
	Yes	No	**0**(Control)	Yes	-	-
	Yes	Yes	**5.0**(Treat.)	Yes	-	-
3 (different variables)	-	-	-	Yes	0.03	9.3
	-	-	-	No	3.4	22.2
4 (prior knowledge)	B causally affects A: B $\dashrightarrow$ A					

Figure 2: Tabular depiction of the different studies (data sets). Study 1 is a random sample aiming at predicting D and identifying risk factors. Study 2 is a Randomized Controlled Trial were the levels of C for a subject are randomly decided and enforced by the experimenter, aiming at identifying a causal relation with cancer. Forced values are denoted by bold font. Study 3 is also an observational study about D, but measuring different variables than Study 1. Prior knowledge provides a piece of causal knowledge but the raw data are not available. Typically, such studies are analyzed independently of each other.

assignment, C and D are not associated. Thus, statistical inferences made based on analyzing Study 2 in isolation probably result in lower statistical power.

- **Study 3** (i.i.d., observational sampling, variables D,E,F): A biologist studies the relation of a couple of proteins in the blood, represented with variables E and F and their relation with breast cancer. She measures in a random sample of women variables $\{D,E,F\}$. As with analyzing Study 1, she develops predictive models for Breast Cancer (based on E and F instead) and checks whether the two proteins are risk factors. These data cannot be pulled together with Studies 1 or 2 because they measure different variables.

- **Prior Knowledge**: A doctor establishes a causal relation between the use of *Contraceptives* (variable B) and the development of *Thrombosis* (variable A), that is, "B causes A" denoted as $B \dashrightarrow A$.[1] Unfortunately, the raw data are not publicly available.

The three studies and prior knowledge are depicted Figure 2. Notice that, treating the empty cells as missing values is meaningless given that it is impossible for an algorithm to estimate the joint distribution between variables never measured together without additional assumptions (see Rubin 1974 for more details).

[1] We use a double arrow $\dashrightarrow$ to denote a causal relation without reference to the context of other variables. This is to avoid confusion with the use of a single arrow $\rightarrow$ in most causal models (e.g., Causal Bayesian Networks) that denotes a *direct* causal relation (or inducing path, see Richardson and Spirtes 2002), where direct causality is defined in the context of the rest of the variables in the model.

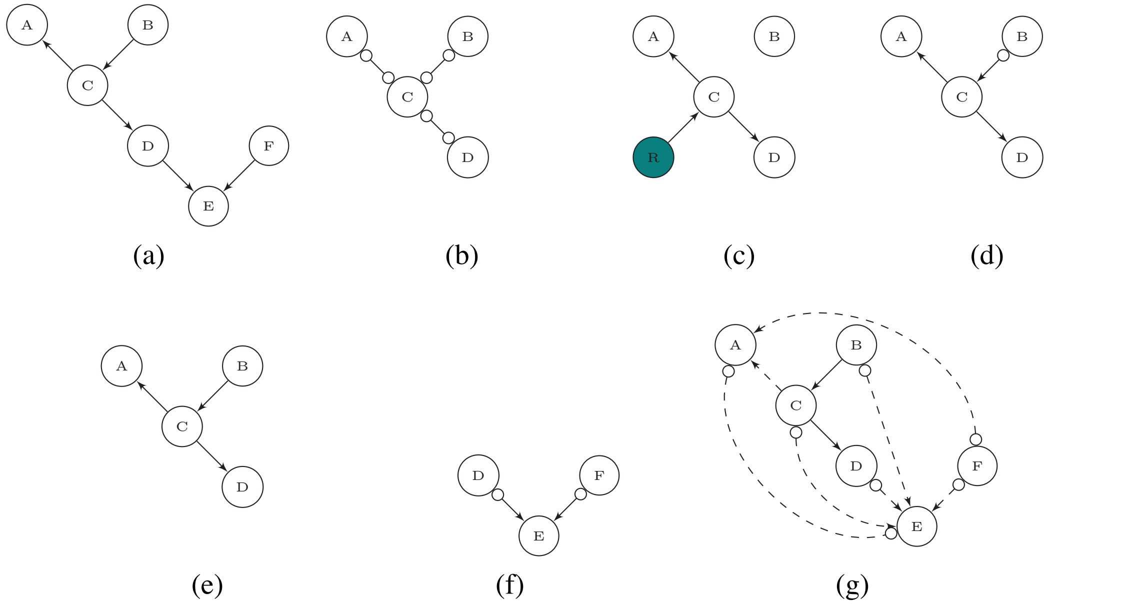


Figure 3: (a) Assumed unknown causal structure. (b) Structure induced by Study 1 alone. (c) Structure induced by Study 2 alone. (d) Structure induced by INCA of Studies 1 and 2. New inference: C is not causing B but they are associated. (e) Structure induced after incorporating knowledge "B causes A". New inference: B causes A and D. (f) Structure induced by Study 3 alone. (g) Structure induced by all studies and knowledge. Dashed edges denote edges whose both existence and absence is consistent with the data. New inference: F and C (two proteins) are not causing each other nor do they have a latent confounder, even though we never measure them together in a study.

We now show informally the reasoning for an integrative causal analysis of the above studies and prior knowledge and compare against independent analysis of the studies. Figure 3(a) shows the presumed true, unknown, causal structure. Figure 3(b-c) shows the causal model induced (asymptotically) by an independent analysis of the data of Study 1 and Study 2 respectively using existing algorithms, such as FCI (Spirtes et al., 2001; Zhang, 2008) and assuming data generated by the true model. The R variable denotes the randomization procedure that assigns patients to control and treatment groups. Notice that it removes any causal link into C since the value of C only depends on the result of the randomization. Figure 3(d) shows the causal model that can be inferred by co-analyzing both studies together. By INCA of Study 1 and 2 it is now additionally inferred that B and C are correlated but C does not cause B: If C was causing B, we would have found the variables dependent in Study 2 (the randomization procedure would not have eliminated the causal link $C \to B$). If we also incorporate prior knowledge that "B causes A" we obtain the graph in Figure 3(e): "B causes A" implies that there has to be at least one directed (causal) path from B to A. Thus, the only possible such path $B \circ\!\!\to C \to A$ becomes directed $B \to C \to A$. In other words using prior knowledge we now additionally infer that "B is causing C": the association found in Study 1 cannot be totally explained by the presence of a latent variable. Analyzing independently Study 3 we obtain the graph of Figure 3(f). In contrast INCA of Study 3 with the rest of data and knowledge results in

Figure 3(g). This type of graph is called the Pairwise Causal Graph (Triantafillou et al., 2010) and is presented in detail in Section 5. The dashed edges denote statistical indistinguishability about the existence of the edge, that is, there exist a consistent causal model with all data and knowledge having the edge, and one without the edge. Among other interesting inferences, notice that *F and C (two proteins) are not causing each other nor do they have a latent confounder, even though we never measure them together*. This is because if $F \to C$, or $C \leftarrow F$, or there exists latent H such that $F \leftarrow H \to C$ it would also imply an association between F and D. These two are found independent however, in Study 3.

4. Running Example

To illustrate the main ideas and concepts, as well as provide a proof-of-concept validation in real data, we have identified the smallest and simplest scenario that we could think of, that makes a testable prediction. Specifically, we identify a special case that predicts an unconditional dependence $Y \not\perp\!\!\!\perp Z|\emptyset$, as well as certain conditional dependencies $Y \not\perp\!\!\!\perp Z|\mathbf{S}$, for some $\mathbf{S} \neq \emptyset$, between two variables not measured in the same samples, based on two data sets, one measuring Y, and one measuring Z.

Example 1 *We assume two i.i.d data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ are provided on variables $\mathbf{O}_1 = \{X,Y,W\}$ and $\mathbf{O}_2 = \{X,Z,W\}$ respectively. We assume that the independence models of the data sets are $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$ and $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$, in other words the one and only independence in $\mathcal{D}_1$ is $X \perp\!\!\!\perp W|Y$, and in $\mathcal{D}_2$ is $X \perp\!\!\!\perp W|Z$. Based on the input data it is possible to induce with existing causal analysis algorithms, such as FCI the following PAGs from each data set respectively:*

$$\mathcal{P}_1 : X \circ\!-\!\circ Y \circ\!-\!\circ W$$

and

$$\mathcal{P}_2 : X \circ\!-\!\circ Z \circ\!-\!\circ W.$$

These are also shown graphically in Figure 4. The problem is to identify one or all MAGs defined on $\mathbf{O} = \{X,Y,Z,W\}$ consistent with the independence models $\mathcal{J}_1$ and $\mathcal{J}_2$, or equivalently, both PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$.

These two PAGs represent all the sound inferences possible about the structure of the data, when analyzing the data sets in isolation and independently of each other. We next develop the theory for their causal co-analysis.

5. Integrative Causal Analysis of Data Sets with Overlapping Variable Sets

In this section, we address the problem of integratively analyzing multiple data sets defined over different variable sets. Co-analyzing these data sets is meaningful (using this approach) only when these variable sets overlap; otherwise, there are no additional inferences to be made unless other information connects the two data sets (e.g., the presence of prior knowledge connecting some variables).

We assume that we are given K data sets $\{\mathcal{D}_i\}_{i=1}^K$ each with samples identically and independently distributed defined over corresponding subsets of variables $\mathbf{O}_i$. From these data we can estimate the independence models $\{\mathcal{J}_i\}_{i=1}^K$ using statistical tests of conditional independence. *A*

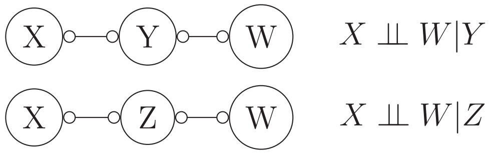


Figure 4: Definition of the co-analysis problem of Example 1: two observational i.i.d. data sets defined on variables $\mathbf{O}_1 = \{X,Y,W\}$ and $\mathbf{O}_2 = \{X,Z,W\}$ are used to identify the independence models $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$ and $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$. These models are represented by PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ shown in the figure. The problem is to identify one or all MAGs defined on $\mathbf{O} = \{X,Y,Z,W\}$ consistent with both $\mathcal{P}_1$ and $\mathcal{P}_2$.

major assumption in the theory and algorithms presented is that the independence models can be identified without any statistical errors. Section 6 discusses how we address this issue when experimenting with real data sets in the presence of statistical errors. We denote the union of all variables as $\mathbf{O} = \cup_{i=1}^{K}\mathbf{O}_i$ and also define $\overline{\mathbf{O}_i} \equiv \mathbf{O} \setminus \mathbf{O}_i$. We now define the problem below:

Definition 8 (Find Consistent MAG) *Assume the distribution of* $\mathbf{O}$ *is faithful. Given independence models* $\{\mathcal{J}(\mathbf{O}_i)\}_{i=1}^{K}$, $\mathbf{O}_i \subseteq \mathbf{O}, i = 1 \ldots K$, *induce a MAG* $\mathcal{M}$ *s.t., for all i*

$$\mathcal{J}(\mathcal{M}[_{\overline{\mathbf{O}_i}}) = \mathcal{J}(P_i)$$

where P_i *is the distribution of* $\mathbf{O}_i$.

In other words, we are looking for a model (graph) $\mathcal{M}$ such that when we consider its marginal graphs over each variable set $\mathbf{O}_i$, each one faithfully represents the observed independence model of that data set. We can reformulate the problem in graph-theoretic terms. Let $\mathcal{P}_i$ be the PAG representing the Markov equivalence class of all MAGs consistent with the independence model $\mathcal{J}_i$. $\mathcal{P}_i$ can be constructed with a sound and complete algorithm such as Fast Causal Inference (FCI) (Spirtes et al., 2001). We can thus recast the problem above as identifying a MAG $\mathcal{M}$ such that,

$$M[_{\overline{\mathbf{O}_i}} \in \mathcal{P}_i, \text{for all } i$$

(abusing the notation to denote with $\mathcal{P}_i$ both the PAG and the equivalence class).

The first algorithm to solve the above problem is ION (Tillman et al., 2008), which identifies the set of PAGs (defined over $\mathbf{O}$) of all consistent MAGs. Subsequently, in Triantafillou et al. (2010), we proposed the algorithm Find Consistent MAG (FCM) that converts the problem to a satisfiability problem for improved computational efficiency. FCM returns one consistent MAG with all input PAGs. Similar ideas have been developed to learn joint structure from marginal structures in decomposable graphs such as undirected graphs (Kim and Lee, 2008) and Bayesian Networks (Kim, 2010). Going back to Example 1, Figure 5 shows all 14 consistent MAGs with the input PAGs in the scenario. The FCM algorithm arbitrarily returns one of them as the solution to the problem (of course, the algorithm can be easily modified to return all solutions). Figure 6 (right) shows the output of ION on the same problem.

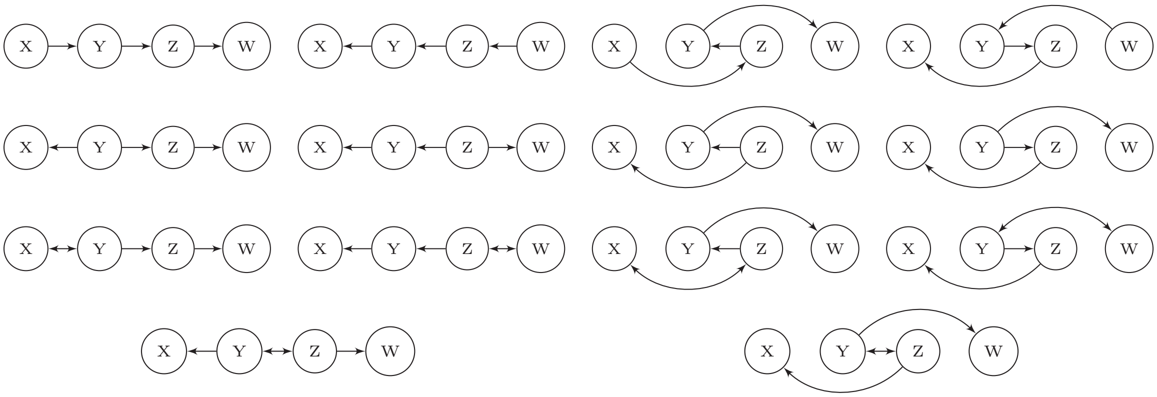


Figure 5: Solution of the co-analysis problem of Example 1: The 14 depicted MAGs are all and only the consistent MAGs with the PAGs shown in Figure 4. In all these MAGs the independencies $X \perp\!\!\!\perp W|Y$ and $X \perp\!\!\!\perp W|Z$ hold (and only them). Notice that, even though the edge $X - Y$ exists in $\mathcal{P}_1$ (Example 1), some of the consistent MAGs (the ones on the right of the figure) do not contain this edge: *adjacencies in the input PAGs do not simply transfer to the solution MAGs*. The FCM algorithm would arbitrarily output one of these MAGs as the solution of the problem of Example 1.

5.1 Representing the Set of Consistent MAGs with Pairwise Causal Graphs

The set of consistent MAGs to a set of PAGs is defined as follows:

Definition 9 (Set of Consistent MAGs) *We call the set of all MAGs $\mathcal{M}$ over variables $\mathbf{O}$ consistent with the set of PAGs $\mathbf{P} = \{\mathcal{P}_i\}_{i=1}^N$ over corresponding variable sets $\mathbf{O}_i$, where $\mathbf{O} = \cup_i \mathbf{O}_i$ as the Set of Consistent MAGs with $\mathbf{P}$ denoted with $\mathbf{M}(\mathbf{P})$.*

Unfortunately, $\mathbf{M}(\mathbf{P})$ cannot in general be represented with a single PAG: the PAG formalism represents a set of equivalent MAGs *when learning from a single data set and its independence model*. In Example 1 though, notice that the MAGs in $\mathbf{M}(\mathbf{P})$ in Figure 5 have a different skeleton (i.e., set of edges ignoring the edge-arrows), so they cannot be represented by a single PAG.

The PAG formalism allows the set of *m*-separations that entail the *m*-separations of all MAGs in the class to be read off its graph in polynomial time. Unfortunately, there is currently no known compact representation of $\mathbf{M}(\mathbf{P})$ such that the *m*-separations that hold for all members of the set can be easily identified (i.e., in polynomial time).

We have introduced (Triantafillou et al., 2010) a new type of graph called the *Pairwise Causal Graph* (PCG) that graphically represents $\mathbf{M}(\mathbf{P})$. However, PCG do not always allow the *m*-separations of each member MAG to be easily identified. A PCG focuses on representing the possible causal pair-wise relations among each pair of variables X and Y in $\mathbf{O}$.

Definition 10 (Pairwise Causal Graph) *We consider the MAGs in $\mathbf{M}(\mathbf{P})$ consistent with the set of PAGs $\mathbf{P} = \{\mathcal{P}_i\}_{i=1}^N$ defined over $\{\mathbf{O}_i\}_{i=1}^N$. A Pairwise Causal Graph $\mathcal{U}$ is a partially oriented mixed graph over $\bigcup_i \mathbf{O}_i$ with two kinds of edges dashed (--) and solid (—) and three kinds of endpoints(>, -, ∘) with the following properties:*

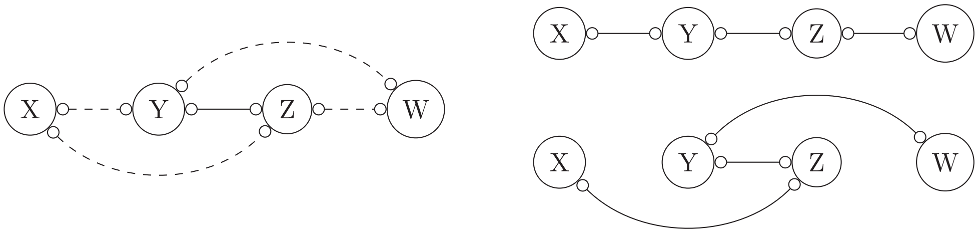


Figure 6: (left) Pairwise Causal Graph (PCG) representing the set of consistent MAGs of Example 1. This PCG is the output of the cSAT+ algorithm on the problem of Example 1. Alternatively, the set of consistent MAGs can be represented with two PAGs (right). This is the output of the ION algorithm on the same problem.

1. *$X - Y$ in $\mathcal{U}$ iff X is adjacent to Y in every consistent $\mathcal{M} \in \mathbf{M}(\mathbf{P})$.*

2. *$X \dashrightarrow\!\!\!\!\!\!- Y$ in $\mathcal{U}$ iff X is adjacent to Y in at least one but not all consistent $\mathcal{M} \in \mathbf{M}(\mathbf{P})$.*

3. *X and Y are not adjacent in $\mathcal{U}$ iff they are not adjacent in any consistent $\mathcal{M} \in \mathbf{M}(\mathbf{P})$.*

4. *The right end-point of edge X -- Y is oriented as $>$, -, or $\circ$ iff X is into Y in all, none, or at least one (but not all) consistent MAG $\mathcal{M} \in \mathbf{M}(\mathbf{P})$ where X and Y are adjacent. Similarly, for the left end-point and for solid edges $X - Y$.*

Solid edges, missing edges, as well as end-points marked with"$>$" and "$-$" show invariant characteristics that hold in all consistent MAGs. Dash edges and "$\circ$"-marked end-points represent uncertainty of the presence of the edge and the type of the end-point.

The PCG of Example 1 is shown in Figure 6 (left). For computing the PCG one can employ the cSAT+ algorithm (Triantafillou et al., 2010). There are several points to notice. The invariant graph features are the solid edge $Y - Z$ and the missing edge between X and W; these are shared by all consistent MAGs. The remaining edges are dashed showing that they are present in at least one consistent MAG. All end-points are marked with "$\circ$" showing that any type of orientation is possible for each of them. The graph fails to graphically represent certain constraints, for example, that there is no MAG that simultaneously contains edges $X - Y$ and $X - Z$; in general, the presence of an edge (or a particular end-point) in a consistent MAG may entail the absence of some other edge (or end-point). It also fails to depict the *m*-separation $X \perp\!\!\!\perp W|Z$ or the fact that any solution has a chain-like structure.

Nevertheless, the graph still conveys valuable information: *the solid edge $X - Y$ along with the Faithfulness condition entails that Y and Z are associated given any subset of the other variables, even though Y and Z are never measured together in any input data set*. This is a testable prediction on which we base the computational experiments in Section 6. Alternatively, the set $\mathbf{M}(\mathbf{P})$ could be represented with *two* PAGs shown in 6 (right), as the set of MAGs consistent with either one them. These PAGs form the output of ION on this problem.

6. Predicting the Presence of Unconditional Dependencies

We now discuss how to implement the identification of the scenario in Example 1 to predict the presence of dependencies.

6.1 Predictions of Dependencies

Recall that, in Example 1 we assume we are given two data sets on variables $\mathbf{O}_1 = \{X,Y,W\}$ and $\mathbf{O}_2 = \{X,Z,W\}$. We then determine, if possible, whether their independence models are respectively $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$ and $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$ by a series of unconditional and conditional tests of independence. If this is the case, we predict an association between Y and Z. The details of determining the independence model are important. Let us denote the p-value of an independence test with null hypothesis $X \perp\!\!\!\perp Y|\mathbf{Z}$ as $p_{X \perp\!\!\!\perp Y|\mathbf{Z}}$. In the algorithms that follow, we make statistical decisions with the following rules:

- If $p_{X \perp\!\!\!\perp Y|\mathbf{Z}} \leq \alpha$ conclude $X \not\perp\!\!\!\perp Y|\mathbf{Z}$ (reject the null hypothesis).
- If $p_{X \perp\!\!\!\perp Y|\mathbf{Z}} \geq \beta$ conclude $X \perp\!\!\!\perp Y|\mathbf{Z}$ (accept the null hypothesis).
- Otherwise, forgo making a decision.

Algorithm 1: Predict Dependency: Full-Testing Rule (**FTR**)

```
Input: Data Sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{X,Y,W\}$ and $\{X,Z,W\}$, respectively
if in $\mathcal{D}_1$ we conclude
        // determine whether $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$
        $X \perp\!\!\!\perp W|Y$ , $X \not\perp\!\!\!\perp Y|\emptyset$ , $Y \not\perp\!\!\!\perp W|\emptyset$ , $X \not\perp\!\!\!\perp W|\emptyset$ , $X \not\perp\!\!\!\perp Y|W$ , $Y \not\perp\!\!\!\perp W|X$
    and in $\mathcal{D}_2$ we conclude
        // determine whether $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$
        $X \perp\!\!\!\perp W|Z$ , $X \not\perp\!\!\!\perp Z|\emptyset$ , $Z \not\perp\!\!\!\perp W|\emptyset$ , $X \not\perp\!\!\!\perp W|\emptyset$ , $X \not\perp\!\!\!\perp Z|W$ , $Z \not\perp\!\!\!\perp W|X$
then
    Predict $Y \not\perp\!\!\!\perp Z|\emptyset$
    Predict either ($X \circ\!-\!\circ Y \circ\!-\!\circ Z \circ\!-\!\circ W$) or ($X \circ\!-\!\circ Z \circ\!-\!\circ Y \circ\!-\!\circ W$) holds
else
    Do not make a prediction
end
```

The details are shown in Algorithm 1 named Full-Testing Rule, or FTR for short. We note a couple of observations. First, the algorithm is opportunistic. It does not produce a prediction whenever possible, but only for the case presented in Example 1. In addition, it makes a prediction only when the p-values of the tests are either too high or too low to relatively safely accept dependencies and independencies. Second, to accept an independence model, for example, that $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$ all possible conditional and unconditional tests among the variables are performed. If any of these tests is inconclusive or contradictory to $\mathcal{J}_1$, the latter is not accepted and no prediction is made. In the terminology of Spirtes et al. (2001), we test for a *detectable failure of faithfulness*. Similar ideas have also been devised in Ramsey et al. (2006) and Spanos (2006). This rule characteristic is important in case one would like to generalize these ideas to larger graphs and sets of variables:

performing all possible tests becomes quickly prohibitive, and the probability of statistical errors increases.

If however, one assumes the Faithfulness Condition holds among variables $\{X,Y,Z,W\}$, then it is not necessary to perform all such tests to determine the independence models. Algorithms for inducing graphical models from data, such as FCI and PC (Spirtes et al., 2001) are based on this observation to gain computational efficiency. The Minimal-Testing Rule, MTR for short, performs only a minimal number of tests that together with Faithfulness may entail that $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$ and $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$ and lead to a prediction. The details are shown in Algorithm 2.

```
Algorithm 2: Predict Dependency Minimal-Testing Rule (MTR)
Input: Data Sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{X,Y,W\}$ and $\{X,Z,W\}$, respectively
if in $\mathcal{D}_1$ we conclude
        // determine whether $\mathcal{J}_1 = \{\langle X,W|Y\rangle\}$
        $X \perp\!\!\!\perp W|Y$ , $X \not\perp\!\!\!\perp Y|\emptyset$ , $Y \not\perp\!\!\!\perp W|\emptyset$
    and in $\mathcal{D}_2$ we conclude
        // determine whether $\mathcal{J}_2 = \{\langle X,W|Z\rangle\}$
        $X \perp\!\!\!\perp W|Z$ , $X \not\perp\!\!\!\perp Z|\emptyset$ , $Z \not\perp\!\!\!\perp W|\emptyset$
then
    Predict $Y \not\perp\!\!\!\perp Z|\emptyset$
    Predict either ($X \circ\!-\!\circ Y \circ\!-\!\circ Z \circ\!-\!\circ W$) or ($X \circ\!-\!\circ Z \circ\!-\!\circ Y \circ\!-\!\circ W$) holds
else
    Do not make a prediction
end
```

6.2 Heuristic Predictions of Dependencies Based on Transitivity

Is it really necessary to develop and employ the theory presented to make such predictions? Could there be other simpler and intuitive rules that are as predictive, or more predictive? For example, a common heuristic inference people are sometimes willing to make is the transitivity rule: if Y is correlated with X and X is correlated with Z, then predict that Y is also correlated with Z. The FTR and MTR rules defined also check these dependencies: $X \not\perp\!\!\!\perp Y$ in $\mathcal{D}_1$ and $X \not\perp\!\!\!\perp Z$ in $\mathcal{D}_1$, so one could object that any success of the rules could be attributed to the transitivity property often holding in Nature. We implement the Transitivity Rule (TR), shown in Algorithm 3 to compare against the INCA-based FTR and MTR rules. Obviously, the Transitivity Rule is not sound in general,[2] but on the other hand, FTR and MTR are also based on the assumption of Faithfulness, which may as well be unrealistic. The verdict will be determined by experimentation.

6.3 Empirical Evaluation of Predicting Unconditional Dependencies

We have applied and evaluated the three rules against each-other as well as random predictions (prior probability of a pair being dependent) on real data, in a way that becomes testable. Specifically, given a data set $\mathcal{D}$ we randomly partition its samples to three data sets of equal size, $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_t$. The latter is hold out for testing purposes. In the first two data sets, we identify quadruples of

[2] The Transitivity Rule should be sound when the marginal of the three variables is faithful to a *Markov Random Field*.

```
Algorithm 3: Predict Dependency Transitivity Rule (TR)
Input: Data Sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{Y,X\}$ and $\{X,Z\}$, respectively
if in $\mathcal{D}_1$: $Y \not\perp\!\!\!\perp X|\emptyset$ and in $\mathcal{D}_2$: $X \not\perp\!\!\!\perp Z|\emptyset$ then
    Predict $Y \not\perp\!\!\!\perp Z|\emptyset$
else
    Do not make a prediction
end
```

Name	Reference	# istances	# vars	Group Size	Vars type	Scient. domain
Covtype	Blackard and Dean (1999)	581012	55	55	N/O	Agricultural
Read	Guvenir and Uysal (2000)	681	26	26	N/C/O	Business
Infant-mortality	Mani and Cooper (2004)	5337	83	83	N	Clinical study
Compactiv	Alcalá-Fdez et al. (2009)	8192	22	22	C	Computer science
Gisette	Guyon et al. (2006a)	7000	5000	50	C	Digit recognition
Hiva	Guyon et al. (2006b)	4229	1617	50	N	Drug discovering
Breast-Cancer	Wang (2005)	286	17816	50	C	Gene expression
Lymphoma	Rosenwald et al. (2002)	237	7399	50	C	Gene expression
Wine	Cortez et al. (2009)	4898	12	12	C	Industrial
Insurance-C	Elkan (2001)	9000	84	84	N/O	Insurance
Insurance-N	Elkan (2001)	9000	86	86	N/O	Insurance
p53	Danziger et al. (2009)	16772	5408	50	C	Protein activity
Ovarian	Conrads (2004)	216	2190	50	C	Proteomics
C&C	Frank and Asuncion (2010)	1994	128	128	C	Social science
ACPJ	Aphinyanaphongs et al. (2006)	15779	28228	50	C	Text mining
Bibtex	Tsoumakas et al. (2010)	7395	1995	50	N	Text mining
Delicious	Tsoumakas et al. (2010)	16105	1483	50	N	Text mining
Dexter	Guyon et al. (2006a)	600	11035	50	N	Text mining
Nova	Guyon et al. (2006b)	1929	12709	50	N	Text mining
Ohsumed	Joachims (2002)	5000	14373	50	C	Text mining

Table 1: Data Sets included in empirical evaluation of Section 6.3. N- Nominal, O - Ordinal, C - Continuous.

variables $\{X,Y,Z,W\}$ for which the Full-Testing and the Minimal-Testing Rules apply. Notice that, the two rules perform tests among variables $\{X,Y,W\}$ in $\mathcal{D}_1$ and among variables $\{X,Z,W\}$ in $\mathcal{D}_2$; *the rules do not access the joint distribution of* Y,Z. Similarly, for the Transitivity Rule we identify triplets $\{X,Y,Z\}$ where the rule applies. Subsequently, we measure the predictive performance of the rules. In more detail:

- *Data Sets*: We selected data sets in an attempt to cover a wide range of sample-sizes, dimensionality (number of variables), types of variables, domains, and tasks. The decision for inclusion depended on availability of the data, ease of parsing and importing them. *No data set was a posteriori removed out of the study, once selected*. Table 1 assembles the list of data sets and their characteristics before preprocessing. Some minimal preprocessing steps were applied to several data sets that are described in Appendix A.

- *Tests of Independence*: For discrete variables we have used the G^2-test (a type of likelihood ratio test) with an adjustment for the degrees-of-freedom used in Tsamardinos et al. (2006)

and presented in detail in Tsamardinos and Borboudakis (2010). For continuous variables we have used a test based on the Fisher z-transform of the partial correlation as described in Spirtes et al. (2001). The two tests employed are typical in the graphical learning literature. In some cases ordinal variables were treated as continuous, while in others the continuous variables were discretized (see Appendix A) so that every possible quadruple $\{X,Y,Z,W\}$ was either treated as all continuous variables or all discrete and one of the two tests above could be applied.

- *Significance Thresholds*: There are two threshold parameters: level α below which we accept dependence and level β above which we accept independence; the TR rule only employs the α parameter. For FTR these thresholds were always set to $\alpha_{FTR} = 0.05$ and $\beta_{FTR} = 0.3$ without an effort to optimize them. Some minimal anecdotal experimentation with FTR showed that the performance of the algorithm is relative insensitive to the values of α_{FTR} and β_{FTR} and the algorithm works without fine-tuning. Notice that FTR requires 10 dependencies and 2 independencies to be identified, while MTR requires 4 dependencies and 2 independencies, and TR requires 2 dependencies to be found. Thus, FTR is more conservative than MTR and TR for the same values of α and β. The Bonferroni correction for MTR dictates that $\alpha_{MTR} = \alpha_{FTR} \times \frac{4}{10} = 0.02$, while for TR we get $\alpha_{TR} = \alpha_{FTR} \times \frac{2}{10} = 0.01$ (TR however, does not require any independencies present so this adjustment may not be conservative enough). We run MTR with threshold values $\alpha_{MTR} \in \{0.05, 0.02, 0.002, 0.0002\}$, that is equal to the threshold of FTR, with the Bonferroni adjustment, and stricter than Bonferroni by one and two orders of magnitude. The β_{MTR} parameter is always set to 0.3. In a similar fashion for TR, we set $\alpha_{TR} \in \{0.05, 0.01, 0.001, 0.0001\}$.

- *Identifying Quadruples*: In low-dimensional data sets (number of variables less than 150), we check the rules on all quadruples of variables. This is time-prohibitive however, for the larger data sets. In such cases, we randomly permute the order of variables and partition them into groups of 50 and consider quadruples only within these groups. The column named "Group Size" in Table 1 notes the actual sizes of the variable groups used.

- *Measuring Performance*: The ground truth for the presence of a predicted correlation is not known. We thus seek to statistically evaluate the predictions. Specifically, for each predicted pair of variables X and Y, we perform a test of independence in the corresponding hold-out test set $\mathcal{D}_t$ and store its p-value $p_{X \perp\!\!\!\perp Y|\emptyset}$. The lower the p-value the higher the probability the pair is truly correlated. We consider as "accurate" a prediction whose p-value is less than a threshold t and we report the accuracy of each rule.

Definition 11 (Prediction Accuracy) *We denote with M_i^R and U_i^R the multiset and set respectively of p-values of the predictions of rule R applied on data set i. The p-values are computed on the hold-out test set. The accuracy of the rule on data set i at threshold t is defined as:*

$$Acc_i^R(t) = \#\{p <= t, p \in M_i^R\} / |M_i^R|.$$

We also define the average accuracy *over all data sets (each data set is weighted the same)*

$$\overline{Acc}^R(t) = \frac{1}{20} \sum_{i=1}^{20} Acc_i^R(t)$$

and the pooled accuracy *over the union of predictions (each prediction is weighted the same)*

$$\underline{Acc}^R(t) = \#\{p <= t, i = 1\ldots 20, p \in M_i^R\} / \sum_i |M_i^R|.$$

The reason M_i^R is defined as a multiset stems from the fact that a dependency $Y \not\perp\!\!\!\perp Z|\emptyset$ may be predicted multiple times if a rule applies to several quadruples $\{X_i, Y, Z, W_i\}$ or triplets $\{X_i, Y, Z\}$ (for the Transitivity Rule). The number of predictions of each rule R (i.e., $|M_i^R|$) is shown in Table 2, while Table 8 in Appendix A reports $|U_i^R|$, the number of pairs $X - Y$ predicted correlated. In some cases (e.g., data sets Read and ACPJ) the Full-Testing Rule does not make any predictions. Overall however, the rules typically make hundreds or even thousands of predictions.

Data Set	$\text{FTR}_{0.05}$	$\text{MTR}_{0.02}$	$\text{TR}_{0.01}$
Covtype	222	33277	54392
Read	0	9	4713
Infant Mortality	22	2038	3736
Compactiv	135	679	3950
Gisette	423	35824	134213
hiva	554	65967	151582
Breast-Cancer	1833	141643	470212
Lymphoma	7712	188216	394572
Wine	4	73	431
Insurance-C	1839	30569	40173
Insurance-N	226	18270	47115
p53	46647	1645476	1995354
Ovarian	539165	1604131	2015133
C&C	99241	416934	301218
ACPJ	0	219	16574
Bibtex	1	3982	25948
Delicious	856	32803	105776
Dexter	0	2	117
Nova	0	124	3473
Ohsumed	0	64	5358

Table 2: Number of predictions $|M_i^R|$ with "Bonferroni" correction for rules FTR, MTR and TR.

Overall Performance: The accuracies at $t = 0.05$, $Acc_i(t)$, $\overline{Acc}(t)$, and $\underline{Acc}(t)$ for the three rules as well as the one achieved by guessing at random are shown in Figure 7. The Bonferroni adjusted thresholds for MTR and TR were used: $\alpha_{FTR} = 0.05, \alpha_{MTR} = 0.02, \alpha_{TR} = 0.01$. Similar figures for all sets of thresholds are shown in Appendix A, Section A.3. Over all predictions, the Full-Testing Rule achieves accuracy 96%, consistently higher than guessing at random, the MTR and the TR. The same results are also depicted in tabular form in Table 3, where additionally, the statistical significance is noted. The null hypothesis is that $Acc_i^{FTR}(0.05) \leq Acc_i^R(0.05)$, for R being MTR or TR. The one-tail Fisher's exact test (Fisher, 1922) is employed when computationally feasible, otherwise the Pearson χ^2 test (Pearson, 1900) is used instead. FTR is typically performing statistically significantly better than all other rules.

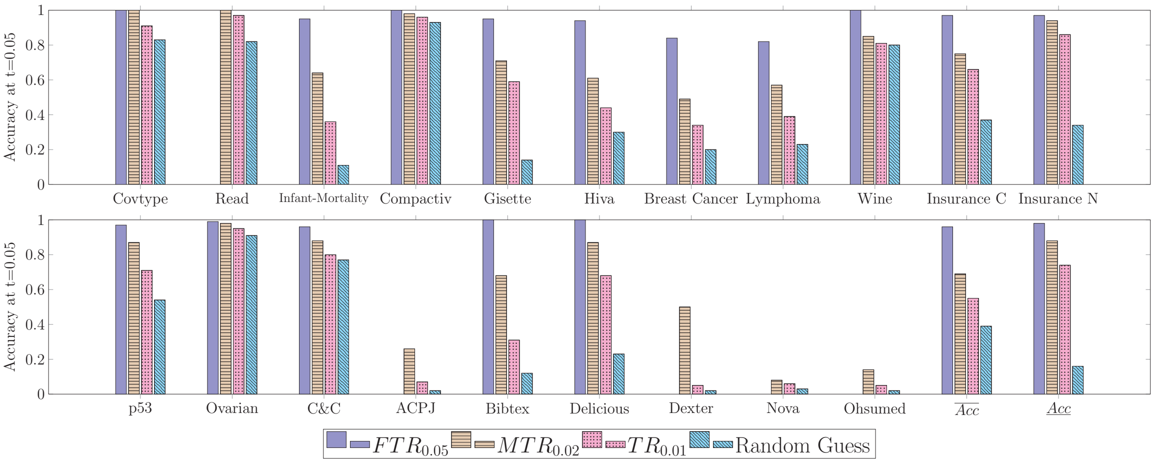


Figure 7: Accuracies Acc_i for each data set, as well as the average accuracy $\overline{Acc}$ (each data set weighs the same) and the pooled accuracy $\underline{Acc}$ (each prediction weighs the same). All accuracies are computed as threshold $t = 0.05$. FTR's accuracy is always above 80% and always higher than MTR, TR, and random guess.

Sensitivity to the α *parameter*: The results are not particularly sensitive to the significance thresholds used for α for MTR and TR. Figures 9 (a-b) show the average accuracy $\overline{Acc}$ and the pooled accuracy $\underline{Acc}$ as a function of the $alpha$ parameter used: no correction, Bonferroni correction, and stricter than Bonferroni by one and two orders of magnitude. The accuracy of MTR and TR improves as they become more conservative but never reaches the one by FTR even for the stricter thresholds of $\alpha_{MTR} = 0.0002$ and $\alpha_{TR} = 0.0001$.

Sensitivity to t: The results are also not sensitive to the particular significance level t used to define accuracy. Figure 8 graphs $Acc_i^R(t)$ over $t = [0, 0.05]$ for two typical data sets as well as $\underline{Acc}(t)$ and $\overline{Acc}(t)$. The situation is similar and consistent across all data sets considered, which are shown in Appendix A. The lines of the Full Testing Rule rise sharply, which indicates that the p-values of its predictions are concentrated close to zero.

Explaining the difference of FTR and MTR: Asymptotically and when the data distribution is faithful to a MAG, the FTR and the MTR rules are both sound (100% accurate). However, when the distribution is not faithful, the performance difference could become large because FTR tests for faithfulness violations as much as possible in an effort to avoid false predictions. This may explain the large differences in accuracies observed in the Infant Mortality, Gisette, Hiva, Breast-Cancer, and Lymphoma data sets. When the distribution is faithful, but the sample is finite, we expect some but small differences. For example when MTR falsely determines that $X \not\perp\!\!\!\perp Y|\emptyset$ due to a false positive test, the FTR rule still has a chance to avoid an incorrect prediction by additionally testing $X \not\perp\!\!\!\perp Y|W$. To support this theoretical analysis we perform experiments with simulated data where the network structure is known. Specifically, we employ the structure of the ALARM (Beinlich et al., 1989), INSURANCE (Binder et al., 1997) and HAILFINDER (Abramson et al., 1996) Bayesian Networks. We sample 20 continuous and 20 discrete pairs of data sets D_1 and D_2 from distributions faithful to the network structure using different randomly chosen parameterizations for the continuous case, and the original network parameters for the discrete case. We do the same for

Data Set	FTR$_{0.05}$	MTR$_{0.02}$	TR$_{0.01}$	Random Guess
Covtype	1.00	1.00	0.91**	0.83**
Read	-	1.00	0.97	0.82
Infant Mortality	0.95	0.64**	0.36**	0.11♠
Compactiv	1.00	0.98	0.96*	0.93**
Gisette	0.95	0.71♠	0.59♠	0.14♠
hiva	0.94	0.61♠	0.44♠	0.30♠
Breast-Cancer	0.84	0.49♠	0.34♠	0.20♠
Lymphoma	0.82	0.57♠	0.39♠	0.23♠
Wine	1.00	0.85	0.81	0.80
Insurance-C	0.97	0.75♠	0.66♠	0.37♠
Insurance-N	0.97	0.94*	0.86**	0.34♠
p53	0.97	0.87♠	0.71♠	0.54♠
Ovarian	0.99	0.98♠	0.95♠	0.91♠
C&C	0.96	0.88♠	0.80♠	0.77♠
ACPJ	-	0.26	0.07	0.02
Bibtex	1.00	0.68	0.31	0.12**
Delicious	1.00	0.87♠	0.68♠	0.23♠
Dexter	-	0.50	0.05	0.02
Nova	-	0.08	0.06	0.03
Ohsumed	-	0.14	0.05	0.02
$\overline{ACC^R}$	0.96	0.69**	0.55**	0.39**
$\underline{\underline{ACC^R}}$	0.98	0.88♠	0.74♠	0.16♠

Table 3: $ACC_i^R(t)$ at $t = 0.05$ with "Bonferroni" correction for rules FTR, MTR, TR and Random Guess. Marks *, **, and ♠ denote a statistically significant difference from FTR at the levels of $0.05, 0.01$, and machine-epsilon respectively.

sample sizes 100, 500, 1000. Subsequently, we apply the FTR and MTR rules with $\alpha_{FTR} = 0.05$ and $\alpha_{MTR} = 0.02$ (Bonferroni adjusted) on each pair of D_1 and D_2 and all possible quadruples of variables. The true accuracy is not computed on a test data set D_t but on the known graph instead by checking whether Y and Z are d-connected given X and W. The mean true accuracies over all samplings are reported in Figure 10. The difference in performance on the faithful, simulated data is usually below 5%. In contrast, the largest difference in performance on the real data sets is over 35% (Breast-Cancer), while the difference of the pooled accuracies is 10%. Thus, violations of faithfulness seem to be the most probable explanation for the large difference in accuracy on the real data.

6.4 Summary, Interpretation, and Conclusions

We now comment and interpret the results of this section:

- Notice that even if all predicted pairs are truly correlated, the accuracy may not reach 100% due to the presence of Type II errors (false negatives) *in the test set*.

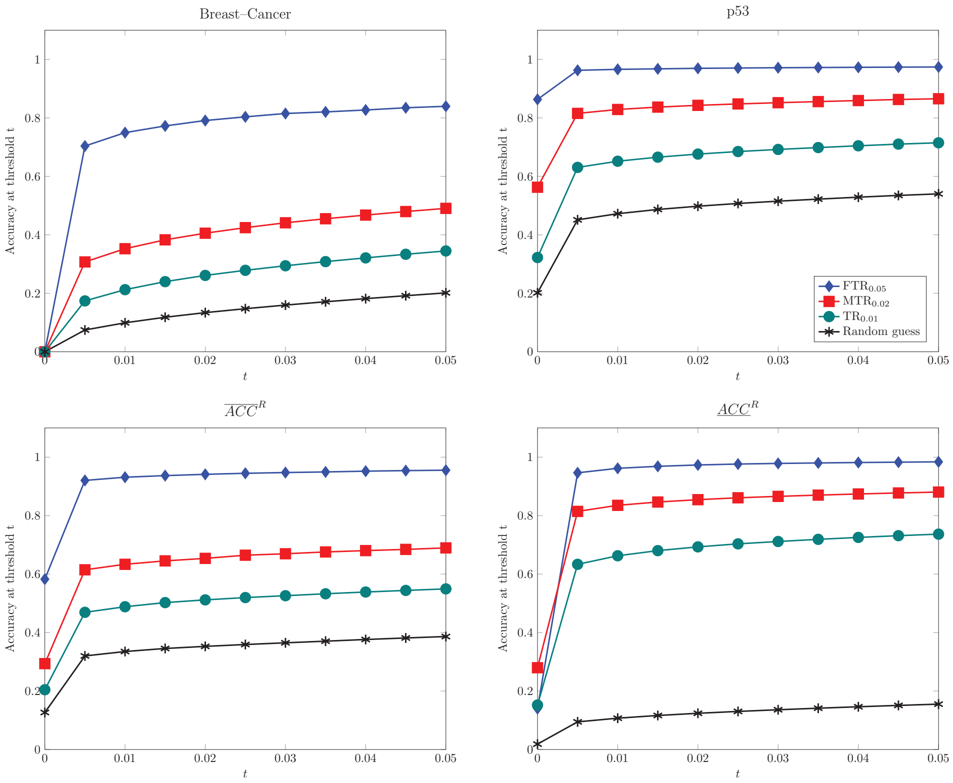


Figure 8: Accuracies $Acc_i^R(t)$ as a function of threshold t for two typical data sets along with $\overline{ACC}^R(t)$ and $\underline{ACC}^R(t)$. The remaining data sets are plot in Appendix A Section A.3. Predicted dependencies have p-values concentrated close to zero. The performance differences are insensitive to the threshold t in the performance definition.

- The FTR rule performs the test for the X-W association independently in both data sets. Given that the data in our experiments come from exactly the same distribution, they could be pooled together to perform a single test; alternatively, if this is not appropriate, the p-values of the tests could be combined to produce a single p-value (Tillman, 2009; Tsamardinos and Borboudakis, 2010).

- The results show that *the Full-Testing Rule accurately predicts the presence of dependencies*, statistically significantly better than random predictions, across all data sets, regardless of the type of data or the idiosyncracies of a domain. The rule is successful in gene-expression data, mass-spectra data measuring proteins, clinical data, images and others. The accuracy of predictions is robustly always above 0.80 and over all predictions it is 0.96; the difference with random predictions is of course more striking in data sets where the percentage of correlations (prior probability) is relatively small, as there is more room for improvement.

- *The Full-Testing Rule is noticeably more accurate than the Minimal-Testing Rule*, due to testing whether the Faithfulness Condition holds in the induced PAGs. The result is important considering that most constraint-based algorithms assume the Faithfulness Condition to induce models, *but do not check whether the induced model is Faithful*. These results indicate that when the latter is not the case, the model (and its predictions) may not be reliable. On the other hand, the FTR rule is also noticeably more conservative: the number of predictions it makes is significantly lower than the one made by MTR. In some data sets (e.g., Compactiv, Insurance-N, and Ovarian) by using the MTR vs. the FTR one sacrifices a small percentage of accuracy (less than 3% in these cases) to gain one order of magnitude more predictions. However, caution should be exercised because in certain data sets MTR is over 35% less accurate than FTR.

- *The Full-Testing Rule is more accurate than the Transitivity Rule*. Thus, the performance of the Full-Testing Rule cannot be attributed to simply performing a super-set of the tests performed by the Transitivity Rule.

- *Predictions are the norm case and not occur in contrived or rare cases only.* Even though there were few or no predictions for a couple of data sets, there are typically hundreds or thousands of predictions for each data set. This is the case despite the fact that we are only looking for a special-case structure and the search for these structures is limited within groups of 50 variables for the larger data sets. The results are consistent with the ones in Triantafillou et al. (2010), where larger structures were induced from simulated data.

- *FTR makes almost no predictions in the text data*:[3] this actually makes sense and is probably evidence for the validity of the method: it is semantically hard to interpret the presence of a word "causing" another word to be present.[4]

- FTR is an opportunistic algorithm that sacrifices completeness to increase accuracy, as well as improve computational efficiency and scalability. General algorithms for co-analyzing data over overlapping variable sets, such as ION (Tillman et al., 2008), IOD (Tillman and Spirtes, 2011) and cSAT (Triantafillou et al., 2010) could presumably make more predictions, and more general types of predictions (e.g., also predict independencies). However, their computational and learning performance on a wide range of domains and high-dimensional data sets is still an open question and an interesting future direction to pursue.

7. Predicting the Presence of Conditional Dependencies

The FTR and the MTR not only predict the presence of the dependency $Y \not\perp\!\!\!\perp Z|\emptyset$ given two data sets on $\mathbf{O}_1 = \{X,Y,W\}$ and $\mathbf{O}_2 = \{X,Z,W\}$; the rules also predict that either $X \circ\!-\!\circ Y \circ\!-\!\circ Z \circ\!-\!\circ W$ or $X \circ\!-\!\circ Z \circ\!-\!\circ Y \circ\!-\!\circ W$ is the model that generated both data sets (see Algorithms 1 and 2). Both of these models also imply the following dependencies:

$$Y \not\perp\!\!\!\perp Z|X,$$

[3] The only predictions in text data are in Bibtex (1 prediction) and in Delicious (856), which are the only text data sets that are actually not purely bag-of-words data sets but include variables corresponding to tags. 66% of the predictions made in Delicious involves tag variables, as well as the single prediction in Bibtex.

[4] However, causality between words is still conceivable in our opinion: deciding to include a word in a document may change a latent variable corresponding to a mental state of the author, which in turn causes her to include some other word.

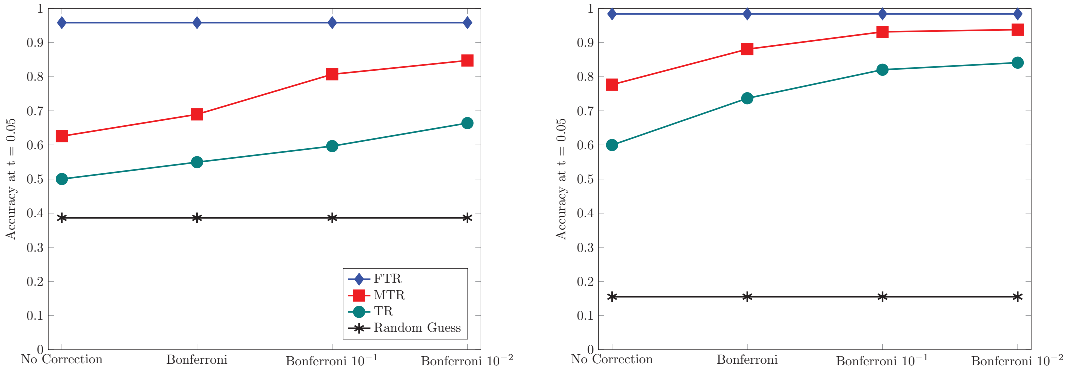


Figure 9: Average accuracy $\overline{Acc}(0.05)$ (left) and pooled accuracy $\underline{Acc}(0.05)$ (right) for each rule as a function of α thresholds used: $\alpha_{MTR} \in \{0.05, 0.02, 0.002, 0.0002\}$ and $\alpha_{TR} \in \{0.05, 0.01, 0.001, 0.0001\}$ corresponding to no correction, Bonferroni correction, and stricter than Bonferroni by one and two orders of magnitude respectively. FTR's performance is higher even when MTR and TR become quite conservative.

$$Y \not\perp\!\!\!\perp Z | W,$$

$$Y \not\perp\!\!\!\perp Z | \{X, W\}.$$

In other words, the rules predict that the dependency between Y and Z is not mediated by either X or W inclusively. To test whether all these predictions hold simultaneously at threshold t we compute:

$$p^* = \max_{\mathbf{S} \subseteq \{X,W\}} p_{Y \perp\!\!\!\perp Z | \mathbf{S}}$$

and test whether $p^* \leq t$. The above dependencies are all the dependencies that are implied by the model but not tested by the FTR given that it has no access to the joint distribution of Y and Z. Note that we forgo providing a value for p^* when any of the conditional dependencies can not be calculated, that is, when there are not enough samples to achieve large enough power, see Tsamardinos and Borboudakis (2010). The accuracy of the predictions for all dependencies in the model, named Structural Accuracy because it scores all the dependencies implied by the structure of model, is defined in a similar fashion to Acc (Definition 11) but based on p^* instead of p:

$$SAcc_i^R(t) = \#\{p^* <= t, p \in M_i^R\} / |M_i^R|.$$

The $SAcc$ for each FTR, MTR (with "Bonferroni" correction) and randomly selected quadruples is shown in Figure 7.1; the remaining data sets are shown in Appendix A. There is no line for the TR as it concerns triplets of variables and makes no predictions about conditional dependencies. Both FTR and MTR have maximum p-values p^* concentrated around zero. The curves do not rise as sharp as those in Figure 8 since the p^* values are always larger than the corresponding $p_{Y \perp\!\!\!\perp Z|\emptyset}$. We also calculate the accuracy at $t = 0.05$ for all data sets (see Table 9 in Appendix A Section A.2). The results closely resemble the ones reported in Table 3, with FTR always outperforming random guess. FTR outperforms MTR on most data sets (and hence $\overline{SACC}^{FTR} > \overline{SACC}^{MTR}$; however, over all predictions their performance is quite similar.

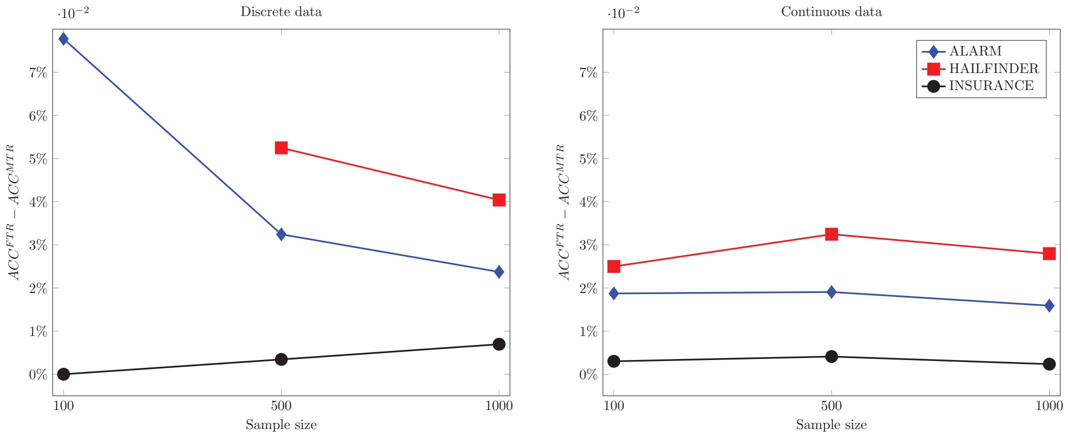


Figure 10: Difference between ACC^{FTR} and ACC^{MTR} for discrete (left) and continuous (right) simulated data sets. Results calculated using the "Bonferroni" correction (i.e., $\text{FTR}_{0.05}$ and $\text{MTR}_{0.02}$). The difference between FTR and MTR is larger than 5% only in two cases with low sample size (ALARM and HAILFINDER networks); however, the difference steeply decreases as the sample size increases. No prediction was made for HAILFINDER with discrete data and 100 samples. The difference between FTR and MTR on faithful data is relatively small.

7.1 Summary, Interpretation, and Conclusions

The results show that both the FTR and MTR rules correctly predict all the dependencies (conditional and unconditional) implied by the models involving the two variables never measured together. These results provide evidence that these rules often correctly identify the data generating structure.

8. Predicting the Strength of Dependencies

In this section, we present and evaluate ideas that turn the qualitative predictions of FTR to quantitative predictions. Specifically, for Example 1 we show *how to predict the strength of dependence* in addition to its existence. In addition to the Faithfulness Condition, we assume that when the FTR applies on quadruple $\{X,Y,Z,W\}$, all dependencies are linear with independent and normally distributed error terms. However, the results of these section could possibly be generalized to more relaxed settings, for example, when some of the error terms are non-Gaussian (Shimizu et al., 2006, 2011). When the Full-Testing Rule applies, we can safely assume the true structure is one of the MAGs shown in Figure 5. Given linear relationships among the variables, we can treat these MAGs as linear Path Diagrams (Richardson and Spirtes, 2002). We also consider normalized versions of the variables with zero mean and standard deviation of one. Let us consider one of the possible MAGs:

$$M_1 : X \xleftarrow{\rho_{XY}} Y \xrightarrow{\rho_{YZ}} Z \xrightarrow{\rho_{ZW}} W$$

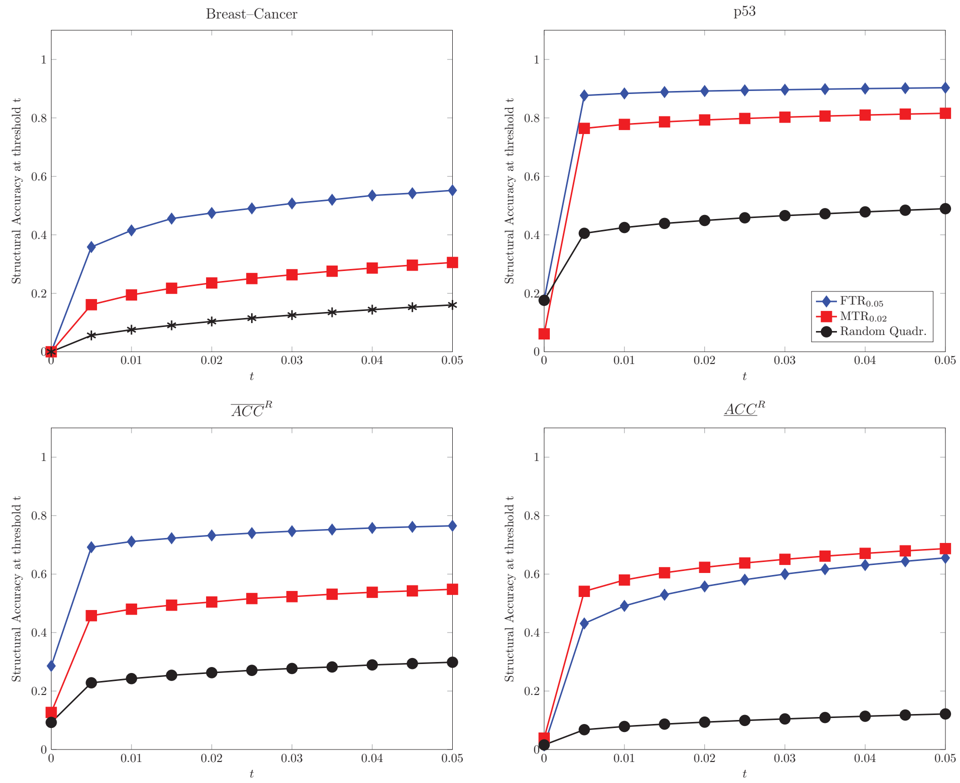


Figure 11: Structural Accuracies $SAcc_i^R(t)$ as a function of threshold t for two typical data sets along with $\overline{SACC}^R(t)$ and $\underline{SACC}^R(t)$. The remaining data sets are plot in Appendix A Section A.2. FTR outperforms MTR on most of the data sets, and thus $\overline{SACC}^{FTR}(t) > \overline{SACC}^{MTR}(t)$. However, since MTR ouperforms FTR on few data sets with a large number of predictions and so $\underline{SACC}^{MTR}(t)$ is slightly better than $\underline{SACC}^{FTR}(t)$ for $t <= 0.05$.

where ρ_{XY} is the *regression coefficient* of regressing X on Y, that is,

$$X = \rho_{XY} Y + \varepsilon$$

and ε is the error term. Since we have standardized the variables, and since the above equation is simple linear regression, ρ_{XY} coincides with the Pearson linear *correlation* between variables X and Y. Thus, there is no need to distinguish the two.[5] Now notice that in all MAGs in Figure 5 there are no colliders. Thus, as in M_1 above, all regressions are simple regressions and all standardized regression coefficients coincide with their respective correlation coefficients, and so, for the rest of the section we will not differentiate between the two.

