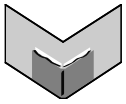
The Journal of Machine Learning Research
Volume 14
Print-Archive Edition

Pages 1–1284



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 14
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 2013.

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

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

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

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

JMLR Editorial Board

Journal of Machine Learning Research

Volume 14, 2013

Journal of Machine Learning Research 14 (2013) 1-37 Submitted 6/12; Revised 11/12; Published 1/13

Global Analytic Solution of Fully-observed Variational Bayesian Matrix Factorization*

Shinichi Nakajima NAKAJIMA.S@NIKON.CO.JP
Optical Research Laboratory
Nikon Corporation
Tokyo 140-8601, Japan

Masashi Sugiyama SUGI@CS.TITECH.AC.JP
Department of Computer Science
Tokyo Institute of Technology
Tokyo 152-8552, Japan

S. Derin Babacan DBABACAN@ILLINOIS.EDU
Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA

Ryota Tomioka TOMIOKA@MIST.I.U-TOKYO.AC.JP
Department of Mathematical Informatics
The University of Tokyo
Tokyo 113-8656, Japan

Editor: Manfred Opper

Abstract

The variational Bayesian (VB) approximation is known to be a promising approach to Bayesian estimation, when the rigorous calculation of the Bayes posterior is intractable. The VB approximation has been successfully applied to *matrix factorization* (MF), offering automatic dimensionality selection for principal component analysis. Generally, finding the VB solution is a non-convex problem, and most methods rely on a local search algorithm derived through a standard procedure for the VB approximation. In this paper, we show that a better option is available for fully-observed VBMF—the global solution can be *analytically* computed. More specifically, the global solution is a reweighted SVD of the observed matrix, and each weight can be obtained by solving a quartic equation with its coefficients being functions of the observed singular value. We further show that the global optimal solution of *empirical* VBMF (where hyperparameters are also learned from data) can also be analytically computed. We illustrate the usefulness of our results through experiments in multi-variate analysis.

Keywords: variational Bayes, matrix factorization, empirical Bayes, model-induced regularization, probabilistic PCA

1. Introduction

The problem of finding a low-rank approximation of a target matrix through *matrix factorization* (MF) recently attracted considerable attention. In this paper, we consider *fully-observed MF* where

*. This paper is a combined and extended version of our earlier conference papers (Nakajima et al., 2010, 2011).

the observed matrix has no missing entry.[1] This formulation includes multivariate analysis techniques such as *principal component analysis* (Hotelling, 1933) and *reduced rank regression* (Reinsel and Velu, 1998). *Canonical correlation analysis* (Hotelling, 1936; Anderson, 1984; Hardoon et al., 2004) and *partial least-squares* (Worsley et al., 1997; Rosipal and Krämer, 2006) are also closely related to MF.

Singular value decomposition (SVD) is a classical method for MF, which gives the optimal low-rank approximation to the target matrix in terms of the squared error. Regularized variants of SVD have been studied for the *Frobenius-norm* penalty (i.e., singular values are regularized by the ℓ_2-penalty) (Paterek, 2007) or the *trace-norm* penalty (i.e., singular values are regularized by the ℓ_1-penalty) (Srebro et al., 2005). Since the Frobenius-norm penalty does not automatically produce a low-rank solution, it should be combined with an explicit low-rank constraint, which is non-convex. In contrast, the trace-norm penalty tends to produce sparse solutions, so a low-rank solution can be obtained without explicit rank constraints. This implies that the optimization problem of trace-norm MF is still convex, and thus the global optimal solution can be obtained. Recently, optimization techniques for trace-norm MF have been extensively studied (Rennie and Srebro, 2005; Cai et al., 2010; Ji and Ye, 2009; Tomioka et al., 2010).

Bayesian approaches to MF have also been actively explored. A *maximum a posteriori* (MAP) estimation, which computes the mode of the posterior distributions, was shown (Srebro et al., 2005) to be equivalent to the ℓ_1-MF when Gaussian priors are imposed on factorized matrices (Salakhutdinov and Mnih, 2008). The *variational Bayesian* (VB) method (Attias, 1999; Bishop, 2006), which approximates the posterior distributions by decomposable distributions, has also been applied to MF (Bishop, 1999; Lim and Teh, 2007; Ilin and Raiko, 2010). The VB-based MF method (VBMF) was shown to perform well in experiments, and its theoretical properties have been investigated (Nakajima and Sugiyama, 2011).

However, the optimization problem of VBMF is non-convex. In practice, the VBMF solution is computed by the *iterated conditional modes* (ICM) algorithm (Besag, 1986; Bishop, 2006), where the mean and the covariance of the posterior distributions are iteratively updated until convergence (Lim and Teh, 2007; Ilin and Raiko, 2010). One may obtain a local optimal solution by the ICM algorithm, but many restarts would be necessary to find a good local optimum.

In this paper, we show that, despite the non-convexity of the optimization problem, the global optimal solution of VBMF can be *analytically* computed. More specifically, the global solution is a reweighted SVD of the observed matrix, and each weight can be obtained by solving a quartic equation with its coefficients being functions of the observed singular value. This is highly advantageous over the standard ICM algorithm since the global optimum can be found without any iterations and restarts. We also consider an *empirical* VB scenario where the hyperparameters (prior variances) are also learned from data. Again, the optimization problem of empirical VBMF is non-convex, but we show that the global optimal solution of empirical VBMF can still be analytically computed. The usefulness of our results is demonstrated through experiments.

Our analysis can be seen as an extension of Nakajima and Sugiyama (2011). The major progress is twofold:

1. Weakened decomposability assumption.

1. This excludes the *collaborative filtering* setup, which is aimed at imputing missing entries of an observed matrix (Konstan et al., 1997; Funk, 2006).

Nakajima and Sugiyama (2011) analyzed the behavior of VBMF under the *column-wise* independence assumption (Ilin and Raiko, 2010), that is, the columns of the factorized matrices are forced to be mutually independent in the VB posterior. This was one of the limitations of the previous work, since the weaker *matrix-wise* independence assumption (Lim and Teh, 2007) is rather standard, and sufficient to derive the ICM algorithm. It was not clear how these different assumptions affect the approximation accuracy to the Bayes posterior. In this paper, we show that the VB solution under the matrix-wise independence assumption is column-wise independent, meaning that the stronger column-wise independence assumption does not degrade the quality of approximation accuracy.

2. Exact analysis for rectangular cases.

   Nakajima and Sugiyama (2011) derived bounds of the VBMF solution (more specifically, bounds of the weights for the reweighed SVD). Those bounds are tight enough to give the exact analytic solution only when the observed matrix is square. In this paper, we conduct a more precise analysis, which results in a quartic equation with its coefficients depending on the observed singular value. Satisfying this quartic equation is a necessary condition for the weight, and further consideration specifies which of the four solutions is the VBMF solution.

In summary, we derive the exact global analytic solution for general rectangular cases under the standard matrix-wise independence assumption.

The rest of this paper is organized as follows. We first introduce the framework of Bayesian matrix factorization and the variational Bayesian approximation in Section 2. Then, we analyze the VB free energy, and derive the global analytic solution in Section 3. Section 4 is devoted to explaining the relation between MF and multivariate analysis techniques. In Section 5, we show practical usefulness of our analytic-form solutions through experiments. In Section 6, we derive simple analytic-form solutions for special cases, discuss the relation between model pruning and spontaneous symmetry breaking, and consider the possibility of extending our results to more general problems. Finally, we conclude in Section 7.

2. Formulation

In this section, we first formulate the problem of probabilistic MF (Section 2.1). Then, we introduce the VB approximation (Section 2.2) and its empirical variant (Section 2.3). We also introduce a simplified variant (Section 2.4), which was analyzed in Nakajima and Sugiyama (2011) and will be shown to be equivalent to the (non-simple) VB approximation in the subsequent section.

2.1 Probabilistic Matrix Factorization

Assume that we have an observation matrix $V \in \mathbb{R}^{L\times M}$, which is the sum of a target matrix $U \in \mathbb{R}^{L\times M}$ and a noise matrix $\mathcal{E} \in \mathbb{R}^{L\times M}$:

$$V = U + \mathcal{E}.$$

In the *matrix factorization* model, the target matrix is assumed to be low rank, and expressed in the following factorized form:

$$U = BA^\top,$$

where $A \in \mathbb{R}^{M\times H}$ and $B \in \mathbb{R}^{L\times H}$. Here, $\top$ denotes the transpose of a matrix or vector. Thus, the rank of U is upper-bounded by $H \leq \min(L,M)$.

We consider the Gaussian probabilistic MF model (Salakhutdinov and Mnih, 2008), given as follows:

$$p(V|A,B) \propto \exp\left(-\frac{1}{2\sigma^2}\|V - BA^\top\|_{\mathrm{Fro}}^2\right), \tag{1}$$

$$p(A) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(AC_A^{-1}A^\top\right)\right), \tag{2}$$

$$p(B) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(BC_B^{-1}B^\top\right)\right), \tag{3}$$

where σ^2 is the noise variance. Here, we denote by $\|\cdot\|_{\mathrm{Fro}}$ the Frobenius norm, and by $\mathrm{tr}(\cdot)$ the trace of a matrix. We assume that $L \leq M$. If $L > M$, we may simply re-define the transpose $V^\top$ as V so that $L \leq M$ holds. Thus this does not impose any restriction. We assume that the prior covariance matrices C_A and C_B are diagonal and positive definite, that is,

$$C_A = \mathrm{diag}(c_{a_1}^2, \ldots, c_{a_H}^2),$$
$$C_B = \mathrm{diag}(c_{b_1}^2, \ldots, c_{b_H}^2),$$

for $c_{a_h}, c_{b_h} > 0, h = 1, \ldots, H$. Without loss of generality, we assume that the diagonal entries of the product $C_A C_B$ are arranged in the non-increasing order, that is, $c_{a_h}c_{b_h} \geq c_{a_{h'}}c_{b_{h'}}$ for any pair $h < h'$.