The rules of path analysis (Wright, 1934) dictate that the correlation between two variables, for example, ρ_{XY} equals the sum of the contribution of every d-connecting path (conditioned on the

[5] If Y was a collider then it would have been regressed on multiple variables; in this case ρ_{XY} should be the partial regression coefficient which in general does not coincide with the partial correlation coefficient, even for standardized variables.

empty set); the contribution of each path is the product of the correlations on its edges. For M_1 the above rule implies (among others):

$$\rho_{XZ} = \rho_{XY} \times \rho_{YZ}$$

because from X to Z there is a single path going through Y. Recall that the 14 consistent MAGs are represented by the following PAGs:

$$P_1 : X \circ\!-\!\circ Y \circ\!-\!\circ Z \circ\!-\!\circ W$$

and

$$P_2 : X \circ\!-\!\circ Z \circ\!-\!\circ Y \circ\!-\!\circ W.$$

All MAGs consistent with P_1 entail the same constraints on the coefficients using path analysis; similarly all MAGs consistent with P_2.[6] Specifically, if P_1 is the true structure we get the constraints

$$\rho_{XZ} = \rho_{XY} \times \rho_{YZ}, \tag{1}$$

$$\rho_{YW} = \rho_{YZ} \times \rho_{ZW}. \tag{2}$$

On the other hand, if P_2 is the true structure we obtain:

$$\rho_{XY} = \rho_{XZ} \times \rho_{YZ}, \tag{3}$$

$$\rho_{ZW} = \rho_{YZ} \times \rho_{YW}. \tag{4}$$

We use ρ, $\hat{r}$, *and* r *to denote actual, predicted, and sample correlations, respectively.* The quantities that we observe are the *sample correlation coefficients*, denoted by r, for the pairs of variables measured together. Thus, we can compute the quantities $r_{XY}, r_{XZ}, r_{YW}, r_{ZW}$ from the data and we would like to predict ρ_{YZ} without available data. From Equations 1, 2, 3, 4 above we obtain four possible estimators:

$$\text{If } P_1 \text{ is true } : \hat{r}^1_{YZ} \approx \frac{r_{XZ}}{r_{XY}} \text{ from Equation 1 and } \hat{r}^2_{YZ} \approx \frac{r_{YW}}{r_{ZW}} \text{ from Equation 2,} \tag{5}$$

$$\text{if } P_2 \text{ is true } : \hat{r}^3_{YZ} \approx \frac{r_{XY}}{r_{XZ}} \text{ from Equation 3 and } \hat{r}^4_{YZ} \approx \frac{r_{ZW}}{r_{YW}} \text{ from Equation 4} \tag{6}$$

where the superscripts correspond to the equation used to produce the estimate. Notice that, each possible PAG provides two equations to predict ρ_{YZ}, that is, the parameter is overidentified. Also, the following important relation holds between the estimators:

$$\hat{r}^1_{YZ} = \frac{1}{\hat{r}^3_{YZ}} \text{ and } \hat{r}^2_{YZ} = \frac{1}{\hat{r}^4_{YZ}}.$$

This observation allows us to distinguish between PAGs P_1 and P_2: if $\hat{r}^1_{YZ}, \hat{r}^2_{YZ} \in [-1,+1]$, then their reciprocals $\hat{r}^3_{YZ}, \hat{r}^4_{YZ} \notin [-1,+1]$ and so, they are not valid estimates for a correlation. Thus, we can infer that P_1 is the true structure and employ only $\hat{r}^1_{YZ}, \hat{r}^2_{YZ}$ for estimation. Otherwise, the reverse holds $\hat{r}^3_{YZ}, \hat{r}^4_{YZ} \in [-1,+1]$, P_2 is the true structure and only $\hat{r}^3_{YZ}, \hat{r}^4_{YZ}$ should be used for estimation.

[6] In general, the consistent MAGs may disagree on the unknown correlations. In this case, these parameters may not identifiable. However, one could analyze all possible MAGs to provide bounds on the unidentifiable quantities in a similar fashion to Balke and Pearl (1997) and Maathuis et al. (2009).

Due to sampling errors it is plausible that we obtain conflicting information: $\hat{r}^1_{YZ} \in [-1,+1]$ but $\hat{r}^2_{YZ} \notin [-1,+1]$ (and so $\hat{r}^3_{YZ} \notin [-1,+1]$ and $\hat{r}^2_{YZ} \in [-1,+1]$). In that case, we forgo making any predictions.

The ramifications of the above analysis are important. In the case where all variables are jointly measured, the distribution is Faithful, the relations are linear and the error terms follow Gaussian distributions, the set of statistically indistinguishable causal graphs is determined completely by the independence model and not by the parameterization of the distribution. However, in the case of incomplete data, where some variable sets are not jointly observed, the set of indistinguishable models also depends on the parameters of the distribution, even for linear relations and Gaussian error terms. In our scenario, by analyzing the estimable parameters we can further narrow down the set of equivalent consistent MAGs.

At this point in our analysis, we are left with two valid estimators, either $\hat{r}^1, \hat{r}^2$ or $\hat{r}^3, \hat{r}^4$. All estimators are computed as ratios. We report the mean of the two valid estimators as the predicted $\hat{r}_{YZ}$ for a more robust estimation. The above procedure is formalized in Algorithm 4, named FTR-S.

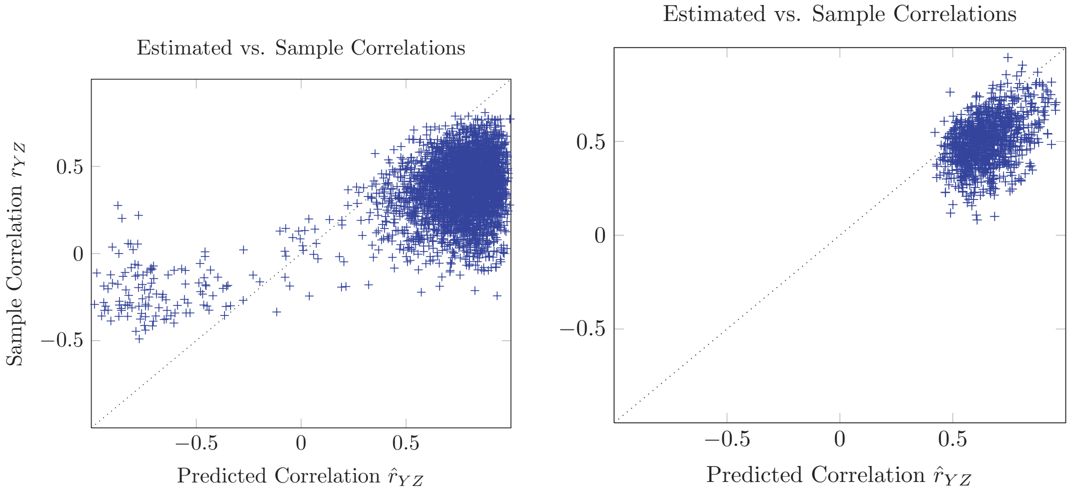
Algorithm 4: Predict Dependency Strength(**FTR-S**)

```
Input: Data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{X,Y,W\}$ and $\{X,Z,W\}$, respectively
if Full-Testing Rule($\mathcal{D}_1$, $\mathcal{D}_2$) does not apply then return;
In $\mathcal{D}_1$ compute $r_{XY}, r_{YW}$;
In $\mathcal{D}_2$ compute $r_{XZ}, r_{ZW}$;
$\hat{r}^1 \leftarrow \frac{r_{XZ}}{r_{XY}}$;
$\hat{r}^2 \leftarrow \frac{r_{YW}}{r_{ZW}}$;
$\hat{r}^3 \leftarrow \frac{r_{XY}}{r_{XZ}}$;
$\hat{r}^4 \leftarrow \frac{r_{ZW}}{r_{YW}}$;
if $\hat{r}^1, \hat{r}^2 \in [-1,1]$ then
    Predict $X \circ - \circ Y \circ - \circ Z \circ - \circ W$;
    Predict correlation $\hat{r}_{YZ} = \frac{1}{2}(\hat{r}^1 + \hat{r}^2)$;
end
else if $\hat{r}^3, \hat{r}^4 \in [-1,1]$ then
    Predict $X \circ - \circ Z \circ - \circ Y \circ - \circ W$;
    Predict correlation $\hat{r}_{YZ} = \frac{1}{2}(\hat{r}^3 + \hat{r}^4)$;
end
else
    Make no prediction
end
```

8.1 Empirical Evaluation of the Predictions of Correlation Strength

As in Section 6, we partition each data set with continuous variables to three data sets $\mathcal{D}_1$, $\mathcal{D}_2$, and a test set $\mathcal{D}_t$. We then apply Algorithm 4 and predict the strength of correlation $\hat{r}_{YZ}$ for various pairs of variables; we compare the predictions with the sample correlation r_{YZ} as estimated in $\mathcal{D}_t$. The results for one representative data set (Lymphoma) are shown in Figure 12(a): there is an apparent trend to overestimate the absolute value of the sample correlation.

There are several possible explanations for the bias of the method, including violations of normality, linearity, faithfulness, and even the known bias in the estimation of sample correlation coefficients (Zimmerman et al., 2003) that are used for making the predictions in Algorithm 4. In order to pinpoint the culprit, we generated data where all assumptions hold from the model M_1 shown in the beginning of this section, where we set the correlations $\rho_{XY}, \rho_{YZ}, \rho_{ZW}$ and the noise terms are independently and normally distributed. We used the entire spectrum of positive correlation coefficients for all three correlations to examine how the bias varies as a function of these correlations. We generated 1000 data sets of different sample sizes of 50, 70 and 100 samples. We then used Equation 1 to estimate r_{YZ} in each experiment. *This set of experiments revealed no significant bias for any of the experimental settings* (results are not shown for brevity).



(a) Lymphoma Data Set (b) Simulated Data where FTR Rule Applies

Figure 12: (a) Predicted ($\hat{r}_{YZ}$) vs sample (r_{YZ}) correlation for the Lymphoma data set. There is an obvious trend to over-estimate the correlation in absolute value. (b) Simulated results from model $\mathcal{M}_1$ when ρ_{XZ} and ρ_{YW} are lower than 0.4 and observed correlations *are found significant* (FTR applies). The FTR constraint that the observed correlations are significant reproduces a similar behavior in the simulated data, explaining the bias.

We next tested whether the bias is an artifact of the filtering by the FTR at Line 1 of the FTR-S algorithm. We re-run this procedure, but this time we kept only the predicted correlations that passed the FTR. By comparing Figure 12(a) produced on real data, and 12(b) on simulated data, we observe a similar behavior, indicating that FTR filtering seems a reasonable explanation for the bias.

An explanation of this phenomenon now follows. Suppose $M_1 : X \xleftrightarrow{\rho_{XY}} Y \xrightarrow{\rho_{YZ}} Z \xrightarrow{\rho_{ZW}} W$ is the data generating MAG. We expect that $\hat{r}_{YZ} = \frac{r_{XZ}}{r_{XY}}$ (the equality $\hat{r}_{YZ} = \frac{r_{YW}}{r_{ZW}}$ also holds but we ignore it to simplify the discussion). When sample correlations among $\{X,Y,Z,W\}$ pass the FTR, this means that both r_{XZ} and r_{XY} are above a cut-off threshold, as given by the Fisher test. For example, for a data set with 70 samples, two variables are considered dependent ($\rho \neq 0$) if their sample correlation is more that 0.2391 (in absolute value), whereas for a data set with 50 samples, this threshold is 0.2852.

Filtering with the Fisher test introduces a bias in the estimation of r. The bias of the estimation without filtering, r_u is $B_{r_u} = E[r_u - \rho] = \overline{r_u} - \rho$, while the bias of the estimation with filtering r_f is $B_{r_f} = E[r_f - \rho] = \overline{r_f} - \rho$, where $|r_f| \geq t$. The threshold t, as mentioned above, is the threshold determined by the Fisher test and depends on sample size. *The lower the sample size, the higher the threshold t, and so the higher the introduced bias B_{r_f}. In addition, the lower the $|\rho|$ the higher the bias B_{r_f}.*

Figure 13 illustrates these points pictorially. In this example, the distribution of the sample correlation r of two variables for sample size 70 when the true correlation is $\rho \in \{0.2, 0.4\}$. For unfiltered estimations, the bias is B_{r_u} is 0.0052 and -0.0011 for ρ equal to 0.2 and 0.4 respectively, whereas for filtered estimations the corresponding values B_{r_f} are 0.1187 and 0.0127.

Going back to the prediction $\hat{r}_{YZ} = \frac{r_{XZ}}{r_{XY}}$ notice that the numerator is always lower (in absolute value) than the denominator. Therefore, when filtered, it is, on average more overestimated than the denominator. This implies that, on average, the fraction leads to overestimating the absolute value of ρ_{YZ}. The lower the values of $|r_{XZ}|$ and $|r_{XY}|$, the larger we expect this bias to be. The situation is similar for all fractions involved in Equations 5 and 6. This hypothesis is confirmed in the data as illustrated in Figure 14 where the predictions are grouped by the mean absolute values of the denominators used in their computation.

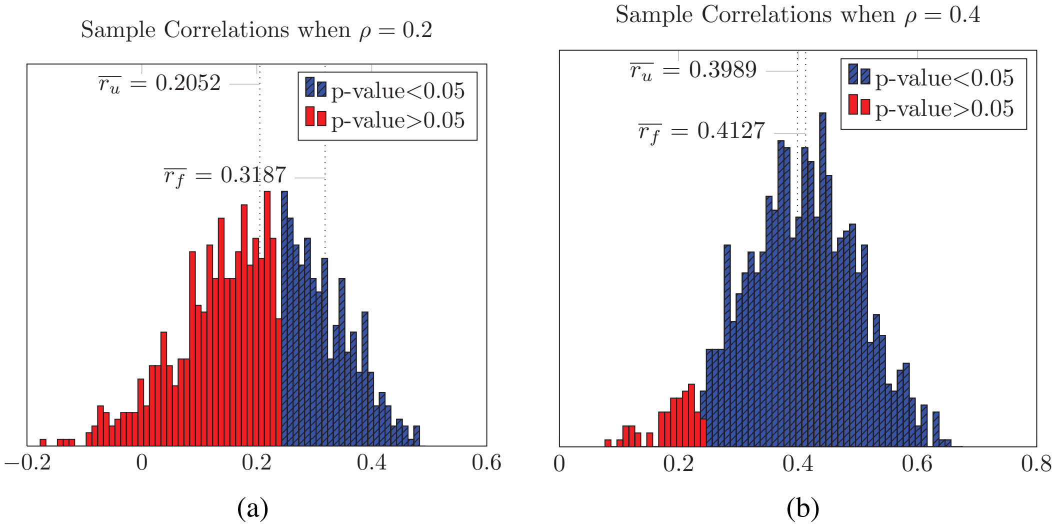


Figure 13: Histograms of the sample correlations for (a) $\rho = 0.2$ and (b) $\rho = 0.4$ for sample size 70. Red bars correspond to cases where the Fisher test returns a p-value > 0.05, whereas blue bars correspond to p-values < 0.05. The dashed lines indicate the mean sample correlation for filtered and unfiltered correlations. The lower the ρ , the more overestimated the sample correlations that pass the Fisher test, therefore the difference between the two means is larger.

The bias should be a function of sample size, the absolute value of the correlations employed for its computation, and the significance thresholds of the FTR rule. However, a full theoretical treatment of the bias is out of the scope of the paper. In the experiments that follow we remove the linear trend to over-estimate (*calibrate*) by regressing the sample correlations r_{YZ} on the predicted $\hat{r}_{YZ}$: the final calibrated prediction is $s \times \hat{r}_{YZ} + i$. For each data set the intercept i and slope s of the regression are estimated by training on the remaining data sets (leave-one-data-set-out validation). The effect of this calibration is shown in Figure 15. To avoid repetition, the detailed set of results is presented in the comparative evaluation to statistical matching in Section 9.

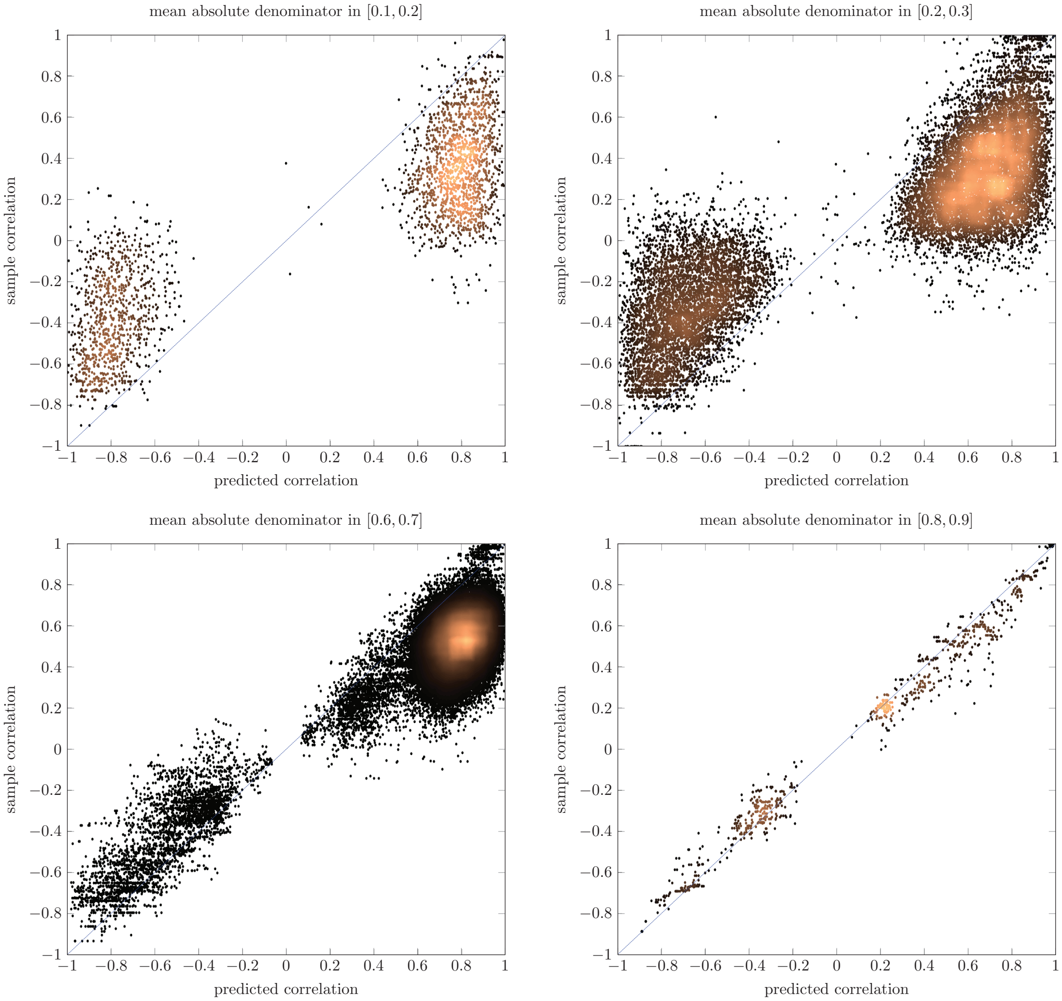
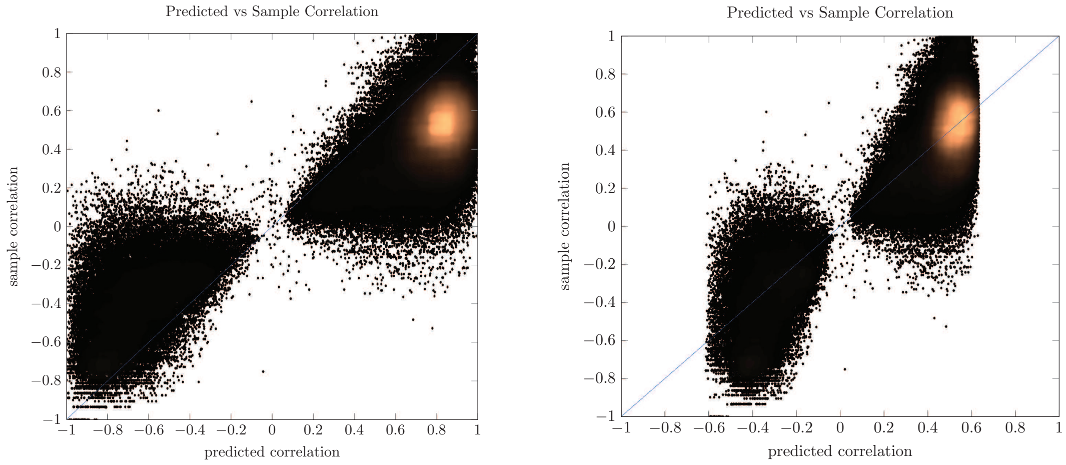


Figure 14: Predicted vs sample correlations over all data sets, grouped by the mean absolute values of the denominators used in their computation: predictions computed based on large correlations have reduced bias. Red regions correspond to higher density areas.

8.2 Summary, Interpretation, and Conclusions

We now comment and interpret the results of this section:



(a) (b)

Figure 15: Predicted vs sample correlations on all data sets (a) before, and (b) after calibration.

- FTR coupled with parametric assumptions can be used to predict the strength of dependency (correlation), providing quantitative predictions. This is equivalent to constructing a prediction model for variables not jointly observed.
- In the case of incomplete data, where some variable sets are not jointly observed, the set of indistinguishable models also depends on the parameters of the distribution, even for linear relations and Gaussian error terms. In contrast, in the case where all variables are jointly measured and the distribution is Faithful the set of statistically indistinguishable causal graphs is completely determined by the independence model (again, also assuming linearity and Gaussian error terms).
- In our simple scenario, *given the correct structure*, path analysis of the induced MAGs provides easy solutions for predicting the strength of dependence. However, *searching for the correct MAG models* by applying the FTR incurs bias on the predictions that should be taken into account.

9. Comparison Against Statistical Matching

Statistical Matching (D’Orazio et al., 2006) is a integrative analysis procedure for data sets defined over overlapping variable sets. Statistical matching addresses two main tasks named the *micro approach* and *the macro approach*. The micro approach aims to impute the missing values and construct a complete synthetic file, whereas the macro approach aims to identify some characteristics of the joint probability distribution of the variables not jointly observed. Naturally, construction of the synthetic data set premises the estimation of the parameters of the joint distribution. We focus on the macro approach as it presents an alternative to the FTR and MTR.

The problem set up is as follows: variables $\mathbf{Y} \cup \mathbf{X}$ are measured in data set $\mathcal{D}_1$, while variables $\mathbf{Z} \cup \mathbf{X}$ are measured in data set $\mathcal{D}_2$. Thus $\mathbf{X}$ are the commonly measured variables. The goal is to estimate the variances and covariances of $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$. The problem cannot be solved without additional assumptions (Rubin, 1974; D’Orazio et al., 2006). Depending on nature of the assumptions, statistical matching is able to produce either intervals or point-estimates for the covariances between $\mathbf{Y}$ and $\mathbf{Z}$. The most typical assumption in the literature able to produce point estimates is the Conditional Independence Assumption: $\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}$. This is an arbitrary assumption that has been long debated. Alternatively, one can limit the shape of the distribution by imposing parametric forms, such as multivariate normality. The latter type of assumptions, for the typical distributions, do not lead to identifiable estimations, but instead provide bounds on the missing covariances. Other approaches do exist that require prior knowledge, for example, Vantaggi (2008) assumes knowledge of structural zeros and Cudeck (2000) of the structure of latent factors; such approaches however, are not directly comparable with FTR and MTR on this task. In this section we briefly present the main theory and techniques used in statistical matching, and then attempt to empirically compare against FTR.

9.1 Statistical Matching Based on the Conditional Independence Assumption

The most common assumption that allows identification of the unknown parameters is the **conditional independence assumption** (CIA): $\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}$. The conditional independence assumption is usually paired with some parametric assumption. The most common assumption for the shape of a continuous distribution of the variables involved in the model is multivariate normality. In this case, the parameters of the jpd are the mean vector and the covariance matrix. The covariance Matrix for $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ can be written as:

$$\Sigma = \begin{bmatrix} \Sigma_{\mathbf{XX}} & \Sigma_{\mathbf{XY}} & \Sigma_{\mathbf{XZ}} \\ \Sigma_{\mathbf{YX}} & \Sigma_{\mathbf{YY}} & \Sigma_{\mathbf{YZ}} \\ \Sigma_{\mathbf{ZX}} & \Sigma_{\mathbf{ZY}} & \Sigma_{\mathbf{ZZ}} \end{bmatrix}$$

where the unknown parameter is $\Sigma_{\mathbf{YZ}}$. The CIA assumption imposes that the covariance matrix of $\mathbf{Y}$ and $\mathbf{Z}$ given $\mathbf{X}$ is null, thus,

$$\Sigma_{\mathbf{YZ}} = \Sigma_{\mathbf{YX}} {\Sigma_{\mathbf{XX}}}^{-1} \Sigma_{\mathbf{XZ}}.$$

In case we have standardized variables, and $\mathbf{Y} = \{Y\}$ and $\mathbf{Z} = \{Z\}$, the covariance matrix becomes

$$\Sigma = \begin{bmatrix} \rho_{\mathbf{XX}} & \rho_{\mathbf{X}Y} & \rho_{\mathbf{X}Z} \\ \rho_{Y\mathbf{X}} & 1 & \rho_{YZ} \\ \rho_{Z\mathbf{X}} & \rho_{ZY} & 1 \end{bmatrix}$$

and so

$$\rho_{YZ} = \rho_{Y\mathbf{X}} {\rho_{\mathbf{XX}}}^{-1} \rho_{\mathbf{X}Z}.$$

This formula can be used to produce a prediction $\hat{r}_{YZ}$ for the correlation coefficient of the not commonly observed variables Y and Z. Recall that, we assume we are given a data set $\mathcal{D}_1$ on variables $\mathbf{X} \cup Y$ and a data set $\mathcal{D}_2$ on $\mathbf{X} \cup Z$. The parameters $\rho_{\mathbf{X}Y}$ and $\rho_{\mathbf{XZ}}$ can be estimated from $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively, while the parameters $\rho_{\mathbf{XX}}$ can be estimated from either or both data sets.

In an applied setting, there is usually also a preprocessing step attempting to identify a subset of the common variables to be used in the matching process. This step serves mainly computational efficiency and interpretability purposes and does not affect the asymptotic properties of the procedure. The main method suggested in D'Orazio et al. (2006) is to disregard all variables in $\mathbf{X}$ that are *independent* with both Y and Z. The details are described in Algorithm 5.

Even though the conditional independence assumption seems quite arbitrary, it is intuitively justified in certain cases. When the number of common variables is large it is unlikely that Y

Algorithm 5: Predict Correlation: Statistical Matching Rule (**SMR**)

```
Input: Data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{\mathbf{V} \cup Y\}$ and $\{\mathbf{V} \cup Z\}$, respectively
$\psi_1 \leftarrow \{V \in \mathbf{V} : V \perp\!\!\!\perp Y | \emptyset\}$ in $\mathcal{D}_1$
$\psi_2 \leftarrow \{V \in \mathbf{V} : V \perp\!\!\!\perp Z | \emptyset\}$ in $\mathcal{D}_2$
$\mathbf{X} \leftarrow \mathbf{V} \setminus (\psi_1 \cap \psi_2)$
Predict $\hat{r}_{YZ} = \hat{\Sigma}_{Y\mathbf{X}} \hat{\Sigma}_{\mathbf{XX}}^{-1} \hat{\Sigma}_{\mathbf{X}Z}$
```

provides *additional* information for Z, than what $\mathbf{X}$ already provides. In other words, we expect $\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X}$ to hold or hold approximately. Using graphical model theory one can better formalize this intuition:

Theorem 12 *Consider a Bayesian Network of maximum degree k faithful to a distribution defined over a set of variables $\mathbf{V} = \mathbf{X} \cup Y \cup Z$, $|\mathbf{V}| = N$. Then, the CIA $Y \perp\!\!\!\perp Z | \mathbf{X}$ holds if and only if $Y \notin Mb(Z)$, where $Mb(Z)$ is the Markov Boundary of Z in the context of variables $\mathbf{V}$; if Y and Z are chosen at random the probability of the CIA being violated is upper bounded by k^2/N.*

Proof In a faithful distribution over $\mathbf{V}$, each variable Y has a unique Markov Boundary $Mb(Y)$ (Pearl, 2000) that coincides with the parents, children, and parents of children (spouses) of Y in any network faithful to the distribution. It is also easy to see that $Y \in Mb(Z) \Leftrightarrow Z \in Mb(Y)$. Finally, the $Mb(Y)$ and any of its supersets d-separates Y from any other node Z. Thus, when $Z \notin Mb(Y)$, then conditioned on the remaining variables (superset of $Mb(Y)$) Y becomes d-separated and independent of Z. Thus, the CIA holds. Conversely, if $Z \in Mb(Y)$ then it is either a neighbor of Y or a spouse. If it is a neighbor it cannot be made independent of Y conditioned on any subset of the variables (Spirtes et al., 2001). If it is a spouse of Y, then conditioned on the remaining variables (which includes the common children) it is d-connected to Y and thus dependent. Thus, the CIA does not hold.

Now, the Markov Boundary of Y is a subset of the nodes that are reachable from Y within two edges. If the network has degree at most k the probability that a randomly chosen Y belongs to the Markov Boundary of Z is less than k^2/N. ■

Thus, when the sparsity remains the same, the probability of a violation of the CIA between two randomly selected variables decreases with the number of participating variables N. The theoretical results is illustrated in Figure 16 on simulated data. The figure shows the results of the statistical matching procedure described in Algorithm 5 for simulated continuous data from a network based on the ALARM Network (Beinlich et al., 1989).[7] To recreate the scenario above we generated two data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ of 1000 samples each from the distribution of the network. We then applied the statistical matching rule described in Algorithm 5 for each pair of variables, considering that the rest of the variables in the network are jointly measured in both data sets. Finally, we generated a third data set to test the predictions of the method. The pairs of variables are partitioned in two categories: pairs of variables that belong to each other's Markov Boundary, and pairs of variables that do not belong to each other's Markov Boundary. As expected, the results are poorer for the

[7] The ALARM network a well-known network with 37 variables. We used the skeleton of ALARM to simulate a conditional linear gaussian network with random parameters.

pairs of variables that belong to each other's Markov Boundary, with a mean absolute error of 0.1649 ± 0.1088, compared to a mean absolute error of 0.0326 ± 0.0271 for pairs that do not belong to each other's Markov Boundary.

In the context of Maximal Ancestral Graphs, defining the Markov Boundary is more complicated and its cardinality cannot be likewise bounded (Pellet and Elisseeff, 2008). Nevertheless, we still expect that, in a sparse network containing a large number of jointly measured variables, the probability that $Y \in Mb(Z)$ is low. We therefore expect that, when the number of common variables is large, the CIA will often hold for randomly-chosen pairs of variables that have not been observed together. If, however, the set of variables measured in common is small, we have no good reason to expect that the conditional independence assumption holds.

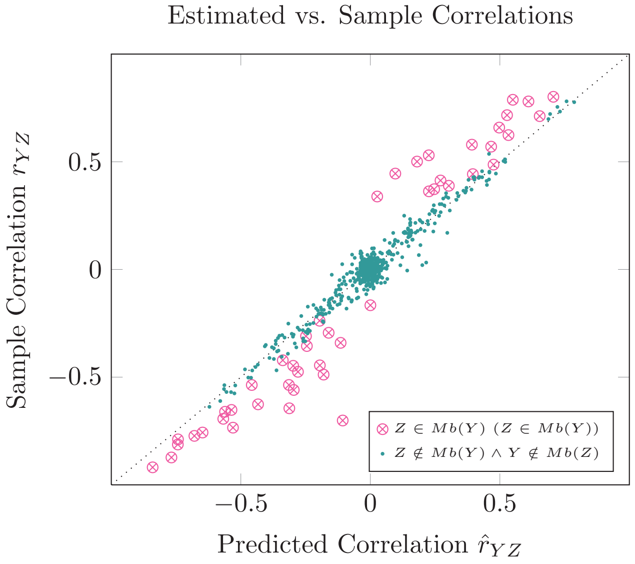


Figure 16: Predicted vs actual sample correlations using the Statistical Matching Rule for simulated data from the ALARM network. For each pair of variables, prediction is based upon the subset of the remaining 35 variables that are determined significantly correlated with either Y or Z at level 0.05 . The CIA holds when $Y \notin Mb(Z)$ in which case the mean absolute error is 0.0326 ± 0.0271; in contrast, when $Y \in Mb(Z)$ the CIA does not hold and the mean absolute error is 0.1649 ± 0.1088.

9.2 Empirical Evaluation of SMR and FTR-S

In this section, we empirically compare the SMR and FTR-S methods for predicting the correlation $\hat{r}_{YZ}$ between two variables Y and Z never jointly observed. Both SMR and FTR-S procedures provide such predictions, however, they follow different approaches that makes their comparison not straightforward:

- SMR provides a prediction for all cases. FTR-S provides a prediction given it identifies a specific structure that entails a significant correlation.

Data Sets	SMR$_G$	SMR$_Q$	FTR-S
ACPJ	445121	509000	0
Breast-Cancer	436093	356000	1005
C&C	5050	1000	70367
Compactiv	231	1000	108
Insurance-C	3486	1000	1372
Lymphoma	180074	147000	3897
Ohsumed	124505	122000	0
Ovarian	52675	43000	273456
Wine	66	495	4
p53	132299	108000	33934

Table 4: Number of predictions

- SMR can be applied to sets X with more than two commonly measured variables and get leverage from all available information. FTR-S on the other hand is applicable only when the number of common variables is two.

We applied the SMR method on all continuous data sets, simulating two scenarios. In the first scenario, SMR is applied on two data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ defined over a quadruple of variables $\{X,Y,Z,W\}$, where only X,W are jointly measured in both. The pairs of $\mathcal{D}_1$, $\mathcal{D}_2$ are simulated by considering randomly chosen variable quadruples from each variable group of each data set of Table 1; as in all experiments, $\mathcal{D}_1$ and $\mathcal{D}_2$ contain a disjoint third of the original samples. This scenario simulates a case where SMR is applied on low dimensional data; we denote it as *SMR$_Q$*. In this case, *SMR has the same information available for making predictions as FTR-S*. Since the number of possible quadruples is computationally prohibitive, we apply *SMR$_Q$* on 1000 randomly chosen quadruples from each variable group of each data set.[8] In the second scenario, SMR is applied to data sets of higher-dimensionality. Specifically, we apply SMR to all pairs of variables in the same group (see Section 6), considering the remaining 48 variables in the group as the common variables $\mathbf{X}$. We name this case *SMR$_G$*. The same leave-one-data-set-out calibration method was used for both SMR cases and FTR-S. Figures 17, 18, 19 and 20 plot the predicted vs. the sample estimates of the correlations for *SMR$_G$*, *SMR$_Q$* and FTR-S for all the continuous data sets used in the study. The figures also present the coefficient of determination R^2, the percentage of variance explained by the predictions. R^2 is also interpreted as the reduction in uncertainty obtained by using a linear function of $\hat{r}$ to predict r vs. predicting r by its expected value $E(r)$. Table 5 shows the correlation between predicted and sample estimates for all methods and data sets. Notice that R^2 is simply computed as the square of the correlation. Other metrics of performance (Mean Absolute Error and Mean Relative Absolute Error) are also presented in the Appendix A, Tables 10, 11.

[8] Notice that FTR is typically executed much more efficiently than SMR$_Q$, because of the possible pruning of the search space, for example, if X and Y are independent, there is no need to test whether the rule applies on any quadruples of the form $\langle X,Y,Z,W\rangle$. For the SMR$_Q$ rule instead, one needs to exhaustively consider all quadruples.

Data Sets	SMR$_G$	SMR$_Q$	FTR-S
ACPJ	0.05 [0.04;0.05]	0.00 [0.00;0.01]	-
Breast-Cancer	0.55 [0.55;0.55]	0.25 [0.24;0.25]	0.88 [0.87;0.90]
C&C	0.99 [0.99;0.99]	0.68 [0.65;0.71]	0.91 [0.91;0.91]
Compactiv	0.97 [0.96;0.98]	0.49 [0.44;0.54]	0.88 [0.83;0.92]
Insurance-C	0.83 [0.82;0.84]	0.47 [0.42;0.51]	0.90 [0.89;0.91]
Lymphoma	0.60 [0.60;0.60]	0.32 [0.31;0.32]	0.50 [0.47;0.52]
Ohsumed	0.02 [0.01;0.03]	0.01 [0.00;0.01]	-
Ovarian	0.62 [0.62;0.63]	0.50 [0.50;0.51]	0.14 [0.14;0.14]
Wine	0.83 [0.74;0.90]	0.58 [0.52;0.64]	0.99 [0.47;1.00]
p53	0.91 [0.91;0.91]	0.45 [0.44;0.45]	0.87 [0.87;0.87]
Mean over data sets	0.64 [0.62;0.65]	0.38 [0.35;0.40]	0.76 [0.68;0.77]
On all predictions	0.73 [0.73;0.73]	0.58 [0.57;0.58]	0.89 [0.89;0.89]

Table 5: Correlations among predicted $\hat{r}_{YZ}$ and sample-estimated r_{YZ}; the 95% confidence intervals are shown in brackets.

9.3 Summary, Interpretation, and Conclusions

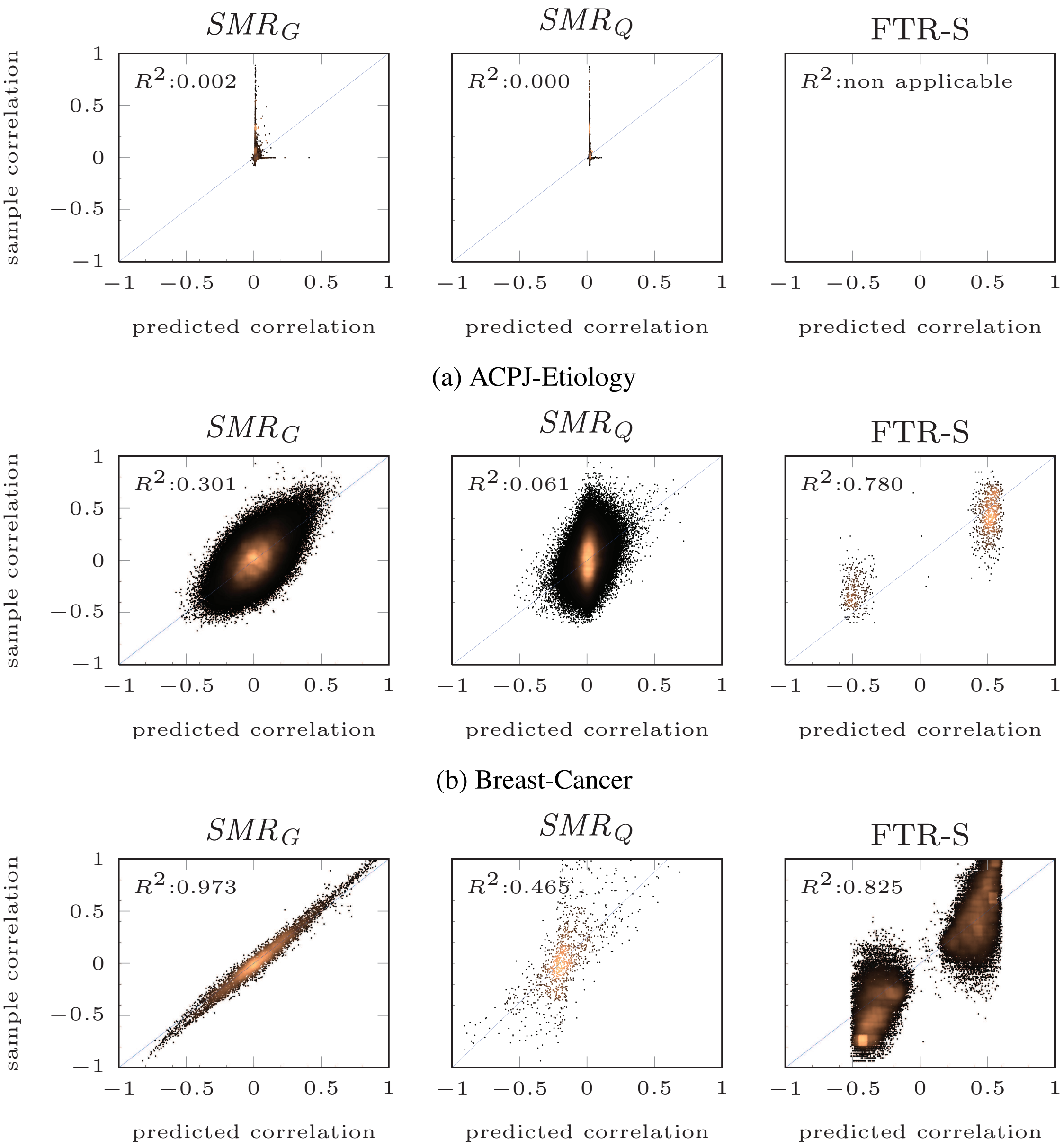
The CIA assumption is the most common assumption in statistical matching to produce point-estimates of the unknown distribution parameters. In comparison to FTR-S, we note the following:

- When predictions are based on only 2 common variables, statistical matching based on the CIA (SMR$_Q$) is unreliable in several data sets and particularly the text categorization ones: the correlation of predicted vs. sample estimates in ACPJ, Breast-Cancer, and Ohsumed is less than 0.3 (Table 4). In general, SMR tends to predict a zero correlation between the two variables Y and Z: the point-clouds in Figures 17, 18, 19 and 20 are vertically oriented around zero. While SMR gives a prediction in every case, it is too liberal in its predictions and the CIA is often violated, as expected by Theorem 12. Over all predictions, the correlation of predicted vs. sample estimates is 0.58.

- When predictions are based on larger sets of common variables statistical matching based on the CIA (SMR$_G$) is more successful. Over all predictions, the correlation of predicted vs. sample estimates is 0.73. The method still fails however, on the text data (ACPJ, Ohsumed) where the predictions are not correlated at all with the sample estimates. On the other hand, FTR-S does not make any predictions on these data sets.

- FTR-S's predictions are highly correlated with sample estimates (0.89 correlation), which is the highest correlation achieved by any of the three methods. However, we point out that these metrics are computed on different sets of predictions and their comparative interpretation is not straightforward (see Appendix A, Section A.2 for more metrics and discussion).

- FTR-S is a novel alternative to statistical matching based on the CIA. FTR-S predictions are better correlated with the sample estimates of the unknown parameters, particularly when the number of common variables is low; we thus recommend that FTR-S should be preferred than existing statistical matching alternatives making the CIA in such cases.



(a) ACPJ-Etiology

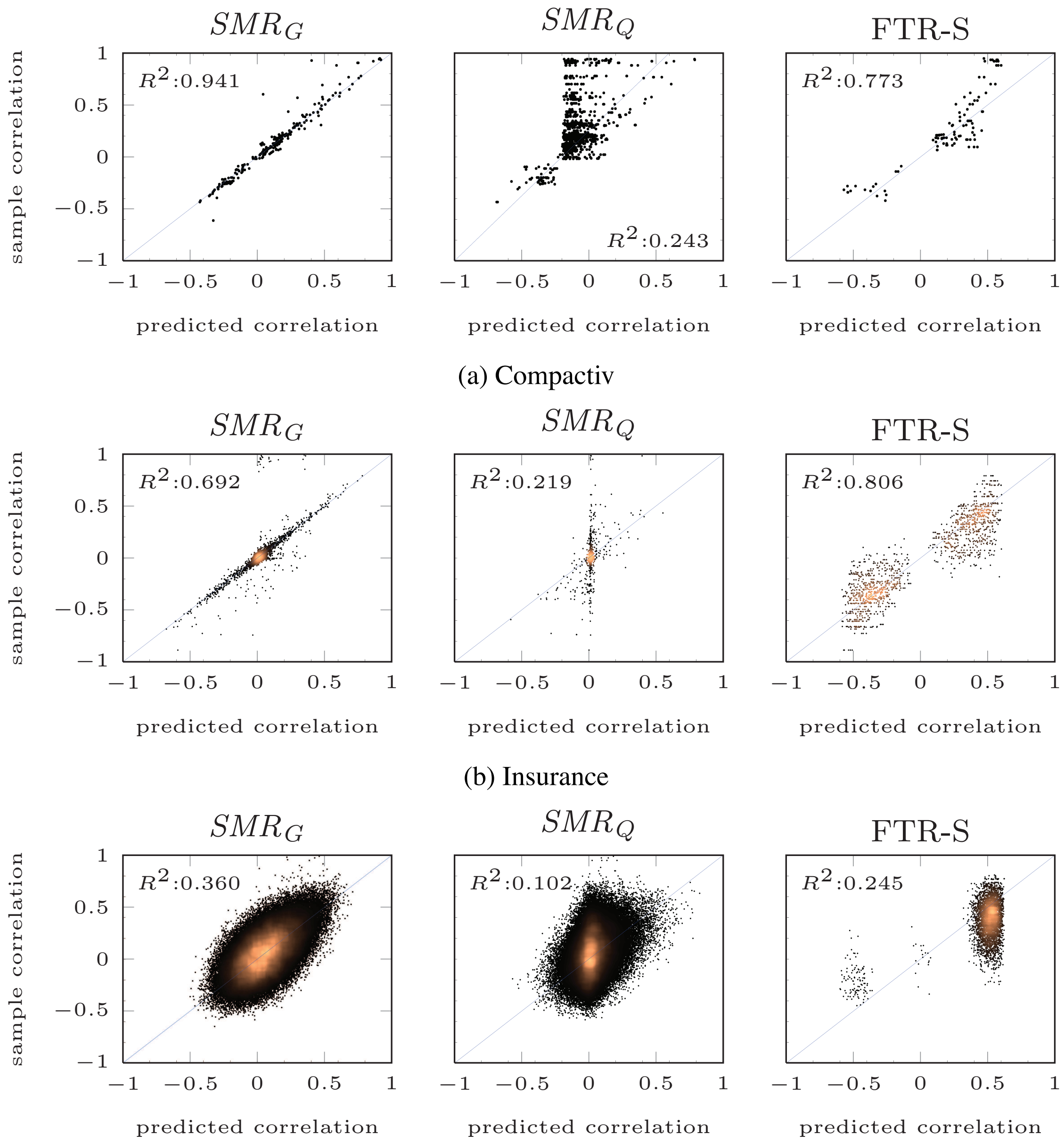
(b) Breast-Cancer

(c) C&C

Figure 17: Predicted vs Sample Correlations for SMR_Q, SMR_G, FTR-S

9.4 Statistical Matching Based on the Assumption of Multivariate Normality

The conditional independence assumption attempts to overcome the lack of joint information of the variables of interest. However, it can often be a misspecified assumption as pointed out in the literature (D'Orazio et al., 2006) and our simulated results above. An alternative approach, is to limit oneself to an assumption involving only the shape of the distribution. The most common distributional assumption adopted by statistical matching techniques for continuous variables is



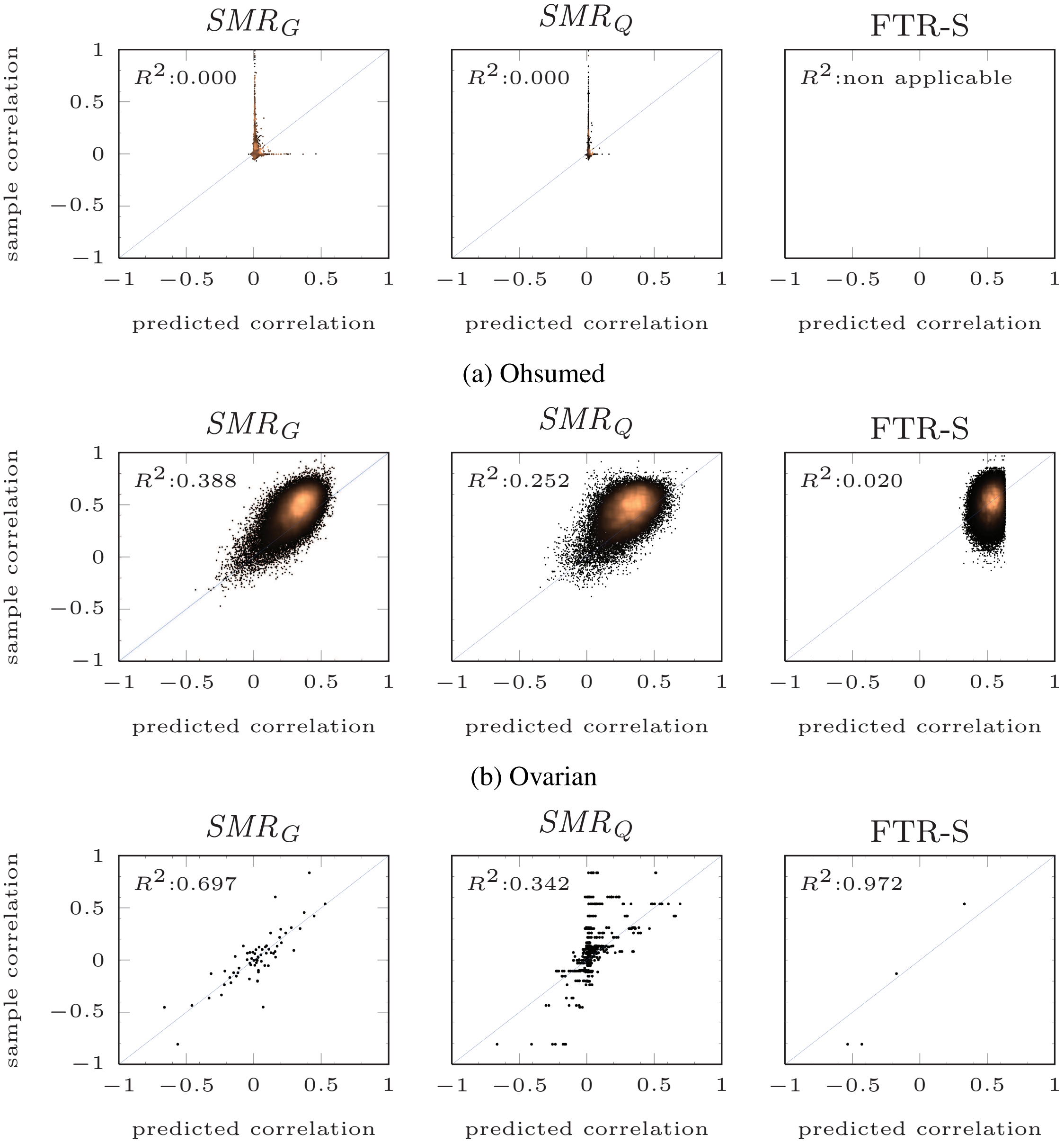
(a) Compactiv

(b) Insurance

(c) Lymphoma

Figure 18: Predicted vs Sample Correlations for SMR_Q, SMR_G, FTR-S

multivariate normality. Of course, multivariate normality alone does not allow the estimation of the parameters of the model. It does, however, impose some constraints on the parameters. These constraints stem from the positive semi-definiteness of the covariance matrix in multivariate normal distributions, thus, they naturally apply to any distribution with a positive semi-definite covariance matrix.



(a) Ohsumed

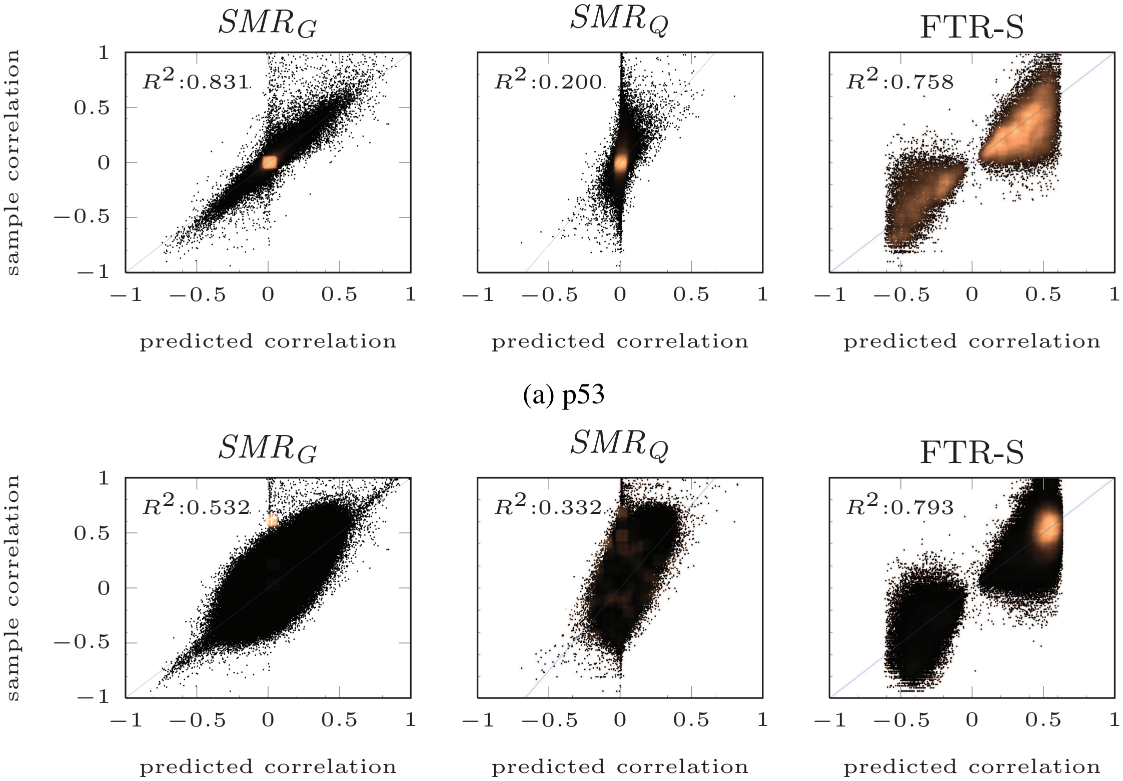
(b) Ovarian

(c) Wine

Figure 19: Predicted vs Sample Correlations for SMR_Q, SMR_G, FTR-S

Let us consider again *standardized* variables $\{\mathbf{X}, Y, Z\}$ and assume their joint is distributed as multivariate normal with correlation / covariance matrix Σ (which is symmetric)

$$\Sigma = \begin{bmatrix} \rho_{\mathbf{XX}} & \rho_{\mathbf{X}Y} & \rho_{\mathbf{X}Z} \\ \rho_{Y\mathbf{X}} & 1 & \rho_{YZ} \\ \rho_{Z\mathbf{X}} & \rho_{ZY} & 1 \end{bmatrix}.$$



(a) p53

(b) All predictions

Figure 20: Predicted vs Sample Correlations for SMR_Q, SMR_G, FTR-S

The unknown quantity in the problem is parameter ρ_{YZ}. One can start from the requirement that Σ must be positive semi-definite to prove that ρ_{YZ} must lie within the interval $C \pm \sqrt{(D)}$ (Moriarity and Scheuren, 2001), where

$$C = \sum_{i=1}^{p} \sum_{j=1}^{p} \rho_{YX_i} \times B^{i,j} \times \rho_{ZX_j}$$

and

$$D = [1 - \sum_{i=1}^{p} \sum_{j=1}^{p} \rho_{YX_i} \times B^{i,j} \times \rho_{YX_j}] \times [1 - \sum_{i=1}^{p} \sum_{j=1}^{p} \rho_{ZX_i} \times B^{i,j} \times \rho_{ZX_j}]$$

where p is the cardinality of set $\mathbf{X}$, and B is the inverse of $\rho_{\mathbf{XX}}$, and $B^{i,j}$ is B's i, j element. This constraint is equivalent stating that the partial correlation $\rho_{YZ|\mathbf{X}}$ parameter can range freely in the interval [-1, 1]. Instead, the CIA specifies that $\rho_{YZ|\mathbf{X}} = 0$, that is, the mid-point of the interval.

The formula above can be applied to quadruples of variables to produce bounds for the unknown parameter ρ_{YZ}. The usefulness of such a prediction depends, of course, on the length of the predicted interval. In case the interval does not include 0, we may also say that the method *predicts an unconditional independence for Y and Z*. This procedure is described in Algorithm 6. In practice, we apply Algorithm 6 using the sample estimates $\hat{r}$ in place of the unknown population parameters ρ. The sample estimates are the maximum likelihood ones. The uncertainty of the estimation could be considered in the computation of the intervals by considering the worst case over all correlation estimates $\hat{r}$ that belong in the 95% confidence interval of their corresponding ρ. However, in this case the algorithm would produce wider intervals and thus fewer predictions.

Algorithm 6: Predict Dependency and Its Strength: Multivariate Normality Rule (**MNR**)

```
Input: Data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ on variables $\{X,Y,W\}$ and $\{X,Z,W\}$, respectively
Compute sample correlation matrix $\Sigma$ (except unknown quantity $\rho_{YZ}$) ;
$MNI \leftarrow [C - \sqrt{(D)}, C + \sqrt{(D)}]$;
if $0 \notin MNI$ then
    Predict $Y \not\perp\!\!\!\perp Z|\emptyset$ ;
end
Predict $\hat{r}_{XY} \in MNI$
```

9.5 Empirical Evaluation and Comparison of MNR and FTR

In order to evaluate how often MNR provides a prediction, we applied Algorithm 6 on real data. Applying Algorithm 6 on all possible combinations of four variables is prohibitive. Thus, to evaluate the MNR we randomly sampled 1000 quadruples from each group of 50 variables in each data set, for all data sets with continuous variables; For the Wine data set we generated all possible 495 quadruples out of its 12 variables.

Table 6 reports MNR performances on the randomly chosen quadruples. The columns of the table present the total number of randomly chosen quadruples (1000 × the number of chunks, except for the Wine data set), the number of predictions made by MNR on these random quadruples, the accuracies Acc^{MNR} and Acc^{FTR} at threshold $t = 0.05$. We then calculate (project) the *expected* number of predictions by the MNR rule, had it been applied on all possible quadruples. The final column presents the ratio of the number of predictions by the FTR rule over the *expected* number of predictions made by the MNR rule on all possible quadruples.

First, notice that MNR, similarly to FTR, does not provide any predictions for the text data sets ACPJ and Ohsumed data sets. Second, the rule is in general, highly accurate and on par with FTR. The most important observation however, is that the MNR does not outperform FTR in the number of predictions. The number of predictions made by FTR ranges from about 25% to 50% of those made by MNR (in four out of eight data sets) to 4 to 6 times more than MNR in the remaining data sets.

To examine whether the predictions of MNR rule overlap with those of FTR, we applied the MNR rule on the quadruples where FTR makes a prediction. The comparison is shown in Table 7. *MNR is able to predict a dependence only for* 1% *to* 25% *of FTR predictions*. The results in both Tables 6 and 7 clearly indicate that the two methods share only a small subset of common predictions, and thus neither method subsumes the other.

9.6 Summary, Interpretation, and Conclusions

We now comment and interpret the results of this section:

- It is possible to predict the presence of dependencies and bound their strength with distributional assumptions other than Faithfulness, such as multivariate normality.

Data Set	# rand. quads sampled	#MNR predictions on sampled quads	ACC^{MNR}	ACC^{FTR}	#FTR predictions / #expected MNR predictions on all quads
Breast-Cancer	356000	2	0.50	0.84	3.98
C&C	1000	45	1.00	0.96	0.02
Compactiv	1000	30	1.00	1.00	0.62
Insurance-C	1000	4	0.75	0.97	0.24
Lymphoma	147000	12	0.67	0.82	2.79
Ovarian	43000	391	0.99	0.99	5.99
p53	108000	39	1.00	0.97	5.19
Wine	495	7	1.00	1.00	0.57

Table 6: A comparison between FTR vs. MNR in predicting unconditional dependencies on randomly sampled quadruples. The columns are: the data set name, the total number of randomly sampled quadruples (1000 × the number of chunks, except for the Wine data set), the number of predictions made by MNR on those, the accuracies Acc^{MNR} and Acc^{FTR} at threshold $t = 0.05$. The final column presents the ratio of the number of predictions by the FTR rule over the *expected* number of predictions made by the MNR rule on all possible quadruples. The number of predictions made by FTR ranges from about 25% to 50% of those made by MNR to 4 to 6 times more than MNR.

Data Set	#FTR predictions	#MNR predictions restricted to cases FTR makes a prediction	% common predictions	*ACC* of both MNR and FTR
Breast-Cancer	1833	32	0.02	1.00
C&C	99241	10640	0.11	1.00
Compactiv	135	28	0.21	1.00
Insurance-C	1839	15	0.01	1.00
Lymphoma	7712	681	0.09	0.97
Ovarian	539165	59327	0.11	1.00
p53	46647	413	0.01	1.00
Wine	4	1	0.25	1.00

Table 7: A comparison between FTR vs. MNR in predicting unconditional dependencies on the cases where both rules apply.

- The sets of predictions entailed by assuming Faithfulness (FTR) and multivariate normality (MNR) do not overlap to a significant degree and neither method subsumes the other and they could be considered complementary. For example, the MNR makes a prediction only in the 1% to 25% of cases where FTR applies. In addition, in some data sets MNR makes only 2% of the number of FTR predictions, while in others MNR makes 6 times more predictions.

10. Related Work

Whole sub-fields have been developed to address the problem of integrative analysis, that we review briefly. Statistical matching has been reviewed, presented, and compared against in Section 9. Meta-Analysis focuses on the co-analysis of studies with similar sampling and experimental design characteristics with the purpose of making inferences about a single association. Meta-Analysis in Statistics (O'Rourke, 2007) combines the results of several studies to address a set of related research hypotheses. While meta-analysis focuses on a pair-wise association of a variable with an outcome of interest, a recent interesting extension addresses the problem of estimating the multivariate associations (for example, in the form of a regression model) with the target variable (Samsa et al., 2005); such methods often appear under the names of meta-regression and univariate synthesis (Zhou et al., 2009). The main idea of the latter is to assume a parametric form of the regression model and estimate the sufficient statistics from several homogeneous (in terms of being conducted on the same population, experimental conditions, sampling, etc.) studies that may not measure all variables (risk factors in this context). Both statistical matching and meta-analysis's scope does not extend to other sources of heterogeneity of the data sets, such as different experimental conditions.

In Computer Science and Machine Learning, the field of Transfer Learning (Pan and Yang, 2010) represents a main effort in integrative analysis. In Transfer Learning, successful search control strategies, model priors, and other characteristics transfer among different domains and/or tasks. When the task (target) is the same but the domains (populations) are different, this type of Transfer Learning is called Domain Adaptation. In this case, typically one would like to translate the estimated conditional distribution $P_s(Y|X)$ used for prediction in a source distribution to a target distribution $P_t(Y|X)$ that may be different (e.g., has a different marginal class distribution). Given that such methods are typically non-causal based, they cannot transfer to data sets where manipulations have been performed (causal methods could transfer predictive models to manipulated distributions as we show in Tsamardinos and Brown 2008, also shown in Maathuis et al. 2010). In addition, the input space for the predictors X has to be common. When the domain is the same (same distribution), but the tasks (target variables) are different, the type of Transfer Learning is called *Multi-Task Learning*. This type of learning attempts to simultaneously build models for several tasks in an effort to use one for leveraging the performance on the others. Typically this is performed by using a shared representation and learning common induced features. Again, these inferences are limited as they can only combine studies under the same sampling and experimental conditions on the same sets of variables.

Other fields may seem related in a first glance, but are orthogonal to the proposed research. The field of Relational Learning (Getoor and Taskar, 2007) does not really address the problem of learning from different data sets/studies over different samples, rather than a single data set (the one stemming from implicitly propositionalizing the database) in the form of relational tables. Similarly, the field of Distributed Learning (Cannataro et al., 2002) is restricted to designing time and communication-efficient analysis of what is essentially a single data set stored in different locations.

Other related work includes efforts to combine models (that may be developed from different data sets) on the same system but on different scales (Gennari et al., 2008). Typically, such methods involve mechanical models using differential equations and are not concerned with statistical models. In addition, these methods concern vertical integration at different temporal or spatial scales, while INCA proposes a horizontal integration of studies.

11. Discussion and Conclusions

We presented the basic idea and concept behind Integrative Causal Analysis (INCA), an approach for co-analyzing data sets that are heterogeneous in several aspects, such as in terms of measured variables and experimental conditions in the context of available prior knowledge. In this approach, one attempts to identify one or all causal models that are consistent with all available data and pieces of prior knowledge, and reason with them. Depending on the assumptions connecting causality with estimable quantities, co-analysis may lead to more inferences than independent analysis of the data sets.

In this paper, we focus on the problem of analyzing data sets over different variable sets. We employ Maximal Ancestral Graphs (MAGs) to model independencies in the data distributions and assume the latter are faithful to some MAG. As a proof-of-concept, we identify the simplest scenario where the INCA idea provides testable predictions, and specifically it predicts the presence and strength of an unconditional dependence, and a chain-like causal structure (entailing several additional conditional dependencies). The idea is implemented in the following algorithms: the Full-Testing Rule (FTR), the Minimal-Testing Rule (MTR) and FTR-S that additionally predicts the strength of the dependence.

The empirical results show that FTR and MTR are able to accurately predict the presence and strength of unconditional dependencies, as well as all the conditional dependencies entailed by the causal model. These predictions are better than chance and cannot be explained by the transitivity of dependencies often holding in Nature. Against typical statistical matching algorithms, FTR-S's predictions are better correlated with sample estimates particularly when the number of common variables is low.

Inducing causal models from observational data has been long debated (Pearl, 2000; Spirtes et al., 2001; Pearl, 2009). In our experiments, we do not employ the causal semantics of the models to predict the effect of manipulations but their ability to represent independencies, based on the assumption of Faithfulness. The results support that graphical models and the assumption of Faithfulness can make testable predictions and can be exploited for novel statistical inferences. While this is not a direct proof in favor of the causal semantics of the models, we do note that both Faithfulness and MAGs have been inspired by theories of probabilistic causality.

The empirical results show that the proposed algorithms' predictions are abundant, indicating the potential of the approach. Extending the theory and algorithms for increased efficiency, statistical robustness, range of tasks, data types, types of prior knowledge, and settings seems a promising direction that may allow the co-analysis of a large part of available studies and data sets.

Acknowledgments

We would like to thank the anonymous reviewers and the Editor for their constructive comments and feedback; We would like to thank Giorgos Borboudakis for his comments and help with proofreading the text; we would also like to thank our affiliated institutions for hosting this research.

Appendix A. Supplementary Material

In this appendix we provide supplementary information for the data sets used in the experiments presented in this paper, as well as some additional results.

A.1 Data Sets Preprocessing

Missing data imputation and discretization were *separately* performed, when necessary, on each sub-data-set $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_t$. Continuous variables X were discretized in three intervals:

- $]-\text{inf};\ mean(X) - std(X)]$
- $[mean(X) + std(X);\ \text{inf}[$
- remaining values.

Missing data were substituted with mean values (continuous, ordinal variables) or encoded as a distinct value (nominal variables). Our implementation of the G^2 test requires that nominal variables with n distinct values are econded as $0 \ldots n-1$. When necessary we re-encoded nominal variables for respecting this convention.

A.1.1 ACPJ

Preprocessing steps: 2765 variables were found constant in at least one sub-data-set and were consequently eliminated from the analysis.
Download information: Aliferis et al. (2010) kindly provided us with the data.

A.1.2 BIBTEX

Preprocessing steps: No particular preprocessing steps.
Download information: The data set is freely available from the MULAN project website: http://sourceforge.net/projects/mulan/ (checked on February 10, 2011).

A.1.3 C&C

Preprocessing steps: The first five attributes were eliminated because they do not carry relevant information. Columns with more than 80% of missing values were removed.
Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime (checked on February 10, 2011).

A.1.4 COMPACTIV

Preprocessing steps: No particular preprocessing steps.
Download information: The data set is freely available from the KEEL software web site: http://sci2s.ugr.es/keel/dataset.php?cod=49 (checked on February 10, 2011).

A.1.5 COVTYPE

Preprocessing steps: Attributes $1 \ldots 10$ were discretized.
Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Covertype
(checked on February 10, 2011).

A.1.6 DELICIOUS

Preprocessing steps: No particular preprocessing steps.

Download information: The data set is freely available from the MULAN project website: http://sourceforge.net/projects/mulan/ (checked on February 10, 2011).

A.1.7 HIVA

Preprocessing steps: No particular preprocessing steps.

Download information: The data set is freely available from the web site: http://www.causality.inf.ethz.ch/al_data/HIVA.html (checked on February 10, 2011).

A.1.8 INSURANCE-C

Preprocessing steps: All variables were considered as continuous; nominal variables (namely, attributes 1 and 5) were eliminated.

Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Insurance+Company+Benchmark+(COIL+2000) (checked on February 10, 2011).

A.1.9 INSURANCE-N

Preprocessing steps: All variables were considered as nominal.

Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Insurance+Company+Benchmark+(COIL+2000) (checked on February 10, 2011).

A.1.10 P53

Preprocessing steps: Samples with missing values were eliminated from the analysis (180 rows in total).

Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/p53+Mutants (checked on February 10, 2011).

A.1.11 READ

Preprocessing steps: Continuous variables (namely attributes 24, 25 and 26) were discretized.

Download information: The data set is freely available from the web site: http://funapp.cs.bilkent.edu.tr/DataSets/ (checked on February 10, 2011).

A.1.12 WINE

Preprocessing steps: Two different data sets are available, respectively about red and white wines. For our experimentation we used only the white wines data set (the one with more samples).

Download information: The data set is freely available from the UCI Machine Learning repository: http://archive.ics.uci.edu/ml/datasets/Wine+Quality (checked on February 10, 2011).

Data Set	$FTR_{0.05}$	$MTR_{0.02}$	$TR_{0.01}$
Covtype	59	810	1431
Read	0	9	260
Infant Mortality	10	427	1170
Compactiv	69	193	231
Gisette	330	12340	31648
hiva	366	16174	34977
Breast-Cancer	1371	68077	228610
Lymphoma	4473	51794	122857
Wine	3	44	66
Insurance-C	394	2212	3264
Insurance-N	95	1002	2527
p53	15181	95195	129372
Ovarian	41600	48376	52646
C&C	4168	5048	5050
ACPJ	0	190	15994
Bibtex	1	1858	16087
Delicious	524	6042	21351
Dexter	0	2	116
Nova	0	115	3280
Ohsumed	0	60	5227

Table 8: Number of unique predictions $|U_i^R|$ with “Bonferroni” correction for rules FTR, MTR, TR and Random Guess

A.1.13 BREAST-CANCER, DEXTER, GISETTE, INFANT-MORTALITY, LYMPHOMA, NOVA, OHSUMED, OVARIAN

Preprocessing steps: No particular preprocessing steps.
Download information: Aliferis et al. (2010) kindly provided us with the data.

A.2 Supplementary Tables

Table 10 presents the performance of the algorithms as measured by the Mean Absolute Error (MAE) of the predictions $\hat{r}_{YZ}$ and the sample-estimates r_{YZ}: $1/N \cdot \sum |\hat{r}^i - r^i|$, where N is the total number of predictions of an algorithm. This metric may favor algorithms that often predict zero correlations on data sets where the number of dependencies is low. This is the case of SMR_G and SMR_Q on the ACPJ data set (see Figure 17a). SMR_G and SMR_Q achieve an MAE of *only* 0.01 and 0.02 respectively because they always predict values close to zero, while failing to detect any high correlation. The corresponding correlations between predictions and sample-estimates on the same data set are low: 0.05 and 0.00 respectively.

Table 11 presents the performance of the algorithms as measured by the Mean Relative Absolute Error (MRAE) of the predictions $\hat{r}_{YZ}$ and the sample-estimates r_{YZ}: $1/N \cdot \sum |\hat{r}^i - r^i|/|r^i|$, where N is the total number of predictions of an algorithm. This metric penalizes more algorithms that attempt predictions of small correlations (such as *SMR*) because even a small absolute error may lead to a high relative error. For example, SMR on the Ovarian data set has a high MRAE (on the order of 10^9 despite a correlation between predictions and sample-estimates of 0.62 .

Data Set	$\text{FTR}_{0.05}$	$\text{MTR}_{0.02}$	Random Quadruple
Covtype	1.00	0.99	$0.74^{\spadesuit}$
Read	-	-	-
Infant Mortality	0.60	0.44	0.10^{**}
Compactiv	0.87	0.93^{*}	0.83
Gisette	0.80	$0.59^{\spadesuit}$	$0.11^{\spadesuit}$
hiva	0.71	$0.47^{\spadesuit}$	$0.22^{\spadesuit}$
Breast-Cancer	0.55	$0.31^{\spadesuit}$	$0.16^{\spadesuit}$
Lymphoma	0.46	$0.34^{\spadesuit}$	$0.18^{\spadesuit}$
Wine	1.00	0.70	0.73
Insurance-C	0.86	$0.65^{\spadesuit}$	$0.42^{\spadesuit}$
Insurance-N	0.57	0.50	0.17^{**}
p53	0.90	$0.82^{\spadesuit}$	$0.49^{\spadesuit}$
Ovarian	0.61	$0.62^{\spadesuit}$	$0.50^{\spadesuit}$
C&C	0.78	$0.73^{\spadesuit}$	$0.66^{\spadesuit}$
ACPJ	-	0.26	0.02
Bibtex	1.00	0.55	0.08^{**}
Delicious	0.99	$0.81^{\spadesuit}$	$0.19^{\spadesuit}$
Dexter	-	0.50	0.02
Nova	-	0.07	0.03
Ohsumed	-	0.14	0.02
$\overline{SACC^R}$	0.78	0.55^{*}	0.30^{**}
$\underline{SACC^R}$	0.66	$0.69^{\spadesuit}$	$0.12^{\spadesuit}$

Table 9: $SACC_i^R(t)$ at $t = 0.05$ with "Bonferroni" correction for rules FTR, MTR and Random Quadruple. Marks *, **, and ♠ denote a statistically significant difference from FTR at the levels of $0.05, 0.01$, and machine-epsilon respectively.

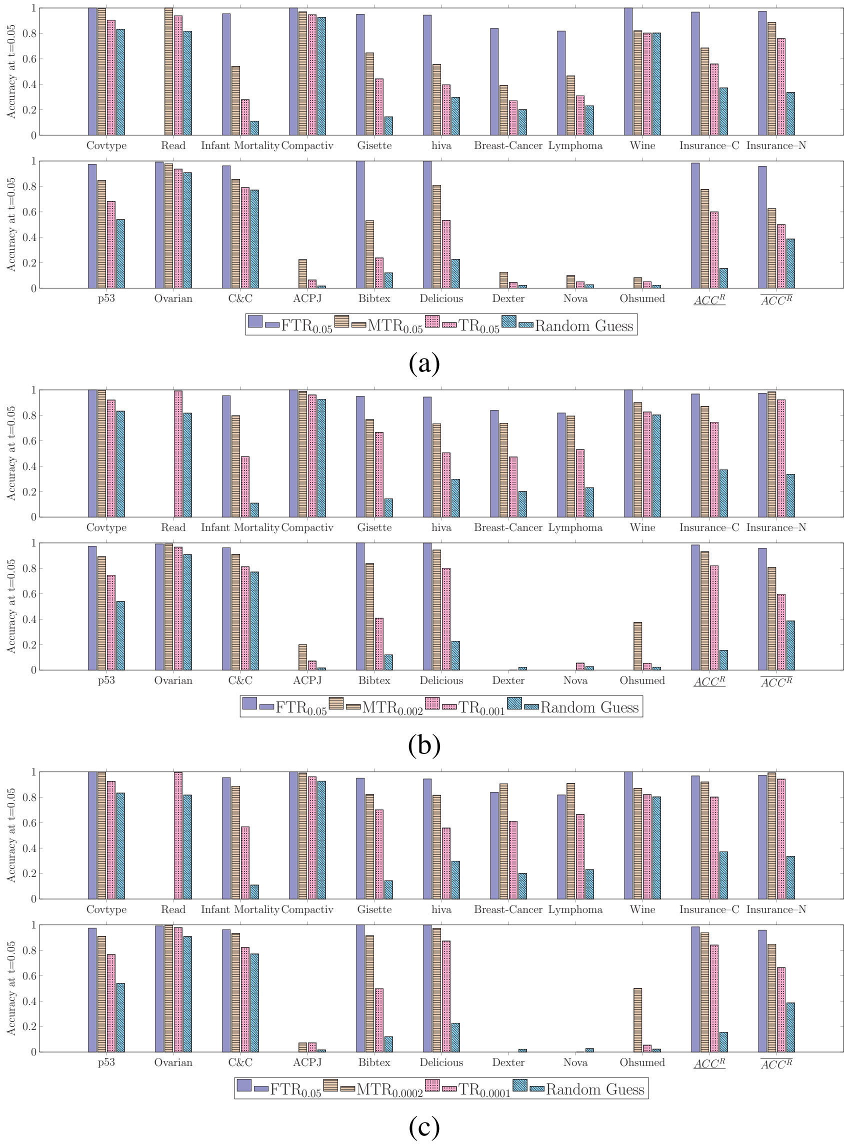
Data Sets	SMR_G	SMR_Q	FTR-S
ACPJ	0.01 ± 0.01	0.02 ± 0.01	-
Breast-Cancer	0.11 ± 0.08	0.13 ± 0.10	0.18 ± 0.13
C&C	0.05 ± 0.03	0.19 ± 0.18	0.18 ± 0.13
Compactiv	0.04 ± 0.06	0.19 ± 0.20	0.14 ± 0.12
Insurance-C	0.03 ± 0.08	0.09 ± 0.14	0.14 ± 0.12
Lymphoma	0.12 ± 0.09	0.14 ± 0.11	0.17 ± 0.14
Ohsumed	0.01 ± 0.02	0.02 ± 0.02	-
Ovarian	0.15 ± 0.10	0.16 ± 0.11	0.09 ± 0.07
Wine	0.09 ± 0.10	0.15 ± 0.17	0.22 ± 0.14
p53	0.03 ± 0.05	0.07 ± 0.10	0.14 ± 0.12
Over data sets	0.06 ± 0.06	0.12 ± 0.11	0.16 ± 0.12
Over predictions	0.07 ± 0.08	0.07 ± 0.09	0.11 ± 0.10

Table 10: Mean Absolute Error (MAE) between the calibrated predictions $\hat{r}_{YZ}$ and sample-estimated r_{YZ} (average value ± standard deviation). SMR_G refers to the Statistical Matching Rule applied on all pairs of variables in the same group, considering the remaining 48 variables in the group as common variables. SMR_Q is the Statistical Matching Rule applied on quadruples of variables randomly chosen from the same group. Finally, FTR-S consists in the Full Testing Rule modified for estimating the strength of the dependency, see Algorithm 4.

Data Sets	SMR$_G$	SMR$_Q$	FTR-S
ACPJ	13.17 ± 87.17	27.22 ± 141.50	-
Breast-Cancer	5.74 ± 624.51	2.79 ± 90.41	1.39 ± 4.51
C&C	1.52 ± 39.16	3.53 ± 44.98	1.30 ± 16.80
Compactiv	0.39 ± 1.43	1.79 ± 9.39	0.46 ± 0.53
Insurance-C	2.79 ± 11.04	2.10 ± 5.15	2.44 ± 18.04
Lymphoma	4.51 ± 182.18	3.66 ± 181.90	5.77 ± 145.88
Ohsumed	4.62 ± 30.53	7.72 ± 8.95	-
Ovarian	$7.32 \times 10^9 \pm 1.68 \times 10^{13}$	0.58 ± 5.51	0.20 ± 0.44
Wine	1.31 ± 2.24	1.78 ± 5.65	0.38 ± 0.06
p53	34.95 ± 7982.92	19.86 ± 4544.32	4.76 ± 290.58
Over data sets	$7.32 \times 10^9 \pm 1.68 \times 10^{13}$	7.10 ± 503.78	2.09 ± 59.61
Over predictions	$2.79 \times 10^9 \pm 3.28 \times 10^{12}$	14.36 ± 1320.98	0.87 ± 87.92

Table 11: Mean Relative Absolute Error (MRAE) between the calibrated predictions $\hat{r}_{YZ}$ and sample-estimated r_{YZ} (average value ± standard deviation) SMR$_G$ refers to the Statistical Matching Rule applied on all pairs of variables in the same group, considering the remaining 48 variables in the group as common variables. SMR$_Q$ is the Statistical Matching Rule applied on quadruples of variables randomly chosen from the same group. Finally, FTR-S consists in the Full Testing Rule modified for estimating the strength of the dependency, see Algorithm 4. For the Ovarian data set the SMR$_G$ rule provides predictions for cases with nearby-zero sample estimated r_{YZ}, and these predictions generate extremely high MRAE values. Once excluded such cases, the SMR$_G$ MRAE on the Ovarian data set is 0.54 ± 12.16, while the MRAE averaged over all data sets and over all predictions is 6.95 ± 897.33 and 10.45 ± 2498.28, respectively.

A.3 Supplementary Figures



(a)

(b)

(c)

Figure 21: Accuracies Acc_i for each data set, as well as the average accuracy $\overline{Acc}$ (each data set weighs the same) and the pooled accuracy $\underline{Acc}$ (each prediction weighs the same). (a) All rules are applied without any correction of significance threshold and all accuracies are computed at $t = 0.05$ (b) Accuracies Acc_i calculated with the "Bonferroni 10^{-1}" significance threshold correction. (c) Accuracies Acc_i calculated with the "Bonferroni 10^{-2}" significance threshold correction.

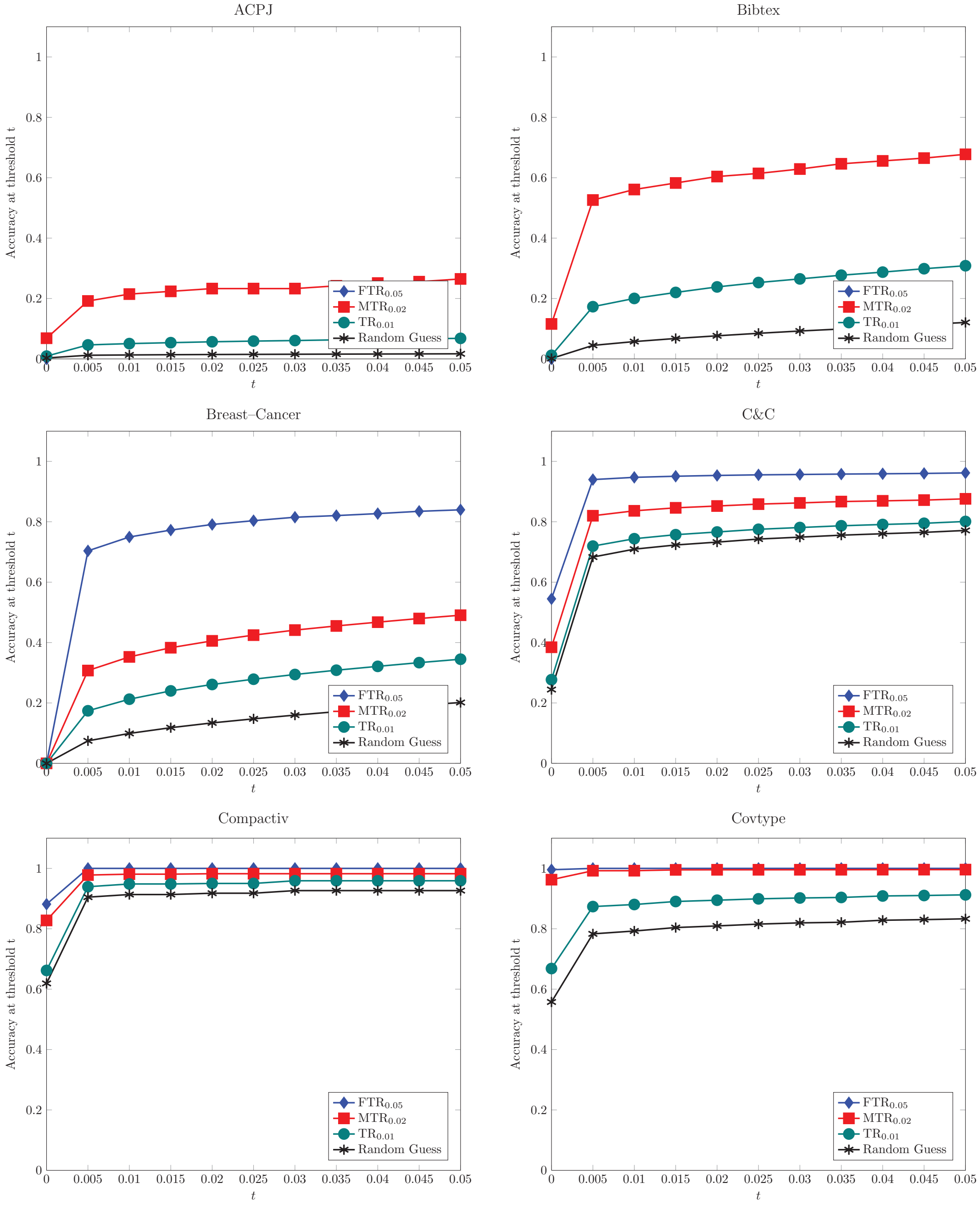


Figure 22: Accuracy at threshold t for data sets ACPJ-Etiology, Bibtex, Breast Cancer and Communities and Crime, Compactiv and Covtype for different rules

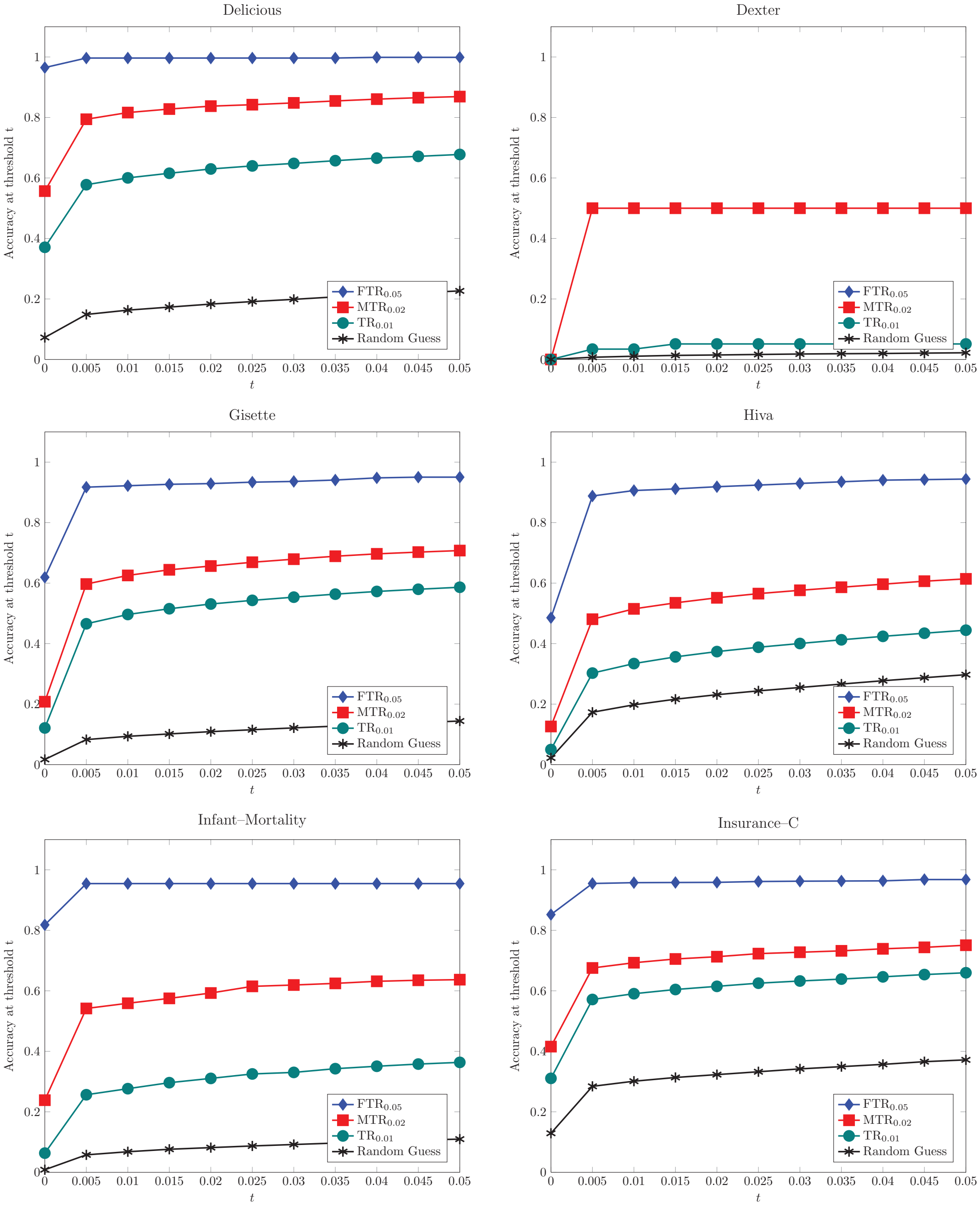


Figure 23: Accuracy at threshold t for data sets Delicious, Dexter, Gisette, Hiva, Infant Mortality and Insurance-C for different rules

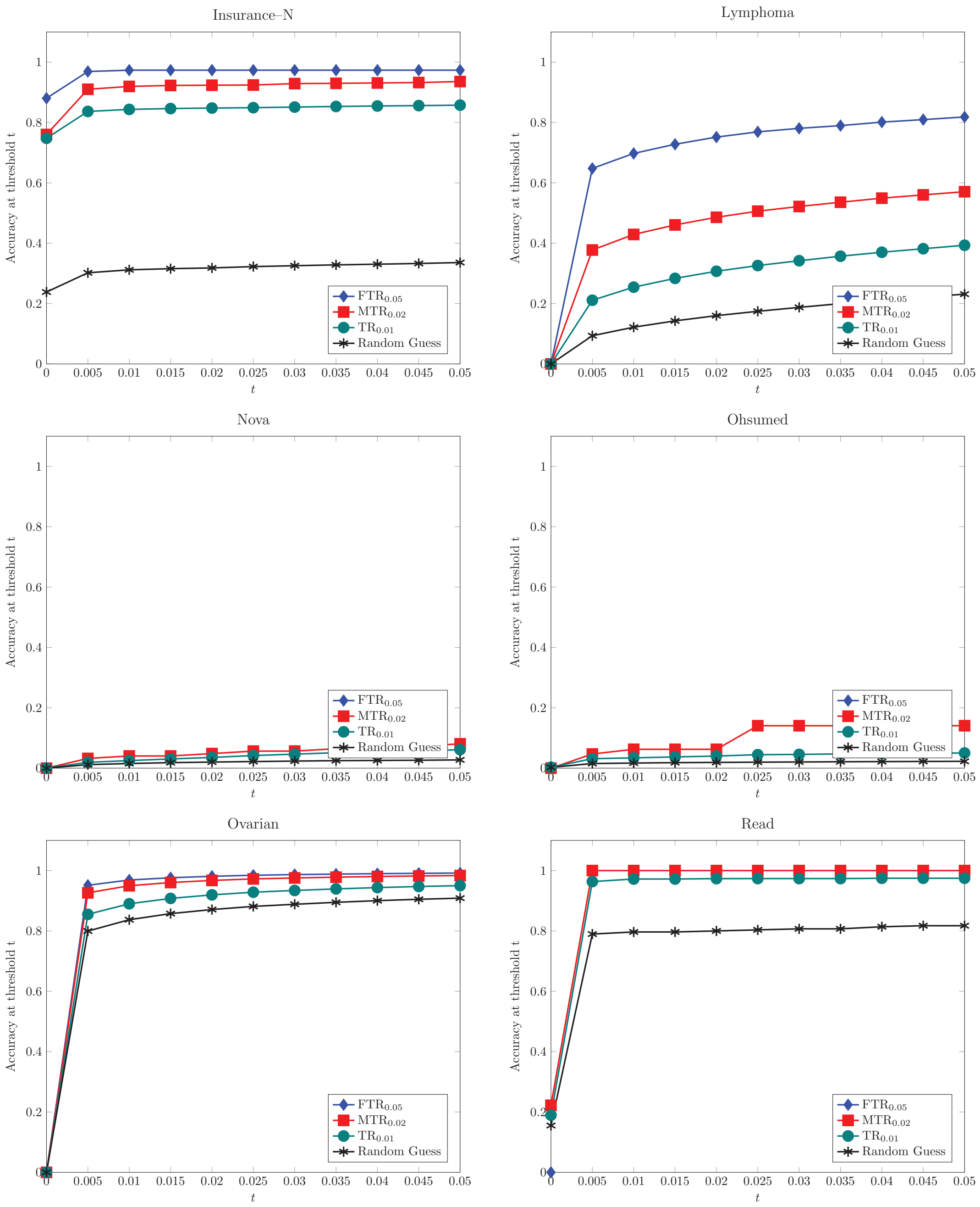


Figure 24: Accuracy at threshold t for data sets Insurance-N, Lymphoma, Nova, Ohsumed, Ovarian, Read and for different rules

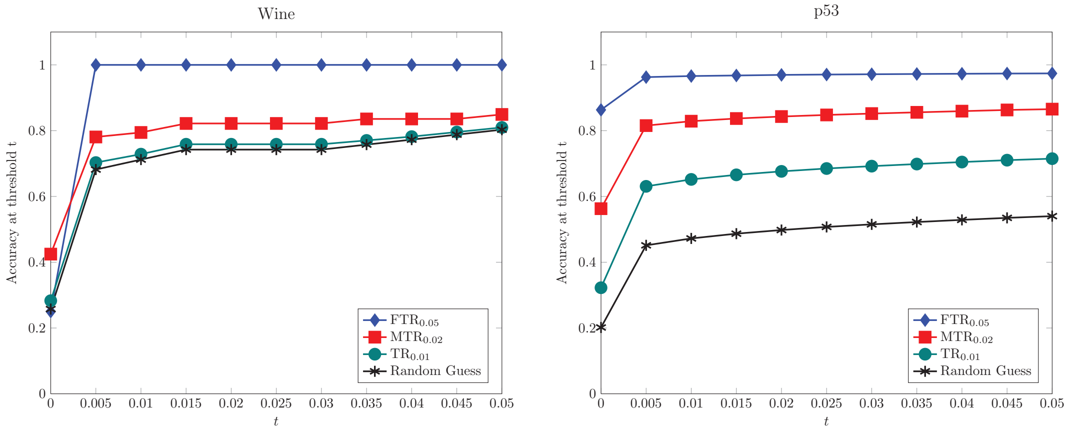


Figure 25: Accuracy at threshold t for data sets Wine, p53

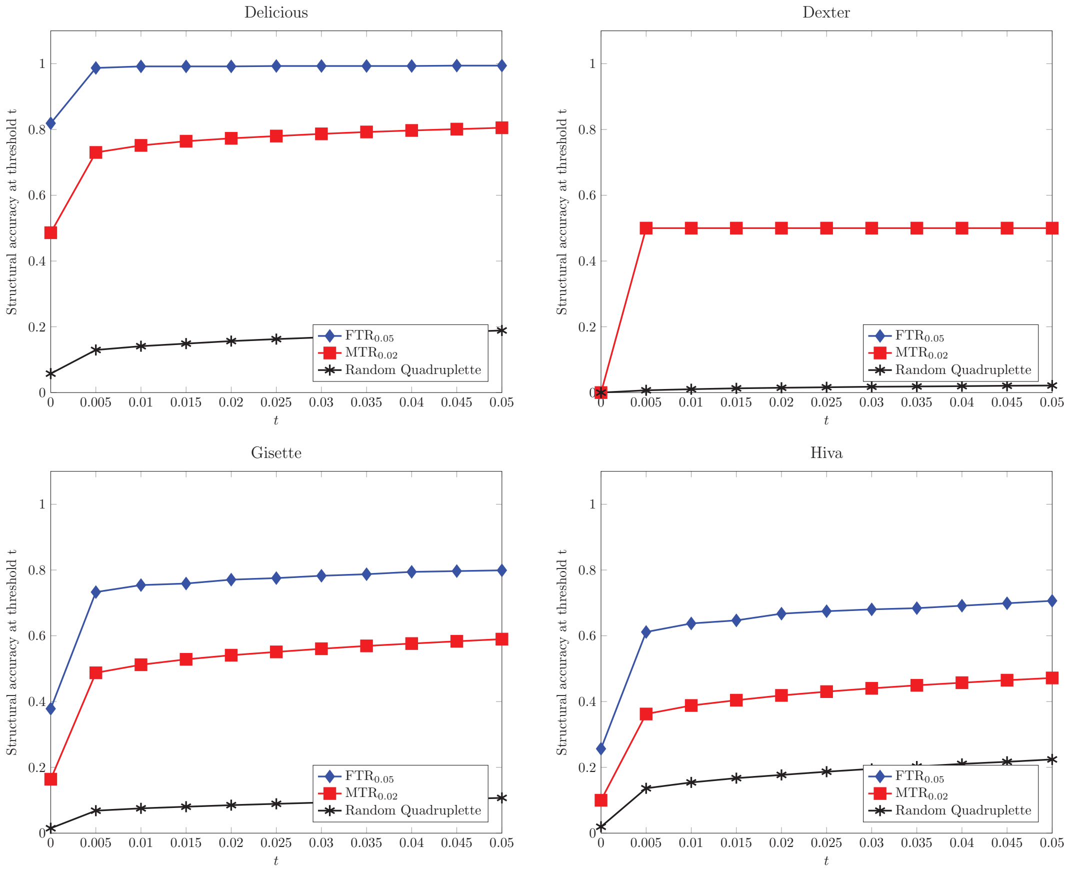


Figure 26: Structural accuracy at threshold t for data sets Delicious, Dexter, Gisette and Hiva for different rules

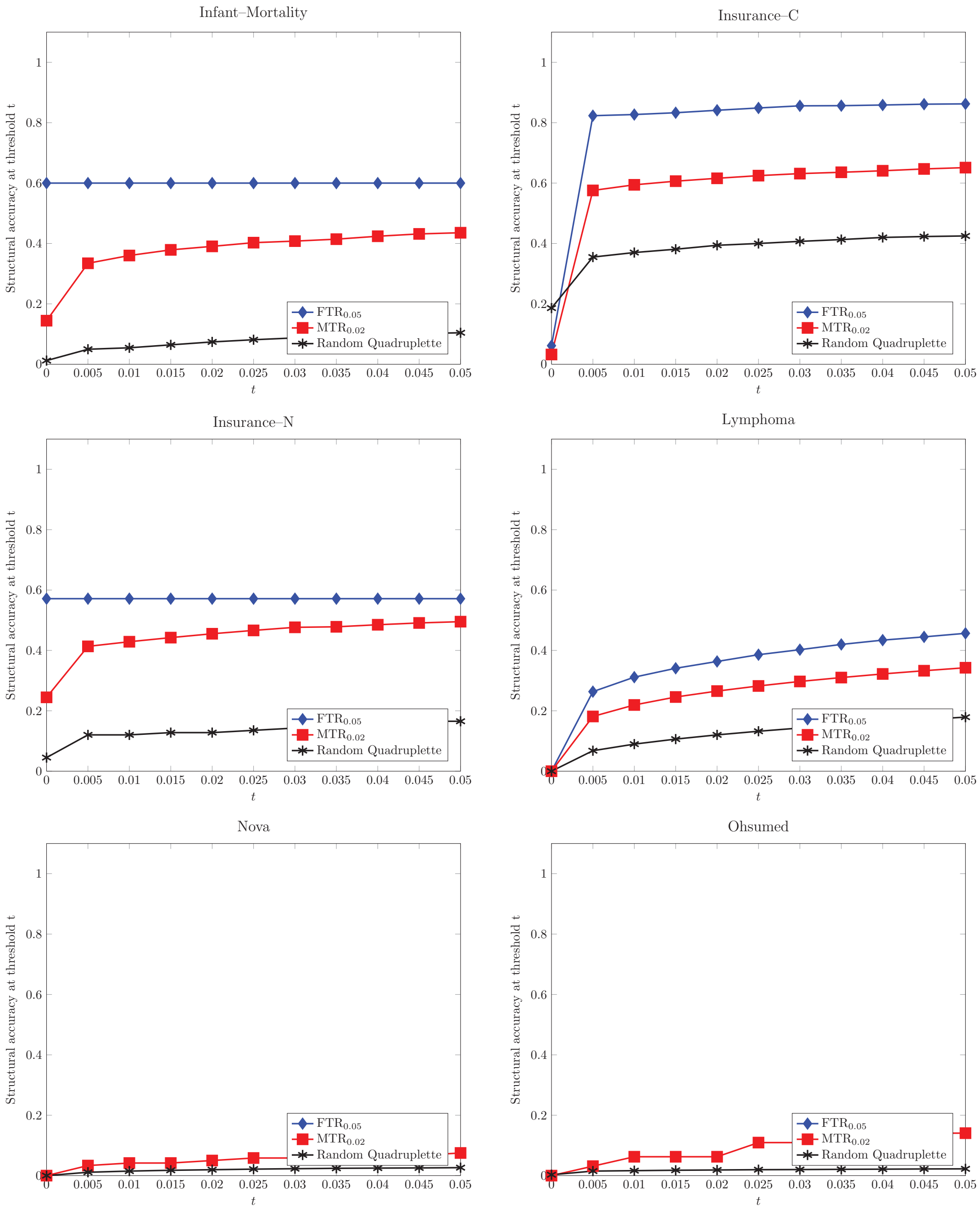


Figure 27: Structural accuracy at threshold t for data sets Infant Mortality, Insurance-C, Insurance-N, Lymphoma, Nova and Ohsumed for different rules

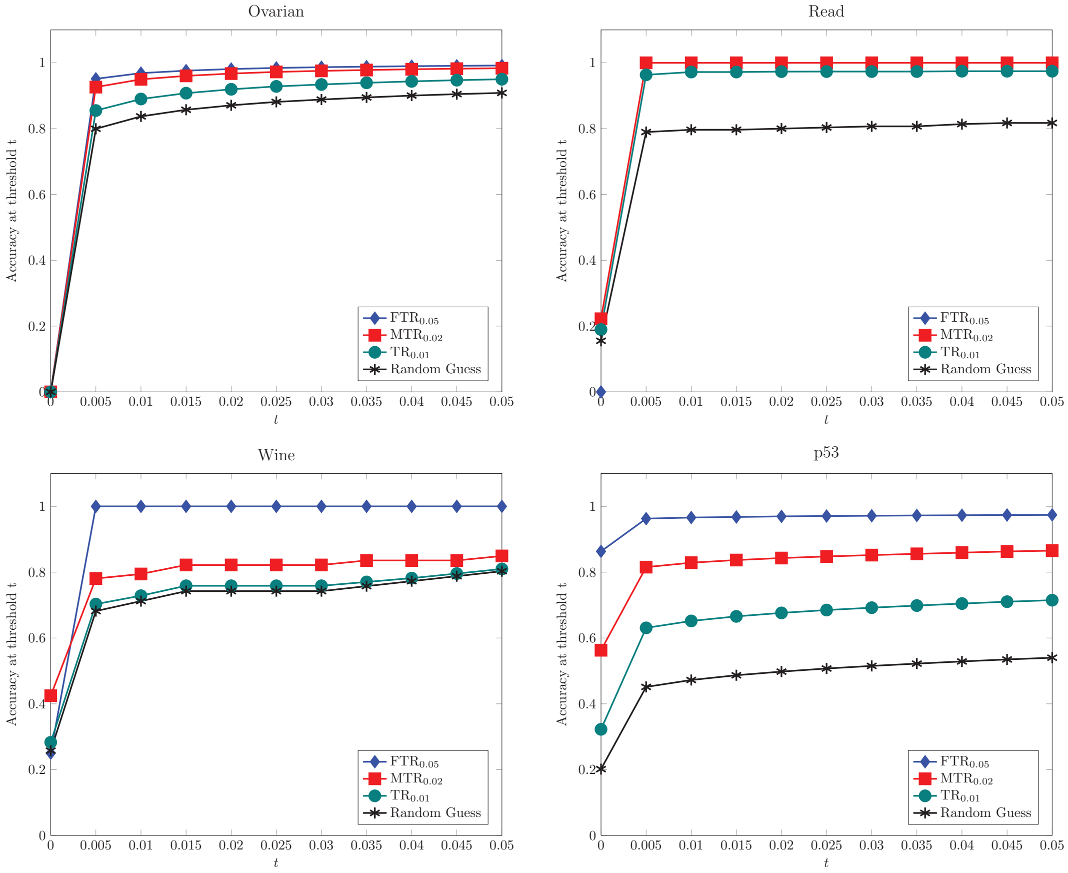


Figure 28: Structural accuracy at threshold t for data sets Ovarian, Read, Wine and p53 for different rules

References

B Abramson, J Brown, W Edwards, A Murphy, and RL Winkler. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

J Alcalá-Fdez, L Sánchez, S García, M J Jesus, S Ventura, J M Garrell, J Otero, C Romero, J Bacardit, V M Rivas, J C Fernández, and F Herrera. KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.

CF Aliferis, A Statnikov, I Tsamardinos, S Mani, and X Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part ii : Analysis and extensions. *Journal of Machine Learning Research*, 11:235–284, 2010.

N Angelopoulos and J Cussens. Bayesian learning of Bayesian networks with informative priors. *Annals of Mathematics and Artificial Intelligence*, 54(1-3):53–98, 2008.

Y Aphinyanaphongs, AR Statnikov, and CF Aliferis. A comparison of citation metrics to machine learning filters for the identification of high quality MEDLINE documents. *JAMIA*, 13(4):446–455, 2006.

A Balke and J Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1172–1176, 1997.

IA Beinlich, HJ Suermondt, RM Chavez, and GF Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Second European Conference on Artificial Intelligence in Medicine*, volume 38, pages 247–256. Springer-Verlag, Berlin, 1989.

J Binder, D Koller, S Russell, and K Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29, 1997.

JA Blackard and DJ Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

G Borboudakis, S Triantafillou, V Lagani, and I Tsamardinos. A constraint-based approach to incorporate prior knowledge in causal models. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2011.

M Cannataro, D Talia, and P Trunfio. Distributed data mining on the grid. *Future Gener. Comput. Syst.*, 18:1101–1112, October 2002.

T Claassen and T Heskes. Causal discovery in multiple models from different experiments. In *Advances in Neural Information Processing Systems (NIPS 2010)*, volume 23, pages 1–9, 2010.

TP et al. Conrads. High-resolution serum proteomic features for ovarian cancer detection. *Endocrine-Related Cancer*, 11(2):163–78, 2004.

GF Cooper and Ch Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of Uncertainty in Artificial Intelligence (UAI 1999)*, volume 10, pages 116–125, 1999.

P Cortez, Ao Cerdeira, F Almeida, T Matos, and J Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, November 2009. ISSN 01679236.

R Cudeck. An estimate of the covariance between variables which are not jointly observed. *Psychometrika*, 65(4):539–546, 2000.

SA Danziger, R Baronio, L Ho, L Hall, K Salmon, GW Hatfield, P Kaiser, and RH Lathrop. Predicting positive p53 cancer rescue regions using most informative positive (MIP) active learning. *PLoS Computational Biology*, 5(9):12, 2009.

M D'Orazio, MD Zio, and M Scanu. *Statistical Matching: Theory and Practice*. Wiley, 2006.

F Eberhardt. A sufficient condition for pooling data. *Synthese*, 163(3):433–442, February 2008.

F Eberhardt, PO Hoyer, and R Scheines. Combining experiments to discover linear cyclic models with latent variables. In *Proceedings of Artificial Intelligence anf Statistics 2010*, volume 9, pages 185–192, 2010.

C Elkan. Magical thinking in data mining: lessons from CoIL challenge 2000. *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–431, 2001.

RA Fisher. On the interpretation of $\chi 2$ from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

A Frank and A Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

JH Gennari, ML Neal, BE Carlson, and DL Cook. Integration of multi-scale biosimulation models via light-weight semantics. *Pacific Symposium On Biocomputing*, 425:414–25, 2008.

L Getoor and B Taskar. *Introduction to Statistical Relational Learning*, volume L. The MIT Press, 2007.

HA Guvenir and I Uysal. Bilkent University function approximation repository, 2000. URL `http://funapp.cs.bilkent.edu.tr`.

I Guyon, S Gunn, M Nikravesh, and L Zadeh. *Feature Extraction, Foundations and Applications*. Springer–Verlag, Berlin, Germany, 2006a.

I Guyon, A Saffari, G Dror, and J Buhmann. Performance prediction challenge. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1649–1656, 2006b.

A Hyttinen, F Eberhardt, and PO Hoyer. Causal discovery for linear cyclic models with latent variables. In *Proccedings of the 5th European Workshop on Probabilistic Graphical Models*, 2010.

A Hyttinen, F Eberhardt, and PO Hoyer. Noisy-OR models with latent confounding. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, 2011.

Th Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. (The Kluwer International Series in Engineering and Computer Science).* Springer, 2002.

SH Kim. Markovian combination of subgraphs of DAGs. In *Proceedings of The 10th IASTED International Conference on Artificial Intelligence and Applications*, pages 90–95, 2010.

SH Kim and S Lee. *New Developments in Robotics, Automation and Control*. In-Tech, Vienna, Austria, 2008.

MH Maathuis, M Kalisch, and P Bühlmann. Estimating high-dimensional intervention effects from observational data. *Annals of Statistics*, 37:3133–3164, 2009.

MH Maathuis, D Colombo, M Kalisch, and P Bühlmann. Predicting causal effects in large-scale systems from observational data. *Nature Methods*, 7(4):247–248, 2010. ISSN 15487105.

S Mani and GF Cooper. Causal discovery using a Bayesian local causal discovery algorithm. *Medinfo 2004*, 11:731–735, 2004.

C Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th Conference of Uncertainty in Aritficial Intelligence*, pages 403–410, 1995.

C Moriarity and F Scheuren. Statistical matching: a paradigm for assessing the uncertainty in the procedure. *Journal of Official Statistics*, 17:407–422, 2001.

RT O'Donnell, AE Nicholson, B. Han, KB Korb, MJ Alam, and LR Hope. Incorporating Expert Elicited Structural Information in the CaMML Causal Discovery Program. Technical report, Clayton School of Information Technology, Monash University, Melbourne, 2006.

K O'Rourke. An historical perspective on meta-analysis: dealing quantitatively with varying study results. *Journal of the Royal Society of Medicine*, 100(12):579–582, 2007.

SJ Pan and Q Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

J Pearl. *Causality: Models, Reasoning and Inference*, volume 113 of *Hardcover*. Cambridge University Press, 2000.

J Pearl. Causal inference in statistics: an overview. *Statistics Surveys*, 3:96–146, 2009.

K Pearson. On a criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can reasonably be supposed to have arisen from random sampling. *Philosophical Magazine*, 50(50):157–175, 1900.

JP Pellet and A Elisseeff. Finding latent causes in causal networks: an efficient approach based on Markov blankets. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, 2008.

J Ramsey, P Spirtes, and J Zhang. Adjacency faithfulness and conservative causal inference. In *Proceedings of Uncertainty in Artificial Intelligence*, 2006.

T Richardson and P Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4): 962–1030, 2002.

A Rosenwald et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *N Engl J Med*, 346(25):1937–1947, 2002.

DG Rubin. Characterizing the estimation of parameters in incomplete-data problems. *Journal of the American Statistical Association*, 69:467–474, 1974.

G Samsa, G Hu, and M Root. Combining information from multiple data sources to create multivariable risk models: Illustration and preliminary assessment of a new method. *Journal of Biomedicine and Biotechnology*, 2005(2):113–123, 2005.

S Shimizu, PO Hoyer, A Hyvärinen, and A Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(2):2003–2030, 2006.

S Shimizu, T Inazumi, Y Sogawa, A Hyvarinen, Y Kawahara, T Washio, PO Hoyer, and K Bollen. DirectLiNGAM: a direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011.

A Spanos. Revisiting the omitted variables argument: Substantive vs. statistical adequacy. *Journal of Economic Methodology*, 13(2):179–218, 2006.

P Spirtes and TS Richardson. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, pages 489–500, 1996.

P Spirtes, C Glymour, and R Scheines. *Causation, Prediction, and Search.* The MIT Press, second edition, January 2001.

J Tian and J Pearl. Causal Discovery from Changes. *Proceedings of UAI*, pages 512–521, 2001.

RE Tillman. Structure learning with independent non-identically distributed data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1041–1048. ACM, 2009.

RE Tillman and P Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15, pages 3–15, 2011.

RE Tillman, David Danks, and Clark Glymour. Integrating locally learned causal structures with overlapping variables. In *Advances in Neural Information Processing Systems (NIPS*, 2008.

S Triantafillou, I Tsamardinos, and IG Tollis. Learning causal structure from overlapping variable sets. In *Proceedings of Artificial Intelligence and Statistics*, volume 9, 2010.

I Tsamardinos and G Borboudakis. Permutation testing improves Bayesian network learning. In *ECML PKDD*, pages 322–337, 2010.

I Tsamardinos and LE Brown. Bounding the false discovery rate in local Bayesian network learning. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI)*, pages 1100–1105, 2008.

I Tsamardinos and S Triantafillou. The possibility of integrative causal analysis: Learning from different datasets and studies. *Journal of Engineering Intelligent Systems*, 17(2/3):163–175, 2009.

I Tsamardinos, LE Brown, and CF Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

G Tsoumakas, I Katakis, and I Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, pages 1–20, 2010.

B Vantaggi. Statistical matching of multiple sources: A look through coherence. *Int. J. Approx. Reasoning*, 49(3):701–711, 2008.

Y et al. Wang. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365(9460):671–679, 2005.

AV Werhli and D Husmeier. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article15, 2007.

S Wright. The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3):161–215, 1934.

J Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

XH Zhou, N Hu, G Hu, and M Root. Synthesis analysis of regression models with a continuous outcome. *Statistics in Medicine*, 28(11):1620–1635, 2009.

DW Zimmerman, BD Zumbo, and RH Williams. Bias in estimation and hypothesis testing of correlation. *Psicoliogica*, 24:133–158, 2003.

Journal of Machine Learning Research 13 (2012) 1159-1187 Submitted 7/11; Revised 2/12; Published 4/12

Hope and Fear for Discriminative Training of Statistical Translation Models

David Chiang CHIANG@ISI.EDU
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292, USA

Editor: Michael Collins

Abstract

In machine translation, discriminative models have almost entirely supplanted the classical noisy-channel model, but are standardly trained using a method that is reliable only in low-dimensional spaces. Two strands of research have tried to adapt more scalable discriminative training methods to machine translation: the first uses log-linear probability models and either maximum likelihood or minimum risk, and the other uses linear models and large-margin methods. Here, we provide an overview of the latter. We compare several learning algorithms and describe in detail some novel extensions suited to properties of the translation task: no single correct output, a large space of structured outputs, and slow inference. We present experimental results on a large-scale Arabic-English translation task, demonstrating large gains in translation accuracy.

Keywords: machine translation, structured prediction, large-margin methods, online learning, distributed computing

1. Introduction

Statistical machine translation (MT) aims to learn models that can predict, given some utterance in a source language, the best translation into some target language. The earliest of these models were generative (Brown et al., 1993; Och et al., 1999): drawing on the insight of Warren Weaver in 1947 that "translation could conceivably be treated as a problem in cryptography" (Locke and Booth, 1955), they treated translation as the inverse of a process in which target-language utterances are generated by a *language model* and then changed into source-language utterances via a noisy channel, the *translation model*.

Och and Ney (2002) first proposed evolving this noisy-channel model into a discriminative log-linear model, which incorporated the language model and translation model as features. This allowed the language model and translation model be to scaled by different factors, and allowed the addition of features beyond these two. Although discriminative models were initially trained by maximum-likelihood estimation, the method that quickly became dominant was minimum-error-rate training or MERT, which directly minimizes some loss function (Och, 2003). The loss function of choice is most often BLEU (rather, $1 - \text{BLEU}$), which is the standard metric of translation quality used in current MT research (Papineni et al., 2002). However, because this loss function is in general non-convex and non-smooth, MERT tends to be reliable for only a few dozen features.

Two strands of research have tried to adapt more scalable discriminative training methods to machine translation. The first uses log-linear probability models, as in the original work of Och and Ney (2002), either continuing with maximum likelihood (Tillmann and Zhang, 2006; Blunsom et al., 2008) or replacing it with minimum risk, that is, expected loss (Smith and Eisner, 2006; Zens et al., 2008; Li and Eisner, 2009; Arun et al., 2010). The other uses linear models and large-margin methods (Liang et al., 2006; Watanabe et al., 2007; Arun and Koehn, 2007); we have followed this approach (Chiang et al., 2008b) and used it successfully with many different kinds of features (Chiang et al., 2009; Chiang, 2010; Chiang et al., 2011).

Here, we provide an overview of large-margin methods applied to machine translation, and describe in detail our approach. We compare MERT and minimum-risk against several online large-margin methods: stochastic gradient descent, the Margin Infused Relaxed Algorithm or MIRA (Crammer and Singer, 2003), and Adaptive Regularization of Weights or AROW (Crammer et al., 2009). Using some simple lexical features, the best of these methods, AROW, yields a sizable improvement of 2.4 BLEU over MERT in a large-scale Arabic-English translation task.

We discuss three novel extensions of these algorithms that adapt them to particular properties of the translation task. *First*, in translation, there is no single correct output, but only a *reference* translation, which is one of many correct outputs. We find that training the model to generate the reference exactly can be too brittle; instead, we propose to update the model towards *hope* translations which compromise between the reference translation and translations that are easier for the model to generate (Section 4). *Second*, translation involves a large space of structured outputs. We try to efficiently make use of this whole space, like most recent work in structured prediction, but unlike much work in statistical MT, which relies on n-best lists of translations instead (Section 5). *Third*, inference in translation tends to be very slow. Therefore, we investigate methods for parallelizing training, and demonstrate a novel method that is expensive, but highly effective (Section 6).

2. Preliminaries

In this section, we outline some basic concepts and notation needed for the remainder of the paper. Most of this material is well-known in the MT literature; only Section 2.4, which defines the loss function, contains new material.

2.1 Setting

In this paper, models are defined over *derivations* d, which are objects that encapsulate an input sentence $f(d)$, an output sentence $e(d)$, and possibly other information.[1] For any input sentence f, let $\mathcal{D}(f)$ be the set of all valid derivations d such that $f(d) = f$.

A model comprises a mapping from derivations d to feature vectors $\mathbf{h}(d)$, together with a vector of feature weights $\mathbf{w}$, which are to be learned. The model score of a derivation d is $\mathbf{w} \cdot \mathbf{h}(d)$. The 1-best or Viterbi derivation of f_i is $\hat{d} = \arg\max_{d \in \mathcal{D}(f_i)} \mathbf{w} \cdot \mathbf{h}(d)$, and the 1-best or Viterbi translation is $\hat{e} = e(\hat{d})$.

We are given a training corpus of input sentences $f_1, \ldots, f_N$, and reference output translations $e_1, \ldots, e_N$ produced by a human translator. Each e_i is not the only correct translation of f_i, but only one of many. For this reason, often multiple reference translations are available for each f_i, but

[1] The variables f and e stand for French and English, respectively, in reference to the original work of Brown et al. (1993).

for notational simplicity, we generally assume a single reference, and describe how to extend to multiple references when necessary.

Note that although the model is defined over derivations, only sentence pairs (f_i, e_i) are observed. There may be more than one derivation of e_i, or there may be no derivations. Nevertheless, assume for the moment that we can choose a *reference derivation* d_i that derives e_i; we discuss various ways of choosing d_i in Section 4.

2.2 Derivation Forests

The methods described in this paper should work with a wide variety of translation models, but, for concreteness, we assume a model defined using a weighted synchronous context-free grammar or related formalism (Chiang, 2007). We do not provide a full definition here, but only enough to explain the algorithms in this paper. In models of this type, derivations can be thought of as trees, and the set of derivations $\mathcal{D}(f)$ is called a *forest*. Although its cardinality can be worse than exponential in $|f|$, it can be represented as a polynomial-sized hypergraph $G = (V, E, r)$, where V is a set of nodes, $r \in V$ is the root node, and $E \subseteq V \times V^*$ is a set of hyperedges. We write a hyperedge as $(v \to \mathbf{v})$. A derivation d is represented by an edge-induced subgraph of G such that $r \in d$ and, for every node $v \in d$, there is exactly one hyperedge $(v \to \mathbf{v})$.

We require that $\mathbf{h}$ (and therefore $\mathbf{w} \cdot \mathbf{h}$) decomposes additively onto hyperedges, that is, $\mathbf{h}$ can be extended to hyperedges such that

$$\mathbf{h}(d) = \sum_{(v \to \mathbf{v}) \in d} \mathbf{h}(v \to \mathbf{v}).$$

This allows us to find the Viterbi derivation efficiently using dynamic programming.

2.3 BLEU

The standard metric for MT evaluation is currently BLEU (Papineni et al., 2002). Since we use this metric not only for evaluation but during learning, it is necessary to describe it in detail.

For any string e, let $g_k(e)$ be the multiset of all k-grams of e. Let K be the maximum size k-grams we will consider; $K = 4$ is standard. For any multiset A, let $\#_A(x)$ be the multiplicity of x in A, let $|A| = \sum_x \#_A(x)$, and define the multisets $A \cap B$, $A \cup B$, and A^* such that

$$\begin{aligned}
\#_{A \cap B}(x) &= \min(\#_A(x), \#_B(x)), \\
\#_{A \cup B}(x) &= \max(\#_A(x), \#_B(x)), \\
\#_{A^*}(x) &= \begin{cases} \infty & \text{if } \#_A(x) > 0, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Let c be the candidate translation to be evaluated and let r be the reference translation. Then define a vector of *component scores*

$$\mathbf{b}(c, r) = [m_1, \ldots m_K, n_1, \ldots n_K, \rho]$$

where

$$\begin{aligned}
m_k &= |g_k(c) \cap g_k(r)|, \\
n_k &= |g_k(c)|, \\
\rho &= |r|.
\end{aligned}$$

If there is a set of multiple references R, then

$$m_k = \left| g_k(c) \cap \bigcup_{r \in R} g_k(r) \right|, \tag{1}$$

$$\rho = \arg\min_{r \in R} \big| |r| - |c| \big| \tag{2}$$

where ties are resolved by letting ρ be the length of the shorter reference.

The component scores are additive, that is, the component score vector for a set of sentences $c_1, \ldots, c_N$ with references $r_1, \ldots, r_N$ is $\sum_i \mathbf{b}(c_i, r_i)$. Then the BLEU score is defined in terms of the component scores:

$$\text{BLEU}(\mathbf{b}) = \exp\left(\frac{1}{K} \sum_{k=1}^{K} \frac{m_k}{n_k} + \min\left(0, 1 - \frac{\rho}{n_1}\right) \right).$$

2.4 Loss Function

Our learning algorithms assume a loss function $\ell_i(e, e_i)$ that indicates how bad it is to guess e instead of the reference e_i. Our loss function is based on BLEU, but because our learning algorithms are online, we need to be able to evaluate the loss for a single sentence, whereas BLEU was designed to be used on whole data sets. If we try to compute it on a single sentence, several problems arise. If n_k is zero, the BLEU score is undefined; if any of the m_k are zero, the whole BLEU score is zero. Even barring such problems, a BLEU score for a single sentence may not accurately reflect the impact of that sentence on the whole test set (Chiang et al., 2008a).

The standard solution to these problems is to add pseudocounts (Lin and Och, 2004):

$$\text{BLEU}(\overline{\mathbf{b}} + \mathbf{b}) = \exp\left(\frac{1}{K} \sum_{k=1}^{K} \frac{\overline{m}_k + m_k}{\overline{n}_k + n_k} + \min\left(0, 1 - \frac{\overline{\rho} + \rho}{\overline{n}_1 + n_1}\right) \right)$$

where $\overline{\mathbf{b}} = [\overline{m}_1, \ldots, \overline{m}_K, \overline{n}_1, \ldots, \overline{n}_K, \overline{\rho}]$ are pseudocounts that must be set appropriately.

Watanabe et al. (2007) score a sentence in the context of all previously seen 1-best translations, which they call the *oracle document*. We follow this approach here, but in order to reduce dependence on the distant past, we use an exponential decay. That is, after processing each training example (f_i, e_i), we update the oracle document using the 1-best translation $\hat{e}$:

$$\overline{\mathbf{b}} \leftarrow 0.9 \cdot (\overline{\mathbf{b}} + \mathbf{b}(\hat{e}, e_i)).$$

Then we define a per-sentence metric B that measures the impact that adding a new input and output sentence will have on the BLEU score of the oracle document:

$$B(\mathbf{b}) = \overline{n}_1 \cdot \left(\text{BLEU}(\overline{\mathbf{b}} + \mathbf{b}) - \text{BLEU}(\overline{\mathbf{b}}) \right). \tag{3}$$

The reason for the scaling factor $\overline{n}_1$, which is the size of the oracle document, is to try to correct for the fact that if the oracle document is small, then adding a new sentence will have a large effect on its BLEU score, and vice versa.