Throughout the paper, we denote a column vector of a matrix by a bold small letter, and a row vector by a bold small letter with a tilde, namely,

$$A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_H) = (\widetilde{\boldsymbol{a}}_1, \ldots, \widetilde{\boldsymbol{a}}_M)^\top \in \mathbb{R}^{M\times H},$$
$$B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_H) = \left(\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_L\right)^\top \in \mathbb{R}^{L\times H}.$$

2.2 Variational Bayesian Approximation

The Bayes posterior is written as

$$p(A,B|V) = \frac{p(V|A,B)p(A)p(B)}{p(V)}, \tag{4}$$

where $p(V) = \langle p(V|A,B)\rangle_{p(A)p(B)}$ is the marginal likelihood. Here, $\langle\cdot\rangle_p$ denotes the expectation over the distribution p. Since the Bayes posterior (4) is computationally intractable, the VB approximation was proposed (Bishop, 1999; Lim and Teh, 2007; Ilin and Raiko, 2010).

Let $r(A,B)$, or r for short, be a trial distribution. The following functional with respect to r is called the free energy:

$$\begin{aligned} F(r|V) &= \left\langle \log \frac{r(A,B)}{p(V|A,B)p(A)p(B)} \right\rangle_{r(A,B)} \\ &= \left\langle \log \frac{r(A,B)}{p(A,B|V)} \right\rangle_{r(A,B)} - \log p(V). \end{aligned} \tag{5}$$

The first term in Equation (5) is the Kullback-Leibler (KL) distance from the trial distribution to the Bayes posterior, and the second term is a constant. Therefore, minimizing the free energy (5) amounts to finding the distribution closest to the Bayes posterior in the sense of the KL distance. In the VB approximation, the free energy (5) is minimized over some restricted function space.

A standard constraint for the MF model is *matrix-wise* independence (Bishop, 1999; Lim and Teh, 2007), that is,

$$r^{\mathrm{VB}}(A,B) = r_{\mathrm{A}}^{\mathrm{VB}}(A) r_{\mathrm{B}}^{\mathrm{VB}}(B). \tag{6}$$

This constraint breaks the entanglement between the parameter matrices A and B, and leads to a computationally-tractable iterative algorithm, called the *iterated conditional modes* (ICM) algorithm (Besag, 1986; Bishop, 2006). The resulting distribution is called the *VB posterior*.

Using the variational method, we can show that the VB posterior minimizing the free energy (5) under the constraint (6) can be written as

$$r^{\mathrm{VB}}(A,B) = \prod_{m=1}^{M} \mathcal{N}_H(\widetilde{a}_m; \widetilde{\widehat{a}}_m, \Sigma_A) \prod_{l=1}^{L} \mathcal{N}_H(\widetilde{b}_l; \widetilde{\widehat{b}}_l, \Sigma_B), \tag{7}$$

where the parameters satisfy

$$\widehat{A} = \left(\widetilde{\widehat{a}}_1, \ldots, \widetilde{\widehat{a}}_M\right)^\top = V^\top \widehat{B} \frac{\Sigma_A}{\sigma^2}, \tag{8}$$

$$\widehat{B} = \left(\widetilde{\widehat{b}}_1, \ldots, \widetilde{\widehat{b}}_L\right)^\top = V \widehat{A} \frac{\Sigma_B}{\sigma^2}, \tag{9}$$

$$\Sigma_A = \sigma^2 \left(\widehat{B}^\top \widehat{B} + L\Sigma_B + \sigma^2 C_A^{-1}\right)^{-1}, \tag{10}$$

$$\Sigma_B = \sigma^2 \left(\widehat{A}^\top \widehat{A} + M\Sigma_A + \sigma^2 C_B^{-1}\right)^{-1}. \tag{11}$$

Here, $\mathcal{N}_d(\cdot; \boldsymbol{\mu}, \Sigma)$ denotes the d-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. Note that, in the VB posterior (7), the rows $\{\widetilde{a}_m\}$ ($\{\widetilde{b}_l\}$) of A (B) are independent of each other, and share a common covariance Σ_A (Σ_B) (Bishop, 1999).

The ICM for VBMF iteratively updates the parameters $(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B)$ by Equations (8)–(11) until convergence, allowing one to obtain a local minimum of the free energy (5). Finally, the VB estimator of U is computed as

$$\widehat{U}^{\mathrm{VB}} = \widehat{B}\widehat{A}^\top.$$

2.3 Empirical VB Approximation

The free energy minimization principle also allows us to estimate the hyperparameters C_A and C_B from data. This is called the *empirical* Bayesian scenario. In this scenario, C_A and C_B are updated in each iteration by the following formulas:

$$c_{a_h}^2 = \|\widehat{a}_h\|^2 / M + (\Sigma_A)_{hh}, \tag{12}$$

$$c_{b_h}^2 = \|\widehat{b}_h\|^2 / L + (\Sigma_B)_{hh}. \tag{13}$$

When the noise variance σ^2 is unknown, it can also be estimated based on the free energy minimization. The update rule for σ^2 is given by

$$\sigma^2 = \frac{\|V\|_{\mathrm{Fro}}^2 - \mathrm{tr}(2V^\top \widehat{B}\widehat{A}^\top) + \mathrm{tr}\left((\widehat{A}^\top \widehat{A} + M\Sigma_A)(\widehat{B}^\top \widehat{B} + L\Sigma_B)\right)}{LM}, \tag{14}$$

which should be applied in each iteration of the ICM algorithm.

2.4 SimpleVB Approximation

A simplified variant, called the SimpleVB approximation, assumes *column-wise* independence of each matrix (Ilin and Raiko, 2010; Nakajima and Sugiyama, 2011), that is,

$$r^{\mathrm{SVB}}(A,B) = \prod_{h=1}^{H} r_{a_h}^{\mathrm{SVB}}(\boldsymbol{a}_h) \prod_{h=1}^{H} r_{b_h}^{\mathrm{SVB}}(\boldsymbol{b}_h). \tag{15}$$

Note that the *column-wise* independence constraint (15) is stronger than the *matrix-wise* independence constraint (6), that is, any column-wise independent distribution is matrix-wise independent.

The SimpleVB posterior can be written as

$$r^{\mathrm{SVB}}(A,B) = \prod_{h=1}^{H} \mathcal{N}_M(\boldsymbol{a}_h; \widehat{\boldsymbol{a}}_h^{\mathrm{SVB}}, \sigma_{a_h}^{2\,\mathrm{SVB}} I_M) \prod_{h=1}^{H} \mathcal{N}_L(\boldsymbol{b}_h; \widehat{\boldsymbol{b}}_h^{\mathrm{SVB}}, \sigma_{b_h}^{2\,\mathrm{SVB}} I_L),$$

where the parameters satisfy

$$\widehat{\boldsymbol{a}}_h^{\mathrm{SVB}} = \frac{\sigma_{a_h}^{2\,\mathrm{SVB}}}{\sigma^2} \left(V - \sum_{h' \neq h} \widehat{\boldsymbol{b}}_{h'}^{\mathrm{SVB}} \widehat{\boldsymbol{a}}_{h'}^{\mathrm{SVB}\top} \right)^\top \widehat{\boldsymbol{b}}_h^{\mathrm{SVB}}, \tag{16}$$

$$\widehat{\boldsymbol{b}}_h^{\mathrm{SVB}} = \frac{\sigma_{b_h}^{2\,\mathrm{SVB}}}{\sigma^2} \left(V - \sum_{h' \neq h} \widehat{\boldsymbol{b}}_{h'}^{\mathrm{SVB}} \widehat{\boldsymbol{a}}_{h'}^{\mathrm{SVB}\top} \right) \widehat{\boldsymbol{a}}_h^{\mathrm{SVB}}, \tag{17}$$

$$\sigma_{a_h}^{2\,\mathrm{SVB}} = \sigma^2 \left(\|\widehat{\boldsymbol{b}}_h^{\mathrm{SVB}}\|^2 + L\sigma_{b_h}^{2\,\mathrm{SVB}} + \sigma^2 c_{a_h}^{-2} \right)^{-1}, \tag{18}$$

$$\sigma_{b_h}^{2\,\mathrm{SVB}} = \sigma^2 \left(\|\widehat{\boldsymbol{a}}_h^{\mathrm{SVB}}\|^2 + M\sigma_{a_h}^{2\,\mathrm{SVB}} + \sigma^2 c_{b_h}^{-2} \right)^{-1}. \tag{19}$$

Here, I_d denotes the d-dimensional identity matrix. The constraint (15) restricts the covariances Σ_A and Σ_B in Equation (7) to be diagonal, and thus reduces necessary memory storage and computational cost (Ilin and Raiko, 2010).

Iterating Equations (16)–(19) until convergence, we can obtain a local minimum of the free energy. Equations (14), (12), and (13) are similarly applied if the noise variance σ^2 is unknown and in the empirical Bayesian scenario, respectively.

The column-wise independence (15) also simplifies theoretical analysis. Thanks to this simplification, Nakajima and Sugiyama (2011) showed that the SimpleVBMF solution is a reweighted SVD, and successfully derived theoretical bounds of the weights. Their analysis revealed interesting properties of VBMF, called *model-induced regularization*. However, it has not been clear how restrictive the column-wise independence assumption is. In Section 3, we theoretically show that the column-wise independence assumption has actually no effect, before deriving the exact global analytic solution.

3. Theoretical Analysis

In this section, we first prove the equivalence between VBMF and SimpleVBMF (Section 3.1). After that, starting from a proposition given in Nakajima and Sugiyama (2011), we derive the global analytic solution for VBMF (Section 3.2). Finally, we derive the global analytic solution for the empirical VBMF (Section 3.3).