Finally, we can define the loss of a translation e relative to e' as the difference between their B scores, following Watanabe et al. (2007):

$$\ell_i(e, e') = B(\mathbf{b}(e, e_i)) - B(\mathbf{b}(e', e_i))$$

and, as shorthand,

$$\ell_i(d,e') \equiv \ell_i(e(d),e'),$$
$$\ell_i(d,d') \equiv \ell_i(e(d),e(d')).$$

3. Learning Algorithms

In large-margin methods, we want to ensure that the difference, or *margin*, between the correct label and an incorrect label exceeds some minimum; in *margin scaling* (Crammer and Singer, 2003), this minimum is equal to the loss. That is, our learning problem is to minimize:

$$L(\mathbf{w}) = \frac{1}{N}\sum_i L_i(\mathbf{w}) \tag{4}$$

where

$$L_i(\mathbf{w}) = \max_{d\in\mathcal{D}(f_i)} v_i(\mathbf{w},d,d_i),$$
$$v_i(\mathbf{w},d,d_i) = \ell_i(d,d_i) - \mathbf{w}\cdot(\mathbf{h}(d_i)-\mathbf{h}(d)).$$

Note that since $d_i \in \mathcal{D}(f_i)$ and $v_i(\mathbf{w},d_i,d_i) = 0$, $L_i(\mathbf{w})$ is always nonnegative. We now review the derivations of several existing algorithms for optimizing (4) for structured models.

3.1 Stochastic Gradient Descent

An easy way to optimize the objective function $L(\mathbf{w})$ is stochastic (sub)gradient descent (SGD) (Ratliff et al., 2006; Shalev-Shwartz et al., 2007). In SGD, we consider one component $L_i(\mathbf{w})$ of the objective function at a time and update $\mathbf{w}$ by the subgradient:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla L_i(\mathbf{w}), \tag{5}$$
$$\nabla L_i(\mathbf{w}) = -(\mathbf{h}(d_i)-\mathbf{h}(d^+))$$

where

$$d^+ = \arg\max_{d\in\mathcal{D}(f_i)} v_i(\mathbf{w},d,d_i).$$

If, as an approximation, we restrict $\mathcal{D}(f_i)$ to just the 1-best derivation of f_i, then we get the structured perceptron algorithm (Rosenblatt, 1958; Freund and Schapire, 1999; Collins, 2002). Otherwise, we get Algorithm 1. Note that, as is common practice with the perceptron, the final weight vector is the average of the weight vector at each iteration. (Line 6 as implemented here can be inefficient; in practice, we use the trick of Daumé III (2006, p. 19) to average efficiently.)

The derivation d^+ is the worst violator of our constraint that the margin be greater than or equal to the loss, and appears frequently in large-margin learning algorithms. We call d^+ the *fear derivation*.[2] An easy way to approximate the fear derivation would be to generate an n-best list and select the derivation from it that maximizes v_i. In Section 5 we discuss better ways to search for the fear derivation.

[2] The terminology of *fear* derivations and *hope* derivations to be defined below are due to Kevin Knight.

Algorithm 1 Stochastic gradient descent

```
Require: training examples $(f_1,e_1),\ldots,(f_N,e_N)$
$\mathbf{w} \leftarrow \mathbf{0}$
$\mathbf{s} \leftarrow \mathbf{0}, t \leftarrow 0$
while not converged do
    for $i \in \{1,\ldots,N\}$ in random order do
        UPDATEWEIGHTS($\mathbf{w}, i$)
        $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{w}$
        $t \leftarrow t + 1$
$\mathbf{w} \leftarrow \mathbf{s}/t$

procedure UPDATEWEIGHTS($\mathbf{w}$, i)
    $d^+ \leftarrow \arg\max_{d \in \mathcal{D}(f_i)} v_i(\mathbf{w}, d, d_i)$
    $\mathbf{w} \leftarrow \mathbf{w} + \eta(\mathbf{h}(d_i) - \mathbf{h}(d^+))$
```

3.2 MIRA

Kivinen and Warmuth (1996) derive SGD from the following update:

$$\mathbf{w} \leftarrow \arg\min_{\mathbf{w}'} \left(\frac{1}{2\eta} \|\mathbf{w}' - \mathbf{w}\|^2 + L_i(\mathbf{w}') \right) \tag{6}$$

where the first term, the *conservativity* term, prevents us from moving too far in a single iteration. Taking partial derivatives and setting to zero, we get

$$\mathbf{w}' - \mathbf{w} + \eta \nabla L_i(\mathbf{w}') = 0.$$

If we make the approximation $\nabla L_i(\mathbf{w}') \approx \nabla L_i(\mathbf{w})$, we get the gradient-descent update again:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla L_i(\mathbf{w}).$$

But the advantage of using (6) without approximation is that it will not overshoot the optimum if the step size η happens to be too large. This is the Margin Infused Relaxed Algorithm (MIRA) of Crammer and Singer (2003).

The MIRA update (6) replaces the procedure UPDATEWEIGHTS in Algorithm 1. It is more commonly presented as a quadratic program (QP):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2\eta} \|\mathbf{w}' - \mathbf{w}\|^2 + \xi_i \\ \text{subject to} \quad & v_i(\mathbf{w}', d, d_i) - \xi_i \leq 0 \qquad \forall d \in \mathcal{D}(f_i) \end{aligned}$$

where ξ_i is a slack variable.[3] (Note that $\xi_i \geq 0$ since $d_i \in \mathcal{D}(f_i)$ and $v_i(\mathbf{w}', d_i, d_i) = 0$.) The Lagrangian is:

$$\mathcal{L} = \frac{1}{2\eta} \|\mathbf{w}' - \mathbf{w}\|^2 + \xi_i + \sum_{d \in \mathcal{D}(f_i)} \alpha_d (v_i(\mathbf{w}', d, d_i) - \xi_i). \tag{7}$$

[3] Watanabe et al. (2007) use a different slack variable ξ_{id} for each hypothesis d, which leads to a different update than the one derived below.

Setting partial derivatives to zero gives:

$$\mathbf{w}' = \mathbf{w} + \eta \sum_{d \in \mathcal{D}(f_i)} \alpha_d (\mathbf{h}(d_i) - \mathbf{h}(d))$$

$$\sum_{d \in \mathcal{D}(f_i)} \alpha_d = 1.$$

Substituting back into (7), we get the following dual problem:

$$\begin{aligned} \text{maximize} \quad & -\frac{\eta}{2} \left\| \sum_{d \in \mathcal{D}(f_i)} \alpha_d (\mathbf{h}(d_i) - \mathbf{h}(d)) \right\|^2 + \sum_{d \in \mathcal{D}(f_i)} \alpha_d v_i(\mathbf{w}, d, d_i) \\ \text{subject to} \quad & \sum_{d \in \mathcal{D}(f_i)} \alpha_d = 1 \\ & \alpha_d \geq 0 \qquad \forall d \in \mathcal{D}(f_i). \end{aligned}$$

In machine translation, and in structured prediction in general, the number of hypotheses in $\mathcal{D}(f_i)$, and therefore the number of constraints in the QP, can be exponential or worse. Watanabe et al. (2007) use the 1 best or 10 best hypotheses. In an earlier version of this work (Chiang et al., 2008b), we used the top 10 fear derivations.[4] Here, we use the cutting-plane algorithm of Tsochantaridis et al. (2004), which repeatedly recomputes the fear derivation and adds it to a working set $\mathcal{S}_i$ of derivations on which the QP is optimized (Algorithm 2). A new fear derivation is added to the working set only if it is a worse violator by a certain margin (ε); otherwise, the algorithm terminates.

The procedure OPTIMIZESET solves the QP restricted to $\mathcal{S}_i$ by sequential minimal optimization (Platt, 1998), in which we repeatedly select a pair of derivations d', d'' and optimize their dual variables $\alpha_{d'}, \alpha_{d''}$. The function SELECTPAIR uses the heuristics suggested by Taskar (2004, p. 80) to select a pair of constraints: one must violate one of the KKT conditions ($\alpha_d(v_i(\mathbf{w}', d, d_i) - \xi_i) = 0$), and the other must allow the objective to be improved. The procedure OPTIMIZEPAIR optimizes a single pair of dual variables. This optimization is exact and can be derived as follows. Suppose we have current suboptimal weights $\mathbf{w}(\alpha) = \mathbf{w} + \eta \sum \alpha_d (\mathbf{h}(d_i) - \mathbf{h}(d))$, and we want to increase $\alpha_{d'}$ by δ and decrease $\alpha_{d''}$ by δ. Then we get the following optimization in a single variable, δ:

$$\begin{aligned} \text{maximize} \quad & -\frac{\eta}{2} \left\| \sum_d \alpha_d (\mathbf{h}(d_i) - \mathbf{h}(d)) + \delta(-\mathbf{h}(d') + \mathbf{h}(d'')) \right\|^2 + \delta(v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)) \\ \text{subject to} \quad & -\alpha_{d'} \leq \delta \leq \alpha_{d''}. \end{aligned} \tag{8}$$

Setting the partial derivative with respect to δ equal to zero, we get

$$\begin{aligned} \delta &= \frac{\eta \sum_d \alpha_d (\mathbf{h}(d_i) - \mathbf{h}(d)) \cdot (\mathbf{h}(d') - \mathbf{h}(d'')) + v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)}{\eta \|\mathbf{h}(d') - \mathbf{h}(d'')\|^2} \\ &= \frac{(\mathbf{w}(\alpha) - \mathbf{w}) \cdot (\mathbf{h}(d') - \mathbf{h}(d'')) + v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)}{\eta \|\mathbf{h}(d') - \mathbf{h}(d'')\|^2} \\ &= \frac{v_i(\mathbf{w}(\alpha), d', d_i) - v_i(\mathbf{w}(\alpha), d'', d_i)}{\eta \|\mathbf{h}(d') - \mathbf{h}(d'')\|^2}. \end{aligned}$$

[4] More accurately, we took the union of the 10 best derivations, the top 10 fear derivations, and the top 10 hope derivations (to be defined below).

Algorithm 2 MIRA weight update (Tsochantaridis et al., 2004; Platt, 1998; Taskar, 2004)

```
procedure UpdateWeights($\mathbf{w}$, i)
    $\varepsilon = 0.01$
    $\mathcal{S}_i \leftarrow \{d_i\}$
    $again \leftarrow$ true
    while $again$ do
        $again \leftarrow$ false
        $d^+ \leftarrow \arg\max_{d \in \mathcal{D}(f_i)} v_i(\mathbf{w}, d, d_i)$
        if $v_i(\mathbf{w}, d^+, d_i) > \max_{d \in \mathcal{S}_i} v_i(\mathbf{w}, d, d_i) + \varepsilon$ then
            $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{d^+\}$
            OptimizeSet($\mathbf{w}, i$)
            $again \leftarrow$ true

procedure OptimizeSet($\mathbf{w}, i$)
    $\alpha_d \leftarrow 0$ for $d \in \mathcal{S}_i$
    $\alpha_{d_i} \leftarrow 1$
    $iterations \leftarrow 0$
    while $iterations < 1000$ do
        $iterations \leftarrow iterations + 1$
        $d', d'' \leftarrow$ SelectPair($\mathbf{w}, i$)
        if $d', d''$ not defined then
            return
        OptimizePair($\mathbf{w}, i, d', d''$)

function SelectPair($\mathbf{w}, i$)
    $\varepsilon = 0.01$
    for $d' \in \mathcal{S}_i$ do
        $v_{max} \leftarrow \max_{d'' \neq d'} v_i(\mathbf{w}, d'', d_i)$
        if $\alpha_{d'} = 0$ and $v_i(\mathbf{w}, d', d_i) > v_{max} + \varepsilon$ then
            if $\exists d'' \neq d'$ such that $\alpha_{d''} > 0$ then
                return $d', d''$
        if $\alpha_{d'} > 0$ and $v_i(\mathbf{w}, d', d_i) < v_{max} - \varepsilon$ then
            if $\exists d'' \neq d'$ such that $v_i(\mathbf{w}, d'', d_i) > v_i(\mathbf{w}, d', d_i)$ then
                return $d', d''$
    return undefined

procedure OptimizePair($\mathbf{w}, i, d', d''$)
    $\delta \leftarrow \dfrac{v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)}{\eta \|\mathbf{h}(d') - \mathbf{h}(d'')\|^2}$
    $\delta \leftarrow \max(-\alpha_{d'}, \min(\alpha_{d''}, \delta))$
    $\alpha_{d'} \leftarrow \alpha_{d'} + \delta$; $\alpha_{d''} \leftarrow \alpha_{d''} - \delta$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta\delta(\mathbf{h}(d') - \mathbf{h}(d''))$
```

But in order to maintain constraint (8), we clip δ to the interval $[-\alpha_{d'}, \alpha_{d''}]$ (line 35).

At the end of training, following McDonald et al. (2005), we average all the weight vectors obtained at each iteration, just as in the averaged perceptron.

3.3 AROW

The conservativity term in (6) assumes that it is equally risky to move $\mathbf{w}$ in any direction, but this is not the case in general. For example, even a small change in the language model weights could result in a large change in translation length and fluency, whereas large changes in features like those attached to number-translation rules have a relatively small effect.

Imagine that we choose a feature of our model, h_j, and replace it with the feature $h_j \cdot c$ while replacing its weight with w_j/c. This change has no effect on the scores assigned to derivations or the translations generated, so intuitively one would hope that it also has no effect on learning. However, it is easy to see that our online algorithms in fact apply updates that are c times bigger, and relative to the new weight, c^2 times bigger.

A number of approaches are suggested in the literature to address this problem, for example, the second-order perceptron (Cesa-Bianchi et al., 2005), confidence-weighted learning (Dredze et al., 2008), and Adaptive Regularization of Weights or AROW (Crammer et al., 2009). AROW replaces the weight vector $\mathbf{w}$ with a Gaussian distribution over weight vectors, $\mathcal{N}(\mathbf{w}, \Sigma)$. The conservativity term in (6) accordingly changes from a Euclidean distance to a Kullback-Leibler distance. In addition, a new term is introduced that causes the confidence in the weights to increase over time (in AROW's predecessor (Dredze et al., 2008), it was motivated as the variance of L_i).

$$\mathbf{w}, \Sigma \leftarrow \arg\min_{\mathbf{w}', \Sigma'} \left(\mathrm{KL}\left(\mathcal{N}(\mathbf{w}', \Sigma') \,\|\, \mathcal{N}(\mathbf{w}, \Sigma)\right) + L_i(\mathbf{w}') + \frac{\lambda}{2} \mathbf{x}^\mathsf{T} \Sigma' \mathbf{x} \right).$$

In the original formulation of AROW for binary classification, $\mathbf{x}$ is the instance vector. Here, we set it to $\sum_{d \in \mathcal{S}_i} \alpha_d \left(\mathbf{h}(d_i) - \mathbf{h}(d)\right)$, even though the α_d aren't known in advance; in practice, they are known by the time they are needed.

With the KL distance between the two Gaussians written out explicitly, the quantity we want to minimize is:

$$\frac{1}{2} \left(\log \frac{\det \Sigma}{\det \Sigma'} + \mathrm{Tr}\left(\Sigma^{-1} \Sigma'\right) + (\mathbf{w}' - \mathbf{w})^\mathsf{T} \Sigma^{-1} (\mathbf{w}' - \mathbf{w}) - D \right) + L_i(\mathbf{w}') + \frac{\lambda}{2} \mathbf{x}^\mathsf{T} \Sigma' \mathbf{x}$$

where D is the number of features. We minimize with respect to $\mathbf{w}'$ and Σ' separately. If we drop terms not depending on $\mathbf{w}'$, we get:

$$\mathbf{w} \leftarrow \arg\min_{\mathbf{w}'} \frac{1}{2} (\mathbf{w}' - \mathbf{w})^\mathsf{T} \Sigma^{-1} (\mathbf{w}' - \mathbf{w}) + L_i(\mathbf{w}')$$

which is the same as MIRA (6) except that Σ has taken the place of η. This leads to Algorithm 3, which modifies Algorithm 2 in two ways. First, line 34 is replaced with:

$$\delta \leftarrow \frac{v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)}{(\mathbf{h}(d') - \mathbf{h}(d''))\Sigma(\mathbf{h}(d') - \mathbf{h}(d''))}$$

and line 37 is replaced with:

$$\mathbf{w} \leftarrow \mathbf{w} - \Sigma \delta (\mathbf{h}(d') - \mathbf{h}(d'')).$$

Next, we turn to Σ. Setting partial derivatives with respect to Σ' to zero, and using the fact that Σ' is symmetric, we get (Petersen and Pedersen, 2008):

$$\frac{1}{2}\left(-\Sigma'^{-1} + \Sigma^{-1}\right) + \frac{\lambda}{2}\mathbf{x}^{\mathsf{T}}\mathbf{x} = 0.$$

This leads to the AROW update, which follows the update for $\mathbf{w}$ (line 5 in Algorithm 1):

$$\Sigma^{-1} \leftarrow \Sigma^{-1} + \lambda\mathbf{x}^{\mathsf{T}}\mathbf{x}.$$

We initialize Σ to $\eta_0 I$ and then update it at each iteration using this update; following Crammer et al. (2009), we keep only the diagonal elements of Σ.

Algorithm 3 AROW (Crammer et al., 2009)

```
Require: training examples $(f_1, e_1), \ldots, (f_N, e_N)$
$\mathbf{w} \leftarrow \mathbf{0}$
$\Sigma \leftarrow \eta_0 I$
$\mathbf{s} \leftarrow \mathbf{0}, t \leftarrow 0$
while not converged do
    for $i \in \{1, \ldots, N\}$ in random order do
        UPDATEWEIGHTS($\mathbf{w}, i$)                              ▷ Algorithm 2
        $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{w}$
        $t \leftarrow t + 1$
        $\mathbf{x} \leftarrow \sum_{d \in \mathcal{S}_i} \alpha_d\,(\mathbf{h}(d_i) - \mathbf{h}(d))$
        $\Sigma^{-1} \leftarrow \Sigma^{-1} + \lambda\,\mathrm{diag}(x_1^2, \ldots, x_n^2)$
$\mathbf{w} \leftarrow \mathbf{s}/t$

procedure OPTIMIZEPAIR($\mathbf{w}, i, d', d''$)
    $\delta \leftarrow \frac{v_i(\mathbf{w}, d', d_i) - v_i(\mathbf{w}, d'', d_i)}{(\mathbf{h}(d') - \mathbf{h}(d''))\Sigma(\mathbf{h}(d') - \mathbf{h}(d''))}$
    $\delta \leftarrow \max(-\alpha_{d'}, \min(\alpha_{d''}, \delta))$
    $\alpha_{d'} \leftarrow \alpha_{d'} + \delta$
    $\alpha_{d''} \leftarrow \alpha_{d''} - \delta$
    $\mathbf{w} \leftarrow \mathbf{w} - \Sigma\delta(\mathbf{h}(d') - \mathbf{h}(d''))$
.
```

4. The Reference Derivation

We have been assuming that d_i is the derivation of the reference translation e_i. However, this is not always possible or even desirable. In this section, we discuss some alternative choices for d_i.

4.1 Bold/Max-BLEU Updating

It can happen that there does not exist any derivation of e_i, for example, if e_i contains a word never seen before in training. In this case, Liang et al. (2006), in the scheme they call *bold updating*, simply skip the sentence. Another approach, called *max*-BLEU *updating* (Tillmann and Zhang, 2006; Arun and Koehn, 2007), is to try to find the derivation with the highest BLEU score. However, Liang et al. find that even when it is possible to find a d_i that exactly generates e_i, it is not necessarily desirable to update the model towards it, because it may be a *bad derivation* of a *good translation*.

For example, consider the following Arabic sentence (written left-to-right in Buckwalter romanization) with English glosses:

sd	qTEp	mn	AlkEk	AlmmlH	“	brytzl	”	Hlqh	.
blocked	piece	of	biscuit	salted	“	pretzel	”	his-throat	.

A very literal translation might be,

> A piece of a salted biscuit, a “pretzel,” blocked his throat.

But the reference translation is in fact:

> A pretzel, a salted biscuit, became lodged in his throat.

While accurate, this translation swaps grammatical roles in a way that is still difficult for statistical MT systems to model. If the system happens to have some bad rules that translate *sd qTEp mn* as *a pretzel* and *“ brytzl ”* as *became lodged in*, then it can use these bad rules to obtain a perfect translation, but using this derivation as the reference derivation would only reinforce the use of these bad rules. A derivation of the more literal translation would probably serve better as the reference translation. What we need is a *good derivation* of a *good translation*.

4.2 Local Updating

The most common way to do this has been to generate the *n*-best derivations according to the model and to choose the one with the lowest loss (Och and Ney, 2002). Liang et al. (2006) call this *local updating*. Watanabe et al. (2007) generate a 1000-best list and select either the derivation with lowest loss or the 10 derivations with lowest loss. The idea is that restricting to derivations with a higher model score will filter out derivations that use bad, low-probability rules. Normally one uses an *n*-best list as a proxy for the whole space of derivations, so that the larger *n* is, the better; in this case, however, as *n* increases, local updating approaches max-BLEU updating, which is what we are trying to avoid. It is not clear what the optimal *n* is, and whether it depends on factors such as sentence length or pruning.

4.3 Hope Derivations

Here, we propose an approach that ties the choice of d_i more closely to the model. We suppose that for each f_i, the reference derivation d_i is unknown, and it doesn't necessarily derive the reference translation e_i, but we add a term to the objective function that says that we want d_i to have low loss relative to e_i.

$$\mathbf{w} \leftarrow \arg\min_{\mathbf{w}'} \min_{d_i \in \mathcal{D}(f_i)} \left(\frac{1}{2\eta} \|\mathbf{w}' - \mathbf{w}\|^2 + \max_{d \in \mathcal{D}(f_i)} v_i(\mathbf{w}', d, d_i) + (1-\mu)\ell_i(d_i, e_i) \right).$$

The parameter $\mu < 0$ controls how strongly we want d_i to have low loss.

We first optimize with respect to d_i, holding $\mathbf{w}'$ constant. Then the optimization reduces to

$$d_i = \arg\max_{d \in \mathcal{D}(f_i)} \left(\mu \ell_i(d, e_i) + \mathbf{w} \cdot \mathbf{h}(d)\right). \tag{9}$$

Then, we optimize with respect to $\mathbf{w}'$, holding d_i constant. Since this is identical to (6), we can use any of the algorithms presented in Section 3.

We call d_i chosen according to (9) the *hope* derivation. Unlike the fear derivation, it is parameterized by μ. If we let $\mu = -1$, the definition of the hope derivation becomes conveniently symmetric with the fear derivation:

$$d_i = \arg\max_{d \in \mathcal{D}(f_i)} (-\ell_i(d, e_i) + \mathbf{w} \cdot \mathbf{h}(d)).$$

Both the hope and fear derivations try to maximize the model score, but the fear derivation maximizes the loss whereas the hope derivation minimizes the loss.

5. Searching for Hope and Fear

As mentioned above, one simple way of approximating either the hope or fear derivation is to generate an *n*-best list and choose from it the derivation that maximizes (9) or v_i, respectively. But Figure 1 shows that this approximation can be quite poor in practice, because the *n*-best list covers such a small portion of the entire search space. Increasing *n* would help (and, unlike with local updating, the larger *n* is, the better), but could become inefficient.

Instead, we use a dynamic program, analogous to the Viterbi algorithm, to directly search for the hope/fear derivations in the forest. (For efficiency, we reuse the forest that is previously used to search for the Viterbi derivation—an approximation, because this forest is pruned using the model score.) If our loss function were decomposable onto hyperedges, this would be a simple matter of setting the hyperedge weights to $\mathbf{w} \cdot \mathbf{h}(v \to \mathbf{v}) \pm \ell_i(v \to \mathbf{v})$ and running the Viterbi algorithm. However, our loss function is not hyperedge-decomposable, so we must resort to approximations.

5.1 Towards Hyperedge-level BLEU

We begin by attempting to decompose the component scores $\mathbf{b}$ onto hyperedges. First, we need to be able to calculate $g_k(v \to \mathbf{v})$, the set of k-grams introduced by the hyperedge $(v \to \mathbf{v})$. This turns out to be fairly easy, because nearly all decoder implementations have a mechanism for scoring a k-gram language model, which is a feature of the form

$$h_{\mathrm{LM}_k}(d) = \sum_{w_1 \cdots w_k \in g_k(e(d))} \log P(w_k \mid w_1 \cdots w_{k-1}).$$

Since h_{LM_k} is decomposable onto hyperedges by assumption, it is safe to assume that g_k is also decomposable onto hyperedges, and so is n_k, which is the cardinality of g_k.

But m_k is not as easy to decompose, because of "clipping" of k-gram matches. Suppose our reference sentence is

Australia is one of the few countries that have diplomatic relations with North Korea

and we have two partial translations

the few

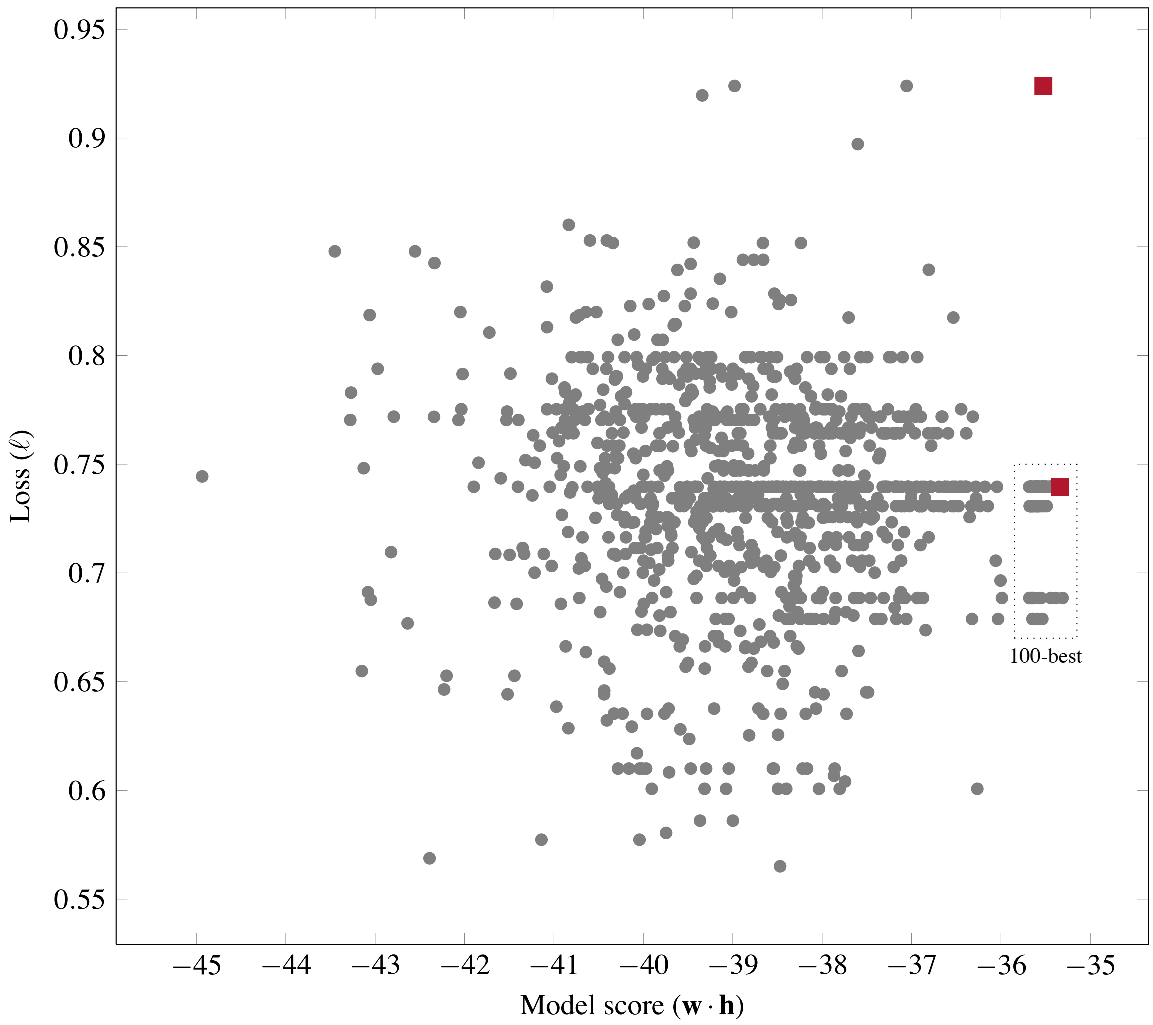


Figure 1: Using loss-augmented inference to search for fear translations in the whole forest is better than searching in the *n*-best list. Each point represents a derivation. The red square in the upper-right is the fear derivation obtained by loss-augmented inference, whereas the red square inside the box labeled "100-best" is the fear derivation selected from the 100-best list. (The gray circles outside the box are 100 random samples from the forest.)

the countries

then for both, $m_1 = 2$. But if we combine them into

the few the countries

then m_1 is not $2+2=4$, but 3, because *the* only occurs once in the reference sentence. In order to decompose m_k exactly, we would have to structure the forest hypergraph so that subderivations with different g_k are rooted at different nodes, resulting in an exponential blowup. Therefore, following Dreyer et al. (2007), we use *unclipped* counts of n-gram matches, which are not limited to the number of occurrences in the reference(s), in place of (1):

$$m_k = |g_k(c) \cap g_k(r)^*|.$$

These counts are easily decomposable onto hyperedges.

Finally, in order to decompose ρ, if there are multiple references, we can't use the standard definition of ρ in (2); instead we use the average reference length. Then we can apportion ρ among hyperedges according to how much of the input sentence they consume:

$$\rho(v \to \mathbf{v}) = \frac{\rho}{|f_i|} \left(|f(v)| - \sum_{v' \in \mathbf{v}} |f(v')| \right) \tag{10}$$

where $f(v)$ is the part of the input sentence covered by the subderivation rooted at v.

5.2 Forest Reranking

Appendix A.3, following Tromble et al. (2008), describes a way to fully decompose BLEU onto hyperedges. Here, however, we follow Dreyer et al. (2007), who use a special case of forest reranking (Huang, 2008). To search for the hope or fear derivation, we use the following dynamic program:

$$\begin{aligned} \mathit{vderiv}(v) &= \mathop{\arg\max}_{d \in \{\mathit{vderiv}(v \to \mathbf{v})\}} \phi(d) \\ \mathit{vderiv}(v \to \mathbf{v}) &= \{v \to \mathbf{v}\} \cup \bigcup_{v' \in \mathbf{v}} \mathit{vderiv}(v') \end{aligned}$$

where ϕ is one of the following:

$$\begin{aligned} \phi(d) &= \mathbf{w} \cdot \mathbf{h}(d) + B(\mathbf{b}(d, e_i)) && \text{(hope)}, \\ \phi(d) &= \mathbf{w} \cdot \mathbf{h}(d) - B(\mathbf{b}(d, e_i)) && \text{(fear)}. \end{aligned}$$

Note that maximizing $\mathbf{w} \cdot \mathbf{h}(d) + B(\mathbf{b}(d, e_i))$ is equivalent to maximizing $\mathbf{w} \cdot \mathbf{h}(d) - \ell_i(d, e_i)$, since they differ by only a constant; likewise, maximizing $\mathbf{w} \cdot \mathbf{h}(d) - B(\mathbf{b}(d, e_i))$ is equivalent to maximizing $\mathbf{w} \cdot \mathbf{h}(d) + \ell_i(d, e_i)$.

This algorithm is not guaranteed to find the optimum, however. We illustrate with a counterexample, using BLEU-2 (i.e., $K = 2$) instead of BLEU-4 for simplicity. Suppose our reference sentence is as above, and we have two partial candidate sentences

1. one of the few nations which maintain ties with the DPRK has been

2. North Korea with relations diplomatic have that countries few the of one is

Translation #1 has 4 unigram matches and 3 bigram matches, for a BLEU-2 score of $\sqrt{12/156}$; translation #2 has 13 unigram matches and 1 bigram match, for a BLEU-2 score of $\sqrt{13/156}$. If we extend both translations, however, with the word *Australia*, giving them each an extra unigram match, then translation #1 gets a BLEU-2 score of $\sqrt{15/156}$, and translation #2, $\sqrt{14/156}$. Though it does not always find the optimum, it works well enough in practice. After we find a hope or fear derivation, we recalculate its exact BLEU score, without any of the approximations described in this section.

6. Parallelization

Because inference is so slow for the translation task, and especially for the CKY-based decoder we are using, parallelization is critical. Batch learning algorithms like MERT are embarrassingly parallel, but parallelization of online learning is an active research area. Two general strategies have been proposed for SGD. The simpler strategy is to run p learners in parallel and then average their final weight vectors afterward (Mann et al., 2009; McDonald et al., 2010; Zinkevich et al., 2010). The more communication-intensive option, known as *asynchronous* SGD, is to maintain a single weight vector and for p parallel learners to update it simultaneously (Langford et al., 2009; Gimpel et al., 2010). It is not actually necessary for a learner to wait for the others to finish computing their updates; it can simply update the weight vector and move to the next example.

6.1 Iterative Parameter Mixing

A compromise between the two is *iterative parameter mixing* (McDonald et al., 2010), in which a master node periodically averages the weight vectors of the learners. At the beginning of each epoch, a master node broadcasts the same initial weight vector to p learners, which run in parallel over the training data and send their weight vectors back to the master node. The master averages the p weight vectors together to obtain the initial weight vector for the next epoch. At the end of training, the weight vectors from each iteration of each learner are all averaged together to yield the final weight vector.

6.2 Asynchronous MIRA/AROW

In asynchronous SGD, when multiple learners make simultaneous updates to the master weight vector, the updates are simply summed. Our experience is that this works, but requires carefully throttling back the learning rate η. Here, we focus on asynchronous parallelization of MIRA/AROW. The basic idea is to build forests for several examples in parallel, and optimize the QP over all of them together. However, this would require keeping the forests of all the examples in a shared memory, which would probably be too expensive. Instead, the solution we have adopted (Algorithm 4) is for the learners to broadcast just the working sets $\mathcal{S}_i$ to one another, rather than whole forests. Thus, when each learner works on a training example (f_i, e_i), it optimizes the QP on it along with all of the working sets it received from other nodes. It can grow the working set $\mathcal{S}_i$, but not the working sets it received from other nodes. For AROW, each node maintains its own Σ in addition to its own $\mathbf{w}$.

Algorithm 4 Asynchronous MIRA

```
$\mathbf{w}_k \leftarrow \mathbf{0}$ **for** each node $k$
$\mathbf{s}_k \leftarrow \mathbf{0}, t_k \leftarrow 0$ **for** each node $k$
**while** not converged **do**
    $T \leftarrow$ training data
    **for** each node $k$ in parallel **do**
        **while** $T \neq \emptyset$ **do**
            pick a random $(f_i, e_i)$ from $T$ and remove it
            receive working sets $\{\mathcal{S}_{i'} \mid i' \in I\}$ from other nodes
            $\textsc{UpdateWeights}(\mathbf{w}_k, i, I)$
            broadcast $\mathcal{S}_i$ to other nodes
            $\mathbf{s}_k \leftarrow \mathbf{s}_k + \mathbf{w}_k$
            $t_k \leftarrow t_k + 1$
$\mathbf{w} \leftarrow \dfrac{\sum_k \mathbf{s}_k}{\sum_k t_k}$
**procedure** $\textsc{UpdateWeights}(\mathbf{w}, i, I)$
    $\varepsilon = 0.01$
    $\mathcal{S}_i \leftarrow \{d_i\}$
    $\mathit{again} \leftarrow$ true
    **while** $\mathit{again}$ **do**
        $\mathit{again} \leftarrow$ false
        $d^+ \leftarrow \arg\max_{d \in \mathcal{D}(f_i)} v_i(\mathbf{w}, d, d_i)$
        **if** $v_i(\mathbf{w}, d^+, d_i) > \max_{d \in \mathcal{S}_i} v_i(\mathbf{w}, d, d_i) + \varepsilon$ **then**
            $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{d^+\}$
            $\mathit{again} \leftarrow$ true
        **if** $\mathit{again}$ **then**
            $\textsc{OptimizeSets}(\mathbf{w}, \{i\} \cup I)$

**procedure** $\textsc{OptimizeSets}(\mathbf{w}, I)$
    **for** $i \in I$ **do**
        $\alpha_d \leftarrow 0$ **for** $d \in \mathcal{S}_i$
        $\alpha_{d_i} \leftarrow 1$
    $\mathit{again} \leftarrow$ true
    $\mathit{iterations} \leftarrow 0$
    **while** $\mathit{again}$ and $\mathit{iterations} < 1000$ **do**
        $\mathit{again} \leftarrow$ false
        $\mathit{iterations} \leftarrow \mathit{iterations} + 1$
        **for** $i \in I$ **do**
            $d', d'' \leftarrow \textsc{SelectPair}(\mathbf{w}, i)$                    ▷ Algorithm 2
            **if** $d', d''$ defined **then**
                $\textsc{OptimizePair}(\mathbf{w}, i, d', d'')$
                $\mathit{again} \leftarrow$ true
```

7. Experiments

We experimented with the methods described above on the hierarchical phrase-based translation system Hiero (Chiang, 2005, 2007), using two feature sets. The *small* model comprises 13 features: 7 inherited from Pharaoh (Koehn et al., 2003), a second language model, and penalties for the glue rule, identity rules, unknown-word rules, and two kinds of number/name rules. The *large* model additionally includes the following lexical features:

- $\text{lex}(e)$ fires when an output word e is generated
- $\text{lex}(f,e)$ fires when an output word e is generated aligned to a input word f
- $\text{lex}(\text{NULL},e)$ fires when an output word e is generated unaligned

In all these features, f and e are limited to words occurring 10,000 times or more in the parallel data; less-frequent words are replaced with the special symbol UNK. Typically, this results in 10,000–20,000 features.

Our training data were all drawn from the constrained track of the NIST 2009 Open Machine Translation Evaluation. We extracted an Arabic-English grammar from all the allowed parallel data (152+175M words), and we trained two 5-gram language models, one on the combined English sides of the Arabic-English and Chinese-English tracks (385M words), and another on 2 billion words of English.

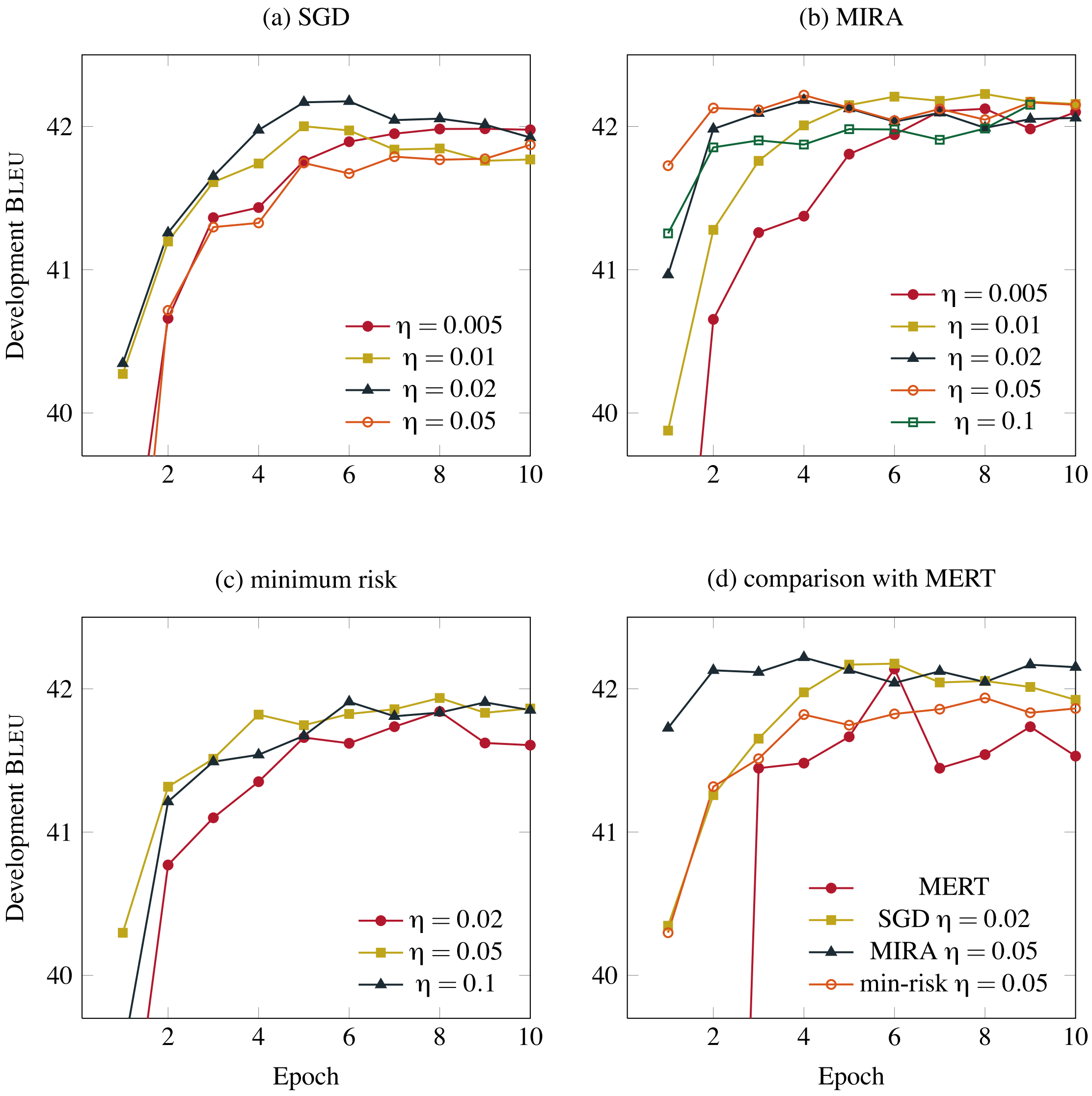
We ran discriminative training on 3011 lines (67k Arabic words) of newswire and web data drawn from the NIST 2004 and 2006 evaluations and newsgroup data from the GALE program (LDC2006E92). After each epoch (pass through the discriminative-training data), we used the averaged weights to decode our development data, which was from the NIST 2008 evaluation (1357 lines, 36k Arabic words). After 10 epochs, we chose the weights that yielded the highest BLEU on the development data and decoded the test data, which was from the NIST 2009 evaluation (1313 lines, 34k Arabic words).

Except where noted, the following default settings were used:

- Learning rate $\eta = 0.01$
- Hope derivations with $\mu = -1$
- Forest reranking for hope/fear derivations
- Iterative parameter mixing on 20 processors

A few probability features have to be initialized carefully: the two language models and the two phrase translation probability models. If these features are given negative weights, extremely long and disfluent translations result, and we find that the learner has difficulty recovering. So we initialize their weights to 1 instead of 0, and in AROW, we initialize their learning rates to 0.01 instead of η_0.

The learning curves in the figures referenced below show the BLEU score obtained on the development data (disjoint from the discriminative-training data) over time. Figure 2abc shows learning curves for SGD, MIRA, and minimum risk (see Appendix A) for several values of the learning rate η, using the small model. Generally, all the methods converged to the same performance level, and SGD and minimum risk were surprisingly not very sensitive to the learning rate η. MIRA, on the



(a) SGD

(b) MIRA

(c) minimum risk

(d) comparison with MERT

Figure 2: Learning curves of various algorithms on the development data, using the small model. Graphs (a), (b), and (c) show the effect of the learning rate η on SGD, MIRA, and minimum risk. SGD and min-risk seem relatively insensitive to η, while MIRA converges faster with higher η. Graph (d) compares the three online methods against MERT. The online algorithms converge more quickly and smoothly than MERT does, with MIRA slightly better than the others. The first two epochs of MERT, not shown here, had scores of 10.6 and 31.6.

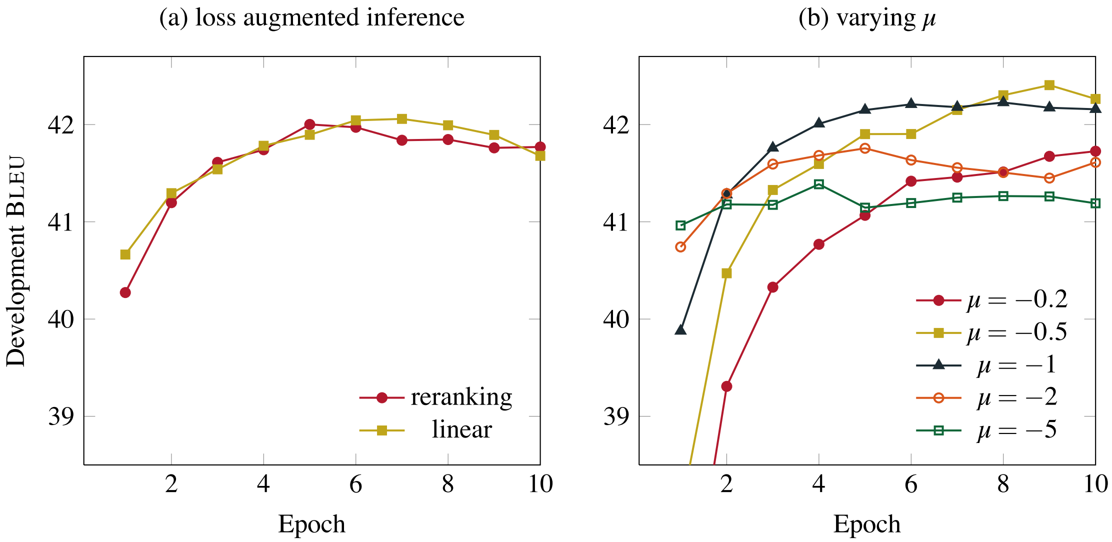


Figure 3: Variations on selecting hope/fear derivations, using the small model. (a) Linear BLEU performs as well as or slightly better than forest reranking. SGD, $\eta = 0.01$. (b) More negative values of the loss weight μ for hope derivations lead to higher initial performance, whereas less negative loss weights lead to higher final performance. MIRA, $\eta = 0.01$.

other hand, converged faster with higher learning rates up to $\eta = 0.05$. Since our past experience suggests that on tasks with lower BLEU scores (namely, Chinese-English web and speech), lower learning rates are better, our default $\eta = 0.01$ seems like a generally safe value.

Figure 2d compares all three algorithms with MERT (20 random restarts). The online algorithms converge more quickly and smoothly than MERT does, with MIRA converging faster than the others. However, on the test set (Table 1), MERT outperformed the other algorithms. Using bootstrap resampling with 1000 samples (Koehn, 2004; Zhang et al., 2004), only the difference with minimum risk was significant ($p < 0.05$).

One possible confounding factor in our comparison with minimum risk is that it must use linear BLEU to compute the gradient. To control for this, we ran SGD (on the hinge loss) using both forest reranking and linear BLEU to search for hope/fear derivations (Figure 3a). We found that their performance is quite close, strengthening our finding that the hinge loss performs slightly better than minimum risk.

Figure 3b compares several values of the parameter μ that controls how heavily to weight the loss function when computing hope derivations. Higher loss weights lead to higher initial performance, whereas lower loss weights lead to higher final performance (the exception being $\mu = -0.2$, which perhaps would have improved with more time). A weight of $\mu = -1$ appears to be a good tradeoff, and is symmetrical with the weight of 1 used when computing fear derivations. It would be interesting, however, to investigate decaying the loss weight over time, as proposed by McAllester et al. (2010).

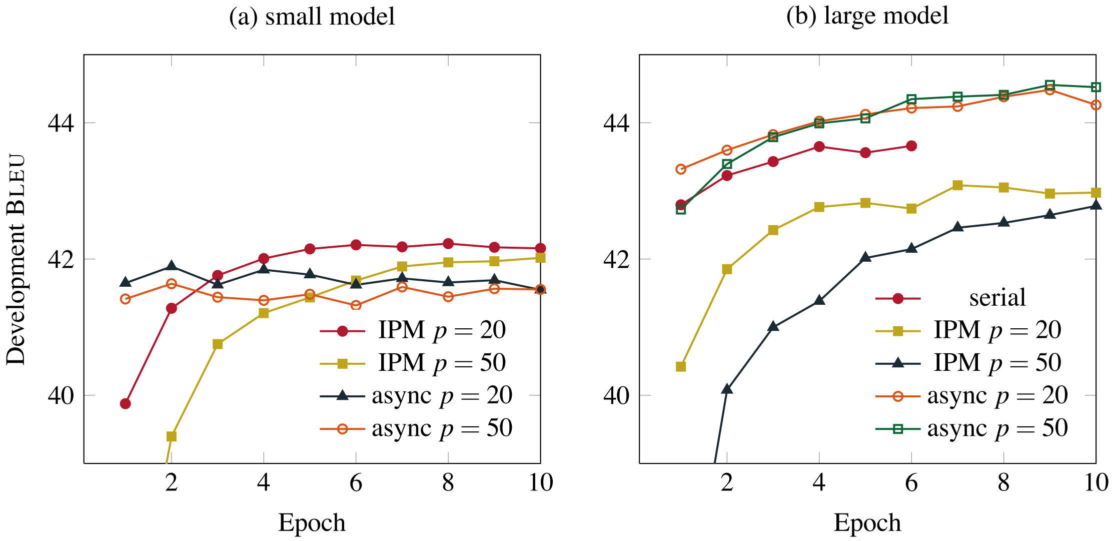


Figure 4: On the small model, asynchronous MIRA does not perform well compared to iterative parameter mixing. But on the large model, asynchronous MIRA strongly outperforms iterative parameter mixing. Increasing the number of processors to 50 provides little benefit to iterative parameter mixing in either case, whereas asynchronous MIRA gets a near-linear speedup.

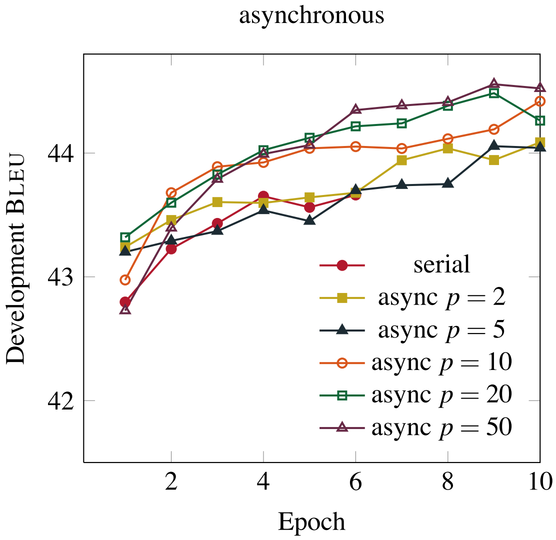


Figure 5: Taking a closer look at asynchronous sharing of working sets, we see that, at each epoch, greater parallelization generally gives better performance.

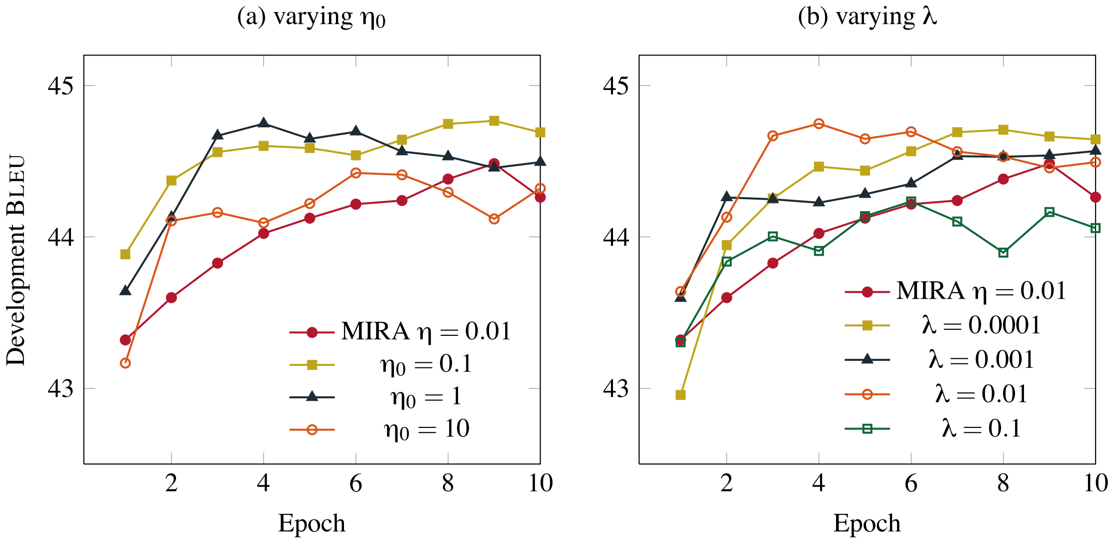


Figure 6: (a) With $\lambda = 0.01$, AROW seems relatively insensitive to the choice of η_0 in the range of 0.1 to 1, but performs much worse outside that range. (b) With $\eta_0 = 1$, AROW converges faster for larger values of λ up to 0.01; at 0.1, however, the algorithm appears to be unable to make progress.

We then compared the two methods of parallelization (Figure 4). These experiments were run on a cluster of nodes communicating by MPI (Message Passing Interface) over Myrinet, a high-speed local area networking system. In these graphs, the x-axis continues to be the number of epochs; wallclock time is roughly proportional to the number of epochs divided by p, but mixed hardware unfortunately prevented us from performing direct comparisons of wallclock time.

One might expect that, at each epoch, the curves with greater p underperform the curves with lower p only slightly. With iterative parameter mixing, for both the small and large models, we see that increasing p from 20 to 50 degrades performance considerably. It would appear that there is very little speedup due to parallelization, probably because the training data is so small (3011 sentences).

Asynchronous MIRA using the small model starts off well but afterwards does not do as well as iterative parameter mixing. On the large model, however, asynchronous MIRA performs dramatically better. Taking a closer look at its performance for varying p (Figure 5), we see that, at each epoch, the curves with greater p actually tend to outperform the curves with lower p.

Next, we tested the AROW algorithm. We held λ fixed to 0.01 and compared different values of the initial learning rate η_0 (Figure 6a), finding that the algorithm performed well for $\eta_0 = 0.1$ and 1 and was fairly insensitive to the choice of η_0 in that range; larger and smaller values, however, performed worse. We then held $\eta_0 = 1$ and compared different values of λ (Figure 6b), finding that higher values converged faster, but $\lambda = 0.1$ did much worse.

The scores on the test set (Table 1) using the large model generally confirm what was already observed on the development set. In total, the improvement over MERT on the test set is 2.4 BLEU.

model	obj	alg	approx	par	epoch	BLEU dev	BLEU test
small	$1 - \text{BLEU}$	MERT	–	–	6	42.1	45.2
small	hinge	SGD $\eta = 0.02$	rerank	IPM	6	42.2	44.9
small	risk	SGD $\eta = 0.05$	linear	IPM	8	41.9	44.8
small	hinge	MIRA $\eta = 0.05$	rerank	IPM	4	42.2	44.9
large	hinge	SGD $\eta = 0.01$	rerank	IPM	5	42.4	45.2
large	hinge	MIRA $\eta = 0.01$	rerank	IPM	7	43.1	45.9
large	hinge	MIRA $\eta = 0.01$	rerank	async	9	44.5	47.3
large	hinge	AROW $\eta_0 = 1$ $\lambda = 0.01$	rerank	async	4	44.7	47.6

Table 1: Final results. Key to columns: **model** = features used, **obj** = objective function, **alg** optimization algorithm, **approx** = approximation for calculating the loss function on forests, **par** = parallelization method, **epoch** = which epoch was selected on the development data, **dev** and **test** = (case-insensitive IBM) BLEU score on development and test data (NIST 2008 and 2009, respectively).

8. Conclusion

We have surveyed several methods for online discriminative training and the issues that arise in adapting these methods to the task of statistical machine translation. Using SGD, we found that the large-margin objective performs slightly better than minimum risk. Then, using the large-margin objective, we found that MIRA does better than SGD, and AROW, better still. We extended all of these methods in novel ways to cope with the large structured search space of the translation task, that is, to use as much of the translation forest as possible.

An apparent disadvantage of the large-margin objective is its requirement of a single correct derivation, which does not exist. We showed that the *hope* derivation serves this purpose well. We demonstrated that the highest-BLEU derivation is not in general the right choice, by showing that performance drops for very negative values of μ. We also raised the possibility, as yet unexplored, of decaying μ over time, as has been suggested by McAllester et al. (2010).

The non-decomposability of BLEU as a loss function is a nuisance that must be dealt with carefully. However, the choice of approximation (forest reranking versus linear BLEU) for loss-augmented inference or expectations turned out not to be very important. Past experience shows that linear BLEU sometimes outperforms and sometimes underperforms forest reranking, but since it is faster and easier to implement, it may be the better choice.

The choice of parallelization method turned out to be critical. We found that asynchronous sharing of working sets in MIRA/AROW not only gave speedups that were nearly linear in the number of processors, but also gave dramatically higher final BLEU scores than iterative parameter mixing. It is not clear yet whether this is because iterative parameter mixing was not able to converge in only 10 epochs or because aggregating working sets confers an additional advantage.

Although switching from MERT to online learning initially hurt performance, by adding some very simple features to the model, we ended up with a gain of 2.4 BLEU over MERT. When these online methods are implemented with due attention to translation forests, the nature of the translation problem, the idiosyncrasies of BLEU, and parallelization, they are a highly effective vehicle for exploring new extensions to discriminative models for translation.

Acknowledgments

This work evolved over time to support several projects adding new features to the ISI machine translation systems, and would not have been possible without my collaborators on those projects: Steve DeNeefe, Kevin Knight, Yuval Marton, Michael Pust, Philip Resnik, and Wei Wang. I also thank Michael Bloodgood, Michael Collins, John DeNero, Vladimir Eidelman, Kevin Gimpel, Chun-Nan Hsu, Daniel Marcu, Ryan McDonald, Fernando Pereira, Fei Sha, and the anonymous reviewers for their valuable ideas and feedback. This work was supported in part by DARPA under contract HR0011-06-C-0022 (subcontract to BBN Technologies), HR0011-09-1-0028, and DOI-NBC D11AP00244. *S.D.G.*

Appendix A. Minimum Risk Training

In this appendix, we describe minimum risk (expected loss) training (Smith and Eisner, 2006; Zens et al., 2008; Li and Eisner, 2009; Arun et al., 2010) and some notes on its implementation.

A.1 Objective Function

Define a probabilistic version of the model,

$$P_T(d \mid f_i) \propto \exp \frac{1}{T} \mathbf{w} \cdot \mathbf{h}(d)$$

where T is a temperature parameter, and for any random variable X over derivations, define

$$E_T[X \mid f_i] = \sum_{d \in \mathcal{D}(f_i)} P_T(d \mid f_i) X(d).$$

In minimum-risk training, we want to minimize $\sum_i E_T[\ell_i(d, d_i) \mid f_i]$ for $T = 1$. In annealed minimum-risk training (Smith and Eisner, 2006), we let $T \to 0$, in which case the expected loss approaches the loss.

This objective function is differentiable everywhere (unlike in MERT), though not convex (as maximum likelihood is). The gradient for a single example is:

$$\nabla E_T[\ell_i(d, d_i) \mid f_i] = \frac{1}{T} \left(E_T[\ell_i \mathbf{h} \mid f_i] - E_T[\ell_i \mid f_i] E_T[\mathbf{h} \mid f_i] \right)$$

or, in terms of B:

$$\begin{aligned} \nabla E_T[\ell_i(d, d_i) \mid f_i] &= -\nabla E_T[B(\mathbf{b}(d, e_i)) \mid f_i] \\ &= -\frac{1}{T} \left(E_T[B\mathbf{h} \mid f_i] - E_T[B \mid f_i] E_T[\mathbf{h} \mid f_i] \right). \end{aligned} \tag{11}$$

A major advantage that minimum-risk has over the large-margin methods explored in this paper is that it does not require a reference derivation, or a hope derivation as a proxy for the reference derivation. The main challenge with minimum-risk training is that we must calculate expectations of B and $B\mathbf{h}$. We discuss how this is done below.

A.2 Relationship to Hope/fear Derivations

There is an interesting connection between the risk and the generalized hinge loss (4). McAllester et al. (2010) show that for applications where the input space is continuous (as in speech processing), a perceptron-like update using the hope and 1-best derivations, or the 1-best and fear derivations, approaches the gradient of the loss. We provide here an analogous argument for the discrete input case.