3.1 Equivalence between VBMF and SimpleVBMF

Under the *matrix-wise* independence constraint (6), the free energy (5) can be written as

$$\begin{aligned} F^{\mathrm{VB}} &= \langle \log r_A(A) + \log r_B(B) - \log p(V|A,B)p(A)p(B) \rangle_{r(A)r(B)} \\ &= \frac{\|V\|_{\mathrm{Fro}}^2}{2\sigma^2} + \frac{LM}{2}\log\sigma^2 + \frac{M}{2}\log\frac{|C_A|}{|\Sigma_A|} + \frac{L}{2}\log\frac{|C_B|}{|\Sigma_B|} \\ &\quad + \frac{1}{2}\mathrm{tr}\left\{ C_A^{-1}\left(\widehat{A}^\top\widehat{A} + M\Sigma_A\right) + C_B^{-1}\left(\widehat{B}^\top\widehat{B} + L\Sigma_B\right) \right. \\ &\quad \left. + \sigma^{-2}\left(-2\widehat{A}^\top V^\top \widehat{B} + \left(\widehat{A}^\top\widehat{A} + M\Sigma_A\right)\left(\widehat{B}^\top\widehat{B} + L\Sigma_B\right)\right)\right\} + \mathrm{const.}, \end{aligned} \tag{20}$$

where $|\cdot|$ denotes the determinant of a matrix. Note that Equations (8)–(11) together form the stationarity condition of Equation (20) with respect to $(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B)$.

We say that two points $(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B)$ and $(\widehat{A}', \widehat{B}', \Sigma_A', \Sigma_B')$ are *equivalent* if both give the same free energy and $\widehat{B}\widehat{A}^\top = \widehat{B}'\widehat{A}'^\top$ holds. We obtain the following theorem (its proof is given in Appendix A):

Theorem 1 *When $C_A C_B$ is non-degenerate (i.e., $c_{a_h} c_{b_h} > c_{a_{h'}} c_{b_{h'}}$ for any pair $h < h'$), any solution minimizing the free energy* (20) *has diagonal Σ_A and Σ_B. When $C_A C_B$ is degenerate, any solution has an* equivalent *solution with diagonal Σ_A and Σ_B.*

The result that Σ_A and Σ_B become diagonal would be natural because we assumed the independent Gaussian priors on A and B: the fact that any V can be decomposed into orthogonal singular components may imply that the observation V cannot convey any preference for singular-component-wise correlation. Note, however, that Theorem 1 does not necessarily hold when the observed matrix has missing entries.

Obviously, any VBMF solution (minimizer of the free energy (20)) with diagonal covariances is a SimpleVBMF solution (minimizer of the free energy (20) under the constraint that the covariances are diagonal). Theorem 1 states that, if $C_A C_B$ is non-degenerate, the set of VBMF solutions and the set of SimpleVBMF solutions are identical. In the case when $C_A C_B$ is degenerate, the set of VBMF solutions is the union of the set of SimpleVBMF solutions and the set of their *equivalent* solutions with non-diagonal covariances. Actually, any VBMF solution can be obtained by rotating its *equivalent* SimpleVBMF solution (VBMF solution with diagonal covariances) (see Appendix A.4). In practice, it is however sufficient to focus on the SimpleVBMF solutions, since *equivalent* solutions share the same free energy F^{VB} and the same mean prediction $\widehat{B}\widehat{A}^\top$. In this sense, we can conclude that the stronger *column-wise* independence constraint (15) does not degrade approximation accuracy, and the VBMF solution under the *matrix-wise* independence (6) *essentially* agrees with the SimpleVBMF solution.

Since we have shown the equivalence between VBMF and SimpleVBMF, we can use the results obtained in Nakajima and Sugiyama (2011), where SimpleVBMF was analyzed, for pursuing the global analytic solution for (non-simple) VBMF.

3.2 Global Analytic Solution for VBMF

Here, we derive an analytic-form expression of the VBMF global solution. We denote by $\mathbb{R}^d_{++}$ the set of the d-dimensional vectors with positive elements, and by $\mathbb{S}^d_{++}$ the set of $d \times d$ symmetric positive-definite matrices. We solve the following problem:

$$\begin{aligned} \text{Given} \quad & (c_{a_h}^2, c_{b_h}^2) \in \mathbb{R}^2_{++} \ (\forall h = 1, \ldots, H), \ \sigma^2 \in \mathbb{R}_{++}, \\ \min \quad & F^{\mathrm{VB}}(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B) \\ \text{s.t.} \quad & \widehat{A} \in \mathbb{R}^{M \times H}, \ \widehat{B} \in \mathbb{R}^{L \times H}, \ \Sigma_A \in \mathbb{S}^H_{++}, \ \Sigma_B \in \mathbb{S}^H_{++}, \end{aligned}$$

where $F^{\mathrm{VB}}(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B)$ is the free energy given by Equation (20). This is a non-convex optimization problem, but we show that the global optimal solution can still be analytically obtained.

We start from the following proposition, which is obtained by summarizing Lemma 11, Lemma 13, Lemma 14, Lemma 15, and Lemma 17 in Nakajima and Sugiyama (2011):

Proposition 2 *(Nakajima and Sugiyama, 2011) Let γ_h (≥ 0) be the h-th largest singular value of V, and let $\boldsymbol{\omega}_{a_h}$ and $\boldsymbol{\omega}_{b_h}$ be the associated right and left singular vectors:*

$$V = \sum_{h=1}^{L} \gamma_h \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top.$$

Then, the global SimpleVB solution (under the column-wise independence (15)*) can be expressed as*

$$\widehat{U}^{\mathrm{SVB}} \equiv \langle BA^\top \rangle_{r^{\mathrm{SVB}}(A,B)} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{SVB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top.$$

Let

$$\widetilde{\gamma}_h = \sqrt{\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2} + \sqrt{\left(\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM\sigma^4}}.$$

When

$$\gamma_h \leq \widetilde{\gamma}_h,$$

the SimpleVB solution for the h-th component is $\widehat{\gamma}_h^{\mathrm{SVB}} = 0$. When

$$\gamma_h > \widetilde{\gamma}_h, \tag{21}$$

$\widehat{\gamma}_h^{\mathrm{SVB}}$ is given as a positive real solution of

$$\widehat{\gamma}_h^2 + q_1(\widehat{\gamma}_h) \cdot \widehat{\gamma}_h + q_0 = 0, \tag{22}$$

where

$$\begin{aligned} q_1(\widehat{\gamma}_h) &= \frac{-(M-L)^2(\gamma_h - \widehat{\gamma}_h) + (L+M)\sqrt{(M-L)^2(\gamma_h - \widehat{\gamma}_h)^2 + \frac{4\sigma^4 LM}{c_{a_h}^2 c_{b_h}^2}}}{2LM}, \\ q_0 &= \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} - \left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h^2. \end{aligned}$$

When Inequality (21) *holds, Equation* (22) *has only one positive real solution, which lies in*

$$0 < \widehat{\gamma}_h < \gamma_h.$$

In Nakajima and Sugiyama (2011), it was shown that any SimpleVBMF solution is a stationary point, and Equation (22) was derived from the stationarity condition (16)–(19). Bounds of $\widehat{\gamma}_h^{\mathrm{SVB}}$ were obtained by approximating Equation (22) with a quadratic equation (more specifically, by bounding $q_1(\widehat{\gamma}_h)$ by constants). This analysis revealed interesting properties of VBMF, including the model-induced regularization effect and the sparsity induction mechanism. Thanks to Theorem 1, almost the same statements as Proposition 2 hold for VBMF (Lemma 8 in Appendix B).

In this paper, our purpose is to obtain the exact solution, and therefore, we should treat Equation (22) more precisely. If $q_1(\widehat{\gamma}_h)$ were a constant, Equation (22) would be quadratic with respect to $\widehat{\gamma}_h$, and its solutions could be easily obtained. However, Equation (22) is not even polynomial, because $q_1(\widehat{\gamma}_h)$ depends on the square root of $\widehat{\gamma}_h$. With some algebra, we can convert Equation (22) to a quartic equation, which has four solutions in general. By examining which solution corresponds to the positive solution of Equation (22), we obtain the following theorem (the proof is given in Appendix B):

Theorem 3 *Let* $\widehat{\gamma}_h^{\mathrm{second}}$ *be the* second *largest real solution of the following quartic equation with respect to* $\widehat{\gamma}_h$*:*

$$f(\widehat{\gamma}_h) := \widehat{\gamma}_h^4 + \xi_3\widehat{\gamma}_h^3 + \xi_2\widehat{\gamma}_h^2 + \xi_1\widehat{\gamma}_h + \xi_0 = 0, \tag{23}$$

where the coefficients are defined by

$$\begin{aligned}
\xi_3 &= \frac{(L-M)^2\gamma_h}{LM}, \\
\xi_2 &= -\left(\xi_3\gamma_h + \frac{(L^2+M^2)\eta_h^2}{LM} + \frac{2\sigma^4}{c_{a_h}^2 c_{b_h}^2}\right), \\
\xi_1 &= \xi_3\sqrt{\xi_0}, \\
\xi_0 &= \left(\eta_h^2 - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}\right)^2, \\
\eta_h^2 &= \left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h^2.
\end{aligned}$$

Then, the global VB solution can be expressed as

$$\widehat{U}^{\mathrm{VB}} \equiv \langle BA^\top \rangle_{r^{\mathrm{VB}}(A,B)} = \widehat{B}\widehat{A}^\top = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{VB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top,$$

where

$$\widehat{\gamma}_h^{\mathrm{VB}} = \begin{cases} \widehat{\gamma}_h^{\mathrm{second}} & \text{if } \gamma_h > \widetilde{\gamma}_h, \\ 0 & \text{otherwise.} \end{cases}$$

The coefficients of the quartic equation (23) are analytic, so $\widehat{\gamma}_h^{\text{second}}$ can also be obtained analytically, for example, by *Ferrari's method* (Hazewinkel, 2002).[2] Therefore, the global VB solution can be analytically computed.[3] This is a strong advantage over the standard ICM algorithm since many iterations and restarts would be necessary to find a good solution by ICM.