Consider a single training example (f_i, e_i), so that we can simply write ℓ for ℓ_i and $E_T[X]$ for $E_T[X \mid f_i]$. Define a loss-augmented model:

$$P_\mu(d \mid f_i) \propto \exp \frac{1}{\mu} (\mathbf{w} \cdot \mathbf{h}(d) + \mu \ell(d, d_i))$$

and define

$$E_\mu[X] = \sum_{d \in \mathcal{D}(f_i)} P_\mu(d \mid f_i) X(d).$$

As before, the gradient with respect to $\mathbf{w}$ is:

$$\nabla_{\mathbf{w}} E_\mu[\ell] = \frac{1}{T} (E_\mu[\ell \mathbf{h}] - E_\mu[\ell] E_\mu[\mathbf{h}])$$

and, by the same reasoning, the partial derivative of $E[\mathbf{h}]$ with respect to μ comes out to be the same:

$$\frac{\partial}{\partial \mu} E_\mu[\mathbf{h}] = \frac{1}{T} (E_\mu[\mathbf{h} \ell] - E_\mu[\mathbf{h}] E_\mu[\ell]).$$

Therefore we have

$$\begin{aligned} \nabla_{\mathbf{w}} E[\ell] &= \left. \frac{\partial E_\mu[\mathbf{h}]}{\partial \mu} \right|_{\mu=0} \\ &= \lim_{\mu \to 0} \frac{1}{2\mu} (E_\mu[\mathbf{h}] - E_{-\mu}[\mathbf{h}]) \end{aligned}$$

which suggests the following update rule:

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta'}{2\mu} (E_\mu[\mathbf{h}] - E_{-\mu}[\mathbf{h}])$$

with μ decaying over time. But if we let $\mu = 1$ (that is, to approximate the tangent with a secant), and $\eta' = 2\eta$, we get:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta (E_{+1}[\mathbf{h}] - E_{-1}[\mathbf{h}]).$$

Having made this approximation, there is no harm in letting $T = 0$, so that the expectations of $\mathbf{h}$ become the value of $\mathbf{h}$ at the mode of the underlying distribution:

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} - \eta (\mathbf{h}(d_{+1}) - \mathbf{h}(d_{-1})), \\ d_{+1} &= \arg\max_d (\mathbf{w} \cdot \mathbf{h}(d) + \ell(d, d_i)), \\ d_{-1} &= \arg\max_d (\mathbf{w} \cdot \mathbf{h}(d) - \ell(d, d_i)). \end{aligned}$$

But this is exactly the SGD update on the generalized hinge loss (5), with $d^+ = d_{+1}$ being the fear derivation and $d_i = d_{-1}$ being the hope derivation.

A.3 Linear BLEU

In order to calculate the expected loss from a forest of derivations, we must make the loss fully decomposable onto hyperedges. Tromble et al. (2008) define a linear approximation to BLEU which they use for minimum Bayes risk decoding. We present here a version that includes the brevity penalty.

Suppose we have some fixed document with component scores $\overline{\mathbf{b}}$ and add a sentence to it that has component scores $\mathbf{b}$. How does adding the new sentence affect the BLEU score? Form a first-order Taylor approximation around $\overline{\mathbf{b}}$:

$$\begin{aligned}\text{BLEU}(\overline{\mathbf{b}}+\mathbf{b}) &\approx \text{BLEU}(\overline{\mathbf{b}}) + \mathbf{b}\cdot\nabla\text{BLEU}(\overline{\mathbf{b}}) \\ &= \text{BLEU}(\overline{\mathbf{b}})\left(1+\sum_{k=1}^{K}\left(\frac{m_k}{K\overline{m}_k}-\frac{n_k}{K\overline{n}_k}\right)+H\left(\overline{\rho}-\overline{n}_1\right)\left(\frac{\overline{\rho}n_1}{\overline{n}_1^2}-\frac{\rho}{\overline{n}_1}\right)\right)\end{aligned}$$

where

$$H(x)=\begin{cases}1 & \text{if } x\geq 0\\ 0 & \text{if } x<0.\end{cases}$$

Note that although the brevity penalty is not differentiable at $\overline{n}_1=\overline{\rho}$, we have filled in an arbitrary value (which is easier than smoothing the brevity penalty and works well in practice).

Since this approximation is linear in the m_k and n_k, it is decomposable onto hyperedges. The term involving ρ is the same for all derivations, so we don't need to decompose it and can also skip (10).

The approximation is highly dependent on $\overline{\mathbf{b}}$; Tromble et al. use a fixed $\overline{\mathbf{b}}$ but we use the oracle document defined in Section 2.4. Then B, defined as in (3) but using the linear approximation to BLEU, is decomposable down to hyperedges, making it possible to compute $E[B]$ as well as $E[b\mathbf{h}]$ over the entire forest.

A.4 Calculating the Risk and its Gradient

To calculate the expected loss, we can use the expectation semiring of Eisner (2002); we give a slightly modified definition that renormalizes intermediate values in such a way that they can be stored directly instead of as signed logarithms:

$$\begin{aligned}
expect_B(v) &= \sum_{(v\to\mathbf{v})\in E}\frac{inside_{\mathbf{w}\cdot\mathbf{h}}(v\to\mathbf{v})}{inside_{\mathbf{w}\cdot\mathbf{h}}(v)}expect_B(v\to\mathbf{v}), && (12)\\
expect_B(v\to\mathbf{v}) &= B(v\to\mathbf{v})+\sum_{v'\in\mathbf{v}}expect_B(v'), && (13)\\
inside_{\mathbf{w}\cdot\mathbf{h}}(v) &= \sum_{(v\to\mathbf{v})\in E} inside_{\mathbf{w}\cdot\mathbf{h}}(v\to\mathbf{v}),\\
inside_{\mathbf{w}\cdot\mathbf{h}}(v\to\mathbf{v}) &= \exp\mathbf{w}\cdot\mathbf{h}(v\to\mathbf{v})\times\prod_{v'\in\mathbf{v}} inside_{\mathbf{w}\cdot\mathbf{h}}(v').
\end{aligned}$$

To calculate the expected product $E_T[B\mathbf{h} \mid f_i]$ in the gradient (11), we use the second-order expectation semiring (Li and Eisner, 2009), similarly modified:

$$\begin{aligned} \mathit{expect}_{B\mathbf{h}}(v) &= \sum_{(v\to\mathbf{v})\in E} \frac{\mathit{inside}_{\mathbf{w}\cdot\mathbf{h}}(v\to\mathbf{v})}{\mathit{inside}_{\mathbf{w}\cdot\mathbf{h}}(v)} \mathit{expect}_{B\mathbf{h}}(v\to\mathbf{v}), \\ \mathit{expect}_{B\mathbf{h}}(v\to\mathbf{v}) &= \mathit{expect}_{B}(v\to\mathbf{v})\mathit{expect}_{\mathbf{h}}(v\to\mathbf{v}) \\ &\quad + \sum_{v'\in\mathbf{v}} \big(\mathit{expect}_{B\mathbf{h}}(v') - \mathit{expect}_{B}(v')\mathit{expect}_{\mathbf{h}}(v')\big) \end{aligned}$$

where $\mathit{expect}_{\mathbf{h}}$ is calculated analogously to expect_B (12–13).

References

Abhishek Arun and Philipp Koehn. Online learning methods for discriminative training of phrase based statistical machine translation. In *Proceedings of MT Summit XI*, 2007.

Abhishek Arun, Barry Haddow, and Philipp Koehn. A unified approach to minimum risk training and decoding. In *Proceedings of the Fifth Workshop on Statistical Machine Translation*, 2010.

Phil Blunsom, Trevor Cohn, and Miles Osborne. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL*, 2008.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19: 263–311, 1993.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, 2005.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2), 2007.

David Chiang. Learning to translate with source and target syntax. In *Proceedings of ACL*, 2010.

David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of EMNLP*, 2008a.

David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of EMNLP*, 2008b.

David Chiang, Wei Wang, and Kevin Knight. 11,001 new features for statistical machine translation. In *Proceedings of NAACL HLT*, 2009.

David Chiang, Steve DeNeefe, and Michael Pust. Two easy improvements to lexical weighting. In *Proceedings of ACL HLT*, 2011.

Michael Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, 2002.

Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, 2009.

Harold Charles Daumé III. *Practical Structured Learning Techniques for Natural Language Processing*. PhD thesis, University of Southern California, 2006.

Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of ICML*, 2008.

Markus Dreyer, Keith Hall, and Sanjeev Khudanpur. Comparing reordering constraints for SMT using efficient BLEU oracle computation. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, 2007.

Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of ACL*, 2002.

Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296, 1999.

Kevin Gimpel, Dipanjan Das, and Noah A. Smith. Distributed asynchronous online learning for natural language processing. In *Proceedings of CoNLL*, 2010.

Liang Huang. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*, 2008.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, 1996.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, 2004.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*, 2003.

John Langford, Alexander J. Smola, and Martin Zinkevich. Slow learners are fast. In *Advances in Neural Information Processing Systems 22*, 2009.

Zhifei Li and Jason Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of EMNLP*, 2009.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of COLING-ACL*, 2006.

Chin-Yew Lin and Franz Josef Och. ORANGE: A method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING*, 2004.

William N. Locke and A. Donald Booth, editors. *Machine Translation of Languages: Fourteen Essays*. Technology Press of MIT, Cambridge, MA, 1955.

Gideon Mann, Ryan McDonald, Mehryar Mohri, Nathan Silberman, and Daniel D. Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems 22*, 2009.

David McAllester, Tamir Hazan, and Joseph Keshet. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems 23*, 2010.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of ACL*, 2005.

Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Proceedings of NAACL HLT*, 2010.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, 2003.

Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, 2002.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of EMNLP*, 1999.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, 2002.

Kaare Brandt Petersen and Michael Syskind Pedersen. *The Matrix Cookbook*. 2008. `http://matrixcookbook.com`.

John C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 195–208. MIT Press, 1998.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Subgradient methods for maximum margin structured learning. In *Proceedings of the ICML Workshop on Learning in Structured Output Spaces*, 2006.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of ICML*, 2007.

David A. Smith and Jason Eisner. Minimum risk annealing for training log-linear models. In *Proceedings of COLING/ACL*, 2006. Poster Sessions.

Ben Taskar. *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University, 2004.

Christoph Tillmann and Tong Zhang. A discriminative global training algorithm for statistical MT. In *Proceedings of COLING-ACL*, 2006.

Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. Lattice minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of EMNLP*, 2008.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of ICML*, 2004.

Taro Watanabe, Jun Suzuki, Hajime Tsukuda, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of EMNLP*, 2007.

Richard Zens, Saša Hasan, and Hermann Ney. A systematic comparison of training criteria for statistical machine translation. In *Proceedings of EMNLP*, 2008.

Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, 2004.

Martin A. Zinkevich, Markus Weimer, Alex Smola, and Lihong Li. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*, 2010.

Journal of Machine Learning Research 13 (2012) 1189-1219 Submitted 3/11; revised 2/12; Published 4/12

A Multi-Stage Framework for Dantzig Selector and LASSO

Ji Liu JI.LIU@ASU.EDU
Peter Wonka PETER.WONKA@ASU.EDU
Jieping Ye JIEPING.YE@ASU.EDU
Arizona State University
699 South Mill Avenue
Tempe, AZ 85287-8809, USA

Editor: Tong Zhang

Abstract

We consider the following sparse signal recovery (or feature selection) problem: given a design matrix $X \in \mathbb{R}^{n\times m}$ $(m \gg n)$ and a noisy observation vector $y \in \mathbb{R}^n$ satisfying $y = X\beta^* + \varepsilon$ where ε is the noise vector following a Gaussian distribution $N(0,\sigma^2 I)$, how to recover the signal (or parameter vector) β^* when the signal is sparse?

The Dantzig selector has been proposed for sparse signal recovery with strong theoretical guarantees. In this paper, we propose a multi-stage Dantzig selector method, which iteratively refines the target signal β^*. We show that if X obeys a certain condition, then with a large probability the difference between the solution $\hat{\beta}$ estimated by the proposed method and the true solution β^* measured in terms of the ℓ_p norm $(p \geq 1)$ is bounded as

$$\|\hat{\beta}-\beta^*\|_p \leq \left(C(s-N)^{1/p}\sqrt{\log m}+\Delta\right)\sigma,$$

where C is a constant, s is the number of nonzero entries in β^*, the risk of the oracle estimator Δ is independent of m and is much smaller than the first term, and N is the number of entries of β^* larger than a certain value in the order of $O(\sigma\sqrt{\log m})$. The proposed method improves the estimation bound of the standard Dantzig selector approximately from $Cs^{1/p}\sqrt{\log m}\sigma$ to $C(s-N)^{1/p}\sqrt{\log m}\sigma$ where the value N depends on the number of large entries in β^*. When $N = s$, the proposed algorithm achieves the oracle solution with a high probability, where the oracle solution is the projection of the observation vector y onto true features. In addition, with a large probability, the proposed method can select the same number of correct features under a milder condition than the Dantzig selector. Finally, we extend this multi-stage procedure to the LASSO case.

Keywords: multi-stage, Dantzig selector, LASSO, sparse signal recovery

1. Introduction

The sparse signal recovery problem has been studied in many areas including machine learning (Zhang, 2009b; Zhao and Yu, 2006), signal processing (Donoho et al., 2006; Romberg, 2008; Wainwright, 2009), and mathematics/statistics (Bunea et al., 2007; Candès and Plan, 2009; Candès and Tao, 2007; Koltchinskii and Yuan, 2008; Lounici, 2008; Meinshausen et al., 2006; Ravikumar et al., 2008; Zhang, 2009a). In the sparse signal recovery problem, one is mainly interested in the signal recovery accuracy, that is, the distance between the estimation $\hat{\beta}$ and the original signal or the true solution β^*. If the design matrix X is considered as a feature matrix, that is, each column is a feature vector, and the observation y as a target object vector, then the sparse signal recovery problem is equivalent to feature selection (or model selection). In feature selection, one concerns the feature selection accuracy. Typically, a group of features corresponding to the coefficient values in $\hat{\beta}$ larger than a threshold form the supporting feature set. The difference between this set and the true supporting set (i.e., the set of features corresponding to nonzero coefficients in the original signal) measures the feature selection accuracy.

Two well-known algorithms for learning sparse signals include LASSO (Tibshirani, 1996) and Dantzig selector (Candès and Tao, 2007):

$$\textbf{LASSO} \quad \min_{\beta} : \frac{1}{2}\|X\beta - y\|_2^2 + \lambda' ||\beta||_1,$$

$$\textbf{Dantzig Selector} \quad \min_{\beta} : ||\beta||_1 \quad s.t. : \|X^T(X\beta - y)\|_\infty \leq \lambda.$$

Strong theoretical results concerning LASSO and Dantzig selector have been established in the literature (Cai et al., 2009; Candès and Plan, 2009; Candès and Tao, 2007; Wainwright, 2009; Zhang, 2009a; Zhao and Yu, 2006).

1.1 Contributions

In this paper, we propose a multi-stage procedure based on the Dantzig selector, which estimates the supporting feature set F_0 and the signal $\hat{\beta}$ iteratively. The intuition behind the proposed multi-stage method is that feature selection and signal recovery are tightly correlated and they can benefit from each other: a more accurate estimation of the supporting features can lead to a better signal recovery and a more accurate signal recovery can help identify a better set of supporting features. In the proposed method, the supporting set F_0 starts from an empty set and its size increases by one after each iteration. At each iteration, we employ the basic framework of Dantzig selector and the information about the current supporting feature set F_0 to estimate the new signal $\hat{\beta}$. In addition, we select the supporting feature candidates in F_0 among all features in the data at each iteration, thus allowing to remove incorrect features from the previous supporting feature set.

The main contributions of this paper lie in the theoretical analysis of the proposed method. Specifically, we show: 1) the proposed method can improve the estimation bound of the standard Dantzig selector approximately from $Cs^{1/p}\sqrt{\log m}\sigma$ to $C(s-N)^{1/p}\sqrt{\log m}\sigma$ where the value N depends on the number of large entries in β^*; 2) when $N = s$, the proposed algorithm can achieve the oracle solution $\bar{\beta}$ with a high probability, where the oracle solution is the projection of the observation vector y onto true features (see Equation (1) for the explicit description of $\bar{\beta}$); 3) with a high probability, the proposed method can select the same number of correct features under a milder condition than the standard Dantzig selector method; 4) this multi-stage procedure can be easily extended to the LASSO case. The numerical experiments validate these theoretical results.

1.2 Related Work

Sparse signal recovery without observation noise was studied by Candès and Tao (2005), which showed under the restricted isometry property (RIP) sparse signals can be perfectly recovered by solving an ℓ_1 norm minimization problem. LASSO and Dantzig selector can be considered as its noisy versions. Zhao and Yu (2006) proved the feature selection consistency of LASSO under the irrepresentable condition. It was also shown by Candès and Plan (2009) that if the true signal is strong enough together with some additional assumptions on its supporting set and signs, the mutual incoherence property (MIP) (or incoherence condition) can guarantee the feature selection consistency and the sign consistency with a high probability. A comprehensive analysis for LASSO, including the recovery accuracy in an arbitrary ℓ_p norm ($p \geq 1$) and the feature selection consistency, was presented in Zhang (2009a). Candès and Tao (2007) proposed the Dantzig selector (which is a linear programming problem) for sparse signal recovery and presented a bound of recovery accuracy with the same order as LASSO under the uniform uncertainty principle (UUP). An approximate equivalence between the LASSO estimator and the Dantzig selector was given by Bickel et al. (2009). Lounici (2008) studied the ℓ_∞ convergence rate for LASSO and Dantzig estimators in a high-dimensional linear regression model under MIP. James et al. (2009) provided conditions on the design matrix X under which the LASSO and Dantzig selector coefficient estimates are identical for certain tuning parameters. Please refer to recent papers (Zhang, 2009a; Fan and Lv, 2010) for a more comprehensive overview of LASSO and Dantzig selector.

Since convex regularization methods like LASSO and Dantzig selector give biased estimation due to convex regularization, many heuristic methods have been proposed to correct the bias of convex relaxation recently, including orthogonal matching pursuit (OMP) (Tropp, 2004; Donoho et al., 2006; Zhang, 2009b, 2011a; Cai and Wang, 2011), two stage LASSO (Zhang, 2009a), multiple thresholding LASSO (Zhou, 2009), adaptive LASSO (Zou, 2006), adaptive forward-backward greedy method (FoBa) (Zhang, 2011b), and nonconvex regularization methods (Zhang, 2010b; Fan and Lv, 2011; Lv and Fan, 2009; Zhang, 2011b). They have been shown to outperform the standard convex methods in many practical applications. It was shown that under exact recovery condition (ERC) (similar to MIP) the solution of OMP guarantees the feature selection consistency in the noiseless case (Tropp, 2004). The results of Tropp (2004) were extended to the noisy case by Zhang (2009b). Very recently, Zhang (2011a) showed that under RIP (weaker than MIP and ERC), OMP can stably recover a sparse signal in 2-norm under measurement noise. A multiple thresholding procedure was proposed to refine the solution of LASSO or Dantzig selector (Zhou, 2009). The FoBa algorithm was proposed by Zhang (2011b), and it was shown that under RIP the feature selection consistency is achieved if the minimal nonzero entry in the true solution is larger than $O(\sigma\sqrt{\log m})$. The adaptive LASSO was proposed to adaptively tune the weight value for the ℓ_1 norm penalty, and it was shown to enjoy the oracle properties (Zou, 2006). Zhang (2010b) proposed a general multi-stage convex regularization method (MSCR) to solve a nonconvex sparse regularization problem. It was also shown that a specific case "least square loss + nonconvex sparse regularization" can eliminate the bias in signal recovery (Zhang, 2010b) and achieve the feature selection consistency (Zhang, 2011c) under the sparse eigenvalue condition (SEC) if the true signal is strong enough. More related work about nonconvex regularization methods can be found in a recent paper by Zhang and Zhang (2012).

Conditions mentioned above can be classified into two classes: 1) the ℓ_2 conditions including RIP, UUP, and SEC; 2) the ℓ_∞ conditions including ERC and MIP. Overall, the ℓ_2 conditions are considered to be weaker than the ℓ_∞ conditions, since the ℓ_∞ conditions require about $O(s^2 \log m)$ random projections while the ℓ_2 conditions only need $O(s \log m)$ random projections.

1.3 Definitions, Notations, and Basic Assumptions

We use $X \in \mathbb{R}^{n\times m}$ to denote the design matrix and focus on the case $m \gg n$, that is, the signal dimension is much larger than the observation dimension. The correlation matrix A is defined as

$A = X^T X$ with respect to the design matrix. The noise vector ε follows the multivariate normal distribution $\varepsilon \sim N(0, \sigma^2 I)$. The observation vector $y \in \mathbb{R}^n$ satisfies $y = X\beta^* + \varepsilon$, where β^* denotes the original signal (or true solution). $\hat{\beta}$ is used to denote the solution of the proposed algorithm. The α-supporting set ($\alpha \geq 0$) for a vector β is defined as

$$supp_\alpha(\beta) = \{j : \ |\beta_j| > \alpha\}.$$

The "supporting" set of a vector refers to the 0-supporting set. F denotes the supporting set of the original signal β^*. For any index set S, $|S|$ denotes the size of the set and $\bar{S}$ denotes the complement of S in $\{1, 2, 3, ..., m\}$. In this paper, s is used to denote the size of the supporting set F, that is, $s = |F|$. We use β_S to denote the subvector of β consisting of the entries of β in the index set S. The ℓ_p norm of a vector v is computed by $\|v\|_p = (\sum_i |v_i|^p)^{1/p}$, where v_i denotes the ith entry of v. The oracle solution $\bar{\beta}$ is defined as

$$\bar{\beta}_F = (X_F^T X_F)^{-1} X_F^T y \text{ and } \bar{\beta}_{\bar{F}} = 0. \tag{1}$$

We employ the following notation to measure some properties of a PSD matrix $M \in \mathbb{R}^{K\times K}$ (Zhang, 2009a):

$$\begin{aligned} \mu^{(p)}_{M,k} &= \inf_{u \in \mathbb{R}^k, |I|=k} \frac{\|M_{I,I}u\|_p}{\|u\|_p}, \quad \rho^{(p)}_{M,k} = \sup_{u \in \mathbb{R}^k, |I|=k} \frac{\|M_{I,I}u\|_p}{\|u\|_p}, \\ \theta^{(p)}_{M,k,l} &= \sup_{u \in \mathbb{R}^l, |I|=k, |J|=l, I\cap J=\varnothing} \frac{\|M_{I,J}u\|_p}{\|u\|_p}, \quad \gamma_M = \max_{i\neq j} |M_{ij}|, \end{aligned}$$

where $p \in [1, \infty]$, I and J are disjoint subsets of $\{1, 2, ..., K\}$, and $M_{I,J} \in \mathbb{R}^{|I|\times|J|}$ is a submatrix of M with rows from the index set I and columns from the index set J. One can easily verify that $\mu^{(\infty)}_{A,k} \geq 1 - \gamma_A(k-1)$, $\rho^{(\infty)}_{A,k} \leq 1 + \gamma_A(k-1)$, and $\theta^{(\infty)}_{A,k,l} \leq l\gamma_A$, if all columns of X are normalized to have a unit length.

Additionally, we use the following notation to denote two probabilities:

$$\eta_1' = \eta_1(\pi \log((m-s)/\eta_1))^{-1/2}, \quad \eta_2' = \eta_2(\pi \log(s/\eta_2))^{-1/2},$$

where η_1 and η_2 are two factors between 0 and 1. In this paper, if we say "large", "larger" or "the largest", it means that the absolute value is large, larger or the largest. For simpler notation in the computation of sets, we sometimes use "$S_1 + S_2$" to indicate the union of two sets S_1 and S_2, and use "$S_1 - S_2$" to indicate the removal of the intersection of S_1 and S_2 from the first set S_1. In this paper, the following assumption is always admitted.

Assumption 1 *We assume that $s = |supp_0(\beta^*)| < n$, the variable number is much larger than the feature dimension (i.e., $m \gg n$), each column vector is normalized as $X_i^T X_i = 1$ where X_i indicates the ith column (or feature) of X, and the noise vector ε follows the Gaussian distribution $N(0, \sigma^2 I)$.*

In the literature, it is often assumed that $X_i^T X_i = n$, which is essentially identical to our assumption. However, this may lead to a slight difference of a factor $\sqrt{n}$ in some conclusions. We have automatically transformed conclusions from related work according to our assumption when citing them in our paper.

1.4 Organization

The rest of the paper is organized as follows. We present our multi-stage algorithm in Section 2. The main theoretical results are summarized in Section 3 with detailed proofs given in Appendix A (for Dantzig selector) and Appendix B (for LASSO). The numerical simulation is reported in Section 4. Finally, we conclude the paper in Section 5.

2. The Multi-Stage Dantzig Selector Algorithm

In this section, we introduce the multi-stage Dantzig selector algorithm. In the proposed method, we update the support set F_0 and the estimation $\hat{\beta}$ iteratively; the supporting set F_0 starts from an empty set and its size increases by one after each iteration. At each iteration, we employ the basic framework of Dantzig selector and the information about the current supporting set F_0 to estimate the new signal $\hat{\beta}$ by solving the following linear program:

$$\begin{aligned} &\min \|\beta_{\bar{F}_0}\|_1 \\ &s.t.\ \|X_{\bar{F}_0}^T(X\beta - y)\|_\infty \leq \lambda \\ &\qquad \|X_{F_0}^T(X\beta - y)\|_\infty = 0. \end{aligned} \tag{2}$$

Since the features in F_0 are considered as the supporting candidates, it is natural to enforce them to be orthogonal to the residual vector $X\beta - y$, that is, one should make use of them for reconstructing the overestimation y. This is the rationale behind the constraint: $\|X_{F_0}^T(X\beta - y)\|_\infty = 0$. The other advantage is when all correct features (i.e., the true feature set F) are chosen, the proposed algorithm can be shown to converge to the oracle solution. In other words, the oracle solution satisfies this constraint with F. The detailed procedure is formally described in **Algorithm 1** below. Apparently, when $F_0^{(0)} = \varnothing$ and $N = 0$, the proposed method is identical to the standard Dantzig selector.

Algorithm 1 Multi-Stage Dantzig Selector

```
Require: $F_0^{(0)}, \lambda, N, X, y$
Ensure: $\hat{\beta}^{(N)}, F_0^{(N)}$
while i=0; i≤N; i++ do
    Obtain $\hat{\beta}^{(i)}$ by solving the problem (2) with $F_0 = F_0^{(i)}$;
    Form $F_0^{(i+1)}$ as the index set of the $i+1$ largest elements of $\hat{\beta}^{(i)}$;
end while
```

3. Main Results

This section introduces the main results of this paper and discusses some of their implications. The proofs are provided in the Appendix.

3.1 Motivation

To motivate the proposed multi-stage algorithm, we first consider a simple case where some knowledge about the supporting features is known in advance. In standard Dantzig selector, we assume

$F_0 = \varnothing$. If we assume that the features belonging to a set F_0 are known as supporting features, that is, $F_0 \subset F$, we have the following result:

Theorem 1 *Assume that Assumption 1 holds. Take* $F_0 \subset F$ *and* $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ *in the optimization problem* (2). *If there exists some* l *such that*

$$\mu^{(p)}_{A,s+l} - \theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0 - \bar{F}|}{l}\right)^{1-1/p} > 0$$

holds, then with a probability larger than $1-\eta'_1$*, the* ℓ_p *norm* ($1 \le p \le \infty$) *of the difference between* $\hat{\beta}$*, the solution of the problem* (2)*, and the oracle solution* $\bar{\beta}$ *is bounded as*

$$\|\hat{\beta} - \bar{\beta}\|_p \le \frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0 - \bar{F}| + l2^p)^{1/p}}{\mu^{(p)}_{A,s+l} - \theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda \tag{3}$$

and with a probability larger than $1-\eta'_1-\eta'_2$*, the* ℓ_p *norm* ($1 \le p \le \infty$) *of the difference between* $\hat{\beta}$*, the solution of the problem* (2) *and the true solution* β^* *is bounded as*

$$\begin{aligned}\|\hat{\beta} - \beta^*\|_p \le & \frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0 - \bar{F}| + l2^p)^{1/p}}{\mu^{(p)}_{A,s+l} - \theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda + \\ & \frac{s^{1/p}}{\mu^{(p)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.\end{aligned} \tag{4}$$

It is clear that both bounds (for any $1 \le p \le \infty$) are monotonically increasing with respect to the value of $|\bar{F}_0 - \bar{F}|$. In other words, the larger F_0 is, the lower these bounds are. This coincides with our motivation that more knowledge about the supporting features can lead to a better signal estimation. Most related literatures directly estimate the bound of $\|\hat{\beta} - \beta^*\|_p$. Since β^* may not be a feasible solution of problem (2), it is not easy to directly estimate the distance between $\hat{\beta}$ and β^*.

The bound given in the inequality (4) consists of two terms. Since $m \gg n > s$, we have $\sqrt{2\log((m-s)/\eta_1)} \gg \sqrt{2\log(s/\eta_2)}$ if $\eta_1 \approx \eta_2$. When $p = 2$, the following holds:

$$\mu^{(2)}_{A,s+l} - \theta^{(2)}_{A,s+l,l}\left(\frac{|\bar{F}_0 - \bar{F}|}{l}\right)^{1-1/2} \le \mu^{(2)}_{(X_F^T X_F)^{1/2},s}$$

due to the following relationships:

$$\mu^{(2)}_{A,s+l} \le \mu^{(2)}_{A,s} \le \mu^{(2)}_{X_F^T X_F,s} \le \mu^{(2)}_{(X_F^T X_F)^{1/2},s}.$$

From the analysis in the next section, we can see that the first term is the upper bound of the distance from the optimizer to the oracle solution, that is, $\|\hat{\beta} - \bar{\beta}\|_p$ and the second term is the upper bound of the distance from the oracle solution to the true solution, that is, $\|\bar{\beta} - \beta^*\|_p$.[1] Thus, the first term may be much larger than the second term under the assumption $m \gg n > s$.

[1] The presented bound for $\|\bar{\beta} - \hat{\beta}\|_p$ can be sharper for a particular value of p, for example, $\|\bar{\beta} - \beta^*\|_2 \le O(\sigma\sqrt{s})$, $\|\bar{\beta} - \beta^*\|_\infty \le O(\sigma\sqrt{\log s})$ (Zhang, 2009b). For simplicity, a general bound $\|\bar{\beta} - \beta^*\|_p \le O(\sigma s^{1/p}\sqrt{\log s})$ is used in this paper.

3.2 Comparison with Dantzig Selector

We first compare our estimation bound with the one derived by Candès and Tao (2007) for $p = 2$. For convenience of comparison, we rewrite their theorem (Candès and Tao, 2007) equivalently as:

Theorem 2 *Suppose* $\beta \in \mathbb{R}^m$ *is any s-sparse vector of parameters obeying* $\delta_{2s} + \theta^{(2)}_{A,s,2s} < 1$. *Setting* $\lambda_p = \sigma\sqrt{2\log(m/\eta)}$ $(0 < \eta \leq 1)$*, with a probability at least* $1 - \eta(\pi \log m)^{-1/2}$*, the solution of the standard Dantzig selector* $\hat{\beta}_D$ *obeys*

$$\|\hat{\beta}_D - \beta^*\|_2 \leq \frac{4}{1 - \delta_{2s} - \theta^{(2)}_{A,s,2s}} s^{1/2}\sigma\sqrt{2\log(m/\eta)}, \tag{5}$$

where $\delta_{2s} = \max(\rho^{(2)}_{A,2s} - 1, 1 - \mu^{(2)}_{A,2s})$.

In order to compare Theorem 1 with the result above, taking $l = |\bar{F}_0 - \bar{F}| \leq s$, $p = 2$, $\eta_1 = \frac{m-s}{m}\eta$, and $\eta_2 = \frac{s}{m}\eta$ in Theorem 1, we obtain that

$$\|\hat{\beta} - \beta^*\|_2 \leq \left(\frac{\sqrt{10l}}{\mu^{(2)}_{A,s+l} - \theta^{(2)}_{A,s+l,l}} + \frac{\sqrt{s}}{\mu^{(2)}_{(X_F^T X_F)^{1/2},s}} \right) \sigma\sqrt{2\log(m/\eta)} \tag{6}$$

holds with probability larger than $1 - \eta(\pi \log m)^{-1/2}$. It is easy to verify that

$$1 - \delta_{2s} - \theta^{(2)}_{A,s,2s} \leq \mu^{(2)}_{A,s+l} - \theta^{(2)}_{A,s+l,s} \leq \mu^{(2)}_{A,2s} \leq \mu^{(2)}_{(X_F^T X_F),s} = \left(\mu^{(2)}_{(X_F^T X_F)^{1/2},s}\right)^2 \leq \mu^{(2)}_{(X_F^T X_F)^{1/2},s} \leq 1.$$

When $F_0 = \varnothing$, the bound in (6) is comparable to the one in (5). Since $\mu^{(2)}_{A,s+l} - \theta^{(2)}_{A,s+l,l}$ in Equation (6) is a decreasing function in terms of l, if F_0 is nonempty, particularly if F_0 is close to F (i.e., l is close to 0), the condition $\mu^{(2)}_{A,s+l} - \theta^{(2)}_{A,s+l,l} > 0$ required in Equation (6) is much easier to satisfy than the condition $1 - \delta_{2s} - \theta^{(2)}_{A,s,2s} > 0$ required in Equation (5).

3.3 Feature Selection

The estimation bounds in Theorem 1 assume that a set F_0 is given. In this section, we show how the supporting set can be estimated. Similar to previous work (Candès and Plan, 2009; Zhang, 2009b), $|\beta^*_j|$ for $j \in F$ is required to be larger than a threshold value. As is clear from the proof in **Appendix A**, the threshold value α_0 is actually proportional to the value of $\|\hat{\beta} - \beta^*\|_\infty$. We essentially employ the result with $p = \infty$ in Theorem 1 to estimate the threshold value. It shows that the value of $\|\hat{\beta} - \beta^*\|_\infty$ is bounded by $O(\lambda)$, which is consistent with the result of Lounici (2008). In the following, we first consider the simple case when $N = 0$. We have shown in the last section that the estimation bound in this case is similar to the one for Dantzig selector.

Theorem 3 *Under the Assumption 1, if there exist a nonempty set*

$$\Omega = \{l \mid \mu^{(\infty)}_{A,s+l} - \theta^{(\infty)}_{A,s+l,l} \left(\frac{s}{l}\right) > 0\}$$

and an index set J such that $|\beta_j^*| > \alpha_0$ for any $j \in J$, where

$$\begin{aligned}\alpha_0 =& \|\hat{\beta}^{(0)} - \beta^*\|_\infty + \|\hat{\beta}^{(0)} - \bar{\beta}\|_\infty \\ \leq& 4 \min_{l \in \Omega} \frac{\max\left(1, \frac{s}{l}\right)}{\mu_{A,s+l}^{(\infty)} - \theta_{A,s+l,l}^{(\infty)}\left(\frac{s}{l}\right)} \lambda + \frac{1}{\mu_{(X_F^T X_F)^{1/2},s}^{(\infty)}} \sigma\sqrt{2\log(s/\eta_2)},\end{aligned}$$

then taking $F_0 = \varnothing$, $N = 0$, $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ into the problem (2) *(equivalent to Dantzig selector), the largest $|J|$ elements of $\hat{\beta}_{std}$ (or $\hat{\beta}^{(0)}$) belong to F with probability larger than $1 - \eta_1' - \eta_2'$.*

The theorem above indicates that under the given condition, if $\min_{j \in J} |\beta_j^*| > O(\sigma\sqrt{\log m})$ (assuming that there exists $l \geq s$ such that $\mu_{A,s+l}^{(\infty)} - \theta_{A,s+l,l}^{(\infty)}\left(\frac{s}{l}\right) > 0$), then with high probability the selected $|J|$ features by Dantzig selector belong to the true supporting set. In particular, if $|J| = s$, then the consistency of feature selection is achieved. In order to build up a link to the previous work, we let $l = s$. Note that $\mu_{A,2s}^{(\infty)} - \theta_{A,2s,s}^{(\infty)} \geq 1 - \gamma_A(3s-1)$. If the MIP holds like $\gamma_A s \leq 1/6$ (see Corollary 8.1 in Zhang, 2009a), then the condition required in Theorem 3 is satisfied as well. It means that the condition we require is not stronger than MIP. However, it still belongs to the ℓ_∞ condition like MIP. The result above is comparable to the ones for other feature selection algorithms, including LASSO/two stage LASSO (Candès and Plan, 2009; Zhao and Yu, 2006), OMP (Tropp, 2004; Donoho et al., 2006; Zhang, 2009b), and two stage LASSO (Zhang, 2009a). In all these algorithms, the conditions $\min_{j \in F} |\beta_j^*| \geq C\sigma\sqrt{\log m}$ and an ℓ_∞ condition are required. As pointed out by Zhang and Zhang (2012) and Zhang (2011a), these conditions required by OMP, Dantzig selector, and LASSO in feature selection cannot be improved. If one wants to use the ℓ_2 conditions in feature selection, the minimal nonzero entry of the true solution must be in the order of $O(\sigma\sqrt{s\log m})$, which can be obtained by simply using $\|\hat{\beta}^{(0)} - \beta^*\|_\infty + \|\hat{\beta}^{(0)} - \bar{\beta}\|_\infty \leq \|\hat{\beta}^{(0)} - \beta^*\|_2 + \|\hat{\beta}^{(0)} - \bar{\beta}\|_2$. A similar requirement under the ℓ_2 condition for LASSO (or two stage LASSO) is also implied by Zhang (2009a, Theorem 8.1).

Next, we show that the condition $|\beta_j^*| > \alpha_0$ in Theorem 3 can be relaxed by the proposed multi-stage procedure with $N > 0$, as summarized in the following theorem:

Theorem 4 *Under the Assumption 1, if there exist a nonempty set*

$$\Omega = \{l \mid \mu_{A,s+l}^{(\infty)} - \theta_{A,s+l,l}^{(\infty)}\left(\frac{s}{l}\right) > 0\}$$

and a set J such that $|supp_{\alpha_i}(\beta_J^*)| > i$ holds for all $i \in \{0, 1, ..., |J|-1\}$, where

$$\begin{aligned}\alpha_i =& \|\hat{\beta}^{(i)} - \beta^*\|_\infty + \|\hat{\beta}^{(i)} - \bar{\beta}\|_\infty \\ \leq& 4 \min_{l \in \Omega} \frac{\max\left(1, \frac{s-i}{l}\right)}{\mu_{A,s+l}^{(\infty)} - \theta_{A,s+l,l}^{(\infty)}\left(\frac{s-i}{l}\right)} \lambda + \frac{1}{\mu_{(X_F^T X_F)^{1/2},s}^{(\infty)}} \sigma\sqrt{2\log(s/\eta_2)},\end{aligned}$$

then taking $F_0^{(0)} = \varnothing$, $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ and $N = |J| - 1$ into **Algorithm** *1, the solution after N iterations satisfies $F_0^{(N)} \subset F$ (i.e., $|J|$ correct features are selected) with probability larger than $1 - \eta_1' - \eta_2'$.*

Assume that one aims to select N correct features by the standard Dantzig selector and the multi-stage method. These two theorems show that the standard Dantzig selector requires that at least N of $|\beta^*_j|$'s with $j \in F$ are larger than the threshold value α_0, while the proposed multi-stage method requires that at least i of the $|\beta^*_j|$'s are larger than the threshold value α_{i-1}, for $i = 1, \cdots, N$. Since the upper bounds of $\{\alpha_j\}$'s strictly decrease and the difference of two neighbors is greater than

$$\frac{4\theta^{(\infty)}_{A,s+l,l}}{l\left(\mu^{(\infty)}_{A,s+l} - \theta^{(\infty)}_{A,s+l,l}\left(\frac{s-i}{l}\right)\right)^2}\lambda$$

for some $l \in \Omega$, the proposed multi-stage method requires a strictly weaker condition for selecting N correct features than the standard Dantzig selector. If we consider the ℓ_2 conditions, using $\|\hat{\beta}^{(i)} - \beta^*\|_\infty + \|\hat{\beta}^{(i)} - \bar{\beta}\|_\infty \leq \|\hat{\beta}^{(i)} - \beta^*\|_2 + \|\hat{\beta}^{(i)} - \bar{\beta}\|_2$ to bound α_i, we obtain that $\alpha_i \leq O(\sqrt{(s-i)\log m} + \Delta)\sigma$ where Δ is a small number relying on s. When i is close to s, the order of α_i approaches $O(\sigma\sqrt{\log m})$. Recall that the FoBa algorithm (Zhang, 2011b), MSCR (Zhang, 2011c), and MC+ (Zhang, 2010a) require an ℓ_2 condition and the threshold value is in the order of $O(\sigma\sqrt{\log m})$ for the feature selection consistency while the standard LASSO or Dantzig selector requires the threshold value in the order of $O(\sigma\sqrt{s\log m})$. Therefore, our condition lies between them.

3.4 Signal Recovery

In this section, we derive the estimation bound of the proposed multi-stage method by combining results from Theorems 1, 3, and 4.

Theorem 5 *Under the Assumption 1, if there exist l such that*

$$\mu^{(\infty)}_{A,s+l} - \theta^{(\infty)}_{A,s+l,l}\left(\frac{s}{l}\right) > 0 \;\; \textit{and} \;\; \mu^{(p)}_{A,2s} - \theta^{(p)}_{A,2s,s} > 0,$$

and a set J such that $|supp_{\alpha_i}(\beta^*_J)| > i$ *holds for all* $i \in \{0, 1, ..., |J| - 1\}$, *where the* α_i*'s are defined in Theorem 4, then*

(1) taking $F_0 = \varnothing$, $N = 0$ *and* $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ *into* **Algorithm** *1, with probability larger than* $1 - \eta'_1 - \eta'_2$, *the solution of the Dantzig selector* $\hat{\beta}_D$ *(i.e.,* $\hat{\beta}^{(0)}$*) obeys:*

$$\|\hat{\beta}_D - \beta^*\|_p \leq \frac{(2^{p+1}+2)^{1/p}s^{1/p}}{\mu^{(p)}_{A,2s} - \theta^{(p)}_{A,2s,s}}\lambda + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)};$$

(2) taking $F_0 = \varnothing$, $N = |J|$ *and* $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ *into* **Algorithm** *1, with probability larger than* $1 - \eta'_1 - \eta'_2$, *the solution of the multi-stage method* $\hat{\beta}_{mul}$ *(i.e.,* $\hat{\beta}^{(N)}$*) obeys:*

$$\|\hat{\beta}_{mul} - \beta^*\|_p \leq \frac{(2^{p+1}+2)^{1/p}(s-N)^{1/p}}{\mu^{(p)}_{A,2s-N} - \theta^{(p)}_{A,2s-N,s-N}}\lambda + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.$$

Similar to the analysis in Theorem 1, the first term (i.e., the distance from $\hat{\beta}$ to the oracle solution $\bar{\beta}$) dominates in the estimated bounds. Thus, the performance of the multi-stage method approximately improves the standard Dantzig selector from $Cs^{1/p}\sqrt{\log m}\sigma$ to $C(s-N)^{1/p}\sqrt{\log m}\sigma$. When $p = 2$, our estimation has the same order as FoBa (Zhang, 2011b) and MCSR (Zhang, 2010b), but the conditions involved in our estimation belong to the ℓ_∞ class while they use the ℓ_2 condition.

3.5 The Oracle Solution

The oracle solution $\hat{\beta}$ defined in Equation (1) is the minimum-variance unbiased estimator of the true solution given the noisy observation. We show in the following theorem that the proposed method can obtain the oracle solution with high probability under certain conditions:

Theorem 6 *Under the assumption 1, if there exists l such that $\mu^{(\infty)}_{A,s+l} - \theta^{(\infty)}_{A,s+l,l}\left(\frac{s-i}{l}\right) > 0$, and the supporting set F of β^* satisfies $|supp_{\alpha_i}(\beta^*_F)| > i$ for all $i \in \{0,1,...,s-1\}$, where the α_i's are defined in Theorem 4, then taking $F_0 = \varnothing$, $N = s$ and $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ into* **Algorithm** *1, the oracle solution can be achieved, that is, $F_0^{(N)} = F$ and $\hat{\beta}^{(N)} = \bar{\beta}$, with probability larger than $1 - \eta_1' - \eta_2'$.*

The theorem above shows that when the nonzero elements of the true coefficients vector β^* are large enough, the oracle solution can be achieved with high probability.

3.6 The Multi-Stage LASSO Algorithm

Next we extend the multi-stage procedure to the LASSO case; we expect to achieve similar improvements over the standard LASSO. The multi-stage LASSO algorithm can be obtained by substituting the basic optimization problem, that is, Equation (2) in **Algorithm** 1, by the following problem:

$$\begin{aligned} \min_{\beta} :\ & \frac{1}{2}\|X\beta - y\|_2^2 + \lambda'\|\beta_{\bar{F}_0}\|_1 \\ s.t.:\ & \|X^T_{F_0}(X\beta - y)\|_\infty = 0. \end{aligned} \tag{7}$$

Note that the constraint in Equation (7) is satisfied automatically at the optimal solution by observing the subdifferential of its objective function. Thus, the constraint can be removed from Equation (7) in practice.

We apply the same framework in Dantzig selector to analyze the multi-stage LASSO to obtain a bound estimation for any $p \in [1,\infty]$ and show that similar improvements can be achieved over the standard LASSO. For completeness, we include all proofs and results for multi-stage LASSO in **Appendix B**.

It is worth mentioning that Zhang (2010b, 2011b) recently developed a similar method called MSCR. The main difference is that it uses a threshold value to update the candidate set $F_0^{(i+1)}$ at each iteration and may need to solve LASSO more than s times to converge, while our algorithm needs to solve LASSO less than s times. An advantage of MSCR is that it requires a weaker condition, that is, $\min_{i\in F}|\beta^*| > O(\sigma\sqrt{\log m})$ and an ℓ_2 condition, to achieve the consistency on feature selection and signal recovery.

4. Simulation Study

We have performed simulation studies to verify our theoretical analysis. Our comparison includes two aspects: signal recovery accuracy and feature selection accuracy. The signal recovery accuracy is measured by the relative signal error: $SRA = -20\log_{10}(\|\hat{\beta} - \beta^*\|_2/\|\beta^*\|_2)$, where $\hat{\beta}$ is the solution of a specific algorithm. The feature selection accuracy is measured by the percentage of correct features selected: $FSA = |\hat{F} \cap F|/|F|$, where $\hat{F}$ is the estimated feature candidate set.

We generate an $n \times m$ random matrix X. Each element of X follows an independent standard Gaussian distribution $N(0,1)$. We then normalize the length of the columns of X to be 1.

The $s-$sparse original signal β^* is generated with s nonzero elements independently uniformly distributed from $[-10,10]$. The locations of s nonzero elements are uniformly distributed in $\{1,2,\cdots,m\}$. We form the observation by $y = X\beta^* + \varepsilon$, where the noise vector ε is generated by the Gaussian distribution $N(0,\sigma^2 I)$. All experiments are repeated 100 times and we use their average performance for comparison.

First we compare the standard Dantzig selector and the multi-stage version. For a fair comparison, we choose the same $\lambda = \sigma\sqrt{2\log m}$ in both algorithms. We run the proposed algorithm with $F_0^{(0)} = \varnothing$ with different values of N and let the estimation $\hat{\beta}$ be the output $\hat{\beta}^{(N)}$ in **Algorithm 1**. The feature candidate set $\hat{F}$ is predicted by the index set of the s largest elements in $\hat{\beta}$. Note that $\hat{F}$ identified by $\hat{\beta} = \hat{\beta}^{(N)}$ is different from the output $F_0^{(N)}$ by **Algorithm 1**. The size of $\hat{F}$ is always s while the size of $F_0^{(N)}$ is N. Note that the solution of the standard Dantzig selector algorithm is equivalent to $\hat{\beta}^{(N)}$ with $N = 0$. We report the *SRA* curve of $\hat{\beta}^{(N)}$ with respect to N in the left column of Figure 1. The right column of Figure 1 shows the *FSA* curve with respect to N. We allow $N > s$ in our simulation although this case is beyond our theoretical analysis, since in practice the sparsity number s is usually unknown in advance. We can observe from Figure 1 that 1) the multi-stage method obtains a solution with a smaller distance to the original signal than the standard Dantzig selector method; 2) the multi-stage method selects a larger percentage of correct features than the standard Dantzig selector method; 3) the multi-stage method can achieve the oracle solution with a large probability; and 4) even when $N > s$, the multi-stage algorithm still outperforms the standard Dantzig selector and achieves high accuracy in signal recovery and feature selection. Overall, the recovery accuracy curve increases with an increasing value of N before reaching the sparsity level s and decreases slowly after that, and the feature selection accuracy curve increases while $N \le s$ and becomes flat after N goes beyond s.

Next we apply the multi-stage procedure to the LASSO case and compare the multi-stage LASSO to the standard LASSO and the two-stage LASSO (Zhang, 2009a). The two-stage LASSO algorithm first estimates a support set $F_0 = supp_\alpha(\beta')$ from the solution β' of the standard LASSO where $\alpha > 0$ is the threshold parameter; the second stage estimates the signal by solving the following problem

$$\min_{\beta}: \ \frac{1}{2}\|X\beta - y\|_2^2 + \lambda'\|\beta_{\bar{F}_0}\|_1, \tag{8}$$

which is indeed identical to Equation (7). In order to make it comparable to the proposed multi-stage LASSO algorithm with the parameter N, we properly choose α such that $|F_0| = N$ and use the output $\hat{\beta}'$ from Equation (8) and the feature candidate set by $\hat{\beta}'$ for comparison. Similarly, we use the same $\lambda' = 2\lambda$ in the three algorithms. The comparison reported in Figure 2 also indicates the advantage of the proposed multi-stage procedure.

5. Conclusion

In this paper, we propose a multi-stage procedure to improve the performance of the Dantzig selector and the LASSO by iteratively selecting the supporting features and recovering the original signal. The proposed method makes use of the information of supporting features to estimate the signal and simultaneously makes use of the information of the estimated signal to select the supporting features. Our theoretical analysis shows that the proposed method improves upon the standard

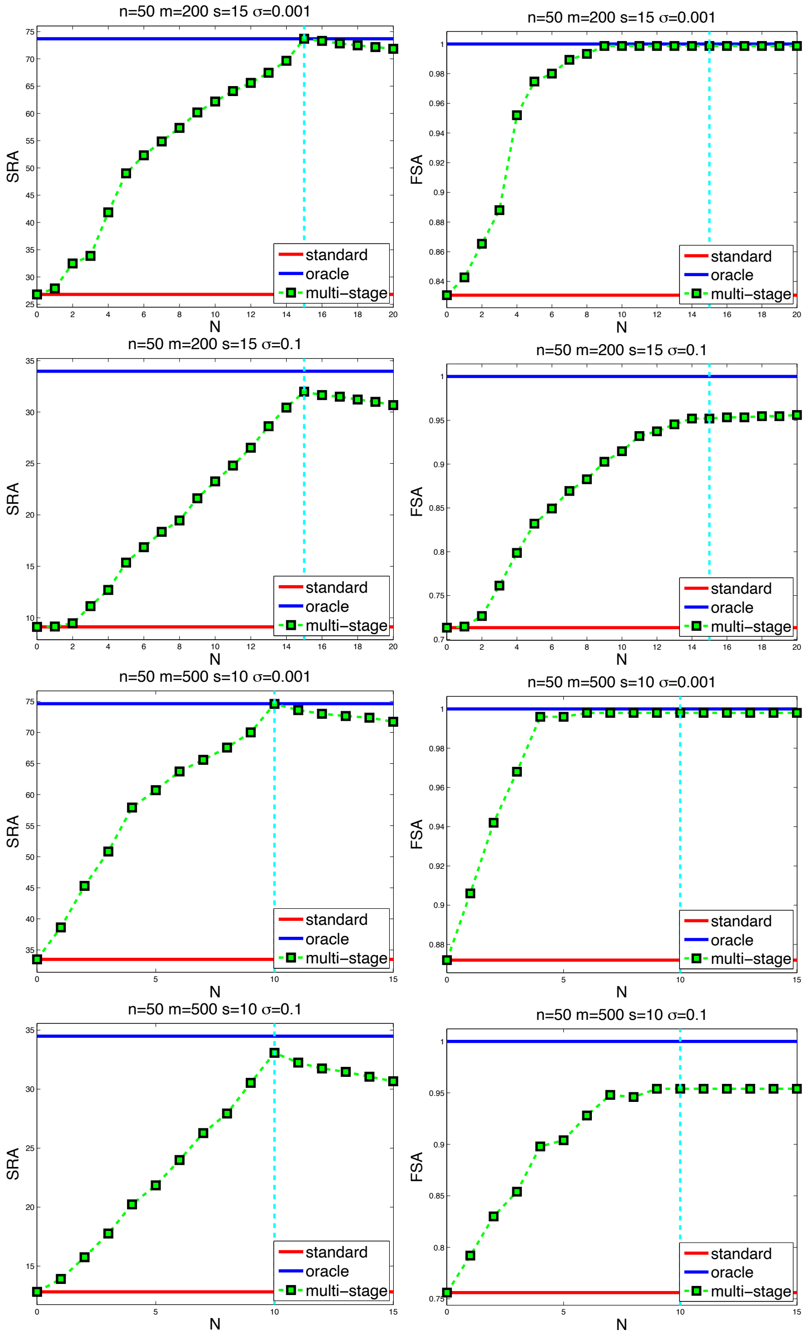


Figure 1: Numerical simulation. We compare the solutions of the standard Dantzig selector method ($N = 0$), the proposed method for different values of $N = 0, 1, \cdots, s, \cdots, s+5$, and the oracle solution. The SRA and FSA comparisons are reported on the left column and the right column, respectively. The red line indicates the SRA (or FSA) value of the standard Dantzig selector method; the blue line indicates the value of the oracle solution; the green curve with black boxes records the results by the proposed method for different values of N; the vertical cyan line distinguishes two cases $N <= s$ and $N > s$.

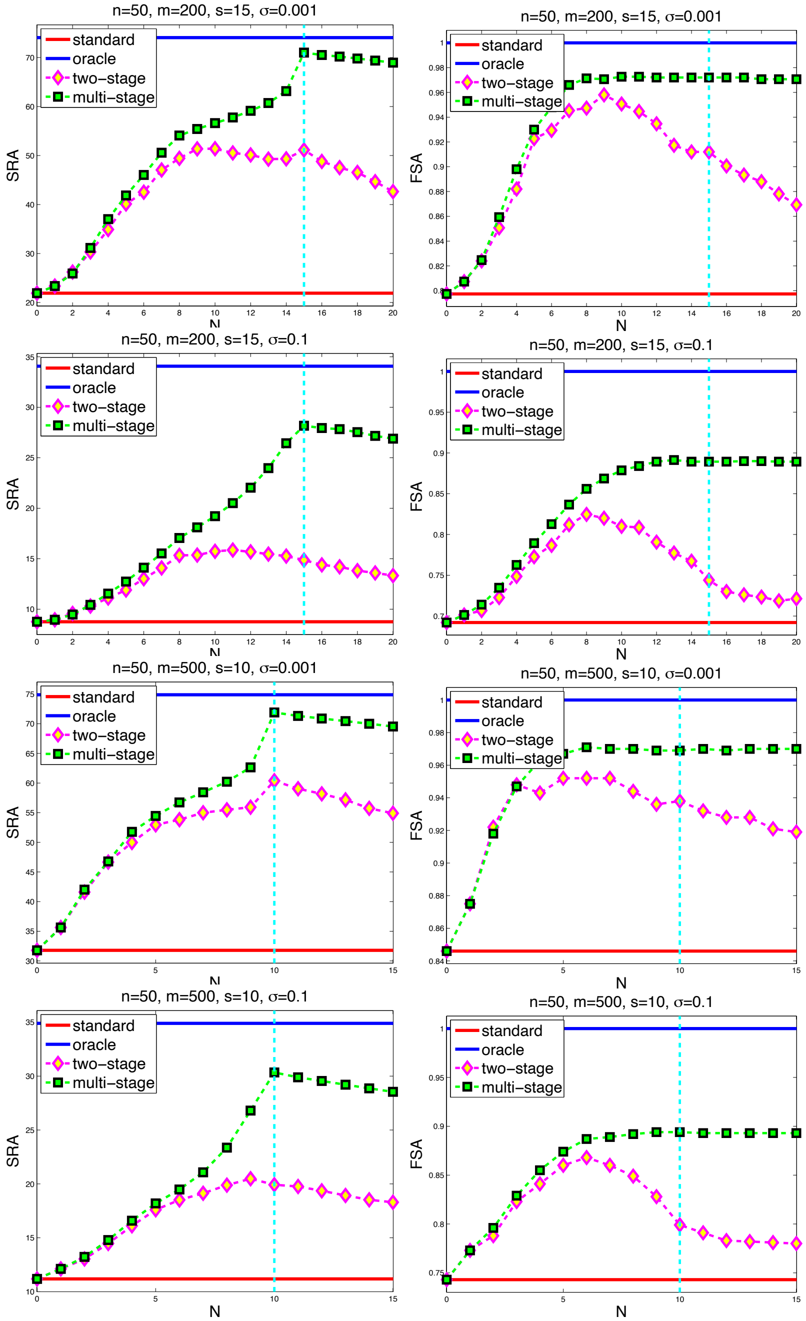


Figure 2: Numerical simulation. We compare the solutions of the standard Dantzig selector method ($N = 0$), the two-stage LASSO algorithm, the proposed method for different values of $N = 0, 1, \cdots, s, \cdots, s+5$, and the oracle solution. The *SRA* and *FSA* comparisons are reported on the left column and the right column, respectively. The red line indicates the *SRA* (or *FSA*) value of the standard Dantzig selector method; the blue line indicates the value of the oracle solution; the green curve with black boxes records the results of the proposed method for different values of N; the magenta curve with yellow diamonds indicates the results of the two-stage LASSO algorithm; the vertical cyan line distinguishes two cases $N <= s$ and $N > s$.

Dantzig selector and the LASSO in both signal recovery and supporting feature selection. The final numerical simulation confirms our theoretical analysis.

Acknowledgments

This work is supported by NSF CCF-0811790, IIS-0953662, and CCF-1025177. We appreciate the constructive comments from the editor and three reviewers.

Appendix A.

Theorem 1 is fundamental for the rest of the theorems. We first highlight a brief architecture for its proof. Theorem 1 estimates $\|\hat{\beta}-\beta^*\|_p$, which is bounded by the sum of two parts: $\|\hat{\beta}-\beta^*\|_p \le \|\hat{\beta}-\bar{\beta}\|_p + \|\bar{\beta}-\beta^*\|_p$. We use the upper bounds of these two parts to estimate the bound of $\|\hat{\beta}-\beta^*\|_p$. The analysis in Section 3.2 shows that the first term $\|\hat{\beta}-\bar{\beta}\|_p$ may be much larger than the second term $\|\bar{\beta}-\beta^*\|_p$. In Lemma 7, we estimate the bound of $\|\bar{\beta}-\beta^*\|_p$ and its holding probability. The remaining part of the proof focuses on the estimation of the bound of $\|\hat{\beta}-\bar{\beta}\|_p$. For convenience, we use h to denote $\hat{\beta}-\bar{\beta}$. h can be divided into $h_{\bar{F}_1-T_1}$ and $h_{F_1+T_1}$, where $F_0 \subset F_1 \subset F$. Lemma 9 studies the relationship between $h_{\bar{F}_1-T_1}$ and $h_{F_1+T_1}$, if $\bar{\beta}$ is feasible (Lemma 8 computes its holding probability). Then, Lemma 11 shows that $\|h\|_p$ can be bounded in terms of $\|h_{F_1+T_1}\|_p$. In Theorem 12, we estimate the bound of $\|h_{F_1+T_1}\|_p$. Finally, letting $F_1 = F$, we prove Theorem 1.

Lemma 7 *With probability larger than* $1-\eta(\pi\log(s/\eta))^{-1/2}$*, the following holds:*

$$\|\bar{\beta}-\beta^*\|_p \le \frac{s^{1/p}\sigma\sqrt{2\log(s/\eta)}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}. \tag{9}$$

Proof According to the definition of $\bar{\beta}$, we have

$$\begin{aligned}\bar{\beta}_F &= (X_F^TX_F)^{-1}X_F^Ty = (X_F^TX_F)^{-1}X_F^T(X\beta^*+\varepsilon) = (X_F^TX_F)^{-1}X_F^T(X_F\beta_F^*+\varepsilon)\\ &= \beta_F^* + (X_F^TX_F)^{-1}X_F^T\varepsilon.\end{aligned}$$

It follows that

$$\bar{\beta}_F-\beta_F^* = (X_F^TX_F)^{-1}X_F^T\varepsilon \sim N(0,(X_F^TX_F)^{-1}\sigma^2).$$

Since $\|\bar{\beta}-\beta^*\|_p = \|\bar{\beta}_F - \beta^*_F\|_p$, we only need to consider the bound for $\|\bar{\beta}_F - \beta^*_F\|_p$. Let $Z = (X_F^T X_F)^{1/2}(\beta^*_F - \bar{\beta}_F)/\sigma \sim N(0,I)$. We have

$$\begin{aligned}
P(\|Z\|_p \geq t) &= (2\pi)^{-s/2}\int_{\|Z\|_p \geq t} e^{-Z^TZ/2}dZ \\
&\leq (2\pi)^{-s/2}\int_{s^{1/p}\|Z\|_\infty \geq t} e^{-Z^TZ/2}dZ \quad (\text{due to } \|Z\|_p \leq s^{1/p}\|Z\|_\infty) \\
&= 1-(2\pi)^{-s/2}\int_{\|Z\|_\infty \leq s^{-1/p}t} e^{-Z^TZ/2}dZ \\
&= 1-\left[(2\pi)^{-1/2}\int_{|Z_i|\leq s^{-1/p}t} e^{-Z_i^2/2}dZ_i\right]^s \\
&= 1-\left[1-2(2\pi)^{-1/2}\int_{s^{-1/p}t}^{\infty} e^{-Z_i^2/2}dZ_i\right]^s \\
&\leq s\left[2(2\pi)^{-1/2}\int_{s^{-1/p}t}^{\infty} e^{-Z_i^2/2}dZ_i\right] \\
&\leq \frac{2s^{1+1/p}}{t(2\pi)^{1/2}}\exp\left[\frac{-t^2}{2s^{2/p}}\right].
\end{aligned}$$

Thus the following bound holds with probability larger than $1-\frac{2s^{1+1/p}}{t(2\pi)^{1/2}}\exp\left[\frac{-t^2}{2s^{2/p}}\right]$:

$$\begin{aligned}
P(\|Z\|_p \leq t) &= P(\|(X_F^TX_F)^{1/2}(\beta^*_F - \bar{\beta}_F)\|_p \leq t\sigma) \\
&\leq P(\mu^{(p)}_{(X_F^TX_F)^{1/2},s}\|\beta^*_F - \bar{\beta}_F\|_p \leq t\sigma) = P(\|\beta^*_F - \bar{\beta}_F\|_p \leq t\sigma/\mu^{(p)}_{(X_F^TX_F)^{1/2},s}).
\end{aligned}$$

Taking $t = \sqrt{2\log(s/\eta)}s^{1/p}$, we prove the claim. Note that the presented bound holds for any $p \geq 1$.
■

Lemma 8 *With probability larger than* $1-\eta(\pi\log\frac{m-s}{\eta})^{-1/2}$*, the following bound holds:*

$$\|X_{\bar{F}}^T(X\bar{\beta}-y)\|_\infty \leq \lambda,$$

where $\lambda = \sigma\sqrt{2\log(m-s)/\eta}$.