Based on the above result, the complete VB posterior can be obtained analytically as follows (the proof is also given in Appendix B):

Theorem 4 *The VB posterior is given by*

$$r^{\text{VB}}(A,B) = \prod_{h=1}^{H} \mathcal{N}_M(\boldsymbol{a}_h; \widehat{\boldsymbol{a}}_h, \sigma_{a_h}^2 I_M) \prod_{h=1}^{H} \mathcal{N}_L(\boldsymbol{b}_h; \widehat{\boldsymbol{b}}_h, \sigma_{b_h}^2 I_L),$$

where, for $\widehat{\gamma}_h^{\text{VB}}$ *being the solution given by Theorem 3,*

$$\widehat{\boldsymbol{a}}_h = \pm\sqrt{\widehat{\gamma}_h^{\text{VB}}\widehat{\delta}_h}\cdot\boldsymbol{\omega}_{a_h},$$

$$\widehat{\boldsymbol{b}}_h = \pm\sqrt{\widehat{\gamma}_h^{\text{VB}}\widehat{\delta}_h^{-1}}\cdot\boldsymbol{\omega}_{b_h},$$

$$\sigma_{a_h}^2 = \frac{-\left(\widehat{\eta}_h^2 - \sigma^2(M-L)\right) + \sqrt{(\widehat{\eta}_h^2 - \sigma^2(M-L))^2 + 4M\sigma^2\widehat{\eta}_h^2}}{2M(\widehat{\gamma}_h^{\text{VB}}\widehat{\delta}_h^{-1} + \sigma^2 c_{a_h}^{-2})},$$

$$\sigma_{b_h}^2 = \frac{-\left(\widehat{\eta}_h^2 + \sigma^2(M-L)\right) + \sqrt{(\widehat{\eta}_h^2 + \sigma^2(M-L))^2 + 4L\sigma^2\widehat{\eta}_h^2}}{2L(\widehat{\gamma}_h^{\text{VB}}\widehat{\delta}_h + \sigma^2 c_{b_h}^{-2})},$$

$$\widehat{\delta}_h = \frac{(M-L)(\gamma_h - \widehat{\gamma}_h^{\text{VB}}) + \sqrt{(M-L)^2(\gamma_h - \widehat{\gamma}_h^{\text{VB}})^2 + \frac{4\sigma^4 LM}{c_{a_h}^2 c_{b_h}^2}}}{2\sigma^2 M c_{a_h}^{-2}},$$

$$\widehat{\eta}_h^2 = \begin{cases} \eta_h^2 & \text{if } \gamma_h > \widetilde{\gamma}_h, \\ \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} & \text{otherwise.} \end{cases}$$

3.3 Global Analytic Solution for Empirical VBMF

Solving the following problem gives the empirical VBMF solution:

$$\begin{aligned} \text{Given} \quad & \sigma^2 \in \mathbb{R}_{++}, \\ \min \quad & F^{\text{VB}}(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B, \{c_{a_h}^2, c_{b_h}^2; h = 1, \ldots, H\}), \\ \text{s.t.} \quad & \widehat{A} \in \mathbb{R}^{M\times H},\ \widehat{B} \in \mathbb{R}^{L\times H},\ \Sigma_A \in \mathbb{S}_{++}^H,\ \Sigma_B \in \mathbb{S}_{++}^H, \\ & (c_{a_h}^2, c_{b_h}^2) \in \mathbb{R}_{++}^2\ (\forall h = 1, \ldots, H), \end{aligned}$$

where $F^{\text{VB}}(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B, \{c_{a_h}^2, c_{b_h}^2; h = 1, \ldots, H\})$ is the free energy given by Equation (20). Although this is again a non-convex optimization problem, the global optimal solution can be obtained analytically. As discussed in Nakajima and Sugiyama (2011), the ratio c_{a_h}/c_{b_h} is arbitrary in empirical VB. Accordingly, we fix the ratio to $c_{a_h}/c_{b_h} = 1$ without loss of generality.

2. In practice, one may solve the quartic equation numerically, for example, by the '`roots`' function in MATLAB®.
3. In our latest work on performance analysis of VBMF, we have derived a simpler-form solution, which does not require to solve a quartic equation (Nakajima et al., 2012b).

Nakajima and Sugiyama (2011) obtained a closed form solution of the optimal hyperparameter value $\widehat{c}_{a_h}\widehat{c}_{b_h}$ for SimpleVBMF. Therefore, we can easily obtain the global analytic solution for empirical VBMF. We have the following theorem (the proof is given in Appendix C):

Theorem 5 *The global empirical VB solution is given by*

$$\widehat{U}^{\mathrm{EVB}} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{EVB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^{\top},$$

where

$$\widehat{\gamma}_h^{\mathrm{EVB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{VB}} & \text{if } \gamma_h > \underline{\gamma}_h \text{ and } \Delta_h \leq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here,

$$\underline{\gamma}_h = (\sqrt{L}+\sqrt{M})\sigma, \tag{24}$$

$$\breve{c}_{a_h}^2 \breve{c}_{b_h}^2 = \frac{1}{2LM}\left(\gamma_h^2 - (L+M)\sigma^2 + \sqrt{\left(\gamma_h^2 - (L+M)\sigma^2\right)^2 - 4LM\sigma^4}\right), \tag{25}$$

$$\Delta_h = M\log\left(\frac{\gamma_h}{M\sigma^2}\breve{\gamma}_h^{\mathrm{VB}}+1\right) + L\log\left(\frac{\gamma_h}{L\sigma^2}\breve{\gamma}_h^{\mathrm{VB}}+1\right) + \frac{1}{\sigma^2}\left(-2\gamma_h\breve{\gamma}_h^{\mathrm{VB}} + LM\breve{c}_{a_h}^2\breve{c}_{b_h}^2\right), \tag{26}$$

and $\breve{\gamma}_h^{\mathrm{VB}}$ is the VB solution for $c_{a_h}c_{b_h} = \breve{c}_{a_h}\breve{c}_{b_h}$.

By using Theorem 3 and Theorem 5, the global empirical VB solution can be computed analytically. This is again a strong advantage over the standard ICM algorithm since ICM would require many iterations and restarts to find a good local minimum. The calculation procedure for the empirical VB solution is as follows: After obtaining $\{\gamma_h\}$ by singular value decomposition of V, we first check if $\gamma_h > \underline{\gamma}_h$ holds for each h, by using Equation (24). If it holds, we compute $\breve{\gamma}_h^{\mathrm{VB}}$ by using Equation (25) and Theorem 3. Otherwise, $\widehat{\gamma}_h^{\mathrm{EVB}} = 0$. Finally, we check if $\Delta_h \leq 0$ holds by using Equation (26).

When the noise variance σ^2 is unknown, it may be estimated by minimizing the VB free energy with respect to σ^2. In practice, this single-parameter minimization may be carried out numerically based on Equation (20) and Theorem 4.

4. Matrix Factorization for Multivariate Analysis

In this section, we explicitly describe the relation between MF and multivariate analysis techniques.

4.1 Probabilistic PCA

The relation to principal component analysis (PCA) (Hotelling, 1933) is straightforward. In probabilistic PCA (Tipping and Bishop, 1999), the observation $\boldsymbol{v} \in \mathbb{R}^L$ is assumed to be driven by a latent vector $\widetilde{\boldsymbol{a}} \in \mathbb{R}^H$ in the following form:

$$\boldsymbol{v} = B\widetilde{\boldsymbol{a}} + \boldsymbol{\varepsilon}.$$

Here, $B \in \mathbb{R}^{L\times H}$ specifies the linear relationship between $\widetilde{\boldsymbol{a}}$ and $\boldsymbol{v}$, and $\boldsymbol{\varepsilon} \in \mathbb{R}^L$ is a Gaussian noise subject to $\mathcal{N}_L(\boldsymbol{0}, \sigma^2 I_L)$. Suppose that we are given M observed samples $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M)$ generated

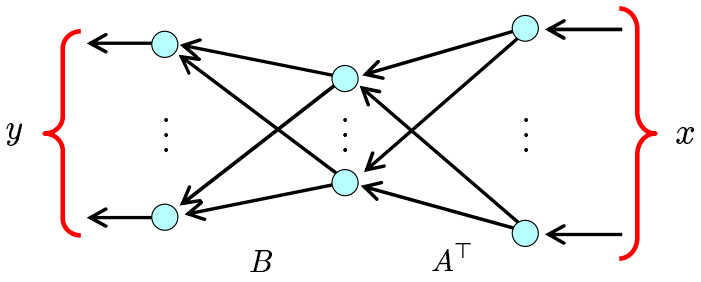


Figure 1: Linear neural network.

from the latent vectors $A^\top = (\widetilde{\boldsymbol{a}}_1, \ldots, \widetilde{\boldsymbol{a}}_M)$, and each latent vector is subject to $\widetilde{\boldsymbol{a}} \sim \mathcal{N}_H(\boldsymbol{0}, I_H)$. Then, the probabilistic PCA model is written as Equations (1) and (2) with $C_A = I_H$.

If we apply Bayesian inference, the intrinsic dimension H is automatically selected without predetermination (Bishop, 1999). This useful property is called *automatic dimensionality selection* (ADS). It was shown that ADS originates from the *model-induced regularization* effect (Nakajima and Sugiyama, 2011).

4.2 Reduced Rank Regression

Reduced rank regression (RRR) (Baldi and Hornik, 1995; Reinsel and Velu, 1998) is aimed at learning a relation between an input vector $\boldsymbol{x} \in \mathbb{R}^M$ and an output vector $\boldsymbol{y} \in \mathbb{R}^L$ by using the following linear model:

$$\boldsymbol{y} = BA^\top \boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{27}$$

where $A \in \mathbb{R}^{M \times H}$ and $B \in \mathbb{R}^{L \times H}$ are parameter matrices, and $\boldsymbol{\varepsilon} \sim \mathcal{N}_L(\boldsymbol{0}, \sigma'^2 I_L)$ is a Gaussian noise vector. This can be expressed as a linear neural network (Figure 1). Thus, we can interpret this model as first projecting the input vector $\boldsymbol{x}$ onto a lower-dimensional latent subspace by $A^\top$ and then performing linear prediction by B.