Proof Let us first consider the probability of $\|X_{\bar{F}}^T(X\bar{\beta}-y)\|_\infty \leq \lambda$. For any $j \in \bar{F}$, define v_j as

$$\begin{aligned}
v_j &= X_j^T(X\bar{\beta}-y) \\
&= X_j^T\left(X_F(X_F^TX_F)^{-1}X_F^T(X_F\beta^*_F+\varepsilon) - X_F\beta^*_F - \varepsilon\right) \\
&= X_j^T\left(X_F(X_F^TX_F)^{-1}X_F^T - I\right)\varepsilon \\
&\sim N(0, X_j^T(I-X_F(X_F^TX_F)^{-1}X_F^T)X_j\sigma^2).
\end{aligned}$$

Since $(I-X_F(X_F^TX_F)^{-1}X_F^T)$ is a projection matrix, we have $X_j^T(I-X_F(X_F^TX_F)^{-1}X_F^T)X_j\sigma^2 \leq \sigma^2$. Thus,

$$P(\|X_{\bar{F}}^T(X\bar{\beta}-y)\|_\infty \geq \lambda) = P(\sup_{j\in\bar{F}}|v_j| \geq \lambda) \leq \frac{2(m-s)\sigma}{\lambda(2\pi)^{1/2}}\exp\{-\lambda^2/2\sigma^2\}.$$

Taking $\lambda = \sigma\sqrt{2\log(m-s)/\eta}$ in the inequality above, we prove the claim. ■

It follows from the definition of $\bar\beta$ that $\|X_F^T(X\bar\beta - y)\|_\infty = 0$ always holds. In the following discussion, we assume that the following assumption holds:

Assumption 2 *$\bar\beta$ is a feasible solution of the problem* (2), *if $F_0 \subset F$.*

Under the assumption above, both $\|X_{\bar F}^T(X\bar\beta - y)\|_\infty \le \lambda$ and $\|X_F^T(X\bar\beta - y)\|_\infty = 0$ hold.

Note that this assumption is just used to simplify the description for following proofs. Our proof for the final theorems will substitute this assumption by the probability it holds.

In the following, we introduce an additional set F_1 satisfying $F_0 \subset F_1$ (Zhang, 2009a).

Lemma 9 *Let $F_0 \subset F$. Assume that Assumption 2 holds. Given any index set F_1 such that $F_0 \subset F_1$, we have the following conclusions:*

$$\begin{aligned}\|h_{\bar F_0 - \bar F_1}\|_1 + 2\|\bar\beta_{\bar F_1}\|_1 \ge& \|h_{\bar F_1}\|_1\\ \|X_{F_0}^T Xh\|_\infty =& 0\\ \|X_{\bar F}^T Xh\|_\infty \le& 2\lambda\\ \|X_{\bar F_0 - \bar F}^T Xh\|_\infty \le& \lambda.\end{aligned}$$

Proof Since $\bar\beta$ is a feasible solution, the following holds

$$\begin{aligned}\|\hat\beta_{\bar F_0}\|_1 &\le \|\bar\beta_{\bar F_0}\|_1\\ \|\hat\beta_{\bar F_0 - \bar F_1}\|_1 + \|\hat\beta_{\bar F_1}\|_1 &\le \|\bar\beta_{\bar F_0 - \bar F_1}\|_1 + \|\bar\beta_{\bar F_1}\|_1\\ \|\hat\beta_{\bar F_1}\|_1 &\le \|h_{\bar F_0 - \bar F_1}\|_1 + \|\bar\beta_{\bar F_1}\|_1\\ \|h_{\bar F_1} + \bar\beta_{\bar F_1}\|_1 &\le \|h_{\bar F_0 - \bar F_1}\|_1 + \|\bar\beta_{\bar F_1}\|_1\\ \|h_{\bar F_1}\|_1 &\le \|h_{\bar F_0 - \bar F_1}\|_1 + 2\|\bar\beta_{\bar F_1}\|_1.\end{aligned}$$

Thus, the first inequality holds. Since

$$X_{F_0}^T Xh = X_{F_0}^T X(\hat\beta - \bar\beta) = X_{F_0}^T(X\hat\beta - y) - X_{F_0}^T(X\bar\beta - y),$$

the second inequality can be obtained as follows:

$$\|X_{F_0}^T Xh\|_\infty \le \|X_{F_0}^T(X\hat\beta - y)\|_\infty + \|X_{F_0}^T(X\bar\beta - y)\|_\infty = 0.$$

The third inequality holds since

$$\|X_{\bar F}^T Xh\|_\infty \le \|X_{\bar F}^T(X\hat\beta - y)\|_\infty + \|X_{\bar F}^T(X\bar\beta - y)\|_\infty \le 2\lambda.$$

Similarly, the fourth inequality can be obtained as follows:

$$\|X_{\bar F_0 - \bar F}^T Xh\|_\infty \le \|X_{\bar F_0 - \bar F}^T(X\hat\beta - y)\|_\infty + \|X_{\bar F_0 - \bar F}^T(X\bar\beta - y)\|_\infty \le \lambda.$$

■

Lemma 10 *Given any $v \in \mathbb{R}^m$, its index set T is divided into a group of subsets T_j's ($j = 1, 2, \ldots$) without intersection such that $\bigcup_j T_j = T$. If* $\max_j |T_j| \leq l$ *and* $\max_{i \in T_{j+1}} |v_{T_{j+1}}[i]| \leq \|v_{T_j}\|_1 / l$ *hold for all j's, then we have*

$$\|v_{\bar{T}_1}\|_p \leq \|v\|_1 l^{1/p-1}.$$

Proof Since $|v_{T_{j+1}}[i]| \leq \|v_{T_j}\|_1 / l$, we have

$$\begin{aligned} \|v_{T_{j+1}}\|_p^p = & \sum_{i \in T_{j+1}} |v_{T_{j+1}}^p[i]| \leq \|v_{T_j}\|_1^p l^{1-p}, \\ \Rightarrow \|v_{T_{j+1}}\|_p \leq & \|v_{T_j}\|_1 l^{1/p-1}. \end{aligned}$$

Thus,

$$\|v_{\bar{T}_1}\|_p \leq \sum_{j \geq 1} \|v_{T_{j+1}}\|_p \leq \sum_{j \geq 1} \|v_{T_j}\|_1 l^{1/p-1} = \|v\|_1 l^{1/p-1},$$

which proves the claim. ■

Note that similar techniques as those in Lemma 10 have been used in the literature (Candès and Tao, 2007; Zhang, 2009a).

Lemma 11 *Assume that $F_0 \subset F$ and $F_0 \subset F_1$. We divide the index set $\bar{F}_1$ into a group of subsets T_j's ($j = 1, 2, \ldots$) such that they satisfy all conditions in Lemma 10 with $v = h$. Then the following holds:*

$$\begin{aligned} \|h_{\bar{F}_1 - T_1}\|_p \leq & l^{1/p-1} \left(|\bar{F}_0 - \bar{F}_1|^{1-1/p} \|h_{\bar{F}_0 - \bar{F}_1}\|_p + 2\|\bar{\beta}_{\bar{F}_1}\|_1 \right), \\ \|h\|_p \leq & \left[1 + \left(\frac{|\bar{F}_0 - \bar{F}_1|}{l} \right)^{p-1} \right]^{1/p} \|h_{F_1 + T_1}\|_p + 2l^{1/p-1} \|\bar{\beta}_{\bar{F}_1}\|_1. \end{aligned}$$

Proof Using Lemma 10 with $T = \bar{F}_1$, the first inequality can be obtained using the first inequality in lemma 9 as follows:

$$\begin{aligned} \|h_{\bar{F}_1 - T_1}\|_p \leq & l^{1/p-1} \|h_{\bar{F}_1}\|_1 \leq l^{1/p-1} \left(\|h_{\bar{F}_0 - \bar{F}_1}\|_1 + 2\|\bar{\beta}_{\bar{F}_1}\|_1 \right) \\ \leq & l^{1/p-1} \left(|\bar{F}_0 - \bar{F}_1|^{1-1/p} \|h_{\bar{F}_0 - \bar{F}_1}\|_p + 2\|\bar{\beta}_{\bar{F}_1}\|_1 \right). \end{aligned}$$

For any $x \geq 0$, $y \geq 0$, $p \geq 1$, and $a \geq 0$, it can be easily verified that

$$(x^p + (ax + y)^p)^{1/p} \leq (1 + a^p)^{1/p} x + y. \tag{10}$$

It follows that

$$\begin{aligned} \|h\|_p = & \left[\|h_{F_1 + T_1}\|_p^p + \|h_{\bar{F}_1 - T_1}\|_p^p \right]^{1/p} \\ \leq & \left[\|h_{F_1 + T_1}\|_p^p + \left[\left(\frac{|\bar{F}_0 - \bar{F}_1|}{l} \right)^{1-1/p} \|h_{\bar{F}_0 - \bar{F}_1}\|_p + 2l^{1/p-1} \|\bar{\beta}_{\bar{F}_1}\|_1 \right]^p \right]^{1/p} \\ \leq & \left[1 + \left(\frac{|\bar{F}_0 - \bar{F}_1|}{l} \right)^{p-1} \right]^{1/p} \|h_{F_1 + T_1}\|_p + 2l^{1/p-1} \|\bar{\beta}_{\bar{F}_1}\|_1. \end{aligned}$$

The first inequality is due to the first claim in this lemma; the second inequality is due to $\|h_{\bar{F}_0-\bar{F}_1}\|_p \leq \|h_{F_1+T_1}\|_p$ and (10). We complete the proof for the second claim. ■

Theorem 12 *Under Assumption 1, taking $F_0 \subset F$ and $\lambda = \sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ into the optimization problem* (2)*, for any given index set F_1 satisfying $F_0 \subset F_1 \subset F$, if there exists some l such that $\mu^{(p)}_{A,s_1+l} - \theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p} > 0$ holds where $s_1 = |F_1|$, then with probability larger than $1-\eta_1'$, the ℓ_p norm ($1 \leq p \leq \infty$) of the difference between the optimizer of the problem* (2) *and the oracle solution is bounded as*

$$\|\hat{\beta}-\bar{\beta}\|_p \leq \frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+2^p l)^{1/p}\lambda + 2\theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - \theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} + 2l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1$$

and with probability larger than $1-\eta_1'-\eta_2'$, the ℓ_p norm ($1 \leq p \leq \infty$) of the difference between the optimizer of the problem (2) *and the true solution is bounded as*

$$\|\hat{\beta}-\beta^*\|_p \leq \frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+2^p l)^{1/p}\lambda + 2\theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - \theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} + 2l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1 + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.$$

Proof First, we assume Assumption 2 and the inequality (9) hold. Divide $\bar{F}_1$ into a group of subsets T_j's ($j = 1,2,...$) without intersection such that $\bigcup_j T_j = \bar{F}_1$, $\max_j |T_j| \leq l$ and $\max_{i\in T_{j+1}} h_{T_{j+1}}[i] \leq \|h_{T_j}\|_1/l$ hold. Note that such a partition always exists. Simply, let T_1 be the index set of the largest l elements in h, T_2 be the index set of the largest l elements among the remaining elements, and so on (the size of the last set may be less than l). It is easy to verify that this group of sets satisfy all conditions above. For convenience of presentation, we denote $T_0=\bar{F}_0-\bar{F}_1$ and $T_{01}=T_0+T_1$. Since

$$\begin{aligned}
&\|X^T_{T_{01}+F_0}Xh\|_p\\
=&\|X^T_{T_{01}+F_0}X_{T_{01}+F_0}h_{T_{01}+F_0}+\sum_{j\geq 2}X^T_{T_{01}+F_0}X_{T_j}h_{T_j}\|_p\\
\geq&\mu^{(p)}_{A,s_1+l}\|h_{T_{01}+F_0}\|_p-\sum_{j\geq 2}\theta^{(p)}_{A,s_1+l,l}\|h_{T_j}\|_p\\
\geq&\mu^{(p)}_{A,s_1+l}\|h_{T_{01}+F_0}\|_p-\theta^{(p)}_{A,s_1+l,l}\sum_{j\geq 2}\|h_{T_j}\|_p\\
\geq&\mu^{(p)}_{A,s_1+l}\|h_{T_{01}+F_0}\|_p-\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|h_{\bar{F}_1}\|_1\quad\text{(due to lemma 10)}\\
\geq&\mu^{(p)}_{A,s_1+l}\|h_{T_{01}+F_0}\|_p-\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\left(\|h_{T_0}\|_1+2\|\bar{\beta}_{\bar{F}_1}\|_1\right)\quad\text{(due to lemma 9)}\\
\geq&\mu^{(p)}_{A,s_1+l}\|h_{T_{01}+F_0}\|_p-\theta^{(p)}_{A,s_1+l,l}\left(\frac{l}{|T_0|}\right)^{1/p-1}\|h_{T_0}\|_p-2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\\
\geq&\left(\mu^{(p)}_{A,s_1+l}-\theta^{(p)}_{A,s_1+l,l}\left(\frac{l}{|T_0|}\right)^{1/p-1}\right)\|h_{T_{01}+F_0}\|_p-2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1
\end{aligned}$$

and

$$\begin{aligned}
&\|X^T_{T_{01}+F_0}Xh\|_p^p\\
=&\|X^T_{F_0}Xh\|_p^p+\|X^T_{T_{01}\cap F}Xh\|_p^p+\|X^T_{T_{01}\cap\bar{F}}Xh\|_p^p\\
\leq&|T_{01}\cap F|\lambda^p+|T_{01}\cap\bar{F}|(2\lambda)^p\quad\text{(due to lemma 9)}\\
\leq&|T_0\cap F|\lambda^p+|T_1\cap F|\lambda^p+|T_0\cap\bar{F}|(2\lambda)^p+|T_1\cap\bar{F}|(2\lambda)^p\quad\text{(due to }F_1\subset F)\\
\leq&|T_0|\lambda^p+l(2\lambda)^p,\quad\text{(due to }T_0\cap\bar{F}=\varnothing)
\end{aligned}$$

we have

$$\begin{aligned}
\|h_{F_1+T_1}\|_p=\|h_{T_{01}+F_0}\|_p\leq&\frac{(|T_0|+2^pl)^{1/p}\lambda+2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1}{\mu^{(p)}_{A,s_1+l}-\theta^{(p)}_{A,s_1+l,l}\left(\frac{l}{|T_0|}\right)^{1/p-1}}\\
=&\frac{(|\bar{F}_0-\bar{F}_1|+2^pl)^{1/p}\lambda+2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1}{\mu^{(p)}_{A,s_1+l}-\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}}.
\end{aligned}$$

Due to the second inequality in Lemma 11, we have

$$\begin{aligned}
\|h\|_p\leq&\left[1+\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\|h_{F_1+T_1}\|_p+2l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\\
=&\frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+2^pl)^{1/p}\lambda+2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l}-\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}}+\\
&2l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1.
\end{aligned}$$

Thus, we can bound $\|\hat{\beta}-\beta^*\|_p$ as

$$\begin{aligned}\|\hat{\beta}-\beta^*\|_p \leq &\|\hat{\beta}-\bar{\beta}\|_p+\|\bar{\beta}-\beta^*\|_p \\ \leq &\frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+2^p l)^{1/p}\lambda+2\theta^{(p)}_{A,s_1+l,l}l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l}-\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} \\ &+2l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1+\frac{s^{1/p}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.\end{aligned}$$

Finally, taking $\lambda=\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$, Lemma 8 with $\eta=\eta_1$ implies that Assumption 2 holds with probability larger than $1-\eta_1'$ and Lemma 7 with $\eta=\eta_2$ implies that (9) holds with probability larger than $1-\eta_2'$. Thus, these two bounds above hold with probabilities larger than $1-\eta_1'$ and $1-\eta_1'-\eta_2'$, respectively. ■

Remark 13 *Candès and Tao (2007) provided a more general upper bound for the Dantzig selector solution in the order of* $O\Big(k^{1/2}\sigma\sqrt{\log m}+r_k^{(2)}(\beta^*)\sqrt{\log m}\Big)$, *where* $1\leq k\leq s$ *and* $r_k^{(p)}(\beta)=\left(\sum_{i\in L_k}|\beta_i|^p\right)^{1/p}$ *(L_k is the index set of the k largest entries in β). We argue that the result in Theorem 12 potentially implies a tighter bound for Dantzig selector. Setting* $F_0=\varnothing$ *(equivalent to the standard Dantzig selector) and* $l=k$ *with* $k=|\bar{F}_1|$ *in Theorem 12, it is easy to verify that the order of the bound for* $\|\hat{\beta}_D-\bar{\beta}\|_p$ *is determined by* $O\left(k^{1/p}\sigma\sqrt{\log m}+k^{1/p-1}r_k^{(1)}(\bar{\beta})\right)$, *or* $O\left(k^{1/p}\sigma\sqrt{\log m}+k^{1/p-1}r_k^{(1)}(\beta^*)\right)$ *due to Lemma 7. This bound achieves the same order as the bound of the LASSO solution given by Zhang (2009a), which is the sharpest bound for LASSO to our knowledge.*

We are now ready to prove Theorem 1.

Proof of Theorem 1: Taking $F_1=F$ in theorem 12 which indicates that $\bar{\beta}_{\bar{F}_1}=0$, we conclude that

$$\|\hat{\beta}-\bar{\beta}\|_p\leq\frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0-\bar{F}|+l2^p)^{1/p}}{\mu^{(p)}_{A,s+l}-\theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda$$

holds with probability larger than $1-\eta_1'$ and

$$\begin{aligned}\|\hat{\beta}-\beta^*\|_p \leq &\|\hat{\beta}-\bar{\beta}\|_p+\|\bar{\beta}-\beta^*\|_p \\ \leq &\frac{\left[1+\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0-\bar{F}|+l2^p)^{1/p}}{\mu^{(p)}_{A,s+l}-\theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda+\frac{s^{1/p}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}\end{aligned}$$

holds with probability larger than $1-\eta_1'-\eta_2'$. ■

Proof of Theorem 3: From the proof in Theorem 12, the bounds (3) and (4) in Theorem 1 hold with probability 1 if Assumption 2 and the inequality (9) hold. It is easy to verify by Theorem 1 that for any $j \in J$, the following holds: $|\beta_j^*| > \alpha_0 \geq \|\hat{\beta}-\bar{\beta}\|_\infty + \|\hat{\beta}-\beta^*\|_\infty$. For any $j \in J$, we have

$$|\hat{\beta}_j| \geq |\beta_j^*| - |\hat{\beta}_j - \beta_j^*| > \|\hat{\beta}-\bar{\beta}\|_\infty + \|\hat{\beta}-\beta^*\|_\infty - |\hat{\beta}_j - \beta_j^*| \geq \|\hat{\beta}-\bar{\beta}\|_\infty \geq \|\hat{\beta}_{\bar{F}}\|_\infty.$$

Thus, there exist at least $|J|$ elements of $\hat{\beta}_F$ larger than $\|\hat{\beta}_{\bar{F}}\|_\infty$. If we pick up the largest $|J|$ elements in $\hat{\beta}$, then all of them correspond to the location of nonzero entries in the true solution β^*. Since Assumption 2 and the inequality (9) hold, the bounds (3) and (4) in Theorem 1 hold with probability larger than $1-\eta_1'-\eta_2'$. Thus the claim above holds with probability larger than $1-\eta_1'-\eta_2'$. Note that the probability will not accumulate, as we only need the holding probability of Assumption 2 and the inequality (9). The proofs below follow the same principle. ■

Proof of Theorem 4: From the proof in Theorem 12, the bounds (3) and (4) in Theorem 1 hold with probability 1 if assumption 2 and the inequality (9) hold. In the multi-stage algorithm, the problem in (2) is solved N times. It is easy to verify that the following holds:

$$\alpha_0 \geq \|\hat{\beta}^{(0)}-\bar{\beta}\|_\infty + \|\hat{\beta}^{(0)}-\beta^*\|_\infty.$$

Since $|supp_{\alpha_0}(\beta_J^*)| > 0$, there exists at least 1 element in $\hat{\beta}_J^{(0)}$ larger than $\|\hat{\beta}_{\bar{F}}^{(0)}\|_\infty$. Thus, $F_0^{(1)}$ must be a subset of F. Then, we can verify that

$$\alpha_1 \geq \|\hat{\beta}^{(1)}-\bar{\beta}\|_\infty + \|\hat{\beta}^{(1)}-\beta^*\|_\infty,$$

and $|supp_{\alpha_1}(\beta_J^*)| > 1$ guarantee that there exist at least 2 elements in $\hat{\beta}_J^{(1)}$ larger than $\|\hat{\beta}_{\bar{F}}^{(1)}\|_\infty$. Thus, $F_0^{(2)}$ must be a subset of F. Similarly, we can show that $F_0^{(N)}$ is guaranteed to be a subset of F. Since the bounds (3) and (4) in Theorem 1 hold with probability larger than $1-\eta_1'-\eta_2'$, the claim $F_0^{(N)} \subset F$ holds with probability larger than $1-\eta_1'-\eta_2'$. ■

Proof of Theorem 5: From Theorem 1, the first conclusion holds with probability larger than $1-\eta_1'-\eta_2'$ by choosing $F_0 = \varnothing$ and $l = s$.

Assuming Assumption 2 and the inequality (9) hold, the bounds (3) and (4) in Theorem 1 hold with probability 1. Since the conditions in Theorem 4 are satisfied, the $|J|$ correct features can be selected from the feature set, that is, $F_0^{(|J|)} \subset F$. Using the conclusion in (4) of Theorem 1, the bound of the multi-stage method can be estimated by taking $l = |\bar{F}_0 - \bar{F}|$ as follows:

$$\|\hat{\beta}_{mul}-\beta^*\|_p \leq \frac{(2^{p+1}+2)^{1/p}(s-N)^{1/p}}{\mu_{A,2s-N}^{(p)} - \theta_{A,2s-N,s-N}^{(p)}}\lambda + \frac{s^{1/p}}{\mu_{(X_F^TX_F)^{1/2},s}^{(p)}}\sigma\sqrt{2\log(s/\eta_2)}.$$

Note that since

$$\mu_{A,2s-N}^{(p)} - \theta_{A,2s-N,s-N}^{(p)} \geq \mu_{A,2s}^{(p)} - \theta_{A,2s,s}^{(p)},$$

the following always holds: $\mu_{A,2s-N}^{(p)} - \theta_{A,2s-N,s-N}^{(p)} > 0$. Since Assumption 2 and the inequality (9) hold, the bounds (3) and (4) in Theorem 1 hold with probability larger than $1-\eta_1'-\eta_2'$. Thus the claim above holds with probability larger than $1-\eta_1'-\eta_2'$. ■

Proof of Theorem 6: First, we assume that Assumption 2 and the inequality (9) hold. In this case, the claim in Theorem 4 holds with probability 1. Since all conditions in Theorem 4 are satisfied, after s iterations, s correct features will be selected (i.e., $F_0^{(N)}=F$) with probability 1. Since all correct features are obtained, the optimization problem in the last iteration can be formulated as:

$$\begin{aligned} \min:\ & \|\beta_{\bar{F}}\|_1 \\ s.t.:\ & \|X_{\bar{F}}^T(X\beta-y)\|_\infty \le \lambda \\ & \|X_F^T(X\beta-y)\|_\infty = 0. \end{aligned} \tag{11}$$

The oracle solution minimizes the objective function to 0. Since Assumption 2 indeed implies that the oracle is a feasible solution, the oracle solution is one optimizer. We can also show that it is the unique optimizer. If there is another optimizer $\beta\neq\bar{\beta}$, then $\beta_{\bar{F}}=0$ and $\beta_F=(X_F^TX_F)^{-1}X_F^Ty$, which is identical to the definition of the oracle solution. Thus, we conclude that the oracle is the unique optimizer for the optimization problem (11) with probability 1. Since the holding probability of Assumption 2 and the inequality (9) is larger than $1-\eta_1'-\eta_2'$, the oracle solution can be achieved with the same probability. ■

Appendix B.

In this section, we expound the properties of the multi-stage LASSO which are very similar to the multi-stage Dantzig selector. The complete proof is given below.

In the following discussion, we use $\hat{\beta}'$ to denote the solution in Equation (7) and let $h'=\hat{\beta}'-\bar{\beta}$. We first consider the simple case $F_0\subset F$ as in Section 3.1; we have the following theorem.

Theorem 14 *Assume Assumption 1 holds. Take $F_0\subset F$ and*

$$\lambda'=2\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$$

into the optimization problem (7)*. If there exists some l such that*

$$\mu_{A,s+l}^{(p)}-3\theta_{A,s+l,l}^{(p)}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}>0$$

holds, then with probability larger than $1-\eta_1'$, the ℓ_p norm $(1\le p\le\infty)$ of the difference between $\hat{\beta}'$, the optimizer of the problem (7) *and the oracle solution $\bar{\beta}$ is bounded as*

$$\|\hat{\beta}'-\bar{\beta}\|_p\le\frac{\left[1+3\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0-\bar{F}|+(3/2)^pl)^{1/p}}{\mu_{A,s+l}^{(p)}-3\theta_{A,s+l,l}^{(p)}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda'$$

and with probability larger than $1-\eta_1'-\eta_2'$*, the* ℓ_p *norm* ($1 \le p \le \infty$) *of the difference between* $\hat{\beta}'$*, the optimizer of the problem* (7) *and the true solution* β^* *is bounded as*

$$\|\hat{\beta}'-\beta^*\|_p \le \frac{\left[1+3\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{p-1}\right]^{1/p}(|\bar{F}_0-\bar{F}|+(3/2)^p l)^{1/p}}{\mu^{(p)}_{A,s+l}-3\theta^{(p)}_{A,s+l,l}\left(\frac{|\bar{F}_0-\bar{F}|}{l}\right)^{1-1/p}}\lambda' + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.$$

This theorem is similar to Theorem 1 for the multi-stage Dantzig selector. Like Equations (3) and (4), the two bounds in the above theorem are strictly decreasing in terms of $|\bar{F}_0-\bar{F}|$. Thus, feature selection and signal recovery can benefit from each other. For this reason, the multi-stage LASSO has similar properties as the multi-stage Dantzig selector. We expound them as follows.
(a) First, like Theorem 4, in the LASSO case the multi-stage procedure can lead to a weaker requirement to choose $|J|$ correct features than the standard LASSO as shown in the following theorem.

Theorem 15 *Under Assumption 1, if there exist a nonempty set*

$$\Omega = \{l|\mu^{(p)}_{A,s+l}-3\theta^{(p)}_{A,s+l,l}\left(\frac{s}{l}\right)^{1-1/p}>0\}$$

and a set J such that $|supp_{\alpha_i}(\beta_J^*)|>i$ *holds for all* $i\in\{0,1,...,|J|-1\}$*, where*

$$\alpha_i = \frac{3}{2}\min_{l\in\Omega}\frac{\max(1,\frac{3(s-i)}{l})}{\mu^{(\infty)}_{A,s+l}-3\theta^{(\infty)}_{A,s+l,l}\left(\frac{s-i}{l}\right)}\lambda' + \frac{1}{\mu^{(\infty)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)},$$

then taking $F_0^{(0)}=\varnothing$, $\lambda=\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ *and* $N=|J|-1$ *into the multi-stage algorithm 1, the result after N iterations satisfies* $F_0^{(N)}\subset F$ *(i.e.,* $|J|$ *correct features are chosen) with probability larger than* $1-\eta_1'-\eta_2'$.

It is easy to see that $\alpha_0>\alpha_1>...>\alpha_{|J|-1}$ holds strictly. Referring to the analysis for Theorem 4, we know that the multi-stage method for LASSO requires weaker conditions to obtain $|J|$ correct features than the standard LASSO.
(b) Second, like Theorem 5 the following theorem shows that with a high probability the multi-stage procedure can improve the upper bound of the standard LASSO from $Cs^{1/p}\sqrt{\log m}+\Delta$ to $C(s-N)^{1/p}\sqrt{\log m}+\Delta$, where C is a constant and Δ is a small number independent from m.

Theorem 16 *Under Assumption 1, if there exist l such that* $\mu^{(\infty)}_{A,s+l}-3\theta^{(\infty)}_{A,s+l,l}\left(\frac{s}{l}\right)>0$, $\mu^{(p)}_{A,2s}-3\theta^{(p)}_{A,2s,s}>0$*, and a set J such that* $|supp_{\alpha_i}(\beta_J^*)|>i$ *holds for all* $i\in\{0,1,...,|J|-1\}$*, where* α_i*'s follow the definition in Theorem 15, then taking* $F_0=\varnothing$, $N=|J|$ *and* $\lambda'=2\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ *into the multi-stage LASSO algorithm, the solution* $\hat{\beta}'_{mul}$ *of the multi-stage LASSO obeys*

$$\|\hat{\beta}'_{mul}-\beta^*\|_p \le \frac{4\left(\left(\frac{3}{2}\right)^p+1\right)^{1/p}(s-N)^{1/p}}{\mu^{(p)}_{A,2s-N}-3\theta^{(p)}_{A,2s-N,s-N}}\lambda' + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}$$

with probability larger than $1-\eta_1'-\eta_2'$.

(c) Finally, the proposed method can obtain the oracle solution with high probability under certain conditions:

Theorem 17 *Under Assumption 1, if there exists l such that $\mu^{(\infty)}_{A,s+l} - 3\theta^{(\infty)}_{A,s+l,l}\left(\frac{s-i}{l}\right) > 0$, and the supporting set F of β^* satisfies $|supp_{\alpha_i}(\beta^*_F)| > i$ for all $i \in \{0,1,...,s-1\}$, where α_i follows the definition in theorem 15, then taking $F_0 = \varnothing$, $N = s$ and $\lambda' = 2\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ into the multi-stage LASSO algorithm, the oracle solution can be achieved, that is, $F_0^{(N)} = F$ and $\hat{\beta}'^{(N)} = \bar{\beta}$ with probability larger than $1-\eta_1'-\eta_2'$.*

In the following, we provide the complete proof for the theorems above.

Lemma 18 *Let $\hat{\beta}'$ be defined above. We have*

$$\|X^T_{\bar{F}_0}(X\hat{\beta}'-y)\|_\infty \leq \lambda'.$$

Proof The subdifferential of the objective function in Equation (7) at the optimal solution $\hat{\beta}'$ is given by:

$$X_i^T(X\hat{\beta}'-y) + \lambda' sgn(\hat{\beta}'_i)$$

where $i \in \bar{F}_0$ and

$$sgn(x) = \begin{cases} 1, & x > 0; \\ -1, & x < 0; \\ \text{[-1,1]}, & x = 0. \end{cases}$$

Since 0 must belong to the subdifferential at the optimal solution, we have

$$|X_i^T(X\hat{\beta}'-y)| \leq \lambda',$$

which implies the claim. ■

Let us assume that the oracle solution satisfies the following assumption.

Assumption 3

$$\|X^T_{\bar{F}}(X\bar{\beta}-y)\|_\infty \leq \lambda'/2.$$

This assumption actually plays the same role as Assumption 2 in the Dantzig selector.

In the following, we introduce an additional set F_1 satisfying $F_0 \subset F_1$ (Zhang, 2009a).

Similar to Lemma 9, we have the following results for the LASSO case:

Lemma 19 *Let $F_0 \subset F$. Assume that Assumption 3 holds. Given any index set F_1 such that $F_0 \subset F_1$, we have the following conclusions:*

$$\begin{aligned} 3\|h'_{\bar{F}_0-\bar{F}_1}\|_1 + 4\|\bar{\beta}_{\bar{F}_1}\|_1 &\geq \|h'_{\bar{F}_1}\|_1 \\ \|X^T_{F_0}Xh'\|_\infty &= 0 \\ \|X^T_{\bar{F}}Xh'\|_\infty &\leq \frac{3}{2}\lambda' \\ \|X^T_{\bar{F}_0-\bar{F}}Xh'\|_\infty &\leq \lambda'. \end{aligned}$$

Proof We only show the proof for the first inequality and the rest can be easily proven by following the proof in Lemma 9.

Let $\varepsilon = X\bar{\beta} - y$ and $f(.)$ be the objective function in Equation (7) with respect to β. One can verify that $\varepsilon^T X_F = 0$. Since $\hat{\beta}'$ is the optimal solution of Equation (7), we have

$$\begin{aligned}
0 \geq & f(\hat{\beta}') - f(\bar{\beta}) \\
= & \frac{1}{2}(\|X\hat{\beta}' - y\|_2^2 - \|X\bar{\beta} - y\|_2^2) + \lambda'(\|\hat{\beta}'_{\bar{F}_0}\|_1 - \|\bar{\beta}_{\bar{F}_0}\|_1) \\
= & \frac{1}{2}(Xh')^T(X\hat{\beta}' - y + \varepsilon) + \lambda'(\|\hat{\beta}'_{\bar{F}_0}\|_1 - \|\bar{\beta}_{\bar{F}_0}\|_1) \\
\geq & \varepsilon^T Xh' + \lambda'(\|\hat{\beta}'_{\bar{F}_0}\|_1 - \|\bar{\beta}_{\bar{F}_0}\|_1) \\
\geq & \varepsilon^T(X_F h'_F + X_{\bar{F}} h'_{\bar{F}}) + \lambda'(\|\hat{\beta}'_{\bar{F}_0}\|_1 - \|\bar{\beta}_{\bar{F}_0}\|_1) \\
\geq & -\lambda'/2\|h'_{\bar{F}}\|_1 + \lambda'(\|\hat{\beta}'_{\bar{F}_0 - \bar{F}_1}\|_1 + \|\hat{\beta}'_{\bar{F}_1}\|_1 - \|\bar{\beta}_{\bar{F}_0 - \bar{F}_1}\|_1 - \|\bar{\beta}_{\bar{F}_1}\|_1) \quad \text{(due to Assumption 3)} \\
\geq & -\lambda'/2\|h'_{\bar{F}_0}\|_1 + \lambda'(-\|h'_{\bar{F}_0 - \bar{F}_1}\|_1 + \|h'_{\bar{F}_1}\|_1 - 2\|\bar{\beta}_{\bar{F}_1}\|_1) \\
= & \lambda'/2\|h'_{\bar{F}_1}\|_1 - \frac{3}{2}\lambda'\|h'_{\bar{F}_0 - \bar{F}_1}\|_1 - 2\lambda'\|\bar{\beta}_{\bar{F}_1}\|_1,
\end{aligned}$$

which implies the first inequality. ■

Similar to Lemma 11, the following result holds in the LASSO case:

Lemma 20 *Assume $F_0 \subset F$ and $F_0 \subset F_1$ and the index set $\bar{F}_1$ is divided into a group of subsets T_j's such that they satisfy all conditions in Lemma 10 with $v = h'$. Then the following holds:*

$$\begin{aligned}
\|h'_{\bar{F}_1 - T_1}\|_p \leq & l^{1/p-1}\left(3|\bar{F}_0 - \bar{F}_1|^{1-1/p}\|h'_{\bar{F}_0 - \bar{F}_1}\|_p + 4\|\bar{\beta}_{\bar{F}_1}\|_1\right) \\
\|h'\|_p \leq & \left[1 + 3^p(|\bar{F}_0 - \bar{F}_1|/l)^{p-1}\right]^{1/p}\|h'_{F_1+T_1}\|_p + 4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1,
\end{aligned}$$

where $s_1 = |F_1|$.

Proof Using the claim in Lemma 10 with $v = h'$, we have

$$\begin{aligned}
& \|h'_{\bar{F}_1 - T_1}\|_p \\
\leq & l^{1/p-1}\|h'_{\bar{F}_1}\|_1 \\
\leq & l^{1/p-1}\left(3\|h'_{\bar{F}_0 - \bar{F}_1}\|_1 + 4\|\bar{\beta}_{\bar{F}_1}\|_1\right) \quad \text{(due to the first inequality in Lemma 19)} \\
\leq & l^{1/p-1}\left(3|\bar{F}_0 - \bar{F}_1|^{1-1/p}\|h'_{\bar{F}_0 - \bar{F}_1}\|_p + 4\|\bar{\beta}_{\bar{F}_1}\|_1\right).
\end{aligned}$$

This proves the first inequality. Using this equality, we can obtain the second inequality as follows:

$$\begin{aligned}
& \|h'\|_p \\
= & (\|h'_{F_1+T_1}\|_p^p + \|h'_{\bar{F}_1 - T_1}\|_p^p)^{1/p} \\
\leq & \left[\|h'_{F_1+T_1}\|_p^p + \left(3\left(\frac{|\bar{F}_0 - \bar{F}_1|}{l}\right)^{1-1/p}\|h'_{\bar{F}_0 - \bar{F}_1}\|_p + \frac{4}{l^{1-1/p}}\|\bar{\beta}_{\bar{F}_1}\|_1\right)^p\right]^{1/p} \\
\leq & \left[1 + 3^p(|\bar{F}_0 - \bar{F}_1|/l)^{p-1}\right]^{1/p}\|h'_{F_1+T_1}\|_p + 4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1.
\end{aligned}$$

The last inequality is due to Equation (10). ■

Similar to the Lemma 12, the following result holds in the LASSO case:

Theorem 21 *Under Assumption 1, taking $F_0 \subset F$ and $\lambda' = 2\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)}$ into the optimization problem* (7)*, if for any index set F_1 satisfying $F_0 \subset F_1 \subset F$ there exists some l such that $\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p} > 0$ holds where $s_1 = |F_1|$, then with probability larger than $1-\eta_1'$, the ℓ_p norm ($1 \leq p \leq \infty$) of the difference between the optimizer of the problem* (7) *and the oracle solution is bounded as*

$$
\begin{aligned}
&\|\hat{\beta}' - \bar{\beta}\|_p \\
\leq &\frac{\left[1+3^p\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+(3/2)^p l)^{1/p}\lambda' + \frac{4\theta^{(p)}_{A,s_1+l,l}}{l^{1-1/p}}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} \\
&+4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1
\end{aligned}
$$

and with probability larger than $1-\eta_1'-\eta_2'$, the ℓ_p norm ($1 \leq p \leq \infty$) of the difference between the optimizer of the problem (7) *and the true solution is bounded as*

$$
\begin{aligned}
&\|\hat{\beta}' - \beta^*\|_p \\
\leq &\frac{\left[1+3^p\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1|+(3/2)^p l)^{1/p}\lambda' + \frac{4\theta^{(p)}_{A,s_1+l,l}}{l^{1-1/p}}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} \\
&+4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1 + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^TX_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.
\end{aligned}
$$

Proof The proof follows the same strategy as in Theorem 12. First, we assume that Assumption 3 and the inequality (9) hold. Divide $\bar{F}_1$ into a group of subsets T_j's ($j = 1, 2, \ldots$) without intersection such that $\bigcup_j T_j = \bar{F}_1$, $\max_j |T_j| \leq l$ and $\max_{i \in T_{j+1}} h_{T_{j+1}}[i] \leq \|h_{T_j}\|_1/l$ hold. Since

$$\begin{aligned}
&\|X^T_{T_{01}+F_0} X h'\|_p \\
=&\|X^T_{T_{01}+F_0} X_{T_{01}+F_0} h'_{T_{01}+F_0} + \sum_{j\geq 2} X^T_{T_{01}+F_0} X_{T_j} h'_{T_j}\|_p \\
\geq&\mu^{(p)}_{A,s_1+l}\|h'_{T_{01}+F_0}\|_p - \sum_{j\geq 2}\theta^{(p)}_{A,s_1+l,l}\|h'_{T_j}\|_p \\
\geq&\mu^{(p)}_{A,s_1+l}\|h'_{T_{01}+F_0}\|_p - \theta^{(p)}_{A,s_1+l,l}\sum_{j\geq 2}\|h'_{T_j}\|_p \\
\geq&\mu^{(p)}_{A,s_1+l}\|h'_{T_{01}+F_0}\|_p - \theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|h'_{\bar{F}_1}\|_1 \\
\geq&\mu^{(p)}_{A,s_1+l}\|h'_{T_{01}+F_0}\|_p - \theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\left(3\|h'_{\bar{F}_0-\bar{F}_1}\|_1 + 4\|\bar{\beta}_{\bar{F}_1}\|_1\right) \\
&\quad\text{(due to the first inequality of Lemma 19)} \\
\geq&\mu^{(p)}_{A,s_1+l}\|h'_{T_{01}+F_0}\|_p - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{l}{|T_0|}\right)^{1/p-1}\|h'_{T_0}\|_p - 4\theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1 \\
\geq&\left[\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|T_0|}{l}\right)^{1-1/p}\right]\|h'_{T_{01}+F_0}\|_p - 4\theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1
\end{aligned}$$

and

$$\begin{aligned}
&\|X^T_{T_{01}+F_0} X h'\|^p_p \\
=&\|X^T_{F_0} X h'\|^p_p + \|X^T_{T_{01}\cap F} X h'\|^p_p + \|X^T_{T_{01}\cap \bar{F}} X h'\|^p_p \\
\leq&|T_{01}\cap F|\lambda'^p + |T_{01}\cap\bar{F}|(3\lambda'/2)^p \quad \text{(due to Lemma 19)} \\
\leq&|T_0\cap F|\lambda'^p + |T_1\cap F|\lambda'^p + |T_0\cap\bar{F}|(3\lambda'/2)^p + |T_1\cap\bar{F}|(3\lambda'/2)^p \quad \text{(due to } F_1\subset F\text{)} \\
\leq&|T_0|\lambda'^p + l(3\lambda'/2)^p, \quad \text{(due to } T_0\cap\bar{F}=\varnothing\text{)}
\end{aligned}$$

thus we have

$$\|h'_{F_1+T_1}\|_p = \|h'_{T_{01}+F_0}\|_p \leq \frac{(|\bar{F}_0-\bar{F}_1| + (3/2)^p l)^{1/p}\lambda' + 4\theta^{(p)}_{A,s_1+l,l} l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1}{\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}}.$$

It follows that

$$\begin{aligned}
\|h'\|_p \leq& \left[1+3^p\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\|h'_{F_1+T_1}\|_p + 4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1 \\
\leq& \frac{\left[1+3^p\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{p-1}\right]^{1/p}\left((|\bar{F}_0-\bar{F}_1| + (3/2)^p l)^{1/p}\lambda' + \frac{4\theta^{(p)}_{A,s_1+l,l}}{l^{1-1/p}}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0-\bar{F}_1|}{l}\right)^{1-1/p}} \\
&+4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1,
\end{aligned}$$

and

$$
\begin{aligned}
&\|\hat{\beta}' - \beta^*\|_p \\
\leq & \frac{\left[1 + 3\left(\frac{|\bar{F}_0 - \bar{F}_1|}{l}\right)^{p-1}\right]^{1/p} \left((|\bar{F}_0 - \bar{F}_1| + (3/2)^p l)^{1/p}\lambda' + \frac{4\theta^{(p)}_{A,s_1+l,l}}{l^{1-1/p}}\|\bar{\beta}_{\bar{F}_1}\|_1\right)}{\mu^{(p)}_{A,s_1+l} - 3\theta^{(p)}_{A,s_1+l,l}\left(\frac{|\bar{F}_0 - \bar{F}_1|}{l}\right)^{1-1/p}} \\
&+ 4l^{1/p-1}\|\bar{\beta}_{\bar{F}_1}\|_1 + \frac{s^{1/p}}{\mu^{(p)}_{(X_F^T X_F)^{1/2},s}}\sigma\sqrt{2\log(s/\eta_2)}.
\end{aligned}
$$

Finally, taking

$$
\lambda' = 2\sigma\sqrt{2\log\left(\frac{m-s}{\eta_1}\right)},
$$

Lemma 8 (letting $\eta = \eta_1$) implies that Assumption 3 holds with probability larger than $1 - \eta_1'$ and Lemma 7 (letting $\eta = \eta_2$) implies that Equation (9) holds with probability larger than $1 - \eta_2'$. Thus, these two bounds above hold with probability larger than respectively $1 - \eta_1'$ and $1 - \eta_1' - \eta_2'$. ■

Proof to Theorem 14: By taking $F_1 = F$ in Theorem 21, the claims above can be obtained immediately. ■

Proof to Theorem 15: Please refer to the proof for Theorem 4. ■

Proof to Theorem 16: Please refer to the proof for Theorem 5. ■

Proof to Theorem 17: First, we assume that Assumption 3 and the inequality (9) holds. Then, the claim in Theorem 15 holds with probability 1. Since all conditions in Theorem 15 are satisfied, after s iterations s correct features can be chosen (i.e., $F_0^{(N)} = F$) with probability 1. Since all correct features are obtained, the optimization problem in the last iteration can be formulated as

$$
\min: \frac{1}{2}\|X\beta - y\|_2^2 + \lambda'\|\beta_{\bar{F}}\|_1. \tag{12}
$$

A minimizer should satisfy the following conditions:

$$
\begin{aligned}
0 \in & X_{\bar{F}}^T(X\beta - y) + \lambda' sgn(\beta_{\bar{F}}) \\
0 = & X_F^T(X\beta - y),
\end{aligned} \tag{13}
$$

where the first formula is based on the subdifferential set. Because of Assumption 3, the oracle solution satisfies these two conditions. Since the objective function is not strictly convex, we need to show that the oracle solution is the unique minimizer.

From the second equality in Equation (13), we have $\beta_F = -(X_F^T X_F)^{-1} X_F^T (X_{\bar{F}}\beta_{\bar{F}} - y)$. It follows that the objective function in Equation (12) can be expressed as

$$
f(\beta_{\bar{F}}) = \frac{1}{2}\|(I - X_F(X_F^T X_F)^{-1} X_F^T)(X_{\bar{F}}\beta_{\bar{F}} - y)\|_2^2 + \lambda'\|\beta_{\bar{F}}\|_1.
$$

Because the oracle solution is a minimizer of the Equation (12), "0" should be one of the minimizers of $f(\beta_{\bar{F}})$. Next we show that "0" is the unique minimizer, which implies that the oracle solution is the unique minimizer for Equation (12). We can compute the directional derivative along any direction Δ at the point "0" for the function $f(\beta_{\bar{F}})$ as follows:

$$\begin{aligned}\frac{df(0+t\Delta)}{dt}|_{t=0} =& -y^T(I-X_F(X_F^TX_F)^{-1}X_F^T)X_{\bar{F}}^T\Delta+\lambda'\|\Delta\|_1\\ \geq& \lambda'\|\Delta\|_1-\|\Delta\|_1\|y^T(I-X_F(X_F^TX_F)^{-1}X_F^T)X_{\bar{F}}^T\|_\infty\\ =&\|\Delta\|_1(\lambda'-\|X_{\bar{F}}^T(X\bar{\beta}-y)\|_\infty)\\ >&0. \quad \text{(due to Assumption 3)}\end{aligned}$$

Thus, the directional derivative at "0" is always strictly greater than 0 at arbitrary directions, which shows that "0" should be the unique minimizer for $f(\beta_{\bar{F}})$.

Finally, because the probability of Assumption 3 and the inequality (9) holding is larger than $1-\eta_1'-\eta_2'$, the oracle solution is achieved with the same probability. ■

References

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the Lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011.

T. Cai, G. Xu, and J. Zhang. On recovery of sparse signals via ℓ_1 minimization. *IEEE Transactions on Information Theory*, 55(7):3388–3397, 2009.

E. J. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37 (5A):2145–2177, 2009.

E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

E. J. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

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, 2006.

J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *(invited review article) Statistica Sinica*, 20:101–148, 2010.

J. Fan and J. Lv. Nonconcave penalized likelihood with np-dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484, 2011.

G. M. James, P. Radchenko, and J. Lv. DASSO: connections between the Dantzig selector and Lasso. *Journal of The Royal Statistical Society Series B*, 71(1):127–142, 2009.

V. Koltchinskii and M. Yuan. Sparse recovery in large ensembles of kernel machines on-line learning and bandits. In *Proceedings of the Twenty-First Annual Conference on Learning Theory (COLT)*, pages 229–238, Helsinki, Finland, 2008.

K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electronic Journal of Statistics*, 2:90–102, 2008.

J. Lv and Y. Fan. A unified approach to model selection and sparse recovery using regularized least squares. *Annals of Statistics*, 37(6A):3498–3528, 2009.

N. Meinshausen, P. Bhlmann, and E. Zrich. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

P. Ravikumar, G. Raskutti, M. J. Wainwright, and B. Yu. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized MLE. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1329–1336, Vancouver, British Columbia, Canada, 2008.

J. Romberg. The Dantzig selector and generalized thresholding. In *Proceedings of the Forty-Second Annual Conference on Information Sciences and Systems (CISS)*, pages 22–25, Princeton, New Jersey, USA, 2008.

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

J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

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, 2009.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010a.

C.-H. Zhang and T. Zhang. A general theory of concave regularization for high dimensional sparse estimation problems. Technical report, Department of Statistics, Rutgers University, Piscataway, New Jersey, USA, 2012.

T. Zhang. Some sharp performance bounds for least squares regression with ℓ_1 regularization. *Annals of Statistics*, 37(5A):2109–2114, 2009a.

T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009b.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

T. Zhang. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Transactions on Information Theory*, 57(9):5215–6221, 2011a.

T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011b.

T. Zhang. Multi-stage convex relaxation for feature selection. Technical report, Department of Statistics, Rutgers University, Piscataway, New Jersey, USA, 2011c.

P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

S. Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2304–2312, Vancouver, British Columbia, Canada, 2009.

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

Journal of Machine Learning Research 13 (2012) 1221-1261 Submitted 11/09; Revised 2/12; Published 4/12

A Geometric Approach to Sample Compression

Benjamin I. P. Rubinstein BEN.RUBINSTEIN@MICROSOFT.COM
Microsoft Research
1288 Pear Avenue
Mountain View, CA 94043, USA

J. Hyam Rubinstein RUBIN@MS.UNIMELB.EDU.AU
Department of Mathematics & Statistics
University of Melbourne
Parkville, Victoria 3010, Australia

Editor: Manfred K. Warmuth

Abstract

The Sample Compression Conjecture of Littlestone & Warmuth has remained unsolved for a quarter century. While maximum classes (concept classes meeting Sauer's Lemma with equality) can be compressed, the compression of general concept classes reduces to compressing maximal classes (classes that cannot be expanded without increasing VC dimension). Two promising ways forward are: embedding maximal classes into maximum classes with at most a polynomial increase to VC dimension, and compression via operating on geometric representations. This paper presents positive results on the latter approach and a first negative result on the former, through a systematic investigation of finite maximum classes. Simple arrangements of hyperplanes in hyperbolic space are shown to represent maximum classes, generalizing the corresponding Euclidean result. We show that sweeping a generic hyperplane across such arrangements forms an unlabeled compression scheme of size VC dimension and corresponds to a special case of peeling the one-inclusion graph, resolving a recent conjecture of Kuzmin & Warmuth. A bijection between finite maximum classes and certain arrangements of piecewise-linear (PL) hyperplanes in either a ball or Euclidean space is established. Finally we show that d-maximum classes corresponding to PL-hyperplane arrangements in $\mathbb{R}^d$ have cubical complexes homeomorphic to a d-ball, or equivalently complexes that are manifolds with boundary. A main result is that PL arrangements can be swept by a moving hyperplane to unlabeled d-compress *any* finite maximum class, forming a peeling scheme as conjectured by Kuzmin & Warmuth. A corollary is that some d-maximal classes cannot be embedded into any maximum class of VC-dimension $d+k$, for any constant k. The construction of the PL sweeping involves Pachner moves on the one-inclusion graph, corresponding to moves of a hyperplane across the intersection of d other hyperplanes. This extends the well known Pachner moves for triangulations to cubical complexes.

Keywords: sample compression, hyperplane arrangements, hyperbolic and piecewise-linear geometry, one-inclusion graphs

1. Introduction

Maximum concept classes have the largest cardinality possible for their given VC dimension. Such classes are of particular interest as their special recursive structure underlies all general sample compression schemes known to-date (Floyd, 1989; Warmuth, 2003; Kuzmin and Warmuth, 2007).

It is this structure that admits many elegant geometric and algebraic topological representations upon which this paper focuses.

Littlestone and Warmuth (1986) introduced the study of *sample compression schemes*, defined as a pair of mappings for given concept class C: a *compression function* mapping a C-labeled n-sample to a subsequence of labeled examples and a *reconstruction function* mapping the subsequence to a concept consistent with the entire n-sample. A compression scheme of bounded size—the maximum cardinality of the subsequence image—was shown to imply learnability. The converse—that classes of VC-dimension d admit compression schemes of size d—has become one of the oldest unsolved problems actively pursued within learning theory (Floyd, 1989; Helmbold et al., 1992; Ben-David and Litman, 1998; Warmuth, 2003; Hellerstein, 2006; Kuzmin and Warmuth, 2007; Rubinstein et al., 2007, 2009; Rubinstein and Rubinstein, 2008). Interest in the conjecture has been motivated by its interpretation as the converse to the existence of compression bounds for PAC learnable classes (Littlestone and Warmuth, 1986), the basis of practical machine learning methods on compression schemes (Marchand and Shawe-Taylor, 2003; von Luxburg et al., 2004), and the conjecture's connection to a deeper understanding of the combinatorial properties of concept classes (Rubinstein et al., 2009; Rubinstein and Rubinstein, 2008). Recently Kuzmin and Warmuth (2007) achieved compression of maximum classes without the use of labels. They also conjectured that their elegant min-peeling algorithm constitutes such an unlabeled d-compression scheme for d-maximum classes.

As discussed in our previous work (Rubinstein et al., 2009), maximum classes can be fruitfully viewed as *cubical complexes*. These are also topological spaces, with each cube equipped with a natural topology of open sets from its standard embedding into Euclidean space. We proved that d-maximum classes correspond to *d-contractible complexes*—topological spaces with an identity map homotopic to a constant map—extending the result that 1-maximum classes have trees for one-inclusion graphs. Peeling can be viewed as a special form of contractibility for maximum classes. However, there are many non-maximum contractible cubical complexes that cannot be peeled, which demonstrates that peelability reflects more detailed structure of maximum classes than given by contractibility alone.

In this paper we approach peeling from the direction of simple hyperplane arrangement representations of maximum classes. Kuzmin and Warmuth (2007, Conjecture 1) predicted that d-maximum classes corresponding to simple linear-hyperplane arrangements could be unlabeled d-compressed by sweeping a generic hyperplane across the arrangement, and that concepts are min peeled as their corresponding cell is swept away. We positively resolve the first part of the conjecture and show that sweeping such arrangements corresponds to a new form of *corner peeling*, which we prove is distinct from min peeling. While *min peeling* removes minimum degree concepts from a one-inclusion graph, corner peeling peels vertices that are contained in unique cubes of maximum dimension.

We explore simple hyperplane arrangements in hyperbolic geometry, which we show correspond to a set of maximum classes, properly containing those represented by simple linear Euclidean arrangements. These classes can again be corner peeled by sweeping. Citing the proof of existence of maximum unlabeled compression schemes due to Ben-David and Litman (1998), Kuzmin and Warmuth (2007) ask whether unlabeled compression schemes for infinite classes such as positive half spaces can be constructed explicitly. We present constructions for illustrative but simpler classes, suggesting that there are many interesting infinite maximum classes admitting explicit compression schemes, and under appropriate conditions, sweeping infinite Euclidean, hyperbolic or PL arrangements corresponds to compression by corner peeling.

Next we prove that all maximum classes in $\{0,1\}^n$ are represented as simple arrangements of piecewise-linear (PL) hyperplanes in the n-ball. This extends previous work by Gärtner and Welzl (1994) on viewing simple PL-hyperplane arrangements as maximum classes. The close relationship between such arrangements and their hyperbolic versions suggests that they could be equivalent. Resolving the main problem left open in the preliminary version of this paper (Rubinstein and Rubinstein, 2008), we show that sweeping of d-contractible PL arrangements does compress all finite maximum classes by corner peeling, completing (Kuzmin and Warmuth, 2007, Conjecture 1).

We show that a one-inclusion graph Γ can be represented by a d-contractible PL-hyperplane arrangement if and only if Γ is a strongly contractible cubical complex. This motivates the nomenclature of d-contractible for the class of arrangements of PL hyperplanes. Note then that these one-inclusion graphs admit a corner-peeling scheme of the same size d as the largest dimension of a cube in Γ. Moreover if such a graph Γ admits a corner-peeling scheme, then it is a contractible cubical complex. We give a simple example to show that there are one-inclusion graphs which admit corner-peeling schemes but are not strongly contractible and so are not represented by a d-contractible PL-hyperplane arrangement.

Compressing *maximal classes*—classes which cannot be grown without an increase to their VC dimension—is sufficient for compressing all classes, as embedded classes trivially inherit compression schemes of their super-classes. This reasoning motivates the attempt to embed d-maximal classes into $O(d)$-maximum classes (Kuzmin and Warmuth, 2007, Open Problem 3). We present non-embeddability results following from our earlier counter-examples to Kuzmin & Warmuth's minimum degree conjecture (Rubinstein et al., 2009), and our new results on corner peeling. We explore with examples, maximal classes that can be compressed but not peeled, and classes that are not strongly contractible but can be compressed.

Finally, we investigate algebraic topological properties of maximum classes. Most notably we characterize d-maximum classes, corresponding to simple linear Euclidean arrangements, as cubical complexes homeomorphic to the d-ball. The result that such classes' boundaries are homeomorphic to the $(d-1)$-sphere begins the study of the boundaries of maximum classes, which are closely related to peeling. We conclude with several open problems.

2. Background

We begin by presenting relevant background material on algebraic topology, computational learning theory, and sample compression.

2.1 Algebraic Topology

Definition 1 *A* homeomorphism *is a one-to-one and onto map f between topological spaces such that both f and f^{-1} are continuous. Spaces X and Y are said to be* homeomorphic *if there exists a homeomorphism $f : X \to Y$.*

Definition 2 *A* homotopy *is a continuous map $F : X \times [0,1] \to Y$. The* initial map *is F restricted to $X \times \{0\}$ and the* final map *is F restricted to $X \times \{1\}$. We say that the initial and final maps are* homotopic. *A* homotopy equivalence *between spaces X and Y is a pair of maps $f : X \to Y$ and $g : Y \to X$ such that $f \circ g$ and $g \circ f$ are homotopic to the identity maps on Y and X respectively. We* *say that X and Y have the* same homotopy type *if there is a homotopy equivalence between them. A* deformation retraction *is a special homotopy equivalence between a space X and a subspace $A \subseteq X$. It is a continuous map $r : X \to X$ with the properties that the restriction of r to A is the identity map on A, r has range A and r is homotopic to the identity map on X.*

Definition 3 *A* cubical complex *is a union of solid cubes of the form $[a_1,b_1] \times \ldots \times [a_m,b_m]$, for bounded $m \in \mathbb{N}$, such that the intersection of any two cubes in the complex is either a cubical face of both cubes or the empty-set.*

Definition 4 *A* contractible cubical complex *X is one which has the same homotopy type as a one point space $\{p\}$. X is contractible if and only if the constant map from X to p is a homotopy equivalence.*

Definition 5 *A* simplicial complex *is a union of simplices, each of which is affinely equivalent*[1] *to the convex hull of $k+1$ points $(0,0,\ldots,0),(1,0,\ldots,0),\ldots(0,0,\ldots,1)$ in $\mathbb{R}^k$, for some k. The intersection of any two simplices in the complex is either a face of both simplices or the empty-set. A map $f : X \to Y$ is called* simplicial *if X,Y are simplicial complexes and f maps each simplex of X to a simplex of Y so that vertices are mapped to vertices and the map is affine linear. A* subdivision *of a simplicial complex is a new simplicial complex with the same underlying point-set obtained by cutting up the original simplices into smaller simplices.*

For a more formal treatment of simplicial complexes see (Rourke and Sanderson, 1982). We will need the concepts of piecewise-linear (PL) manifolds and maps.