Suppose we are given n pairs of input and output vectors:

$$\mathcal{V}^n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid \boldsymbol{x}_i \in \mathbb{R}^M, \boldsymbol{y}_i \in \mathbb{R}^L, i = 1, \ldots, n\}. \tag{28}$$

Then, the likelihood of the RRR model (27) is expressed as

$$p(\mathcal{V}^n | A, B) \propto \exp\left(-\frac{1}{2\sigma'^2} \sum_{i=1}^n \|\boldsymbol{y}_i - BA^\top \boldsymbol{x}_i\|^2\right). \tag{29}$$

Let us assume that the samples are centered:

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0} \ \text{ and } \ \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i = \boldsymbol{0}.$$

Furthermore, let us assume that the input samples are *pre-whitened* (Hyvärinen et al., 2001), that is, they satisfy

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top = I_M.$$

Let

$$V = \Sigma_{XY} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{x}_i^\top \tag{30}$$

be the sample *cross-covariance* matrix, and

$$\sigma^2 = \frac{\sigma'^2}{n} \tag{31}$$

be a rescaled noise variance. Then the likelihood (29) can be written as

$$p(\mathcal{V}^n|A,B) \propto \exp\left(-\frac{1}{2\sigma^2}\|V - BA^\top\|_{\mathrm{Fro}}^2\right)\exp\left(-\frac{1}{2\sigma^2}\left(\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{y}_i\|^2 - \|V\|_{\mathrm{Fro}}^2\right)\right). \tag{32}$$

The first factor in Equation (32) coincides with the likelihood of the MF model (1), and the second factor is constant with respect to A and B. Thus, RRR is reduced to MF.

However, the second factor depends on the rescaled noise variance σ^2, and therefore, should be considered when σ^2 is estimated based on the free energy minimization principle. Furthermore, the normalization constant of the likelihood (29) is slightly different from that of the MF model. Taking into account of these differences, the VB free energy of the RRR model (29) with the priors (2) and (3) is given by

$$\begin{aligned}
F^{\mathrm{VB-RRR}} &= \left\langle \log r_A(A) + \log r_B(B) - \log p(\mathcal{V}^n|A,B)p(A)p(B)\right\rangle_{r(A)r(B)} \\
&= \frac{\sum_{i=1}^{n}\|\boldsymbol{y}_i\|^2}{2n\sigma^2} + \frac{nL}{2}\log\sigma^2 + \frac{M}{2}\log\frac{|C_A|}{|\Sigma_A|} + \frac{L}{2}\log\frac{|C_B|}{|\Sigma_B|} \\
&\quad + \frac{1}{2}\mathrm{tr}\left\{C_A^{-1}\left(\widehat{A}^\top\widehat{A} + M\Sigma_A\right) + C_B^{-1}\left(\widehat{B}^\top\widehat{B} + L\Sigma_B\right)\right. \\
&\qquad \left. + \sigma^{-2}\left(-2\widehat{A}^\top V^\top\widehat{B} + \left(\widehat{A}^\top\widehat{A} + M\Sigma_A\right)\left(\widehat{B}^\top\widehat{B} + L\Sigma_B\right)\right)\right\} + \mathrm{const.}
\end{aligned} \tag{33}$$

Note that the difference from Equation (20) exists only in the first two terms. Minimizing Equation (33), instead of Equation (20), gives an estimator for the rescaled noise variance. For the standard ICM algorithm, the following update rule should be substituted for Equation (14):

$$(\sigma^2)^{\mathrm{RRR}} = \frac{n^{-1}\sum_{i=1}^{n}\|\boldsymbol{y}_i\|^2 - \mathrm{tr}(2V^\top\widehat{B}\widehat{A}^\top) + \mathrm{tr}\left((\widehat{A}^\top\widehat{A} + M\Sigma_A)(\widehat{B}^\top\widehat{B} + L\Sigma_B)\right)}{nL}. \tag{34}$$

Once the rescaled noise variance σ^2 is estimated, Equation (31) gives the original noise variance σ'^2 of the RRR model (29).

4.3 Partial Least-Squares

Partial least-squares (PLS) (Worsley et al., 1997; Rosipal and Krämer, 2006) is similar to RRR. In PLS, the parameters A and B are learned so that the squared Frobenius norm of the difference from the sample *cross-covariance matrix* (30) is minimized:

$$(A,B) := \operatorname*{argmin}_{A,B}\|\Sigma_{XY} - BA^\top\|_{\mathrm{Fro}}^2. \tag{35}$$

Clearly, PLS can be seen as the maximum likelihood estimation of the MF model (1).

4.4 Canonical Correlation Analysis

For paired samples (28), the goal of *canonical correlation analysis* (CCA) (Hotelling, 1936; Anderson, 1984) is to seek vectors $\boldsymbol{a} \in \mathbb{R}^M$ and $\boldsymbol{b} \in \mathbb{R}^L$ such that the correlation between $\boldsymbol{a}^\top \boldsymbol{x}$ and $\boldsymbol{b}^\top \boldsymbol{y}$ is maximized. $\boldsymbol{a}$ and $\boldsymbol{b}$ are called *canonical vectors*.

More formally, given the first $(h-1)$ canonical vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{h-1}$ and $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{h-1}$, the h-th canonical vectors are defined as

$$(\boldsymbol{a}_h, \boldsymbol{b}_h) := \operatorname*{argmax}_{\boldsymbol{a},\boldsymbol{b}} \frac{\boldsymbol{a}^\top \Sigma_{XY} \boldsymbol{b}}{\sqrt{\boldsymbol{a}^\top \Sigma_{XX} \boldsymbol{a}} \sqrt{\boldsymbol{b}^\top \Sigma_{YY} \boldsymbol{b}}},$$
$$\text{s.t. } \boldsymbol{a}^\top \Sigma_{XX} \boldsymbol{a}_{h'} = 0 \text{ and } \boldsymbol{b}^\top \Sigma_{YY} \boldsymbol{b}_{h'} = 0 \text{ for } h' = 1, \ldots, h-1,$$

where Σ_{XX} and Σ_{YY} are the sample covariance matrices of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and Σ_{XY} is the sample cross-covariance matrix, defined in Equation (30), of $\boldsymbol{x}$ and $\boldsymbol{y}$. The entire solution $A = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_H)$ and $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_H)$ are given as the H largest singular vectors of $\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}$.

Let us assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are both pre-whitened, that is, $\Sigma_{XX} = I_M$ and $\Sigma_{YY} = I_L$. Then the solutions A and B are given as the singular vectors of Σ_{XY} associated with the H largest singular values. Since the solutions of Equation (35) are also given by the H dominant singular vectors of Σ_{XY} (Stewart, 1993), CCA is reduced to the maximum likelihood estimation of the MF model (1).

5. Experimental Results

In this section, we show experimental results on artificial and benchmark data sets, which illustrate practical usefulness of our analytic solution.

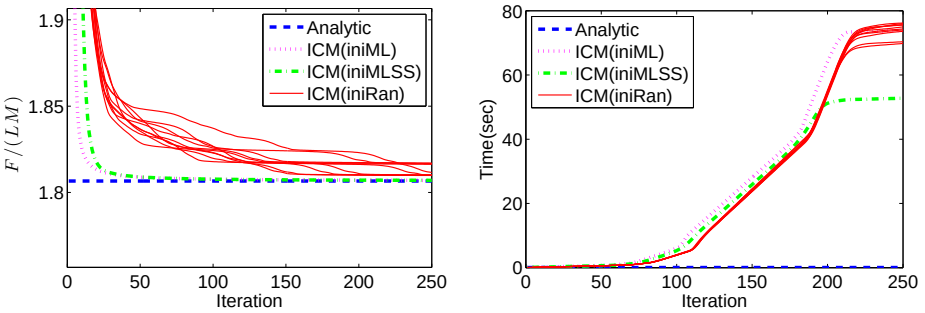
5.1 Experiment on Artificial Data

We compare the standard *ICM* algorithm and the *analytic* solution in the *empirical* VB scenario with unknown noise variance, that is, the hyperparameters (C_A, C_B) and the noise variance σ^2 are also estimated from observation. We use the full-rank model (i.e., $H = \min(L, M)$), and expect the ADS effect to automatically find the true rank H^*.

Figure 2 shows the free energy, the computation time, and the estimated rank over iterations for an artificial (*Artificial1*) data set with the data matrix size $L = 100$ and $M = 300$, and the true rank $H^* = 20$. We randomly created *true* matrices $A^* \in \mathbb{R}^{M \times H^*}$ and $B^* \in \mathbb{R}^{L \times H^*}$ so that each entry of A^* and B^* follows $\mathcal{N}_1(0,1)$. An observed matrix V was created by adding a noise subject to $\mathcal{N}_1(0,1)$ to each entry of $B^* A^{*\top}$.

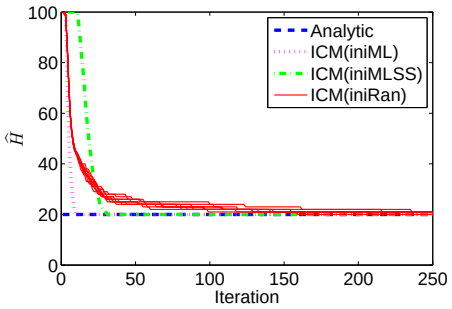
The standard ICM algorithm consists of the update rules (8)–(14). Initial values were set in the following way: $\widehat{A}$ and $\widehat{B}$ are randomly created so that each entry follows $\mathcal{N}_1(0,1)$. Other variables are set to $\Sigma_A = \Sigma_B = C_A = C_B = I_H$ and $\sigma^2 = 1$. Note that we rescale V so that $\|V\|_{\mathrm{Fro}}^2/(LM) = 1$, before starting iterations. We ran the standard ICM algorithm 10 times, starting from different initial points, and each trial is plotted by a solid line (labeled as 'ICM(iniRan)') in Figure 2. The analytic solution consists of applying Theorem 5 combined with a naive 1-dimensional search for the estimation of noise variance σ^2. The analytic solution is plotted by the dashed line (labeled as 'Analytic'). We see that the analytic solution estimates the true rank $\widehat{H} = H^* = 20$ immediately (~ 0.1 sec on average over 10 trials), while the ICM algorithm does not converge in 60 sec.