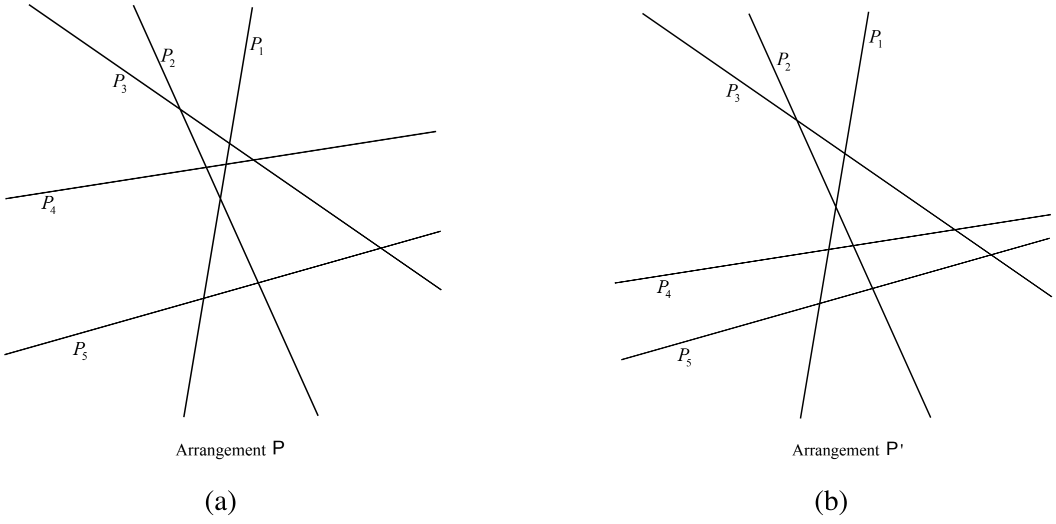
Definition 6 *A mapping $f : X \to Y$ is called* piecewise linear *(PL) if X,Y are simplicial complexes and there are subdivisions $X^\star, Y^\star$ of the respective complexes, so that $f : X^\star \to Y^\star$ is simplicial. A PL homeomorphism $f : X \to Y$ is a bijection so that both f, f^{-1} are PL maps. A PL manifold M is a space which is covered by open sets U_α for $\alpha \in I$ some index set, together with bijections $\phi_\alpha : U_\alpha \to V_\alpha$, where V_α is an open set in $\mathbb{R}^n$. Moreover when $U_\alpha \cap U_\beta \neq \emptyset$, then the transition function $\phi_\beta \circ {\phi_\alpha}^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)$ is a PL homeomorphism. A pair (U_α, ϕ_α) is called a* chart *for M.*

2.2 Pachner Moves

Pachner (1987) showed that triangulations of manifolds which are combinatorially equivalent after subdivision are also equivalent by a series of moves which are now referred to as Pachner moves. For the main result of this paper, we need a version of Pachner moves for cubical structures rather than simplicial ones. The main idea of Pachner moves remains the same.

A *Pachner move* replaces a topological d-ball U divided into d-cubes, with another ball U' with the same $(d-1)$-cubical boundary but with a different interior cubical structure. In dimension $d = 2$, for example, such an initial ball U can be constructed by taking three 2-cubes forming a hexagonal disk and in dimension $d = 3$, four 3-cubes forming a rhombic dodecahedron, which is a polyhedron U with 12 quadrilateral faces in its boundary. The set U' of d-cubes is attached to the same boundary as for U, that is, $\partial U = \partial U'$, as cubical complexes homeomorphic to the $(d-1)$-sphere. Moreover, U' and U are isomorphic cubical complexes, but the gluing between their boundaries produces

[1] The simplices are related via an affine bijection.



(a)

(b)

Figure 1: (a) An example linear-hyperplane arrangement $\mathcal{P}$ and (b) the result of a Pachner move of hyperplane P_4 on $\mathcal{P}$.

the boundary of the 3- or 4-cube, as a 2- or 3-dimensional cubical structure on the 2- or 3-sphere respectively.

To better understand this move, consider the cubical face structure of the boundary V of the $(d+1)$-cube. This is a d-sphere containing $2d+2$ cubes, each of dimension d. There are many embeddings of the $(d-1)$-sphere as a cubical subcomplex into V, dividing it into a pair of d-balls. One ball is combinatorially identical to U and the other to U'.

There are a whole series of Pachner moves in each dimension d, but we are only interested in the ones where the pair of balls U, U' have the same numbers of d-cubes. In Figure 1 a change in a hyperplane arrangement is shown, which corresponds to a Pachner move on the corresponding one-inclusion graph (considered as a cubical complex).

2.3 Concept Classes and their Learnability

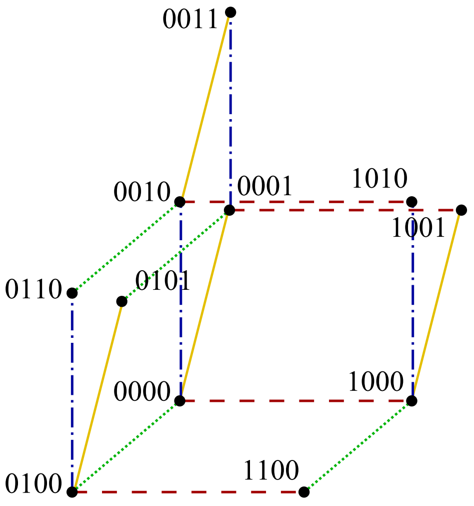
A *concept class* C on *domain* X, is a subset of the power set of set X or equivalently $C \subseteq \{0,1\}^X$. We primarily consider finite domains and so will write $C \subseteq \{0,1\}^n$ in the sequel, where it is understood that $n = |X|$ and the n dimensions or *colors* are identified with an ordering $\{x_i\}_{i=1}^n = X$.

The *one-inclusion graph* $\mathcal{G}(C)$ of $C \subseteq \{0,1\}^n$ is the graph with vertex-set C and edge-set containing $\{u,v\} \subseteq C$ iff u and v differ on exactly one component (Haussler et al., 1994); $\mathcal{G}(C)$ forms the basis of a prediction strategy with essentially-optimal worst-case expected risk. $\mathcal{G}(C)$ can be viewed as a simplicial complex in $\mathbb{R}^n$ by filling in each face with a product of continuous intervals (Rubinstein et al., 2009). Each edge $\{u,v\}$ in $\mathcal{G}(C)$ is labeled by the component on which the two vertices u,v differ.

Example 1 *An example concept class in $\{0,1\}^4$ is enumerated in Figure 2(a). The corresponding one-inclusion graph is visualized in Figure 2(b), making immediately apparent the interpretation of*

	x_1	x_2	x_3	x_4
v_0	0	0	0	0
v_1	1	0	0	0
v_2	0	1	0	0
v_3	0	0	1	0
v_4	1	0	1	0
v_5	1	1	0	0
v_6	0	1	1	0
v_7	0	0	0	1
v_8	1	0	0	1
v_9	0	1	0	1
v_{10}	0	0	1	1

(a)



(b)

Figure 2: (a) A concept class in $\{0,1\}^4$ that is maximum with VC-dim 2 and (b) the one-inclusion graph of the concept class.

the object as a simplicial complex: in this case the concepts form vertices which are connected by edges; these edges bound 2-cubes.

Probably Approximately Correct learnability of a concept class $C \subseteq \{0,1\}^X$ is characterized by the finiteness of the Vapnik-Chervonenkis (VC) dimension of C (Blumer et al., 1989). One key to all such results is Sauer's Lemma.

Definition 7 *The* VC dimension *of concept class* $C \subseteq \{0,1\}^X$ *is defined as* $\mathrm{VC}(C) = \sup\left\{n \,\middle|\, \exists Y \in \binom{X}{n}, \Pi_Y(C) = \{0,1\}^n\right\}$ *where* $\Pi_Y(C) = \{(c(x_1),\ldots,c(x_n)) \mid c \in C\} \subseteq \{0,1\}^n$ *is the* projection *of* C *on sequence* $Y = (x_1,\ldots,x_n)$.

Lemma 8 (Vapnik and Chervonenkis, 1971; Sauer, 1972; Shelah, 1972) *The cardinality of any concept classes* $C \subseteq \{0,1\}^n$ *is bounded by* $|C| \leq \sum_{i=1}^{\mathrm{VC}(C)} \binom{n}{i}$.

Motivated by maximizing concept class cardinality under a fixed VC dimension, which is related to constructing general sample compression schemes (see Section 2.4), Welzl (1987) defined the following special classes.

Definition 9 *Concept class* $C \subseteq \{0,1\}^X$ *is called* maximal *if* $\mathrm{VC}(C \cup \{c\}) > \mathrm{VC}(C)$ *for all* $c \in \{0,1\}^X \backslash C$. *Furthermore if* $\Pi_Y(C)$ *satisfies Sauer's Lemma with equality for each* $Y \in \binom{X}{n}$, *for every* $n \in \mathbb{N}$, *then* C *is termed* maximum. *If* $C \subseteq \{0,1\}^n$ *then* C *is maximum (and hence maximal) if* C *meets Sauer's Lemma with equality.*

Example 2 *The concept class of Example 1 has VC-dimension* 2 *as witnessed by projecting onto any two of the four available axes. Moreover its cardinality of* 11 *exactly meets Sauer's Lemma with equality, so the class is also maximum.*

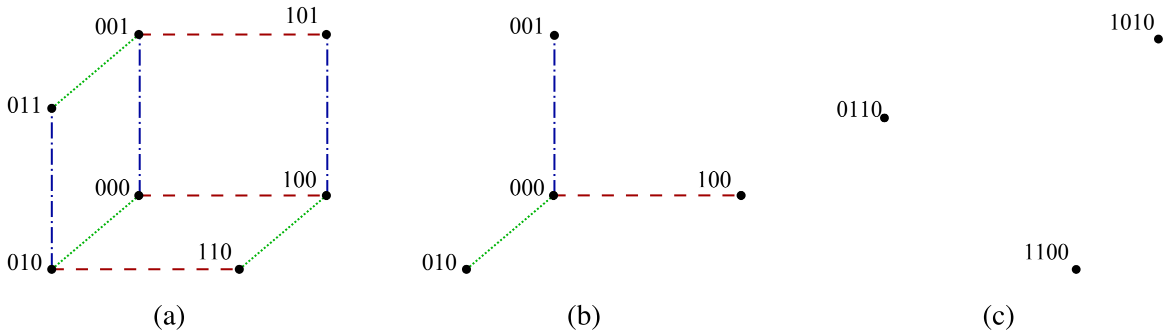


Figure 3: The (a) projection (b) reduction and (c) tail of the concept class of Figure 2 with respect to projecting on to the first three coordinates (i.e., projecting out the fourth coordinate).

The *reduction* of $C \subseteq \{0,1\}^n$ with respect to $i \in [n] = \{1,\ldots,n\}$ is the class $C^i = \Pi_{[n]\setminus\{i\}}\left(\left\{c \in C \mid i \in I_{\mathcal{G}(C)}(c)\right\}\right)$ where $I_{\mathcal{G}(C)}(c) \subseteq [n]$ denotes the labels of the edges incident to vertex c; a *multiple reduction* is the result of performing several reductions in sequence. The *tail* of class C is $\text{tail}_i(C) = \left\{c \in C \mid i \notin I_{\mathcal{G}(C)}(c)\right\}$. Welzl showed that if C is d-maximum, then $\Pi_{[n]\setminus\{i\}}(C)$ and C^i are maximum of VC-dimensions d and $d-1$ respectively.

Example 3 *A projection, reduction and tail of the concept class of Figure 2 are shown in Figures 3(a)—3(c) respectively, when projecting onto coordinates* $\{1,2,3\}$. *In particular note that the reduction, like the projection, is a class in the smaller* 3*-cube while the tail is in the original* 4*-cube. Moreover note that the projection and reduction and maximum with VC-dimensions* 2 *and* 1 *respectively.*

The results presented below relate to other geometric and topological representations of maximum classes existing in the literature. Under the guise of 'forbidden labels', Floyd (1989) showed that maximum $C \subseteq \{0,1\}^n$ of VC-dim d is the union of a maximally overlapping *d-complete collection of cubes* (Rubinstein et al., 2009)—defined as a collection of $\binom{n}{d}$ d-cubes which uniquely project onto all $\binom{n}{d}$ possible sets of d coordinate directions. (An alternative proof was developed by Neylon 2006.) It has long been known that VC-1 maximum classes have one-inclusion graphs that are trees (Dudley, 1985); we previously extended this result by showing that when viewed as complexes, d-maximum classes are contractible d-cubical complexes (Rubinstein et al., 2009). Finally the cells of a simple linear arrangement of n hyperplanes in $\mathbb{R}^d$ form a VC-d maximum class in the n-cube (Edelsbrunner, 1987), but not all finite maximum classes correspond to such Euclidean arrangements (Floyd, 1989).

Example 4 *It is immediately clear from visual inspection that the* 2*-maximum concept classes of Figures 2 and 3(a) are composed of complete collections of* 2*-cubes. Similarly the* 1*-maximum class of Figure 3(c) is a tree with one edge of each color.*

2.4 Sample Compression Schemes

Littlestone and Warmuth (1986) showed that the existence of a compression scheme of finite size is sufficient for learnability of C, and conjectured the converse, that $\text{VC}(C) = d < \infty$ implies a compression scheme of size d. Later Warmuth (2003) weakened the conjectured size to $O(d)$. To-date it is only known that maximum classes can be d-compressed (Floyd, 1989). Unlabeled compression was first explored by Ben-David and Litman (1998); Kuzmin and Warmuth (2007) defined unlabeled compression as follows, and explicitly constructed schemes of size d for maximum classes.

Definition 10 *Let C be a d-maximum class on a finite domain X. A mapping r is called a* representation mapping *of C if it satisfies the following conditions:*

1. *r is a bijection between C and subsets of X of size at most d; and*
2. *[*non-clashing*]:*[2] $\Pi_{r(c)\cup r(c')}(c) \neq \Pi_{r(c)\cup r(c')}(c')$ *for all $c, c' \in C$, $c \neq c'$.*

As with all previously published labeled schemes, all previously known unlabeled compression schemes for maximum classes exploit their special recursive projection-reduction structure and so it is doubtful that such schemes will generalize. Kuzmin and Warmuth (2007, Conjecture 2) conjectured that their *min-peeling* algorithm constitutes an unlabeled d-compression scheme for maximum classes; it iteratively removes minimum degree vertices from $\mathcal{G}(C)$, representing the corresponding concepts by the remaining incident dimensions in the graph. The authors also conjectured that sweeping a hyperplane in general position across a simple linear arrangement forms a compression scheme that corresponds to min peeling the associated maximum class (Kuzmin and Warmuth, 2007, Conjecture 1). A particularly promising approach to compressing general classes is via their maximum-embeddings: a class C embedded in class C' trivially inherits any compression scheme for C', and so an important open problem is to embed maximal classes into maximum classes with at most a linear increase in VC dimension (Kuzmin and Warmuth, 2007, Open Problem 3).

3. Preliminaries

A first step towards characterizing and compressing maximum classes is a process of building them. After describing this process of *lifting* we discuss compressing maximum classes by peeling, and properties of the boundaries of maximum classes.

3.1 Constructing All Maximum Classes

The aim in this section is to describe an algorithm for constructing all maximum classes of VC-dimension d in the n-cube. This process can be viewed as the inverse of mapping a maximum class to its d-maximum projection on $[n]\backslash\{i\}$ and the corresponding $(d-1)$-maximum reduction.

Definition 11 *Let $C, C' \subseteq \{0,1\}^n$ be maximum classes of VC-dimensions $d, d-1$ respectively, so that $C' \subset C$, and let $C_1, C_2 \subset C$ be d-cubes, that is, d-faces of the n-cube $\{0,1\}^n$.*

1. *C_1, C_2 are* connected *if there exists a path in the one-inclusion graph $\mathcal{G}(C)$ with end-points in C_1 and C_2; and*
2. *C_1, C_2 are said to be C'-*connected *if there exists such a connecting path that further does not intersect C'.*

*The C'-*connected components *of C are the equivalence classes of the d-cubes of C under the C'-connectedness relation.*

[2] We abuse notation slightly by applying projections, originally defined to operate on concept classes in Definition 7, to concepts.

Algorithm 1 $\textsc{MaximumClasses}(n,d)$

```
Given: $n \in \mathbb{N}, d \in [n]$
Returns: the set of $d$-maximum classes in $\{0,1\}^n$
1. if $d = 0$ then return $\{\{\mathbf{v}\} \mid \mathbf{v} \in \{0,1\}^n\}$ ;
2. if $d = n$ then return $\{0,1\}^n$ ;
3. $\mathcal{M} \leftarrow \emptyset$ ;
   for each $C \in \textsc{MaximumClasses}(n-1,d)$,
            $C' \in \textsc{MaximumClasses}(n-1,d-1)$ s.t. $C' \subset C$ do
4.    $\{C_1,\ldots,C_k\} \leftarrow C'$-connected components of $C$ ;
5.    $\mathcal{M} \leftarrow \mathcal{M} \cup \bigcup_{\mathbf{p}\in\{0,1\}^k} \left\{(C' \times \{0,1\}) \cup \bigcup_{q\in[k]} C_q \times \{p_q\}\right\}$ ;
   done
6. return $\mathcal{M}$ ;
```

The recursive algorithm for constructing all maximum classes of VC-dimension d in the n-cube, detailed as Algorithm 1, considers each possible d-maximum class C in the $(n-1)$-cube and each possible $(d-1)$-maximum subclass C' of C as the projection and reduction of a d-maximum class in the n-cube, respectively. The algorithm *lifts* C and C' to all possible maximum classes in the n-cube. Then $C' \times \{0,1\}$ is contained in each lifted class; so all that remains is to find the tails from the complement of the reduction in the projection. It turns out that each C'-connected component C_i of C can be lifted to either $C_i \times \{0\}$ or $C_i \times \{1\}$ arbitrarily and independently of how the other C'-connected components are lifted. The set of lifts equates to the set of d-maximum classes in the n-cube that project-reduce to (C,C').

Lemma 12 $\textsc{MaximumClasses}(n,d)$ *(cf. Algorithm 1) returns the set of maximum classes of VC-dimension d in the n-cube for all $n \in \mathbb{N}, d \in [n]$.*

Proof We proceed by induction on n and d. The base cases correspond to $n \in \mathbb{N}, d \in \{0,n\}$ for which all maximum classes, enumerated as singletons in the n-cube and the n-cube itself respectively, are correctly produced by the algorithm. For the inductive step we assume that for $n \in \mathbb{N}, d \in [n-1]$ all maximum classes of VC-dimension d and $d-1$ in the $(n-1)$-cube are already known by recursive calls to the algorithm. Given this, we will show that $\textsc{MaximumClasses}(n,d)$ returns only d-maximum classes in the n-cube, and that all such classes are produced by the algorithm.

Let classes $C \in \textsc{MaximumClasses}(n-1,d)$ and $C' \in \textsc{MaximumClasses}(n-1,d-1)$ be such that $C' \subset C$. Then C is the union of a d-complete collection and C' is the union of a $(d-1)$-complete collection of cubes that are faces of the cubes of C. Consider a concept class $C^\star$ formed from C and C' by Algorithm 1. The algorithm partitions C into C'-connected components $C_1,\ldots,C_k$ each of which is a union of d-cubes. While C' is lifted to $C' \times \{0,1\}$, some subset of the components $\{C_i\}_{i\in S_0}$ are lifted to $\{C_i \times \{0\}\}_{i\in S_0}$ while the remaining components are lifted to $\{C_i \times \{1\}\}_{i\notin S_0}$. Here S_0 ranges over all subsets of $[k]$, selecting which components are lifted to 0; the complement of S_0 specifies those components lifted to 1. By definition $C^\star$ is a d-complete collection of cubes with cardinality equal to $\binom{n}{\le d}$ since $|C^\star| = |C'| + |C|$ (Kuzmin and Warmuth, 2007). So $C^\star$ is d-maximum (Rubinstein et al., 2009, Theorem 34).

If we now consider any d-maximum class $C^\star \subseteq \{0,1\}^n$, its projection on $[n]\backslash\{i\}$ is a d-maximum class $C \subseteq \{0,1\}^{n-1}$ and $C^{\star i}$ is the $(d-1)$-maximum projection $C' \subset C$ of all the d-cubes in $C^\star$

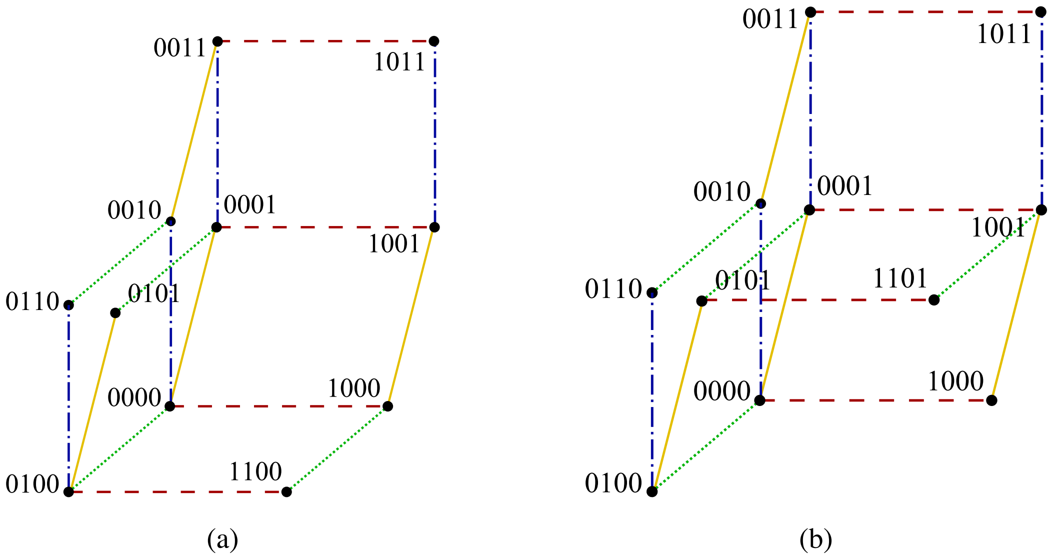


Figure 4: 2-maximum concept classes in $\{0,1\}^4$ constructed by lifting concept class Figure 3(a) as the projection, and concept class Figure 3(b) as the reduction.

which contain color i. It is thus clear that $C^\star$ must be obtained by lifting parts of the C'-connected components of C to the 1 level and the remainder to the 0 level, and C' to $C' \times \{0,1\}$. We will now show that if the vertices of each component are not lifted to the same levels, then while the number of vertices in the lift match that of a d-maximum class in the n-cube, the number of edges are too few for such a maximum class. Define a lifting operator on C as $\ell(v) = \{v\} \times \ell_v$, where $\ell_v \subseteq \{0,1\}$ and

$$|\ell_v| \quad = \quad \begin{cases} 2\ , & \text{if } v \in C' \\ 1, & \text{if } v \in C \backslash C' \end{cases} \quad .$$

Consider now an edge $\{u,v\}$ in $\mathcal{G}(C)$. By the definition of a C'-connected component there exists some C_j such that either $u,v \in C_j \backslash C'$, $u,v \in C'$ or WLOG $u \in C_j \backslash C', v \in C'$. In the first case $\ell(u) \cup \ell(v)$ is an edge in the lifted graph iff $\ell_u = \ell_v$. In the second case $\ell(u) \cup \ell(v)$ contains four edges and in the last it contains a single edge. Furthermore, it is clear that this accounts for all edges in the lifted graph by considering the projection of an edge in the lifted product. Thus any lift other than those produced by Algorithm 1 induces strictly too few edges for a d-maximum class in the n-cube (cf. Kuzmin and Warmuth, 2007, Corollary 7.5). ■

Example 5 *Let C and C' refer to the 2- and 1-maximum concept classes in Figures 3(a) and 3(b) respectively. Then Figures 4(a), 4(b) and 2 make up all possible 2-maximum classes (up to symmetry) resulting from lifting projection C and reduction C'. Figure 2 corresponds to lifting no C'-connected components of C; Figure 4(a) corresponds to lifting just one component; and Figure 4(b) corresponds to lifting two components. (Note that Figure 4(a) and Figure 4(b) are actually equivalent after a symmetry.)*

3.2 Corner Peeling

Kuzmin and Warmuth (2007, Conjecture 2) conjectured that their simple *min-peeling* procedure is a valid unlabeled compression scheme for maximum classes. Beginning with a concept class $C_0 = C \subseteq \{0,1\}^n$, min peeling operates by iteratively removing a vertex v_t of minimum-degree in $\mathcal{G}(C_t)$ to produce the peeled class $C_{t+1} = C_t \backslash \{v_t\}$. The concept class corresponding to v_t is then represented by the dimensions of the edges incident to v_t in $\mathcal{G}(C_t)$, $I_{\mathcal{G}(C_t)}(v_t) \subseteq [n]$. Providing that no-clashing holds for the algorithm, the size of the min-peeling scheme is the largest degree encountered during peeling. Kuzmin and Warmuth predicted that this size is always at most d for d-maximum classes. We explore these questions for a related special case of peeling, where we prescribe which vertex to peel at step t as follows.

Definition 13 *We say that $C \subseteq \{0,1\}^n$ can be* corner peeled *if there exists an ordering $v_1, \ldots, v_{|C|}$ of the vertices of C such that, for each $t \in [|C|]$ where $C_0 = C$,*

1. *$v_t \in C_{t-1}$ and $C_t = C_{t-1} \backslash \{v_t\}$;*
2. *There exists a unique cube C'_{t-1} of maximum dimension over all cubes in C_{t-1} containing v_t;*
3. *The neighbors $\Gamma(v_t)$ of v_t in $\mathcal{G}(C_{t-1})$ satisfy $\Gamma(v_t) \subseteq C'_{t-1}$; and*
4. *$C_{|C|} = \emptyset$.*

The v_t are termed the corner vertices *of C_{t-1} respectively. If d is the maximum degree of each v_t in $\mathcal{G}(C_{t-1})$, then C is d* corner peeled.

Note that we do not constrain the cubes C'_t to be of non-increasing dimension. It turns out that an important property of maximum classes is invariant to this kind of peeling.

Definition 14 *We call a class $C \subseteq \{0,1\}^n$* shortest-path closed *if for any $u, v \in C$, $\mathcal{G}(C)$ contains a path connecting u, v of length $\|u - v\|_1$.*

Lemma 15 *If $C \subseteq \{0,1\}^n$ is shortest-path closed and $v \in C$ is a corner vertex of C, then $C \backslash \{v\}$ is shortest-path closed.*

Proof Consider a shortest-path closed $C \subseteq \{0,1\}^n$. Let c be a corner vertex of C, and denote the cube of maximum dimension in C, containing c, by C'. Consider $\{u, v\} \subseteq C \backslash \{c\}$. By assumption there exists a u-v-path p of length $\|u - v\|_1$ contained in C. If c is not in p then p is contained in the peeled product $C \backslash \{c\}$. If c is in p then p must cross C' such that there is another path of the same length which avoids c, and thus $C \backslash \{c\}$ is shortest-path closed. ■

3.2.1 CORNER PEELING IMPLIES COMPRESSION

Theorem 16 *If a maximum class C can be corner peeled then C can be d-unlabeled compressed.*

Proof The invariance of the shortest-path closed property under corner peeling is key. The corner-peeling unlabeled compression scheme represents each $v_t \in C$ by $r(v_t) = I_{\mathcal{G}(C_{t-1})}(v_t)$, the colors of

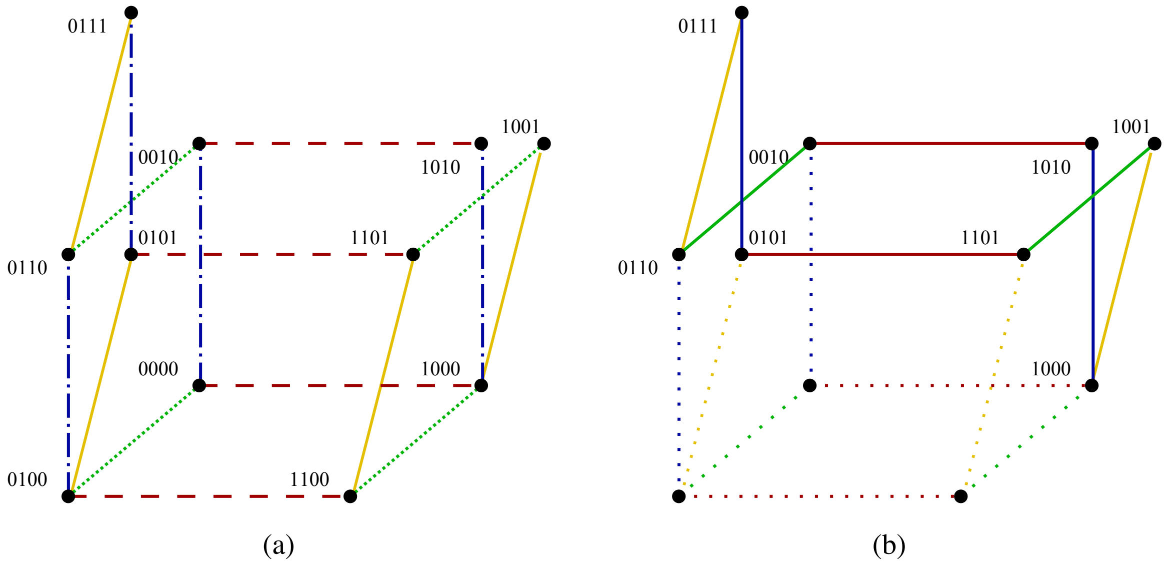


Figure 5: (a) A 2-maximum class in the 4-cube and (b) its boundary highlighted by solid lines.

the cube C'_{t-1} which is deleted from C_{t-1} when v_t is corner peeled. We claim that any two vertices $v_s, v_t \in C$ have non-clashing representatives. WLOG, suppose that $s < t$. The class C_{s-1} must contain a shortest v_s-v_t-path p. Let i be the color of the single edge contained in p that is incident to v_s. Color i appears once in p, and is contained in $r(v_s)$. This implies that $v_{s,i} \neq v_{t,i}$ and that $i \in r(v_s) \cup r(v_t)$, and so $v_s|(r(v_s) \cup r(v_t)) \neq v_t|(r(v_s) \cup r(v_t))$. By construction, $r(\cdot)$ is a bijection between C and all subsets of $[n]$ of cardinality $\leq \mathrm{VC}(C)$. ■

If the oriented one-inclusion graph, with each edge directed away from the incident vertex represented by the edge's color, has no cycles, then that representation's compression scheme is termed *acyclic* (Floyd, 1989; Ben-David and Litman, 1998; Kuzmin and Warmuth, 2007).

Proposition 17 *All corner-peeling unlabeled compression schemes are acyclic.*

Proof We follow the proof that the min-peeling algorithm is acyclic (Kuzmin and Warmuth, 2007). Let $v_1, \ldots, v_{|C|}$ be a corner vertex ordering of C. As a corner vertex v_t is peeled, its unoriented incident edges are oriented away from v_t. Thus all edges incident to v_1 are oriented away from v_1 and so the vertex cannot take part in any cycle. For $t > 1$ assume $V_t = \{v_s \mid s < t\}$ is disjoint from all cycles. Then v_t cannot be contained in a cycle, as all incoming edges into v_t are incident to some vertex in V_t. Thus the oriented $\mathcal{G}(C)$ is indeed acyclic. ■

3.3 Boundaries of Maximum Classes

We now turn to the geometric boundaries of maximum classes, which are closely related to corner peeling.

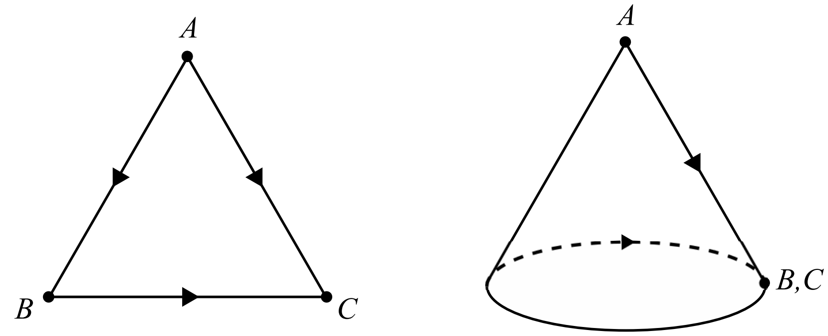


Figure 6: The first steps of building the dunce hat in Example 7.

Definition 18 *The boundary ∂C of a d-maximum class C is defined as all the $(d-1)$-subcubes which are the faces of a single d-cube in C.*

Maximum classes, when viewed as cubical complexes, are analogous to soap films (an example of a minimal energy surface encountered in nature), which are obtained when a wire frame is dipped into a soap solution. Under this analogy the boundary corresponds to the wire frame and the number of d-cubes can be considered the area of the soap film. An important property of the boundary of a maximum class is that all lifted reductions meet the boundary multiple times.

Theorem 19 *Every d-maximum class has boundary containing at least two $(d-1)$-cubes of every combination of $d-1$ colors, for all $d > 1$.*

Proof We use the lifting construction of Section 3.1. Let $C^\star \subseteq \{0,1\}^n$ be a 2-maximum class and consider color $i \in [n]$. Then the reduction $C^{\star i}$ is an unrooted tree with at least two leaves, each of which lifts to an i-colored edge in $C^\star$. Since the leaves are of degree 1 in $C^{\star i}$, the corresponding lifted edges belong to exactly one 2-cube in $C^\star$ and so lie in $\partial C^\star$. Consider now a d-maximum class $C^\star \subseteq \{0,1\}^n$ for $d > 2$, and make the inductive assumption that the projection $C = \Pi_{[n-1]}(C^\star)$ has two of each type of $(d-1)$-cube, and that the reduction $C' = C^{\star n}$ has two of each type of $(d-2)$-cube, in their boundaries. Pick $d-1$ colors $I \subseteq [n]$. If $n \in I$ then consider two $(d-2)$-cubes colored by $I \backslash \{x_n\}$ in $\partial C'$. By the same argument as in the base case, these lift to two I-colored cubes in $\partial C^\star$. If $n \notin I$ then ∂C contains two I-colored $(d-1)$-cubes. For each cube, if the cube is contained in C' then it has two lifts one of which is contained in $\partial C^\star$, otherwise its unique lift is contained in $\partial C^\star$. Therefore $\partial C^\star$ contains at least two I-colored cubes. ■

Example 6 *The one-inclusion graph of a 2-maximum concept class in the 4-cube is depicted in Figure 5(a), along with its boundary of edges in Figure 5(b). Note that all four colors are represented by exactly two boundary edges in this case.*

Having a large boundary is an important property of maximum classes that does not follow from contractibility.

Example 7 *Take a* 2*-simplex with vertices* A,B,C*. Glue the edges* AB *to* AC *to form a cone. Next glue the end loop* BC *to the edge* AB *. The result is a complex* D *with a single vertex, edge and* 2*-simplex, which is classically known as the* dunce hat *(cf. Figure 6). It is not hard to verify that* D *is contractible, but has no (geometric) boundary.*

Although Theorem 19 will not be explicitly used in the sequel, we return to boundaries of maximum complexes later.

4. Euclidean Arrangements

Definition 20 *A* linear arrangement *is a collection of* $n \geq d$ *oriented hyperplanes in* $\mathbb{R}^d$*. Each region or cell in the complement of the arrangement is naturally associated with a concept in* $\{0,1\}^n$*; the side of the* i^{th} *hyperplane on which a cell falls determines the concept's* i^{th} *component. A* simple arrangement *is a linear arrangement in which any subset of* d *planes has a unique point in common and all subsets of* $d+1$ *planes have an empty mutual intersection. Moreover any subset of* $k < d$ *planes meet in a plane of dimension* $d-k$*. Such a collection of* n *planes is also said to be in* general position.

Many of the familiar operations on concept classes in the n-cube have elegant analogues on arrangements.

- Projection on $[n]\backslash\{i\}$ corresponds to removing the i^{th} plane;
- The reduction C^i is the new arrangement given by the intersection of C's arrangement with the i^{th} plane; and
- The corresponding lifted reduction is the collection of cells in the arrangement that adjoin the i^{th} plane.

A k-cube in the one-inclusion graph corresponds to a collection of 2^k cells, all having a common $(d-k)$-face, which is contained in the intersection of k planes, and an edge corresponds to a pair of cells which have a common face on a single plane. The following result is due to Edelsbrunner (1987).

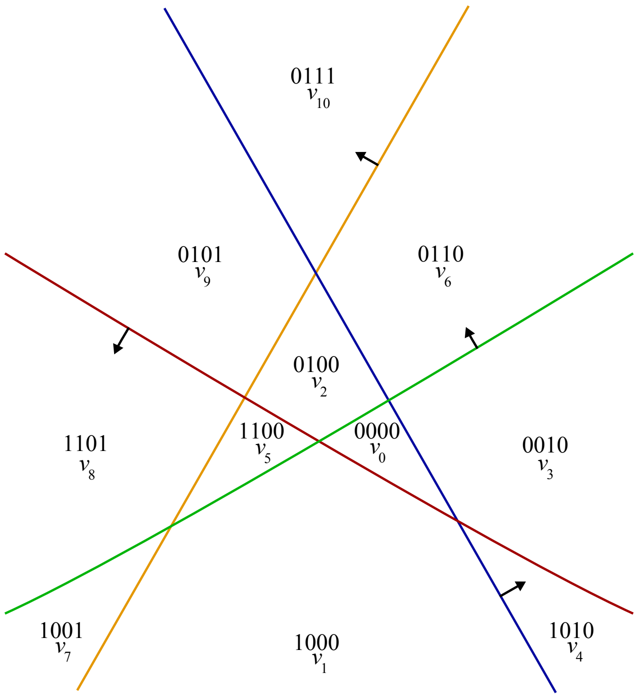
Lemma 21 *The concept class* $C \subseteq \{0,1\}^n$ *induced by a simple linear arrangement of* n *planes in* $\mathbb{R}^d$ *is* d*-maximum.*

Proof Note that C has VC dimension at most d, since general position is invariant to projection, that is, no $d+1$ planes are shattered. Since C is the union of a d-complete collection of cubes (every cell contains d-intersection points in its boundary) it follows that C is d-maximum (Rubinstein et al., 2009). ■

Example 8 *Consider the simple linear arrangement in* $\mathbb{R}^2$ *shown in Figure 7(b). The given labeling of its cells map to the concept class in the* 4*-cube enumerated in Figure 7(a) with one-inclusion graph shown in Figure 5(a). This class is maximum with VC-dimension* 2*.*

	x_1	x_2	x_3	x_4
v_0	0	0	0	0
v_1	1	0	0	0
v_2	0	1	0	0
v_3	0	0	1	0
v_4	1	0	1	0
v_5	1	1	0	0
v_6	0	1	1	0
v_7	1	0	0	1
v_8	1	1	0	1
v_9	0	1	0	1
v_{10}	0	1	1	1

(a)



(b)

Figure 7: (a) The enumeration of the 2-maximum class in $\{0,1\}^4$ in Figure 5(a) and (b) a simple linear line arrangement corresponding to the class, with each cell corresponding to a unique vertex.

Corollary 22 *Let A be a simple linear arrangement of n hyperplanes in $\mathbb{R}^d$ with corresponding d-maximum $C \subseteq \{0,1\}^n$. The intersection of A with a generic hyperplane corresponds to a $(d-1)$-maximum class $C' \subseteq C$. In particular if all d-intersection points of A lie to one side of the generic hyperplane, then C' lies on the boundary of C; and ∂C is the disjoint union of two $(d-1)$-maximum sub-classes.*

Proof The intersection of A with a generic hyperplane is again a simple arrangement of n hyperplanes but now in $\mathbb{R}^{d-1}$. Hence by Lemma 21 C' is a $(d-1)$-maximum class in the n-cube. $C' \subseteq C$ since the adjacency relationships on the cells of the intersection are inherited from those of A.

Suppose that all d-intersections in A lie in one half-space of the generic hyperplane. C' is the union of a $(d-1)$-complete collection. We claim that each of these $(d-1)$-cubes is a face of exactly one d-cube in C and is thus in ∂C. A $(d-1)$-cube in C' corresponds to a line in A where $d-1$ planes mutually intersect. The $(d-1)$-cube is a face of a d-cube in C iff this line is further intersected by a d^{th} plane. This occurs for exactly one plane, which is closest to the generic hyperplane along this intersection line. For once the d-intersection point is reached, when following along the line away from the generic plane, a new cell is entered. This verifies the second part of the result.

Consider two parallel generic hyperplanes h_1, h_2 such that all d-intersection points of A lie in between them. We claim that each $(d-1)$-cube in ∂C is in exactly one of the concept classes induced by the intersection of A with h_1 and A with h_2. Consider an arbitrary $(d-1)$-cube in ∂C. As before this cube corresponds to a region of a line formed by a mutual intersection of $d-1$ planes. Moreover this region is a ray, with one end-point at a d-intersection. Because the ray begins at a point between the generic hyperplanes h_1, h_2, it follows that the ray must cross exactly one of these. ■

Example 9 *To illustrate, consider the* 2*-maximum class in Figure 5(a) that corresponds to the simple linear arrangement in Figure 7(b). The boundary, shown in Figure 5(b) is clearly a disjoint union of two* 1*-maximum classes—in this case sticks.*

Corollary 23 *Let A be a simple linear arrangement of n hyperplanes in $\mathbb{R}^d$ and let $C \subseteq \{0,1\}^n$ be the corresponding d-maximum class. Then C considered as a cubical complex is homeomorphic to the d-ball B^d; and ∂C considered as a $(d-1)$-cubical complex is homeomorphic to the $(d-1)$-sphere S^{d-1}.*

Proof We construct a Voronoi cell decomposition corresponding to the set of d-intersection points inside a very large ball in Euclidean space. By induction on d, we claim that this is a cubical complex and the vertices and edges correspond to the class C. By induction, on each hyperplane, the induced arrangement has a Voronoi cell decomposition which is a $(d-1)$-cubical complex with edges and vertices matching the one-inclusion graph for the tail of C corresponding to the label associated with the hyperplane. It is not hard to see that the Voronoi cell defined by a d-intersection point p on this hyperplane is a d-cube. In fact, its $(d-1)$-faces correspond to the Voronoi cells for p, on each of the d hyperplanes passing through p. We also see that this d-cube has a single vertex in the interior of each of the 2^d cells of the arrangement adjacent to p. In this way, it follows that the vertices of this Voronoi cell decomposition are in bijective correspondence to the cells of the hyperplane arrangement. Finally the edges of the Voronoi cells pass through the faces in the hyperplanes. So these correspond bijectively to the edges of C, as there is one edge for each face of the hyperplanes. Using a very large ball, containing all the d-intersection points, the boundary faces become spherical cells. In fact, these form a spherical Voronoi cell decomposition, so it is easy to replace these by linear ones by taking the convex hull of their vertices. So a piecewise linear cubical complex $\mathbf{C}$ is constructed, which has one-skeleton (graph consisting of all vertices and edges) isomorphic to the one-inclusion graph for C.

Finally we want to prove that $\mathbf{C}$ is homeomorphic to B^d. This is quite easy by construction. For we see that $\mathbf{C}$ is obtained by dividing up B^d into Voronoi cells and replacing the spherical boundary cells by linear ones, using convex hulls of the boundary vertices. This process is clearly given by a homeomorphism by projection. In fact, the homeomorphism preserves the PL-structure so is a PL homeomorphism. ■

Example 10 *Consider again the one-inclusion graph in Figure 5(a) corresponding to a* 2*-maximum concept class in the* 4*-cube. It is trivial to see via inspection that this class, when viewed as a simplicial complex, is homeomorphic to a disc; similarly its boundary, highlighted in Figure 5(b), is homeomorphic to a circle.*

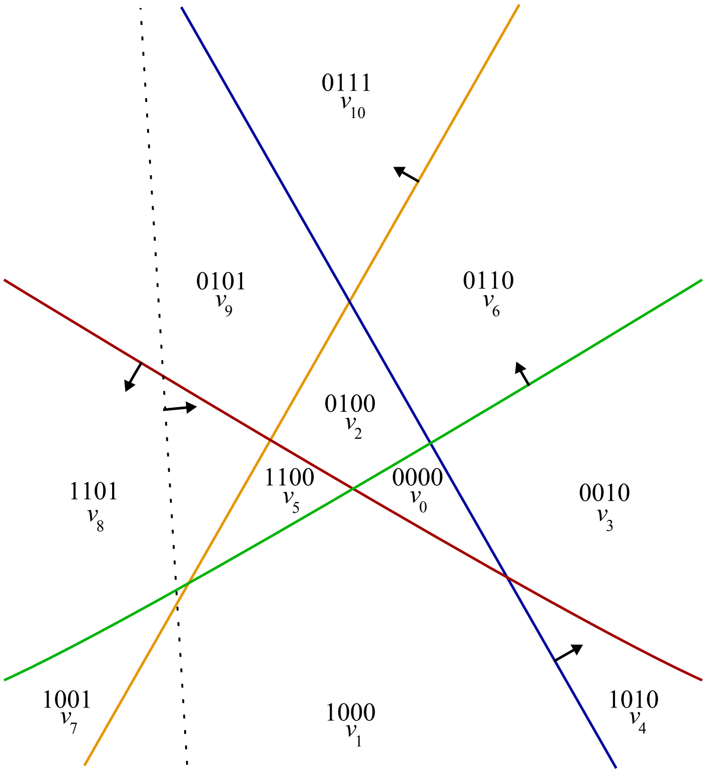


Figure 8: The simple linear line arrangement from Figure 7(b) corresponding to the concept class enumerated in Figure 7(a) and visualized in Figure 5(a). The arrangement is in the process of being swept by the dashed line.

The following example demonstrates that not all maximum classes of VC-dimension d are homeomorphic to the d-ball. The key to such examples is branching.

Example 11 *A simple linear arrangement in $\mathbb{R}$ corresponds to points on the line—cells are simply intervals between these points and so corresponding 1-maximum classes are sticks. Any tree that is not a stick can therefore not be represented as a simple linear arrangement in $\mathbb{R}$ and is also not homeomorphic to the 1-ball which is simply the interval $[-1,1]$.*

As Kuzmin and Warmuth (2007) did previously, consider a generic hyperplane h sweeping across a simple linear arrangement A. h begins with all d-intersection points of A lying in its positive half-space $\mathcal{H}_+$. The concept corresponding to cell c is peeled from C when $|\mathcal{H}_+ \cap c| = 1$, that is, h crosses the last d-intersection point adjoining c. At any step in the process, the result of peeling j vertices from C to reach C_j, is captured by the arrangement $\mathcal{H}_+ \cap A$ for the appropriate h.

Example 12 *Figure 7(a) enumerates the 11 vertices of a 2-maximum class in the 4-cube. Figures 8 and 5(a) display a hyperplane arrangement in Euclidean space and its Voronoi cell decomposition, corresponding to this maximum class. In this case, sweeping the vertical dashed line across the arrangement corresponds to a partial corner peeling of the concept class with peeling sequence v_7, v_8, v_5, v_9, v_2, v_0. What remains is the 1-maximum stick $\{v_1, v_3, v_4, v_6, v_{10}\}$.*

Next we resolve the first half of Kuzmin and Warmuth (2007, Conjecture 1).

Theorem 24 *Any d-maximum class $C \subseteq \{0,1\}^n$ corresponding to a simple linear arrangement A can be corner peeled by sweeping A, and this process is a valid unlabeled compression scheme for C of size d.*

Proof We must show that as the j^{th} d-intersection point p_j is crossed, there is a corner vertex of C_{j-1} peeled away. It then follows that sweeping a generic hyperplane h across A corresponds to corner peeling C to a $(d-1)$-maximum sub-class $C' \subseteq \partial C$ by Corollary 22. Moreover C' corresponds to a simple linear arrangement of n hyperplanes in $\mathbb{R}^{d-1}$.

We proceed by induction on d, noting that for $d = 1$ corner peeling is trivial. Consider h as it approaches the j^{th} d-intersection point p_j. The d planes defining this point intersect h in a simple arrangement of hyperplanes on h. There is a compact cell Δ for the arrangement on h, which is a d-simplex[3] and shrinks to a point as h passes through p_j. We claim that the cell c for the arrangement A, whose intersection with h is Δ, is a corner vertex v_j of C_{j-1}. Consider the lines formed by intersections of $d-1$ of the d hyperplanes, passing through p_j. Each is a segment starting at p_j and ending at h without passing through any other d-intersection points. So all faces of hyperplanes adjacent to c meet h in faces of Δ. Thus, there are no edges in C_{j-1} starting at the vertex corresponding to p_j, except for those in the cube C'_{j-1}, which consists of all cells adjacent to p_j in the arrangement A. So c corresponds to a corner vertex v_j of the d-cube C'_{j-1} in C_{j-1}. Finally, just after the simplex is a point, c is no longer in $\mathcal{H}_+$ and so v_j is corner peeled from C_{j-1}.

Theorem 16 completes the proof that this corner peeling of C constitutes unlabeled compression. ■

Corollary 25 *The sequence of cubes $C'_0, \ldots, C'_{|C|}$, removed when corner peeling by sweeping simple linear arrangements, is of non-increasing dimension. In fact, there are $\binom{n}{d}$ cubes of dimension d, then $\binom{n}{d-1}$ cubes of dimension $d-1$, etc.*

While corner peeling and min peeling share some properties in common, they are distinct procedures. Notice that sweeping produces a monotonic corner-peeling sequence, as cubes are removed in order of non-increasing dimensions.

Example 13 *Consider sweeping a simple linear arrangement corresponding to a 2-maximum class. After all but one 2-intersection point has been swept, the corresponding corner-peeled class C_t is the union of a single 2-cube with a 1-maximum stick. Min peeling applied to C_t would first peel a leaf, while sweeping must peel the 2-cube next.*

A second example is the class in a 3-cube which consists of six vertices, so that two opposite vertices, for example, 000 and 111 are not included. This class cannot be corner peeled as the one-inclusion graph consists of six edges forming a single cycle. On the other hand, it has many min-peeling schemes.

An interesting question is if a class has a corner-peeling scheme, does it always have a min-peeling scheme which is also a corner-peeling scheme? This is given as Question 50 below.

[3] Δ is a topological simplex—the convex hull of $d+1$ affinely independent points in $\mathbb{R}^d$.

The next result follows from our counter-examples to Kuzmin & Warmuth's minimum degree conjecture (Rubinstein et al., 2009).

Corollary 26 *There is no constant c so that all maximal classes of VC-dimension d can be embedded into maximum classes corresponding to simple hyperplane arrangements of dimension $d+c$.*

5. Hyperbolic Arrangements

To motivate the introduction of hyperbolic arrangements, note that linear-hyperplane arrangements can be efficiently described, since each hyperplane is determined by its unit normal and distance from the origin. Similarly, a hyperbolic hyperplane is a hypersphere. So it can be parametrized by its center—a point on the ideal sphere at infinity—and its radius.[4]

However the family of hyperbolic hyperplanes has more flexibility than linear hyperplanes since there are many disjoint hyperbolic hyperplanes, whereas in the linear case only parallel hyperplanes do not meet. Thus we turn to hyperbolic arrangements to represent a larger collection of concept classes than those represented by simple linear arrangements.

We briefly discuss the Klein model of hyperbolic geometry (Ratcliffe, 1994, pg. 7). Consider the open unit ball $\mathbb{H}^k$ in $\mathbb{R}^k$. Geodesics (lines of shortest length in the geometry) are given by intersections of straight lines in $\mathbb{R}^k$ with the unit ball. Similarly planes of any dimension between 2 and $k-1$ are given by intersections of such planes in $\mathbb{R}^k$ with the unit ball. Note that such planes are completely determined by their spheres of intersection with the unit sphere S^{k-1}, which is called the ideal boundary of hyperbolic space $\mathbb{H}^k$. Note that in the appropriate metric, the ideal boundary consists of points which are infinitely far from all points in the interior of the unit ball.

We can now see immediately that a simple hyperplane arrangement in $\mathbb{H}^k$ can be described by taking a simple hyperplane arrangement in $\mathbb{R}^k$ and intersecting it with the unit ball. However we require an important additional property to mimic the Euclidean case. Namely we add the constraint that every subcollection of d of the hyperplanes in $\mathbb{H}^k$ has mutual intersection points inside $\mathbb{H}^k$, and that no $(d+1)$-intersection point lies in $\mathbb{H}^k$. We need this requirement to obtain that the resulting class is maximum.

Definition 27 A simple hyperbolic d-arrangement *is a collection of n hyperplanes in $\mathbb{H}^k$ with the property that every sub-collection of d hyperplanes mutually intersect in a $(k-d)$-dimensional hyperbolic plane, and that every sub-collection of $d+1$ hyperplanes mutually intersect as the empty set.*

Corollary 28 *The concept class C corresponding to a simple d-arrangement of hyperbolic hyperplanes in $\mathbb{H}^k$ is d-maximum in the k-cube.*

Proof The result follows by the same argument as before. Projection cannot shatter any $(d+1)$-cube and the class is a complete union of d-cubes, so is d-maximum. ■

The key to why hyperbolic arrangements represent many new maximum classes is that they allow flexibility of choosing d and k independently. This is significant because the unit ball can be

[4] Note also that hyperbolic hyperplanes are 'linear' in the sense that they are filled by a family of geodesics, which are shortest paths or lines in the hyperbolic metric.

chosen to miss much of the intersections of the hyperplanes in Euclidean space. Note that the new maximum classes are embedded in maximum classes induced by arrangements of linear hyperplanes in Euclidean space.

A simple example is any 1-maximum class. It is easy to see that this can be realized in the hyperbolic plane by choosing an appropriate family of lines and the unit ball in the appropriate position. In fact, we can choose sets of pairs of points on the unit circle, which will be the intersections with our lines. So long as these pairs of points have the property that the smaller arcs of the circle between them are disjoint, the lines will not cross inside the disk and the desired 1-maximum class will be represented.

Corner-peeling maximum classes represented by hyperbolic-hyperplane arrangements proceeds by sweeping, just as in the Euclidean case. Note first that intersections of the hyperplanes of the arrangement with the moving hyperplane appear precisely when there is a first intersection at the ideal boundary. Thus it is necessary to slightly perturb the collection of hyperplanes to ensure that only one new intersection with the moving hyperplane occurs at any time. Note also that new intersections of the sweeping hyperplane with the various lower dimensional planes of intersection between the hyperplanes appear similarly at the ideal boundary. The important claim to check is that the intersection at the ideal boundary between the moving hyperplane and a lower dimensional plane, consisting entirely of d intersection points, corresponds to a corner-peeling move. We include two examples to illustrate the validity of this claim.

Example 14 *In the case of a* 1*-maximum class coming from disjoint lines in* $\mathbb{H}^2$*, a cell can disappear when the sweeping hyperplane meets a line at an ideal point. This cell is indeed a vertex of the tree, that is, a corner-vertex.*

Example 15 *Assume that we have a family of* 2*-planes in the unit* 3*-ball which meet in pairs in single lines, but there are no triple points of intersection, corresponding to a* 2*-maximum class. A corner-peeling move occurs when a region bounded by two half disks and an interval disappears, in the positive half space bounded by the sweeping hyperplane. Such a region can be visualized by taking a slice out of an orange. Note that the final point of contact between the hyperplane and the region is at the end of a line of intersection between two planes on the ideal boundary.*

We next observe that sweeping by generic hyperbolic hyperplanes induces corner peeling of the corresponding maximum class, extending Theorem 24. As the generic hyperplane sweeps across hyperbolic space, not only do swept cells correspond to corners of d-cubes but also to corners of lower dimensional cubes as well. Moreover, the order of the dimensions of the cubes which are corner peeled can be arbitrary—lower dimensional cubes may be corner peeled before all the higher dimensional cubes are corner peeled. This is in contrast to Euclidean sweepouts (cf. Corollary 25). Similar to Euclidean sweepouts, hyperbolic sweepouts correspond to corner peeling and not min peeling.

Theorem 29 *Any d-maximum class $C \subseteq \{0,1\}^n$ corresponding to a simple hyperbolic d-arrangement A can be corner peeled by sweeping A with a generic hyperbolic hyperplane.*

Proof We follow the same strategy of the proof of Theorem 24. For sweeping in hyperbolic space $\mathbb{H}^k$, the generic hyperplane h is initialized as tangent to $\mathbb{H}^k$. As h is swept across $\mathbb{H}^k$, new intersections appear with A just after h meets the non-empty intersection of a subset of hyperplanes of A with

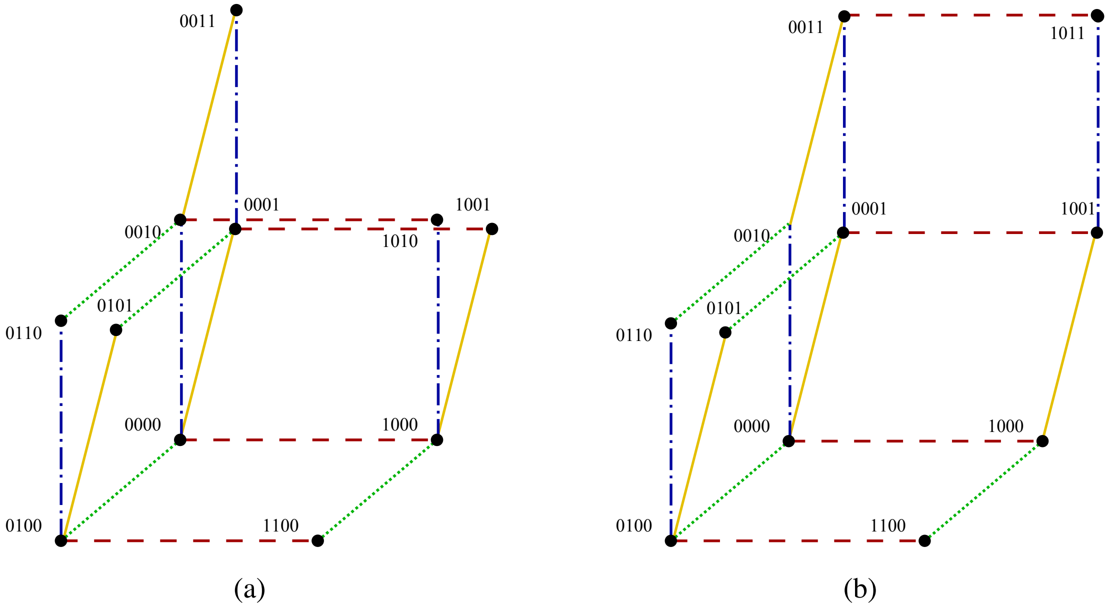


Figure 9: 2-maximum classes in $\{0,1\}^4$ that can be represented as hyperbolic arrangements but not as Euclidean arrangements.

the ideal boundary. Each d-cube C' in C still corresponds to the cells adjacent to the intersection $I_{C'}$ of d hyperplanes. But now $I_{C'}$ is a $(k-d)$-dimensional hyperbolic hyperplane. A cell c adjacent to $I_{C'}$ is corner peeled precisely when h last intersects c at a point of $I_{C'}$ at the ideal boundary. As for simple linear arrangements, the general position of $A \cup \{h\}$ ensures that corner-peeling events never occur simultaneously. For the case $k = d+1$, as for the simple linear arrangements just prior to the corner peeling of c, $\mathcal{H}_+ \cap c$ is homeomorphic to a $(d+1)$-simplex with a missing face on the ideal boundary. And so as in the simple linear case, this d-intersection point corresponds to a corner d-cube. In the case $k > d+1$, $\mathcal{H}_+ \cap c$ becomes a $(d+1)$-simplex (as before) multiplied by $\mathbb{R}^{k-d-1}$. If $k = d$, then the main difference is just before corner peeling of c, $\mathcal{H}_+ \cap c$ is homeomorphic to a k-simplex which may be either closed (hence in the interior of $\mathbb{H}^k$) or with a missing face on the ideal boundary. The rest of the argument remains the same, except for one important observation.

Although swept corners in hyperbolic arrangements can be of cubes of differing dimensions, these dimensions never exceed d and so the proof that sweeping simple linear arrangements induces d-compression schemes is still valid. ■

Example 16 *Constructed with lifting, Figure 9 completes the enumeration, up to symmetry, of the 2-maximum classes in $\{0,1\}^4$ begun with Example 12. These cases cannot be represented as simple Euclidean linear arrangements, since their boundaries do not satisfy the condition of Corollary 23 but can be represented as hyperbolic arrangements as in Figure 10. Figures 11(a) and 11(b) display the sweeping of a general hyperplane across the former arrangement and the corresponding corner peeling. Notice that the corner-peeled cubes' dimensions decrease and then increase.*

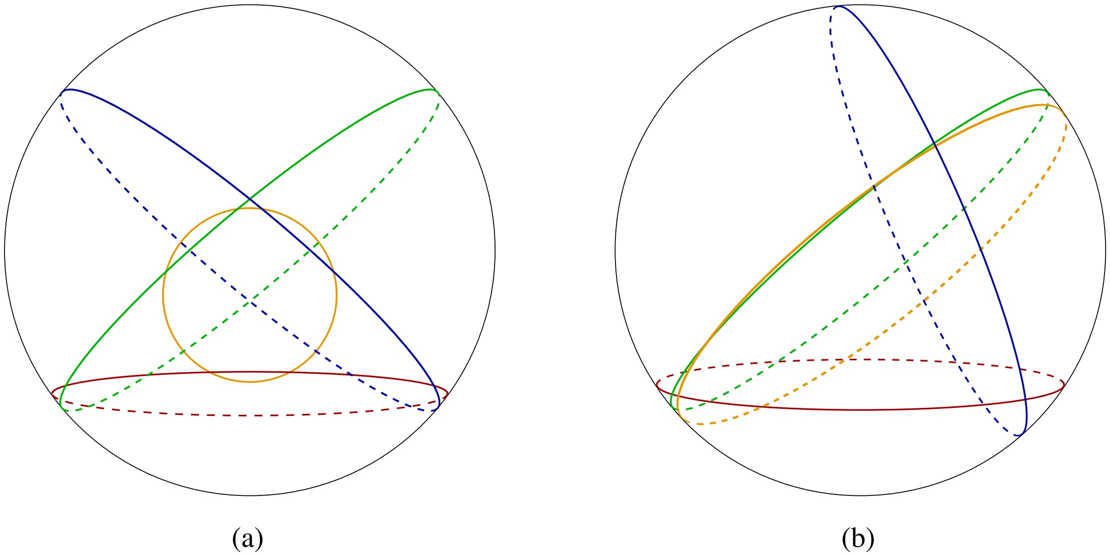


Figure 10: Hyperbolic-hyperplane arrangements corresponding to the classes in Figure 9. In both cases the four hyperbolic planes meet in 6 straight line segments (not shown). The planes' colors correspond to the edges' colors in Figure 9.