Figure 3 shows experimental results on another artificial data set (*Artificial2*) where $L = 70$, $M = 300$, and $H^* = 40$. In this case, all the 10 trials of the ICM algorithm are trapped at local



(a) Free energy

(b) Computation time



(c) Estimated rank

Figure 2: Experimental results for *Artificial1* data set, where the data matrix size is $L = 100$ and $M = 300$, and the true rank is $H^* = 20$.

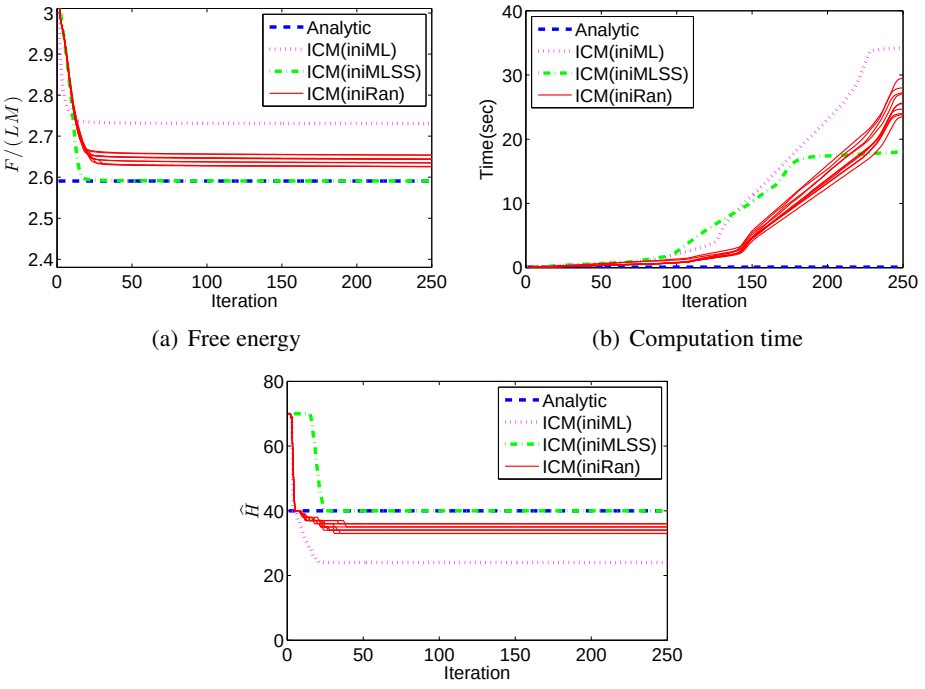
minima. We empirically observed that the local minima problem tends to be more critical, when H^* is large (close to H).

We also evaluated the ICM algorithm with different initialization schemes. The line labeled as 'ICM(iniML)' indicates the ICM algorithm starting from the maximum likelihood (ML) solution: $(\widehat{\boldsymbol{a}}_h, \widehat{\boldsymbol{b}}_h) = (\sqrt{\gamma_h}\boldsymbol{\omega}_{a_h}, \sqrt{\gamma_h}\boldsymbol{\omega}_{b_h})$. The initial values for other variables are the same as the random initialization. Figures 2 and 3 show that the ML initialization generally makes convergence faster than the random initialization, but suffers from the local minima problem more often.

We observed that starting from a small noise variance tends to alleviate the local minima problem at the expense of slightly slower convergence. The line labeled as 'ICM(iniMLSS)' indicates the ICM algorithm starting from the ML solution with a small noise variance $\sigma^2 = 0.0001$. We see in Figures 2 and 3 that this initialization improves quality of solutions, and successfully finds the true rank for these artificial data sets. However, we will show in Section 5.2 that this scheme still suffers from the local minima problem on benchmark data sets.

5.2 Experiment on Benchmark Data

Figures 4–6 show the PCA results on the *Glass*, the *Satimage*, and the *Spectf* data sets available from the UCI repository (Asuncion and Newman, 2007). Similar tendency to the artificial data experiment (Figures 2 and 3) is observed: 'ICM(iniRan)' converges slowly, and is often trapped



(a) Free energy

(b) Computation time

(c) Estimated rank

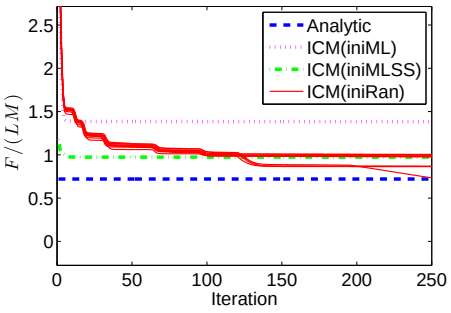
Figure 3: Experimental results for *Artificial2* data set ($L = 70$, $M = 300$, and $H^* = 40$).

at local minima with *wrong* ranks;[4] 'ICM(iniML)' converges slightly faster but to worse local minima; 'ICM(iniMLSS)' tends to give better solutions. Unlike in the artificial data experiment, 'ICM(iniMLSS)' fails to find the *correct* rank with these benchmark data sets. We also conducted experiments on other benchmark data sets, and found that the ICM algorithm generally converges slowly, and sometimes suffers from the local minima problem, while our analytic-form gives the global solution immediately.

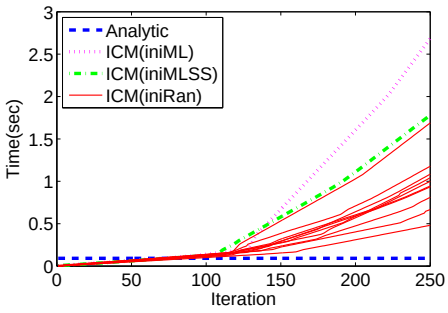
Finally, we applied VBMF to a *reduced rank regression* (RRR) (Reinsel and Velu, 1998) task, and show the results in Figure 7. We centered the $L = 3$-dimensional outputs and the $M = 7$-dimensional inputs of the *Concrete Slump Test* data set, and pre-whitened the inputs. We also standardized the outputs so that the variance of each element is equal to one. Note that we have to minimize Equation (33), instead of Equation (20), for estimating the noise variance in our proposed method with the analytic solution, and use Equation (34), instead of Equation (14), for updating the noise variance in the standard ICM algorithm.

Overall, the proposed global analytic solution is shown to be a useful alternative to the standard ICM algorithm.

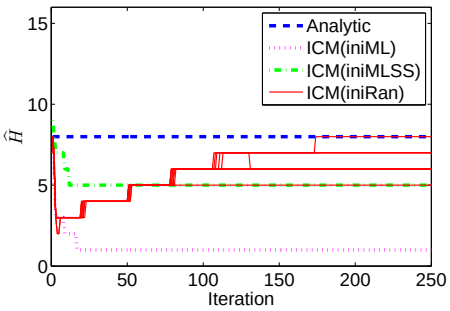
4. Since the *true* ranks of the benchmark data sets are unknown, we mean by a *wrong* rank a rank different from the one given by the global 'Analytic' solution.



(a) Free energy



(b) Computation time



(c) Estimated rank

Figure 4: PCA results for the *Glass* data set ($L = 9, M = 214$).

6. Discussion

In this section, we first derive simple analytic-form solutions for special cases, where the *model-induced regularization* and the *prior-induced* regularization can be clearly distinguished (Section 6.1). Then, we discuss the relation between model pruning by VB and spontaneous symmetry breaking (Section 6.2). Finally, we consider possibilities of extending our results to more general cases (Section 6.3).

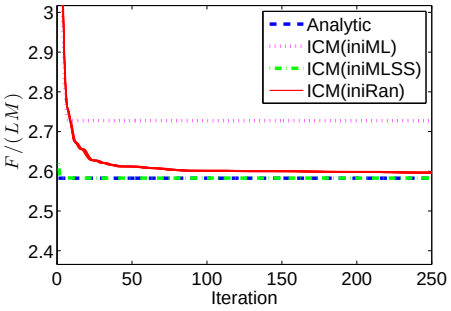
6.1 Special Cases

Here, we consider two special cases, where simple-form solutions are obtained.

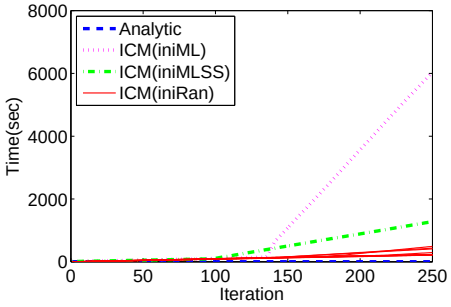
6.1.1 FLAT PRIOR

When $c_{a_h} c_{b_h} \to \infty$ (i.e., the prior is *almost* flat), a simple-form exact solution for SimpleVBMF has been obtained in Nakajima and Sugiyama (2011). Thanks to Theorem 1, the same applies to VBMF under the standard *matrix-wise* independence assumption. This solution can be obtained also by factorizing the quartic equation (23) as follows:

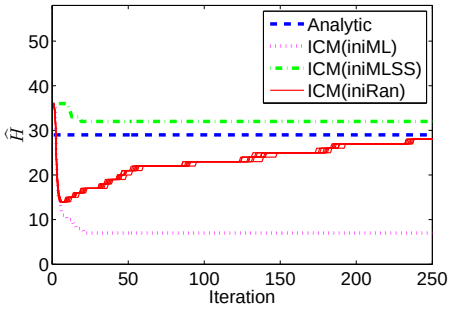
$$\lim_{c_{a_h} c_{b_h} \to \infty} f(\widehat{\gamma}_h) = \left(\widehat{\gamma}_h + \frac{M}{L}\left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\gamma_h\right)\left(\widehat{\gamma}_h + \left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h\right) \cdot \left(\widehat{\gamma}_h - \left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h\right)\left(\widehat{\gamma}_h - \frac{M}{L}\left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\gamma_h\right) = 0.$$



(a) Free energy

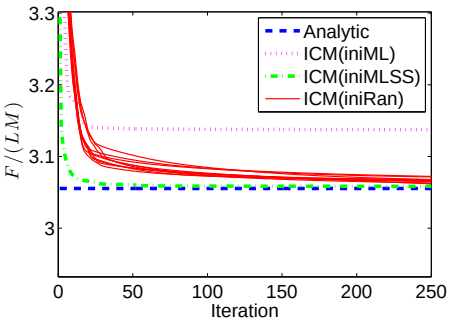


(b) Computation time

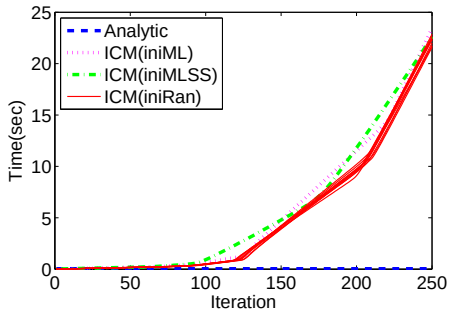


(c) Estimated rank

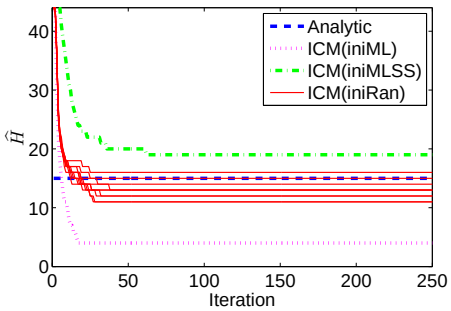
Figure 5: PCA results for the *Satimage* data set ($L = 36, M = 6435$).



(a) Free energy

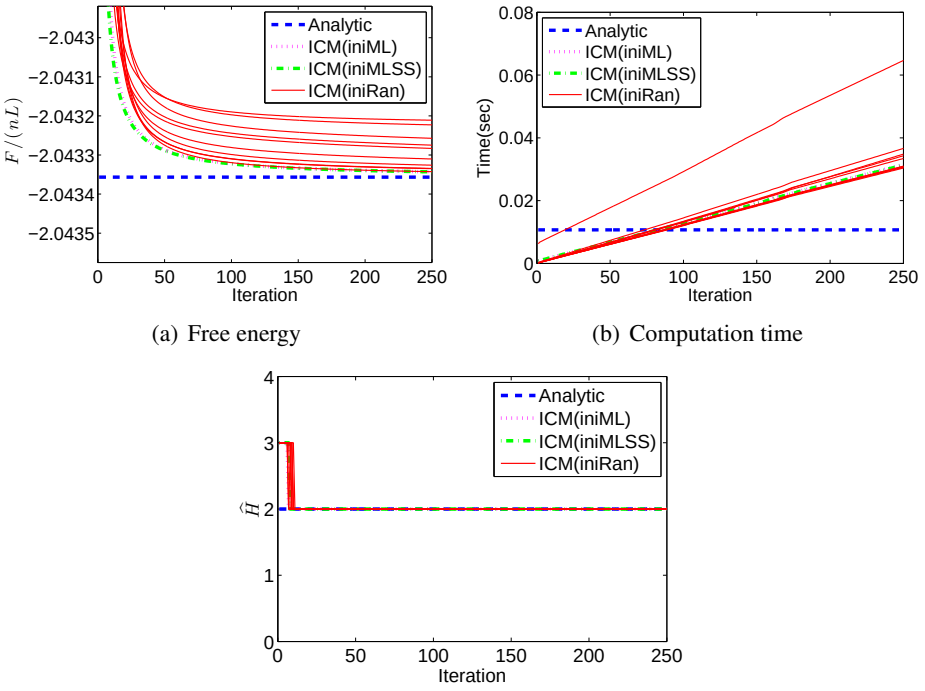


(b) Computation time



(c) Estimated rank

Figure 6: PCA results for the *Spectf* data set ($L = 44, M = 267$).



(a) Free energy

(b) Computation time

(c) Estimated rank

Figure 7: RRR results for the *Concrete Slump Test* data set ($L = 3, M = 7$).

Since Theorem 3 states that its *second* largest solution gives the VB estimator for $\gamma_h > \lim_{c_{a_h} c_{b_h} \to \infty} \widetilde{\gamma}_h = \sqrt{M\sigma^2}$, we have the following corollary:

Corollary 1 *The global VB solution with the almost flat prior (i.e., $c_{a_h} c_{b_h} \to \infty$) is given by*

$$\lim_{c_{a_h} c_{b_h} \to \infty} \widehat{\gamma}_h^{\mathrm{VB}} = \widehat{\gamma}_h^{\mathrm{PJS}} = \begin{cases} \max\left\{0, \left(1 - \dfrac{M\sigma^2}{\gamma_h^2}\right)\gamma_h\right\} & \text{if } \gamma_h > 0, \\ 0 & \text{otherwise}. \end{cases} \tag{36}$$

Equation (36) is the *positive-part James-Stein* (PJS) shrinkage estimator (James and Stein, 1961), operated on each singular component separately. This is actually the upper-bound of the VB solution for arbitrary $c_{a_h} c_{b_h} > 0$. The counter-intuitive fact—a shrinkage is observed even in the limit of flat priors—can be explained by strong non-uniformity of the *volume element of the Fisher metric*, that is, the *Jeffreys* prior (Jeffreys, 1946), in the parameter space. This effect is called *model-induced regularization* (MIR), because it is induced not by priors but by the structure of the model likelihood function (Nakajima and Sugiyama, 2011). MIR was shown to generally appear in Bayesian estimation when the model is *non-identifiable* (i.e., the mapping between parameters and distribution functions is not one-to-one) (Watanabe, 2009). The mechanism how non-identifiability causes MIR and ADS in VBMF was explicitly illustrated in Nakajima and Sugiyama (2011).

6.1.2 SQUARE MATRIX

Also when $L = M$ (i.e., the observation matrix V is square), a simple-form solution can be obtained. Since $\xi_3 = \xi_1 = 0$ (see Theorem 3) in this case, the quartic equation (23) can be solved as a quadratic

equation with respect to $\widehat{\gamma}_h^2$ (Nakajima and Sugiyama, 2011). We can also find the solution by factorizing the quartic equation (23) for $\gamma_h > \sqrt{M\sigma^2}$ as follows:

$$f^{\text{square}}(\widehat{\gamma}_h) = \left(\widehat{\gamma}_h + \widehat{\gamma}_h^{\text{PJS}} + \frac{\sigma^2}{c_{a_h} c_{b_h}}\right)\left(\widehat{\gamma}_h + \widehat{\gamma}_h^{\text{PJS}} - \frac{\sigma^2}{c_{a_h} c_{b_h}}\right) \cdot \left(\widehat{\gamma}_h - \widehat{\gamma}_h^{\text{PJS}} + \frac{\sigma^2}{c_{a_h} c_{b_h}}\right)\left(\widehat{\gamma}_h - \widehat{\gamma}_h^{\text{PJS}} - \frac{\sigma^2}{c_{a_h} c_{b_h}}\right) = 0.$$

Using Theorem 3, we have the following corollary:

Corollary 2 *When $L = M$, the global VB solution is given by*

$$\widehat{\gamma}_h^{\text{VB-square}} = \max\left\{0, \widehat{\gamma}_h^{\text{PJS}} - \frac{\sigma^2}{c_{a_h} c_{b_h}}\right\}. \tag{37}$$

Equation (37) shows that, in this case, MIR and *prior-induced regularization* (PIR) can be completely decomposed—the estimator is equipped with the *model-induced* PJS shrinkage ($\widehat{\gamma}_h^{\text{PJS}}$) and the *prior-induced* trace-norm shrinkage ($-\sigma^2/(c_{a_h} c_{b_h})$).

The empirical VB solution is also simplified in this case. The following corollary is obtained simply by combining Theorem 1 in this paper and Corollary 2 in Nakajima and Sugiyama (2011):

Corollary 3 *When $L = M$, the global empirical VB solution is given by*

$$\widehat{\gamma}_h^{\text{EVB}} = \begin{cases} \left(1 - \dfrac{M\sigma^2}{\gamma_h^2} - \rho_-\right)\gamma_h & \text{if } \gamma_h > \underline{\gamma}_h \text{ and } \Delta'_h \leq 0, \\ 0 & \text{otherwise}, \end{cases}$$

where

$$\underline{\gamma}_h = 2\sqrt{M}\sigma,$$
$$\Delta'_h = \log\left(\frac{\gamma_h^2}{M\sigma^2}(1-\rho_-)\right) - \frac{\gamma_h^2}{M\sigma^2}(1-\rho_-) + \left(1 + \frac{\gamma_h^2}{2M\sigma^2}\rho_+^2\right),$$
$$\rho_\pm = \sqrt{\frac{1}{2}\left(1 - \frac{2M\sigma^2}{\gamma_h^2} \pm \sqrt{1 - \frac{4M\sigma^2}{\gamma_h^2}}\right)}.$$

By using Corollary 2 and Corollary 3, respectively, we can easily compute the VB and the empirical VB solutions in this case without a quartic solver.