Corollary 30 *There is no constant c so that all maximal classes of VC-dimension d can be embedded into maximum classes corresponding to simple hyperbolic-hyperplane arrangements of VC-dimension $d+c$.*

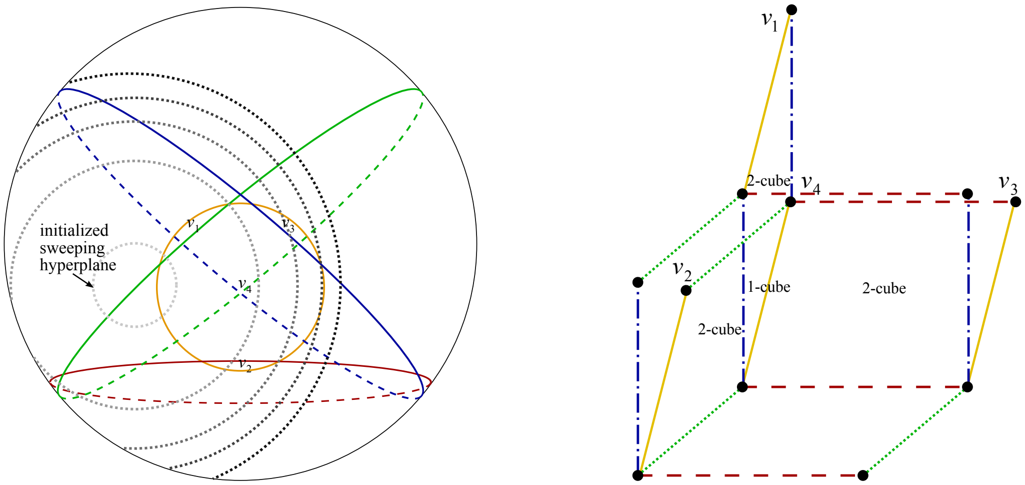
This result follows from our counter-examples to Kuzmin & Warmuth's minimum degree conjecture (Rubinstein et al., 2009).

Corollary 28 gives a proper superset of simple linear-hyperplane arrangement-induced maximum classes as hyperbolic arrangements. We will prove in Section 7 that all maximum classes can be represented as PL-hyperplane arrangements in a ball. These are the topological analogue of hyperbolic-hyperplane arrangements. If the boundary of the ball is removed, then we obtain an arrangement of PL hyperplanes in Euclidean space.

6. Infinite Euclidean and Hyperbolic Arrangements

We consider a simple example of an infinite maximum class which admits corner peeling and a compression scheme analogous to those of previous sections.

Example 17 *Let $\mathcal{L}$ be the set of lines in the plane of the form $L_{2m} = \{(x,y) \mid x = m\}$ and $L_{2n+1} = \{(x,y) \mid y = n\}$ for $m,n \in \mathbb{N}$. Let v_{00}, v_{0n}, v_{m0}, and v_{mn} be the cells bounded by the lines $\{L_2, L_3\}$, $\{L_2, L_{2n+1}, L_{2n+3}\}$, $\{L_{2m}, L_{2m+2}, L_3\}$, and $\{L_{2m}, L_{2m+2}, L_{2n+1}, L_{2n+3}\}$, respectively. Then the cubical complex C, with vertices v_{mn}, can be corner peeled and hence compressed, using a sweepout by the lines $\{(x,y) \mid x+(1+\varepsilon)y = t\}$ for $t \geq 0$ and any small fixed irrational $\varepsilon > 0$. C is a 2-maximum class and the unlabeled compression scheme is also of size 2.*



(a)

(b)

Figure 11: (a) The simple hyperbolic arrangement corresponding to the 2-maximum class in $\{0,1\}^4$ of Figure 9(a)—as shown in Figure 10(a)—with a generic sweeping hyperplane shown in several positions before and after it sweeps past four cells; and (b) the class with the first four corner-vertices peeled by the hyperbolic arrangement sweeping. Notice that three 2-cubes are peeled, then a 1-cube (all shown) followed by 2-cubes.

To verify the properties of this example, notice that sweeping as specified corresponds to corner peeling the vertex v_{00}, then the vertices v_{10}, v_{01}, then the remaining vertices v_{mn}. The lines $x+(1+\varepsilon)y=t$ are generic as they pass through only one intersection point of $\mathcal{L}$ at a time. Additionally, representing v_{00} by $\emptyset$, v_{0n} by $\{L_{2n+1}\}$, v_{m0} by $\{L_{2m}\}$ and v_{mn} by $\{L_{2m}, L_{2n+1}\}$ constitutes a valid unlabeled compression scheme. Note that the compression scheme is associated with sweeping across the arrangement in the direction of decreasing t. This is necessary to pick up the boundary vertices of C last in the sweepout process, so that they have either singleton representatives or the empty set. In this way, similar to Kuzmin and Warmuth (2007), we obtain a compression scheme so that every labeled sample of size 2 is associated with a unique concept in C, which is consistent with the sample. On the other hand to obtain corner peeling, we need the sweepout to proceed with t increasing so that we can begin at the boundary vertices of C.

In concluding this brief discussion, we note that many infinite collections of simple hyperbolic hyperplanes and Euclidean hyperplanes can also be corner peeled and compressed, even if intersection points and cells accumulate. However a key requirement in the Euclidean case is that the concept class C has a non-empty boundary, when considered as a cubical complex. An easy approach is to assume that all the d-intersections of the arrangement lie in a half-space. Moreover, since the boundary must also admit corner peeling, we require more conditions, similar to having all the intersection points lying in an octant.

Example 18 *In $\mathbb{R}^3$, choose the family of planes $\mathcal{P}$ of the form $P_{3n+i} = \{\mathbf{x} \in \mathbb{R}^3 \mid x_{i+1} = 1-1/n\}$ for $n \geq 1$ and $i \in \{0,1,2\}$. A corner-peeling scheme is induced by sweeping a generic plane $\{\mathbf{x} \in \mathbb{R}^3 \mid x_1 + \alpha x_2 + \beta x_3 = t\}$ across the arrangement, where t is a parameter and $1, \alpha, \beta$ are algebraically* *independent (in particular, no integral linear combination is rational) and α, β are both close to 1. This example has similar properties to Example 17: the compression scheme is again given by decreasing t whereas corner peeling corresponds to increasing t. Note that cells shrink to points, as $\mathbf{x} \to \mathbf{1}$ and the volume of cells converge to zero as $n \to \infty$, or equivalently any $x_i \to 1$.*

Example 19 *In the hyperbolic plane $\mathbb{H}^2$, represented as the unit circle centered at the origin in $\mathbb{R}^2$, choose the family of lines $\mathcal{L}$ given by $L_{2n} = \{(x,y) \mid x = 1 - 1/n\}$ and $L_{2n+1} = \{(x,y) \mid x + ny = 1\}$, for $n \geq 1$. This arrangement has corner peeling and compression schemes given by sweeping across $\mathcal{L}$ using the generic line $\{y = t\}$.*

7. Piecewise-Linear Arrangements

PL hyperplanes have the advantage that they can be easily manipulated, by cutting and pasting or isotoping part of a hyperplane to a new position, keeping the rest of the hyperplane fixed. However a disadvantage is that there is no simple way of describing a PL hyperplane, similar to the parametrizations of either linear or hyperbolic hyperplanes. The methods of proof of our main results about representing maximum classes and corner peeling, require PL-hyperplane arrangements. We conjecture that PL-hyperplane arrangements are equivalent to hyperbolic ones. This would give an interesting geometric approach of forming all maximum classes as simple hyperbolic arrangements.

A *PL hyperplane* is the image of a proper piecewise-linear homeomorphism from the $(k-1)$-ball B^{k-1} into B^k, that is, the inverse image of the boundary S^{k-1} of the k-ball is S^{k-2} (Rourke and Sanderson, 1982). A *simple PL d-arrangement* is an arrangement of n PL hyperplanes such that every subcollection of j hyperplanes meet transversely in a $(k-j)$-dimensional PL plane for $2 \leq j \leq d$ and every subcollection of $d+1$ hyperplanes are disjoint.

Corollary 31 *The concept class C corresponding to a simple d-arrangement of PL hyperplanes in B^k is d-maximum in the k-cube.*

Proof The result follows by the same argument as in the linear or hyperbolic cases. Projection cannot shatter any $(d+1)$-cube and the class is a complete union of d-cubes, so is d-maximum. ■

7.1 Maximum Classes are Represented by Simple PL-Hyperplane Arrangements

Our aim is to prove the following theorem by a series of steps.

Theorem 32 *Every d-maximum class $C \subseteq \{0,1\}^n$ can be represented by a simple arrangement of PL hyperplanes in an n-ball. Moreover the corresponding simple arrangement of PL hyperspheres in the $(n-1)$-sphere also represents C, so long as $n > d+1$.*

7.1.1 EMBEDDING A d-MAXIMUM CUBICAL COMPLEX IN THE n-CUBE INTO AN n-BALL

We begin with a d-maximum cubical complex $C \subseteq \{0,1\}^n$ embedded into $[0,1]^n$. This gives a natural embedding of C into $\mathbb{R}^n$. Take a small regular neighborhood $\mathcal{N}$ of C so that the boundary $\partial\mathcal{N}$ of $\mathcal{N}$ will be a closed manifold of dimension $n-1$. Note that $\mathcal{N}$ is contractible because it has a deformation retraction onto C and so $\partial\mathcal{N}$ is a homology $(n-1)$-sphere (by a standard, well-known

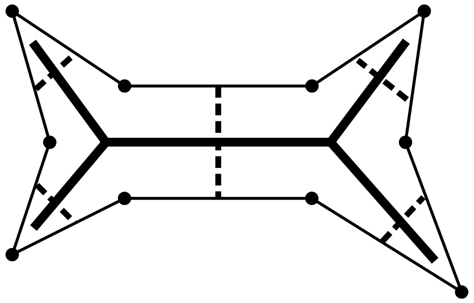


Figure 12: A 1-maximum class (thick solid lines) with its fattening (thin solid lines with points), bisecting sets (dashed lines) and induced complementary cells.

argument from topology due to Mazur 1961). Our aim is to prove that $\partial\mathcal{N}$ is an $(n-1)$-sphere and $\mathcal{N}$ is an n-ball. There are two ways of proving this: show that $\partial\mathcal{N}$ is simply connected and invoke the well-known solution to the generalized Poincaré conjecture (Smale, 1961), or use the cubical structure of the n-cube and C to directly prove the result. We adopt the latter approach, although the former works fine. The advantage of the latter is that it produces the required hyperplane arrangement, not just the structures of $\partial\mathcal{N}$ and $\mathcal{N}$.

7.1.2 BISECTING SETS

For each color i, there is a hyperplane P_i in $\mathbb{R}^n$ consisting of all vectors with i^{th} coordinate equal to $1/2$. We can easily arrange the choice of regular neighborhood $\mathcal{N}$ of C so that $\mathcal{N}_i = P_i \cap \mathcal{N}$ is a regular neighborhood of $C \cap P_i$ in P_i. (We call $\mathcal{N}_i$ a *bisecting set* as it intersects C along the 'center' of the reduction in the i^{th} coordinate direction, see Figure 12.) But then since $C \cap P_i$ is a cubical complex corresponding to the reduction C^i, by induction on n, we can assert that $\mathcal{N}_i$ is an $(n-1)$-ball. Similarly the intersections $\mathcal{N}_i \cap \mathcal{N}_j$ can be arranged to be regular neighborhoods of $(d-2)$-maximum classes and are also balls of dimension $n-2$, etc. In this way, we see that if we can show that $\mathcal{N}$ is an n-ball, then the induction step will be satisfied and we will have produced a PL-hyperplane arrangement (the system of $\mathcal{N}_i$ in $\mathcal{N}$) in a ball.

7.1.3 SHIFTING

To complete the induction step, we use the technique of shifting (Alon, 1983; Frankl, 1983; Haussler, 1995). In our situation, this can be viewed as the converse of lifting. Namely if a color i is chosen, then the cubical complex C has a lifted reduction C' consisting of all d-cubes containing the i^{th} color. By shifting, we can move down any of the lifted components, obtained by splitting C open along C', from the level $x_i = 1$ to the level $x_i = 0$, to form a new cubical complex $C^\star$. We claim that the regular neighborhood of C is a ball if and only if the same is true for $C^\star$. But this is quite straightforward, since the operation of shifting can be thought of as sliding components of C, split open along C', continuously from level $x_i = 1$ to $x_i = 0$. So there is an isotopy of the attaching maps of the components onto the lifted reduction, using the product structure of the latter. It is easy then to check that this does not affect the homeomorphism type of the regular neighborhood and so the claim of shift invariance is proved.

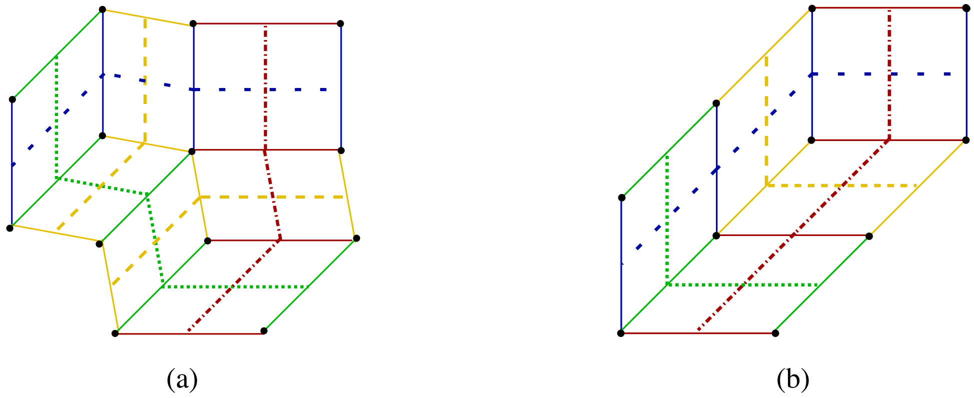


Figure 13: The (a) top and (b) bottom of Figure 9(b) (i.e., the 2-cubes seen from above and below, respectively) both give part of the boundary of a regular neighborhood in $\mathbb{R}^3$.

But repeated shifting finishes with the downwards closed maximum class consisting of all vertices in the n-cube with at most d coordinates being one and the remaining coordinates all being zero. It is easy to see that the corresponding cubical complex $\tilde{C}$ is star-like, that is, contains all the straight line segments from the origin to any point in $\tilde{C}$. If we choose a regular neighborhood $\tilde{\mathcal{N}}$ to also be star-like, then it is obvious that $\tilde{\mathcal{N}}$ is an n-ball, using radial projection. Hence our induction is complete and we have shown that any d-maximum class in $\{0,1\}^n$ can be represented by a family of PL hyperplanes in the n-ball.

7.1.4 IDEAL BOUNDARY

To complete the proof of Theorem 32, let $\partial\mathcal{N} = S^{n-1}$ denote the boundary of the n-ball $\mathcal{N}$ constructed above (cf. Figure 13). Each PL hyperplane intersects this sphere in a PL hypersphere of dimension $n-2$. It remains to show this arrangement of hyperspheres gives the same cubical complex as C, unless $n = d+1$.

Suppose that $n > d+1$. Then it is easy to see that each cell c in the complement of the PL-hyperplane arrangement in $\mathcal{N}$ has part of its boundary on the ideal boundary $\partial\mathcal{N}$. Let $\partial c = \partial c_+ \cup \partial c_-$, where ∂c_+ is the intersection of c with the ideal boundary and ∂c_- is the closure of $\partial c \setminus \partial c_+$.

It is now straightforward to verify that the face structure of ∂c_+ is equivalent to the face structure of ∂c_-. Note that any family of at most d PL hyperplanes meet in a PL ball properly embedded in $\mathcal{N}$. Since $n > d+1$, the smallest dimension of such a ball is two, and hence its boundary is connected. Then ∂c_- has faces which are PL balls obtained in this way of dimension varying between $n-d$ and $n-1$. Each of these faces has boundary a PL sphere which is a face of ∂c_+. So this establishes a bijection between the faces of ∂c_+ and those of ∂c_-. It is easy to check that the cubical complexes corresponding to the PL hyperplanes and to the PL hyperspheres are the same.

Note that if $n = d+1$, then any d-maximum class $C \subseteq \{0,1\}^{d+1}$ is obtained by taking all the d-faces of the $(d+1)$-cube which contain a particular vertex. So C is a d-ball and the ideal boundary of $\mathcal{N}$ is a d-sphere. The cubical complex associated with the ideal boundary is the double $2C$ of C, that is, two copies of C glued together along their boundaries. The proof of Theorem 32 is now complete.

Example 20 *Consider the unique bounded below* 2*-maximum class* $\tilde{C} \subseteq \{0,1\}^5$. *We claim that* $\tilde{C}$ *cannot be realized as an arrangement of PL hyperplanes in the* 3*-ball* B^3. *Note that our method gives* $\tilde{C}$ *as an arrangement in* B^5 *and this example shows that* B^4 *is the best one might hope for in terms of dimension of the hyperplane arrangement.*

For suppose that $\tilde{C}$ *could be realized by any PL-hyperplane arrangement in* B^3. *Then clearly we can also embed* $\tilde{C}$ *into* B^3. *The vertex* $v_0 = \{0\}^5$ *has link given by the complete graph K on* 5 *vertices in* $\tilde{C}$. *(By* link, *we mean the intersection of the boundary of a small ball in* B^3 *centered at* v_0 *with* $\tilde{C}$.*) But as is well known, K is not planar, that is, cannot be embedded into the plane or* 2*-sphere. This contradiction shows that no such arrangement is possible.*

7.2 Maximum Classes with Manifold Cubical Complexes

We prove a partial converse to Corollary 23: if a d-maximum class has a ball as cubical complex, then it can always be realized by a simple PL-hyperplane arrangement in $\mathbb{R}^d$.

Theorem 33 *Suppose that* $C \subseteq \{0,1\}^n$ *is a d-maximum class. Then the following properties of C, considered as a cubical complex, are equivalent:*

(i) There is a simple arrangement A of n PL hyperplanes in $\mathbb{R}^d$ *which represents C.*

(ii) C is homeomorphic to the d-ball.

(iii) C is a d-manifold with boundary.

Proof To prove (i) implies (ii), we can use exactly the same argument as Corollary 23. Next (ii) trivially implies (iii). So it remains to show that (iii) implies (i). The proof proceeds by double induction on n,d. The initial cases where either $d = 1$ or $n = 1$ are very easy.

Assume that C is a manifold. Let p denote the i^{th} coordinate projection. Then $p(C)$ is obtained by collapsing $C^i \times [0,1]$ onto C^i, where C^i is the reduction. As before, let P_i be the linear hyperplane in $\mathbb{R}^n$, where the i^{th} coordinate takes value $1/2$. Viewing C as a manifold embedded in the n-cube, since P_i intersects C transversely, we see that $C^i \times \{1/2\}$ is a proper submanifold of C. But it is easy to check that collapsing $C^i \times [0,1]$ to C^i in C produces a new manifold which is again homeomorphic to C. (The product region $C^i \times [0,1]$ in C can be expanded to a larger product region $C^i \times [-\varepsilon, 1+\varepsilon]$ and so collapsing shrinks the larger region to one of the same homeomorphism type, namely $C^i \times [-\varepsilon, \varepsilon]$). So we conclude that the projection $p(C)$ is also a manifold. By induction on n, it follows that there is a PL-hyperplane arrangement A, consisting of $n-1$ PL hyperplanes in B^d, which represents $p(C)$.

Next, observe that the reduction C^i can be viewed as a properly embedded submanifold M in B^d, where M is a union of some of the $(d-1)$-dimensional faces of the Voronoi cell decomposition corresponding to A, described in Corollary 23. By induction on d, we conclude that C^i is also represented by n PL hyperplanes in B^{d-1}. But then since condition (i) implies (ii), it follows that M is PL homeomorphic to B^{d-1}, since the underlying cubical complex for C^i is a $(d-1)$-ball. So it follows that $A \cup \{M\}$ is a PL-hyperplane arrangement in B^d representing C. This completes the proof that condition (iii) implies (i). ■

8. Corner Peeling 2-Maximum Classes

We give a separate treatment for the case of 2-maximum classes, since it is simpler than the general case and shows by a direct geometric argument, that representation by a simple family of PL hyperplanes or PL hyperspheres implies a corner-peeling scheme.

Theorem 34 *Every* 2*-maximum class can be corner peeled.*

Proof By Theorem 32, we can represent any 2-maximum class $C \subseteq \{0,1\}^n$ by a simple family of PL hyperspheres $\{S_i\}$ in S^{n-1}. Every pair of hyperspheres S_i, S_j intersects in an $(n-3)$-sphere S_{ij} and there are no intersection points between any three of these hyperspheres. Consider the family of spheres S_{ij}, for i fixed. These are disjoint hyperspheres in S_i so we can choose an innermost one S_{ik} which bounds an $(n-2)$-ball B_1 in S_i not containing any other of these spheres. Moreover there are two balls B_2, B_3 bounded by S_{ik} on S_k. We call the two $(n-1)$-balls Q_2, Q_3 bounded by $B_1 \cup B_2$, $B_1 \cup B_3$ respectively in S^{n-1}, which intersect only along B_1, *quadrants*.

Assume B_2 is innermost on S_k. Then the quadrant Q_2 has both faces B_1, B_2 innermost. It is easy to see that such a quadrant corresponds to a corner vertex in C which can be peeled. Moreover, after peeling, we still have a family of PL hyperspheres which give an arrangement corresponding to the new peeled class. The only difference is that cell Q_2 disappears, by interchanging B_1, B_2 on the corresponding spheres S_i, S_k and then slightly pulling the faces apart. (If $n = 3$, we can visualize a pair of disks on two intersecting spheres with a common boundary circle. Then peeling can be viewed as moving these two disks until they coincide and then pulling the first past the second). So it is clear that if we can repeatedly show that a quadrant can be found with two innermost faces, until all the intersections between the hyperspheres have been removed, then we will have corner peeled C to a 1-maximum class, that is, a tree. So peeling will be established.

Suppose neither of the two quadrants Q_2, Q_3 has both faces innermost. Consider Q_2 say and let $\{S_\alpha\}$ be the family of spheres intersecting the interior of the face B_2. Amongst these spheres, there is clearly at least one S_β so that the intersection $S_{k\beta}$ is innermost on S_k. But then $S_{k\beta}$ bounds an innermost ball B_4 in S_k whose interior is disjoint from all the spheres $\{S_\alpha\}$. Similarly, we see that $S_{k\beta}$ bounds a ball B_5 which is the intersection of the sphere S_β with the quadrant Q_2. We get a new quadrant bounded by $B_4 \cup B_5$ which is strictly smaller than Q_2 and has at least one innermost face. But clearly this process must terminate—we cannot keep finding smaller and smaller quadrants and so a smallest one must have both faces innermost. ■

9. Corner Peeling Finite Maximum Classes

Above, simple PL-hyperplane arrangements in the n-ball B^n are defined. For the purposes of this section, we study a slightly more general class of arrangements. Every simple arrangement is in this larger class, but the latter class has many good properties. In Example 23, a maximal class is represented by a 2-contractible hyperbolic-hyperplane arrangement. By contrast, simple hyperplane arrangements always represent maximum classes.

Definition 35 *Suppose that a finite arrangement $\mathcal{P}$ of PL hyperplanes $\{P_\alpha\}$, each properly embedded in an n-ball B^n, satisfies the following conditions:*

i. Each k-subcollection of hyperplanes either intersects transversely in a PL plane of dimension $n-k$, or has an empty intersection; and

ii. The maximum number of hyperplanes which mutually intersect is $d \leq n$.

*Then we say that the arrangement $\mathcal{P}$ is d-*contractible.

The arrangements in Definition 35 are called d-contractible because we prove later that their corresponding one-inclusion graphs are strongly contractible cubical complexes of dimension d. Moreover we now prove that the corresponding one-inclusion graphs have VC dimension exactly d.

Lemma 36 *The one-inclusion graph Γ corresponding to a d-contractible arrangement $\mathcal{P}$ has VC-dimension d.*

Proof We observe first of all, that since $\mathcal{P}$ has a subcollection of d hyperplanes which mutually intersect, the corresponding one-inclusion graph Γ has a d-subcube, when considered as a cubical complex. But then the VC dimension of Γ is clearly at least d. On the other hand, suppose that the VC dimension of Γ was greater than d. Then there is a projection of Γ which shatters some $(d+1)$-cube. But this projection can be viewed as deleting all the hyperplanes of $\mathcal{P}$ except for a subcollection of $d+1$ hyperplanes. However, by assumption, such a collection cannot have any mutual intersection points. It is easy to see that any such an arrangement has at most $2^{d+1}-1$ complementary regions and hence cannot represent the $(d+1)$-cube. This completes the proof. ■

Definition 37 *A one-inclusion graph Γ is* strongly contractible *if it is contractible as a cubical complex and moreover, all reductions and multiple reductions of Γ are also contractible.*

Definition 38 *The* complexity *of a PL-hyperplane arrangement $\mathcal{P}$ is the lexicographically ordered pair (r,s), where r is the number of regions in the complement of $\mathcal{P}$, and s is the smallest number of regions in any half space on one side of an individual hyperplane in $\mathcal{P}$.*

We allow several different hyperplanes to be used for a single sweeping process. So a hyperplane P may start sweeping across an arrangement $\mathcal{P}$. One of the half spaces defined by P can become a new ball B_+ with a new arrangement $\mathcal{P}_+$ defined by restriction of $\mathcal{P}$ to the half space B_+. Then a second generic hyperplane P' can start sweeping across this new arrangement $\mathcal{P}_+$. This process may occur several times. It is easy to see that sweeping a single generic hyperplane as in Theorem 29, applies to such a multi-hyperplane process. Below we show that a suitable multiple sweeping of a PL-hyperplane arrangement $\mathcal{P}$ gives a corner-peeling sequence of all finite maximum classes.

The following states our main theorem.

Theorem 39 *Assume that $\mathcal{P}$ is a d-contractible PL-hyperplane arrangement in the n-ball B^n. Then there is a d-corner-peeling scheme for this collection $\mathcal{P}$.*

Corollary 40 *There is no constant k so that every finite maximal class of VC-dimension d can be embedded into a maximum class of VC-dimension $d+k$.*

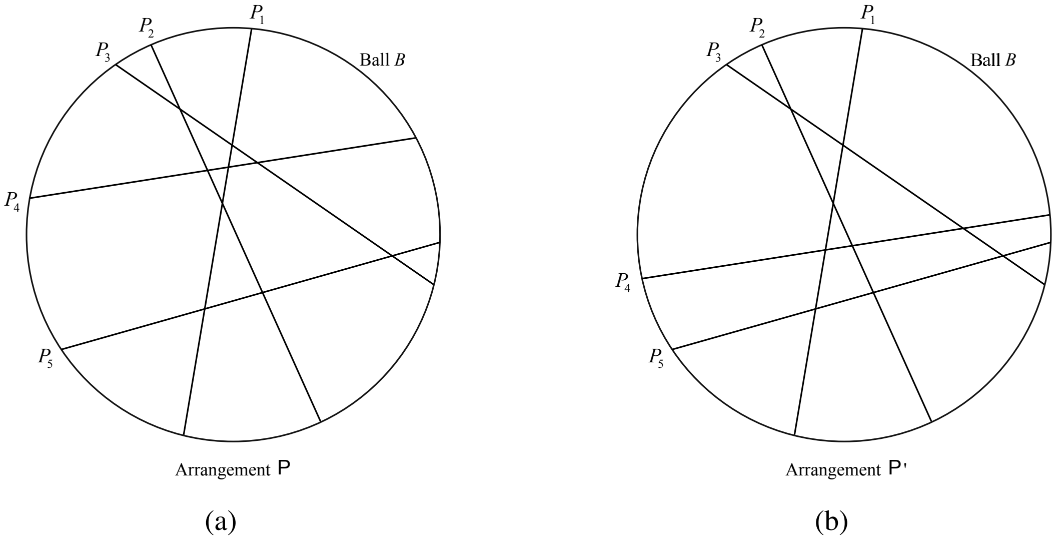


Figure 14: (a) An example PL-hyperplane arrangement $\mathcal{P}$ and (b) the result of a Pachner move of hyperplane P_4 on $\mathcal{P}$.

Proof By Theorem 39, every maximum class has a peeling scheme which successively removes vertices from the one-inclusion graph, so that the vertices being discarded never have degree more than d. But Rubinstein et al. (2007) gave examples of maximal classes of VC-dimension d which have a core of the one-inclusion graph of size $d+k$ for any constant k. Recall that a core is a subgraph and its size is the minimum degree of all the vertices. Having a peeling scheme gives an upper bound on the size of any core and so the result follows. ■

9.1 Proof of Main Theorem

The proof is by induction on the complexity of $\mathcal{P}$. Since we are dealing with the class of d-contractible PL-hyperplane arrangements, it is easy to see that if any such $\mathcal{P}$ is *split open* along some fixed hyperplane P_1 in the arrangement (see Figures 14–15), then the result is two new arrangements $\mathcal{P}_+, \mathcal{P}_-$ each of which contains fewer hyperplanes and also fewer complementary regions than the initial one. The new arrangements have smaller complexity than $\mathcal{P}$ and are $k-, k'$-contractible for some $k, k' \leq d$. This is the key idea of the construction.

To examine this splitting process in detail, first note that each hyperplane P_α of $\mathcal{P}$ is either disjoint from P_1 or splits along P_1 into two hyperplanes P_α^+, P_α^-. We can now construct the new PL-hyperplane arrangements $\mathcal{P}_+, \mathcal{P}_-$ in the balls B_+, B_- obtained by splitting B along P_1. Note that $\partial B_+ = P_1 \cup D_+$ and $\partial B_- = P_1 \cup D_-$ where D_+, D_- are balls of dimension $n-1$ which have a common boundary with P_1. It is easy to verify that $\mathcal{P}_+, \mathcal{P}_-$ satisfy similar hypotheses to the original arrangement. Observe that the maximum number of mutually intersecting hyperplanes in $\mathcal{P}_+, \mathcal{P}_-$ may decrease relative to this number for $\mathcal{P}$, after the splitting operation. The reason is that the hyperplane P_1 'disappears' after splitting and so if all maximum subcollections of $\mathcal{P}$ which

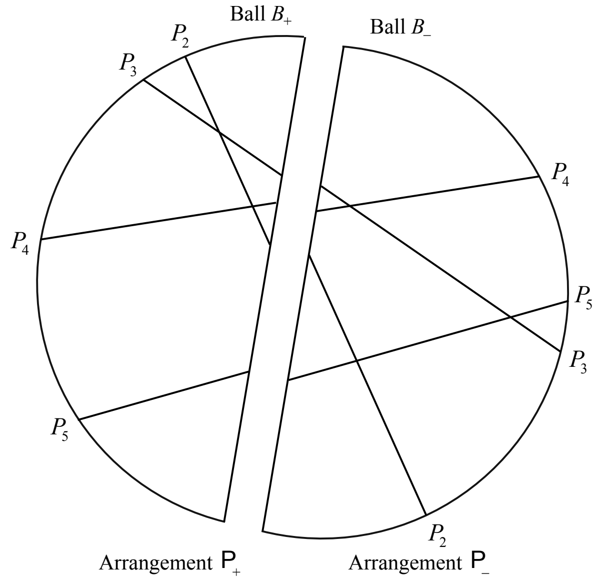


Figure 15: Result of splitting $\mathcal{P}$ in Figure 14(a) along hyperplane P_1.

mutually intersect, all contain P_1, then this number is smaller for $\mathcal{P}_+, \mathcal{P}_-$ as compared to the initial arrangement $\mathcal{P}$. This number shows that $\mathcal{P}_+, \mathcal{P}_-$ can be k- or k'-contractible, for $k, k' < d$ as well as the cases where $k, k' = d$.

Start the induction with any arrangement with one hyperplane. This gives two regions and complexity $(2,1)$. The corresponding graph has one edge and two vertices and obviously can be corner peeled.

We now describe the inductive step. There are two cases. In the first, assume the arrangement has complexity $(r,1)$. The corresponding graph has a vertex which belongs to only one edge, so can be corner peeled. This gives an arrangement with fewer hyperplanes and clearly the complexity has decreased to $(r-1,s)$ for some s. This completes the inductive step for the first case.

For the second case, assume that all d-contractible hyperplane arrangements with complexity smaller than (r,s) have corner-peeling sequences and $s > 1$. Choose any d-contractible hyperplane arrangement $\mathcal{P}$ with complexity (r,s). Select a hyperplane P_1 which splits the arrangement into two smaller arrangements $\mathcal{P}_+, \mathcal{P}_-$ in the balls B_+, B_-. By our definition of complexity, it is easy to see that however we choose P_1, the complexity of each of $\mathcal{P}_+, \mathcal{P}_-$ will be less than that of $\mathcal{P}$. However, a key requirement for the proof will be that we select P_1 so that it has precisely s complementary regions for $\mathcal{P}_+$, that is, P_1 has fewest complementary regions in one of its halfspaces, amongst all hyperplanes in the arrangement.

Since $\mathcal{P}_+$ has smaller complexity than (r,s), by our inductive hypothesis, it can be corner peeled (cf. Figure 16). If any of the corner-peeling moves of $\mathcal{P}_+$ is a corner-peeling move for $\mathcal{P}$, then the argument follows. For any corner-peeling move of $\mathcal{P}$ gives a PL-hyperplane arrangement with fewer complementary cells than $\mathcal{P}$ and thus smaller complexity than (r,s). Hence by the inductive hypothesis, it follows that $\mathcal{P}$ can be corner peeled.

Next, suppose that no corner-peeling move of $\mathcal{P}_+$ is a corner-peeling move for $\mathcal{P}$. In particular, the first corner-peeling move for $\mathcal{P}_+$ must occur for a cell R_+ in the complement of $\mathcal{P}$, which is

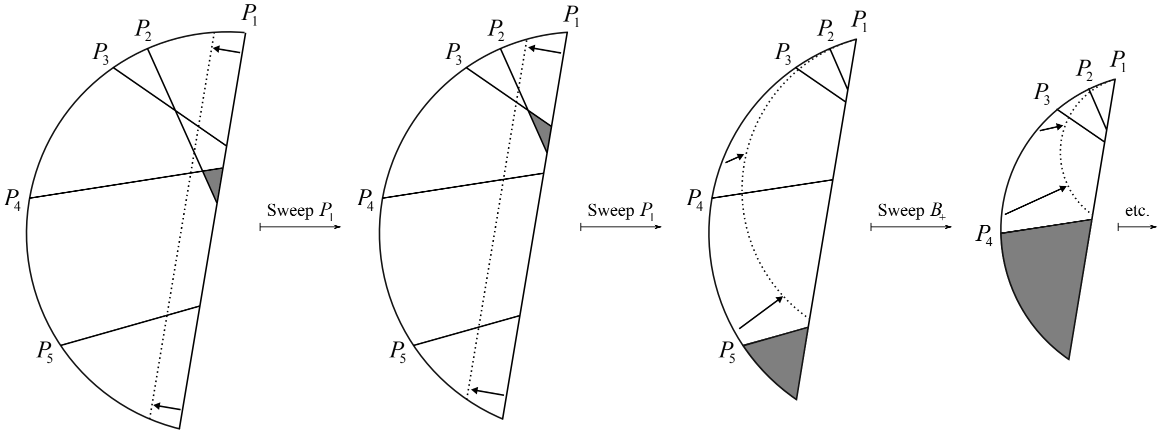


Figure 16: Partial corner-peeling sequence for the (B_+, P_+) arrangement split from the arrangement of Figure 15, in the proof of Theorem 39.

adjacent to P_1. (Clearly any corner-peeling move for $\mathcal{P}_+$, which occurs at a region R_1 with a face on D_+, will be a corner-peeling move for $\mathcal{P}$.) R_+ must be a product of a d'-simplex Δ with a copy of $\mathbb{R}^{n-d'}$, with one face on P_1 and the other faces on planes of $\mathcal{P}$. This is because a corner-peeling move can only occur at a cell with this type of face structure, as described in Theorem 29. The corresponding effect on the one-inclusion graph is peeling of a vertex which is a corner of a d'-cube in the binary class corresponding to the arrangement $\mathcal{P}_+$, where $d' \leq d$.

Now even though such a cell R_+ does not give a corner-peeling move for $\mathcal{P}$, we can push P_1 across R_+. The effect of this is to move the complementary cell R_+ from B^+ to B^-. Moreover, since we assumed that the hyperplane P_1 satisfies $\mathcal{P}^+$ has a minimum number s of complementary regions, it follows that the move pushing P_1 across R_+ produces a new arrangement $\mathcal{P}^\star$ with smaller complexity $(r, s-1)$ than the original arrangement $\mathcal{P}$. Hence by our inductive assumption, $\mathcal{P}^\star$ admits a corner-peeling sequence.

To complete the proof, we need to show that existence of a corner-peeling sequence for $\mathcal{P}^\star$ implies that the original arrangement $\mathcal{P}$ has at least one corner-peeling move. Recall that R_+ has face structure given by $\Delta \times \mathbb{R}^{n-d'}$, with one face on P_1 and the other faces on planes of $\mathcal{P}$. Consider the subcomplex U of the one-inclusion graph consisting of all the regions sharing a vertex or face of dimension k for $1 \leq k \leq n-1$ with R_+. It is not difficult to see that U is a d'-ball consisting of $d'+1$ cubes, each of dimension d'. (As examples, if $d' = 2$, U consists of 3 2-cubes forming a hexagon and if $d' = 3$, U consists of 4 3-cubes with boundary a rhombic dodecahedron.)

Consider the first corner-peeling move on the arrangement $\mathcal{P}^\star$. Note that the one-inclusion graphs of $\mathcal{P}^\star$ and $\mathcal{P}$ differ precisely by replacing U with U', that is, by a Pachner move. Hence this first corner-peeling move must occur at a vertex v_1 whose degree is affected by this replacement, since otherwise, the corner-peeling move would also apply to $\mathcal{P}$ and the proof would be complete. In fact, if v_1 has the same number of adjacent edges before and after the Pachner move, then it must belong to the same single maximum dimension cube before and after the Pachner move. (The only cubes altered by the Pachner move are the ones in U.) It is easy to see that, v_1 must belong to $\partial U = \partial U'$ and must have degree d' in $\mathcal{P}^\star$. So v_1 is a corner of a single d'-cube for U' and does

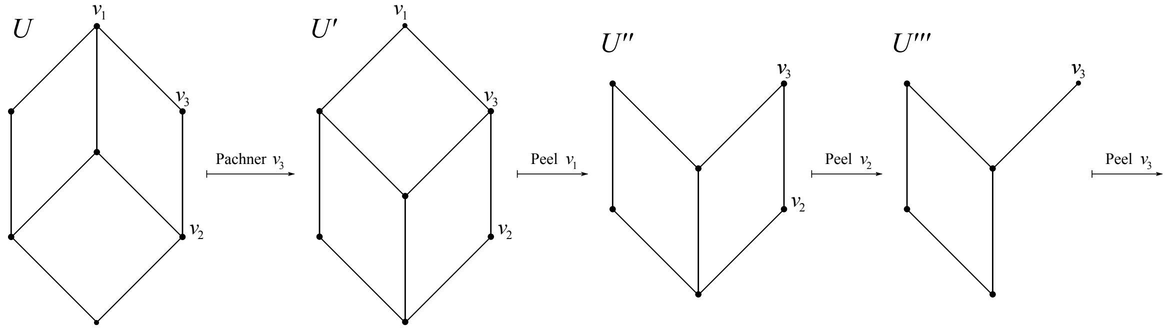


Figure 17: A 2-maximum complex in the 3-cube. After a Pachner move vertices v_1, v_2, v_3, etc. can be corner-peeled.

not belong to any other edges or cubes of the one inclusion graph for $\mathcal{P}^\star$. In U (and hence also in $\mathcal{P}$), v_1 belongs to d'-cubes of dimension d' and so has degree $d'+1$. After peeling away v_1 and its corresponding d'-cube, we still have a d'-ball with only d'-cubes, (cf. Figure 17).

Consider the next corner-peeling move. We claim that it must again be at a vertex v_2 belonging to $\partial U'$. The reason is that only vertices belonging to U' have degree reduced by our first corner-peeling move. So if this second move did not occur at a vertex of U', then it could be used as a corner-peeling move of our initial arrangement $\mathcal{P}$. There may be several choices for v_2. For example, if $d'=2$, then U' is a hexagonal disk and removing one 2-cube from U' gives a choice which could be either of the two vertices which are corners of a single 2-cube in U', (cf. Figure 17). Note that a vertex which is a corner of a single cube in U' remains so after corner peeling at v_1. Note also that v_2 cannot belong to any edges of the one-inclusion graph which are not in U', as for v_1, if v_2 can be used for corner peeling.

We can continue examining corner-peeling moves of $\mathcal{P}^\star$ and find that all must occur at vertices in $\partial U'$, until the unique interior vertex is ready to be peeled, that is, belongs to a single cube. (See Figure 17.) The key to understanding this is that firstly, when we initially peel only vertices in $\partial U'$, these are not adjacent to any vertices of the one-inclusion graph outside U' and so cannot produce any new opportunities for corner peeling of vertices not in U'. Secondly, if the unique interior vertex v of U' can be corner peeled, after sufficiently many vertices in $\partial U'$ have been peeled, then new vertices in $\partial U'$ become candidates for peeling. For although these latter vertices may be adjacent to vertices outside U', after v has been peeled, they may become a corner vertex of a unique maximal cube.

But now a final careful examination of this situation shows that there must be at least one vertex of U which belongs to a single d'-cube in U and to no other edges in $\mathcal{P}$. So this will give our initial corner-peeling move of $\mathcal{P}$.

To elaborate, we can describe U as the set of d'-cubes which share the vertex $(0,0,\ldots,0)$ in the $(d'+1)$-cube $\{0,1\}^{d'+1}$. Then U' consists of all the d'-cubes in $\{0,1\}^{d'+1}$ which contain the vertex $(1,1,\ldots,1)$. Now assume that an initial sequence of corner peeling of vertices in $\partial U'$ allows the next step to be corner peeling of the unique interior vertex v. Note that in the notation above, v corresponds to the vertex $(1,1,\ldots,1)$.

As in Figure 17, we may assume that after the corner peeling corresponding to the initial sequence of vertices in $\partial U'$, that there is a single d'-cube left in U' containing v. Without loss of generality, suppose this is the cube with vertices with $x_1 = 1$ where the coordinates are $x_1, x_2, \ldots, x_{d'+1}$ in $\{0,1\}^{d'+1}$. But then, it follows that there are no vertices outside U' adjacent to any of the initial sequence of vertices, which are all the vertices in $\{0,1\}^{d'+1}$ with $x_1 = 0$, except for $(0,0,\ldots,0)$. But now the vertex $(0,1,\ldots,1)$ has the property that we want - it is contained in a unique d'-cube in U and is adjacent to no other vertices outside U. This completes the proof.

9.2 Peeling Classes with Generic Linear or Generic Hyperbolic Arrangements

In this subsection, we study a special class of d-contractible arrangements. If a collection of hyperplanes in an n-manifold is in general position, then they have the property in the following definition. Then a key idea in differential or PL topology is that any collection can be slightly perturbed to be in general position. See Rourke and Sanderson (1982) for a discussion of these issues in the PL case.

Definition 41 *A linear or hyperbolic-hyperplane arrangement $\mathcal{P}$ in $\mathbb{R}^n$ or $\mathbb{H}^n$ respectively, is called* generic*, if any subcollection of k hyperplanes of $\mathcal{P}$, for $2 \leq k \leq n$ has the property that there are no intersection points or the subcollection intersects transversely in a plane of dimension $n-k$.*

Corollary 42 *Suppose $\mathcal{P}$ is a generic linear or hyperbolic-hyperplane arrangement in $\mathbb{R}^n$ or $\mathbb{H}^n$ and amongst all subcollections of $\mathcal{P}$, the largest with an intersection point in common, has d hyperplanes. Then $\mathcal{P}$ admits a d-corner-peeling scheme.*

Remark 43 *The proof of Corollary 42 is immediate since it is obvious that any generic linear or hyperbolic-hyperplane arrangement is a d-contractible PL-hyperplane arrangement, where d is the cardinality of the largest subcollection of hyperplanes which mutually intersect. Note that many generic linear, hyperbolic or d-contractible PL-hyperplane arrangements do not embed in any simple linear, hyperbolic or PL-hyperplane arrangement. For if there are two hyperplanes in $\mathcal{P}$ which are disjoint, then this is an obstruction to enlarging the arrangement by adding additional hyperplanes to obtain a simple arrangement. Hence this shows that Theorem 39 produces compression schemes, by corner peeling, for a considerably larger class of one-inclusion graphs than just maximum one-inclusion graphs. However it seems possible that d-contractible PL hyperplanes always embed in d-maximum classes, by 'undoing' the operation of sweeping and corner peeling, which pulls apart the hyperplanes.*

10. Peeling Infinite Maximum Classes with Finite-Dimensional Arrangements

We seek infinite classes represented by arrangements satisfying the same conditions as above. Note that any finite subclass of such an infinite class then satisfies these conditions and so can be corner peeled. Hence any such a finite subclass has a complementary region R which has face structure of the product of a d'-simplex with a copy of $\mathbb{R}^{n-d'}$ with one face on the boundary of B^n. To find such a region in the complement of our infinite collection $\mathcal{P}$, we must impose some conditions.

One convenient condition (cf. the proof of Theorem 39) is that a hyperplane P_α in $\mathcal{P}$ can be found which splits B^n into pieces B_+, B_- so that one, say B_+ gives a new arrangement for which the maximum number of mutually intersecting hyperplanes is strictly less than that for $\mathcal{P}$. Assume that the new arrangement satisfies a similar condition, and we can keep splitting until we get to disjoint hyperplanes.

It is not hard to prove that such arrangements always have peeling sequences. Moreover the peeling sequence does give a compression scheme. This sketch establishes the following.

Theorem 44 *Suppose that a countably infinite collection $\mathcal{P}$ of PL hyperplanes $\{P_\alpha\}$, each properly embedded in an n-ball B^n, satisfies the following conditions:*

i. $\mathcal{P}$ satisfies the conditions of d-contractible arrangements as in Definition 35 and

ii. There is an ordering of the planes in $\mathcal{P}$ so that if we split B^n successively along the planes, then at each stage, at least one of the two resulting balls has an arrangement with a smaller maximum number of planes which mutually intersect.

Then there is a d-corner-peeling scheme for $\mathcal{P}$, and this provides a d-unlabeled compression scheme.

Example 21 *Rubinstein and Rubinstein (2008) give an example that satisfies the assumptions of Theorem 44. Namely in $\mathbb{R}^n$ choose the positive octant $O = \{(x_1, x_2, \ldots x_n) : x_i \geq 0\}$. Inside O choose the collection of hyperplanes given by $x_i = m$ for all $1 \leq i \leq n$ and $m \geq 1$ a positive integer. There are many more examples, we present only a very simple model here. Take a graph inside the unit disk D with a single vertex of degree 3 and the three end vertices on ∂D. Now choose a collection of disjoint embedded arcs representing hyperplanes with ends on ∂D and meeting one of the edges of the graph in a single point. We choose finitely many such arcs along two of the graph edges and an infinite collection along one arc. This gives a very simple family of hyperplanes satisfying the hypotheses of Theorem 44. Higher dimensional examples with intersecting hyperplanes based on arbitrary trees can be constructed in a similar manner.*

11. Contractibility, Peeling and Arrangements

In this section, we characterize the concept classes which have one-inclusion graphs representable by d-contractible PL-hyperplane arrangements.

Theorem 45 *Assume that C is a concept class in the binary n-cube and d is the largest dimension of embedded cubes in its one-inclusion graph Γ. The following are equivalent.*

i. Γ is a strongly contractible cubical complex.

ii. There is a d-contractible PL-hyperplane arrangement $\mathcal{P}$ in an n-ball which represents Γ.

Proof To prove that *i* implies *ii*, we use some important ideas in the topology of manifolds. The cubical complex C is naturally embedded into the binary n-cube, which can be considered as an n-ball B^n. A regular neighborhood N of C homotopy retracts onto C and so is contractible. Now we can use a standard argument from algebraic and geometric topology to prove that N is a ball. Firstly, ∂N is simply connected, assuming that $n-d>2$. For given a loop in ∂N, it bounds a disk in N by contractibility. Since C is a d-dimensional complex and $n-d>2$ it follows that this disk can be pushed off C by transversality and then pushed into ∂N. But now we can follow a standard argument using the solution of the Poincaré conjecture in all dimensions (Perelman, 2002; Freedman, 1982; Smale, 1961). By duality, it follows that ∂N is a homotopy $(n-1)$-sphere and so by the Poincaré conjecture, ∂N is an $(n-1)$-sphere. Another application of the Poincaré conjecture shows that N is an n-ball.

Next, the bisecting planes of the binary n-cube meet the n-ball N in neighborhoods of reductions. Hence the assumption that each reduction is contractible enables us to conclude that these intersections are also PL hyperplanes in N. Therefore the PL-hyperplane arrangement has been constructed which represents Γ. It is easy to see that this arrangement is indeed d-contractible, since strong contractibility implies that all multiple reductions are contractible and so intersections of subfamilies of PL hyperplanes are either empty or are contractible and hence planes, by the same argument as the previous paragraph. (Note that such intersections correspond to multiple reductions of Γ.)

Finally to show that *ii* implies *i*, by Theorem 39, a d-contractible PL-hyperplane arrangement $\mathcal{P}$ has a peeling sequence and so the corresponding one-inclusion graph Γ is contractible. This follows since a corner-peeling move can be viewed as a homotopy retraction. But then reductions and multiple reductions are also represented by d'-contractible hyperplane arrangements, since these correspond to the restriction of $\mathcal{P}$ to the intersection of a finite subfamily of hyperplanes of $\mathcal{P}$. It is straightforward to check that these new arrangements are d'-contractible, completing the proof. ■

Remark 46 *Note that any one-inclusion graph Γ which satisfies the hypotheses of Theorem 45 admits a corner-peeling sequence. From the proof above, Γ must be contractible if it has a peeling sequence. However Γ does not have to be strongly contractible. A simple example can be found in the binary 3-cube, with coordinate directions x,y,z. Define Γ to be the union of four edges, labeled x,y,z,x. It is easy to see that Γ has a peeling sequence and is contractible but not strongly contractible. For the bisecting hyperplane transverse to the x direction meets Γ in two points, so the reduction Γ^x is a pair of vertices, which is not contractible.*

Note that all maximum classes are strongly contractible, as are also all linear and hyperbolic arrangements, by Corollary 42 and Theorem 45.

12. Future Directions: Compression Schemes for Maximal Classes

In this section, we compare two maximal classes of VC-dimension 2 in the binary 4-cube. For the first, we show that the one-inclusion graph is not contractible and therefore there is no peeling or corner-peeling scheme. There is an unlabeled compression scheme, but this is not associated with either peeling or a hyperplane arrangement. For the second, the one-inclusion graph is contractible but not strongly contractible. However there are simple corner-peeling schemes and a related compression scheme. Note that the relation between the compression scheme and the corner-peeling scheme is not as straightforward as in our main result above. Finally for the second example, there is a non simple hyperplane arrangement consisting of lines in the hyperbolic plane which represents the class. It would be interesting to know if there are many maximal classes which admit such non simple representations and if there is a general procedure to find associated compression schemes.

Example 22 *Let $\mathcal{C}$ be the maximal class of VC-dimension 2 in the 4-cube with concepts and labels shown in Figure 18(a). This forms an unlabeled compression scheme. Note that the one-inclusion graph is not connected, consisting of four 2-cubes with common vertex at the origin 0000 and an isolated vertex at 1111. So since a contractible complex is connected, the one-inclusion graph*

Concept	**Label**
0000	$\emptyset$
1000	x_1
0100	x_2
0010	x_3
0001	x_4
1100	x_1x_2
0011	x_3x_4
0110	x_2x_3
1001	x_1x_4
1111	$x_1x_3\,,x_2x_4$

(a)

Concept	**Label**
0000	$\emptyset$
1000	x_1
0100	x_2
0010	x_3
1100	x_1x_2
0110	x_2x_3
1010	x_1x_3
1011	x_2x_4
1101	x_3x_4
0111	x_1x_4

(b)

Figure 18: VC-2 maximal classes from (a) Example 22 and (b) Example 23.

cannot be contractible. Moreover any hyperplane arrangement represents a connected complex so there cannot be such an arrangement for this example. This example is the same class (up to flipping coordinate labels) as in Kuzmin and Warmuth (2007, Table 2) but there appear to be some errors there in describing the compression scheme.

Example 23 *Let C be the maximal class of VC-dimension* 2 *in the* 4*-cube with concepts and labels defined in Figure 18(b). The class is enlarged by adding an extra vertex 1111 x_4 to complete the labeling.*

This forms an unlabeled compression scheme and is the same as in Kuzmin and Warmuth (2007, Table 1). The one-inclusion graph is contractible, consisting of three 2*-cubes with common vertex 0100 and three edges attached to these* 2*-cubes. It is easy to form a hyperbolic-line arrangement consisting of three lines meeting in three points forming a triangle and three further lines near the boundary of the hyperbolic plane which do not meet any other line.*

It is easy to see that there is a corner-peeling sequence, but there is not such an obvious way of using this to form a compression scheme. The idea is that the label x_1x_4 comes from picking the origin at 0000 and considering the shortest path to the origin as giving the label. There are numerous ways of corner peeling this one-inclusion complex. The only other comment is that the final vertices 0111, 1011, 1101 and 1111 are labeled in a different manner. Namely putting the origin at 0000 means that 0111 has shortest path with label $x_2x_3x_4$. We replace this by the label x_1x_4 since clearly this satisfies the no-clashing condition. Then the final vertex 1111 has the remaining label x_4 to uniquely specify it.

13. Conclusions and Open Problems

We saw in Corollary 23 that d-maximum classes represented by simple linear-hyperplane arrangements in $\mathbb{R}^d$ have underlying cubical complexes that are homeomorphic to a d-ball. Hence the VC dimension and the dimension of the cubical complex are the same. Moreover in Theorem 33, we proved that d-maximum classes represented by PL-hyperplane arrangements in $\mathbb{R}^d$ are those whose underlying cubical complexes are manifolds or equivalently d-balls.

Question 47 *Does every simple PL-hyperplane arrangement in B^d, where every subcollection of d planes transversely meet in a point, represent the same concept class as some simple linear-hyperplane arrangement?*

Question 48 *What is the connection between the VC dimension of a maximum class induced by a simple hyperbolic-hyperplane arrangement and the smallest dimension of hyperbolic space containing such an arrangement? In particular, can the hyperbolic space dimension be chosen to only depend on the VC dimension and not the dimension of the binary cube containing the class?*

We gave an example of a 2-maximum class in the 5-cube that cannot be realized as a hyperbolic-hyperplane arrangement in $\mathcal{H}^3$. Note that the Whitney embedding theorem (Rourke and Sanderson, 1982) proves that any cubical complex of dimension d embeds in $\mathbb{R}^{2d}$. Can such an embedding be used to construct a hyperbolic arrangement in $\mathcal{H}^{2d}$ or a PL arrangement in $\mathbb{R}^{2d}$?

The structure of the boundary of a maximum class is strongly related to corner peeling. For Euclidean-hyperplane arrangements, the boundary of the corresponding maximum class is homeomorphic to a sphere by Corollaries 22 and 23.

Question 49 *Is there a characterization of the cubical complexes that can occur as the boundary of a maximum class? Characterize maximum classes with isomorphic boundaries.*

Question 50 *Does a corner-peeling scheme exist with corner vertex sequence having minimum degree?*

Theorem 32 suggests the following.

Question 51 *Can any d-maximum class in $\{0,1\}^n$ be represented by a simple arrangement of hyperplanes in $\mathbb{H}^n$?*

Question 52 *Which compression schemes arise from sweeping across simple hyperbolic-hyperplane arrangements?*

Kuzmin and Warmuth (2007) note that there are unlabeled compression schemes that are cyclic. In Proposition 17 we show that corner-peeling compression schemes (like min-peeling) are acyclic. So compression schemes arising from sweeping across simple arrangements of hyperplanes in Euclidean or hyperbolic space are also acyclic. Does acyclicity characterize such compression schemes?

We have established peeling of all finite maximum and a family of infinite maximum classes by representing them as PL-hyperplane arrangements and sweeping by multiple generic hyperplanes. A larger class of arrangements has these properties—namely those which are d-contractible—and we have shown that the corresponding one-inclusion graphs are precisely the strongly contractible ones. Finally we have established that there are d-maximal classes that cannot be embedded in any $(d+k)$-maximum classes for any constant k. Some important open problems along these lines are the following.

Question 53 *Prove peeling of maximum classes using purely combinatorial arguments*

Question 54 *Can all maximal classes be peeled by representing them by hyperplane arrangements and then using a sweeping technique (potentially solving the Sample Compressibility conjecture)? The obvious candidate for this approach is to use d-contractible PL-hyperplane arrangements.*

Question 55 *What about more general collections of infinite maximum classes, or infinite arrangements?*

Question 56 *Is it true that any d-contractible PL-hyperplane arrangement is equivalent to a hyperbolic-hyperplane arrangement?*

Question 57 *Is it true that all strongly contractible classes, with largest dimension d of cubes can be embedded in maximum classes of VC-dimension d?*

Acknowledgments

We thank Peter Bartlett for his very helpful feedback, and gratefully acknowledge the support of the NSF through grants DMS-0434383 and DMS-0707060, and the support of the Siebel Scholars Foundation.

References

N. Alon. On the density of sets of vectors. *Discrete Mathematics*, 46(2):199–202, 1983.

S. Ben-David and A. Litman. Combinatorial variability of Vapnik-Chervonenkis classes with applications to sample compression schemes. *Discrete Applied Mathematics*, 86(1):3–25, 1998.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

R. M. Dudley. The structure of some Vapnik-Chervonenkis classes. In L.M. Le Cam and R.A. Olshen, editors, *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman*, volume II, pages 495–507. Wadsworth, 1985.

H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.

S. Floyd. Space-bounded learning and the Vapnik-Chervonenkis dimension. Technical Report TR-89-061, ICSI, UC Berkeley, 1989.

P. Frankl. On the trace of finite sets. *Journal of Combinatorial Theory (A)*, 34(1):41–45, 1983.

M. Freedman. The topology of four-dimensional manifolds. *Journal of Differential Geometry*, 17: 357–454, 1982.

B. Gärtner and E. Welzl. Vapnik-Chervonenkis dimension and (pseudo-) hyperplane arrangements. *Discrete and Computational Geometry*, 12:399–432, 1994.

D. Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory (A)*, 69:217–232, 1995.

D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$ functions on randomly drawn points. *Information and Computation*, 115(2):284–293, 1994.

L. Hellerstein, 28 June 2006. pers. comm.

D. Helmbold, R. Sloan, and M. K. Warmuth. Learning integer lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

D. Kuzmin and M. K. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8(Sep):2047–2081, 2007.

N. Littlestone and M. K. Warmuth. Relating data compression and learnability. Unpublished manuscript `http://www.cse.ucsc.edu/~manfred/pubs/lrnk-olivier.pdf`, 1986.

M. Marchand and J. Shawe-Taylor. The decision list machine. In *Advances in Neural Information Processing Systems 15*, pages 921–928, 2003.

B. Mazur. A note on some contractible 4-manifolds. *Annals of Mathematics*, 73:221–228, 1961.

T. Neylon. *Sparse Solutions for Linear Prediction Problems*. PhD thesis, NYU, 2006.

U. Pachner. Konstruktionsmethoden und das kombinatorische Homöomorphieproblem für Triangulationen kompakter semilinearer Mannigfaltigkeiten. (german) [Construction methods and the combinatorial homeomorphism problem for triangulations of compact semilinear manifolds]. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 57:69–86, 1987.

G. Perelman. Finite extinction time for the solutions to the Ricci flow on certain 3-manifolds, 2002. `http://arXiv.org/math.DG/0211159v1`.

J. G. Ratcliffe. *Foundations of Hyperbolic Manifolds*. Springer-Verlag, 1994.

C. Rourke and B. Sanderson. *Introduction to Piecewise-Linear Topology*. Springer-Verlag, 1982.

B. I. P. Rubinstein and J. H. Rubinstein. Geometric & topological representations of maximum classes with applications to sample compression. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT'08)*, pages 299–310, 2008.

B. I. P. Rubinstein, P. L. Bartlett, and J. H. Rubinstein. Shifting, one-inclusion mistake bounds and tight multiclass expected risk bounds. In *Advances in Neural Information Processing Systems 19*, pages 1193–1200, 2007.

B. I. P. Rubinstein, P. L. Bartlett, and J. H. Rubinstein. Shifting: one-inclusion mistake bounds and sample compression. *Journal of Computer and System Sciences: Special Issue on Learning Theory 2006*, 75(1):37–59, 2009.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13:145–147, 1972.

S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

S. Smale. Generalized Poincaré conjecture in dimensions greater than four. *Annals of Mathematics*, 74:391–406, 1961.

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

U. von Luxburg, O. Bousquet, and B. Schölkopf. A compression approach to support vector model selection. *Journal of Machine Learning Research*, 5:293–323, 2004.

M. K. Warmuth. Compressing to VC dimension many points. In *Proceedings of the 16th Annual Conference on Learning Theory*, 2003.

E. Welzl. Complete range spaces. Unpublished notes, 1987.