6.2 Model Pruning and Spontaneous Symmetry Breaking

Mackay (2001) pointed out that there are cases when VB prunes model components *inappropriately*, giving a toy example of a mixture of Gaussians. There, *appropriateness* is measured in terms of the similarity to the rigorous Bayesian estimation. He plotted the free energy of the mixture of Gaussians as a function of hidden *responsibility* variables—the probabilities that each sample belongs to each Gaussian component—and argued that VB sometimes favors simpler models too much. In this case, degrees of freedom are pruned when spontaneous symmetry breaking occurs.

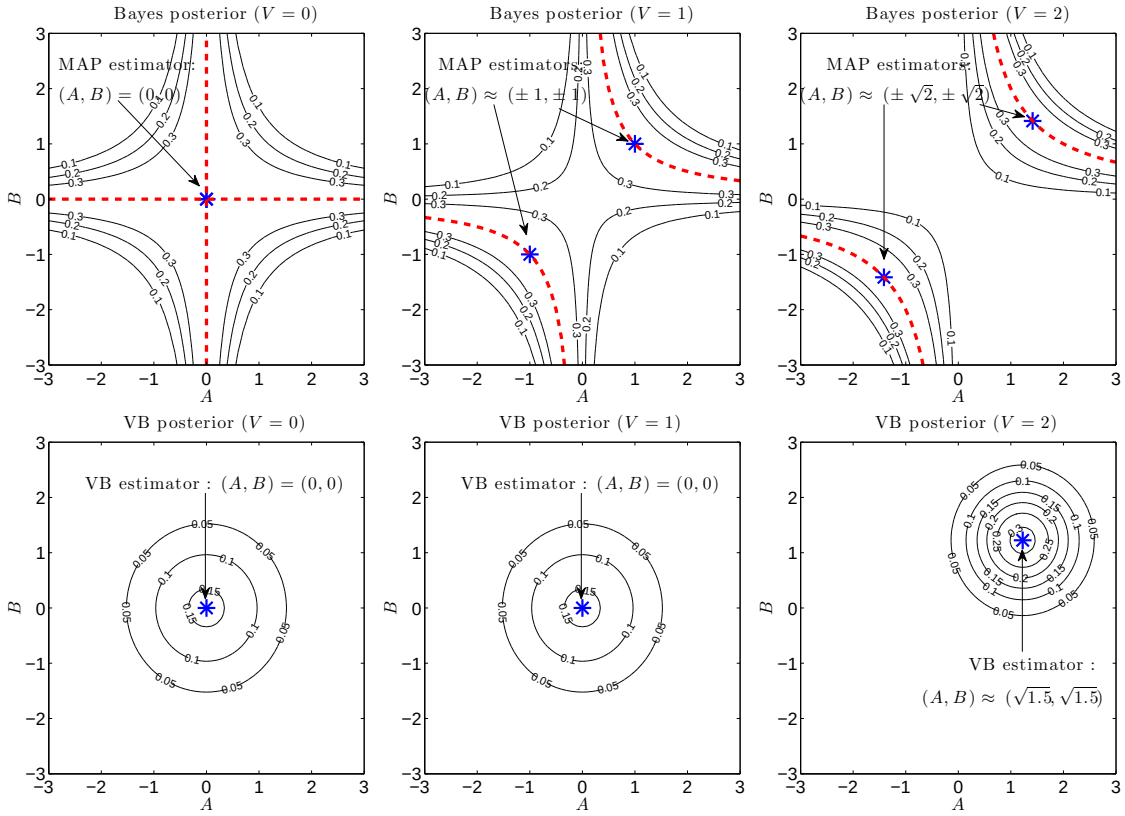


Figure 8: Bayes posteriors (top row) and the VB posteriors (bottom row) of a *scalar factorization* model (i.e., a MF model for $L = M = H = 1$) with $\sigma^2 = 1$ and $c_a = c_b = 100$ (almost flat priors), when the observed values are $V = 0$ (left), $V = 1$ (middle), and $V = 2$ (right), respectively. In the top row, the asterisks indicate the MAP estimators, and the dashed lines the ML estimators (the modes of the contour). In the bottom row, the asterisks indicate the VB estimators. All graphs are quoted from Nakajima and Sugiyama (2011).

In VBMF, degrees of freedom are pruned when spontaneous symmetry breaking does *not* occur. Figure 8 shows the Bayes posteriors (top row) and the VB posteriors (bottom row) of a *scalar factorization* model (i.e., a MF model for $L = M = H = 1$) with $\sigma^2 = 1$ and $c_a = c_b = 100$ (almost flat priors). As we can see in the top row, the Bayes posterior has two modes unless $V = 0$, and the distance between the two modes increases as $|V|$ increases. Since the VB posterior tries to approximate the Bayes posterior with a single uncorrelated distribution, it stays at the origin when $|V|$ is not sufficiently large. When $|V|$ is large enough, the VB posterior approximates one of the modes, as seen in the graphs in the right column (for the case when $V = 2$) of Figure 8 (note that there also exists an *equivalent* VB solution located at $(A,B) \approx (-\sqrt{1.5}, -\sqrt{1.5})$).

Equation (36) implies that symmetry breaking occurs when $V > \widetilde{\gamma}_h \sim \sqrt{M\sigma^2} = 1$, which coincides with the average contribution of noise to the observed singular values over all singular components. In this way, VBMF discards singular components dominated by noise. EVBMF has

a different transition point, and tends to give a sparser solution (see Section 4 in Nakajima and Sugiyama (2011) for further discussion).

Given that the rigorous Bayesian estimator in MF is not sparse (see Figure 10 in Nakajima and Sugiyama, 2011), one might argue that the sparsity of VBMF is *inappropriate*. On the other hand, given that model pruning by VB has been acknowledged as a practically useful property, one might also argue that *appropriateness* should be measured in terms of performance. Motivated by the latter idea, we have conducted performance analysis of EVBMF in our latest work (Nakajima et al., 2012b), and shown that model pruning by EVBMF works perfectly under some condition. Conducting performance analysis in other models would be our future work.

6.3 Extensions

In this paper, we derived the global analytic solution of VBMF, by fully making use of the assumptions that the likelihood and priors are both spherical Gaussian, and that the observed matrix has no missing entry. They were necessary to solve the free energy minimization problem as a reweighted SVD. In this subsection, we discuss possibilities to extend our results to more general problems.

6.3.1 ROBUST PCA

VBMF gives a low-rank solution, which can be seen as a singular-component-wise sparse solution. We can extend our analysis so that a wider variety of sparsity can be handled.

Robust PCA (Candes et al., 2009) has recently gathered a great deal of attention. Equipped with an element-wise sparse term in addition to a low-rank term, robust PCA separates the low dimensional data structure from spiky noise. Its VB variant has also been proposed (Babacan et al., 2012). To obtain the VB solution of robust PCA, we have proposed a novel algorithm where the analytic VBMF solution is applied to partial problems (Nakajima et al., 2012a). Although the global optimality is not guaranteed, this algorithm has been experimentally shown to give a better solution than the standard ICM algorithm. In addition, our proposed algorithm can handle a variety of sparse terms beyond robust PCA.

6.3.2 TENSOR FACTORIZATION

We have shown that the VB solution under *matrix-wise* independence essentially agrees with the SimpleVB solution under *column-wise* independence. We expect that similar *redundancy* would be found also in other models, for example, *tensor factorization* (Kolda and Bader, 2009; Carroll and Chang, 1970; Harshman, 1970; Tucker, 1996). In our preliminary study so far, we saw that the analytic VB solution for tensor factorization is not attainable, at least in the same way as MF. However, we have found that the optimal solution has diagonal covariances for the core tensor in Tucker decomposition (Nakajima, 2012), which would allow us to greatly simplify inference algorithms and reduce necessary memory storage and computational costs.

6.3.3 CORRELATED PRIORS

Our analysis assumed uncorrelated priors. With correlated priors, the posterior is no longer uncorrelated and thus it is not straightforward in general to obtain the global solution from the results obtained in this paper. One exception is the following: Suppose there exists an $H \times H$ non-singular matrix T such that both of $C'_A = TC_AT^\top$ and $C'_B = (T^{-1})^\top C_B T^{-1}$ are diagonal. We can show that

the free energy (20) is invariant under the following transformation for any T:

$$A \to AT^{\top}, \qquad \Sigma_A \to T\Sigma_A T^{\top}, \qquad C_A \to TC_A T^{\top},$$
$$B \to BT^{-1}, \qquad \Sigma_B \to (T^{-1})^T \Sigma_B T^{-1}, \qquad C_B \to (T^{-1})^{\top} C_B T^{-1}.$$

Accordingly, the following procedure gives the global solution analytically: the analytic solution given the diagonal (C'_A, C'_B) is first computed, and the above transformation is then applied.

Handling priors correlated over *rows* of A and B is more challenging and remains as future work. Such a prior allows model correlations in the observation space, and can capture useful characteristics of data, for example, short-term correlation in time-series data and correlation among neighboring pixels in image data.

6.3.4 MISSING ENTRIES PREDICTION

Missing entries prediction is another prototypical application of MF (Konstan et al., 1997; Funk, 2006; Lim and Teh, 2007; Ilin and Raiko, 2010; Salakhutdinov and Mnih, 2008), where finding the global VBMF solution seems a very hard problem. However, one may use our analytic solution as a subroutine, for example, in the *soft-thresholding* step of SOFT-IMPUTE (Mazumder et al., 2010). Along this line, Seeger and Bouchard (2012) have recently proposed an algorithm, which tends to give a better local solution than the standard ICM algorithm for missing entries prediction. They also proposed a way to cope with non-Gaussian likelihood functions.

7. Conclusion

Overcoming the non-convexity of VB methods has been one of the important challenges in the Bayesian machine learning community, since it sometimes prevented us from applying the VB methods to highly complex real-world problems. In this paper, we focused on the matrix factorization (MF) problem with no missing entry, and showed that this weakness could be overcome by *analytically* computing the global optimal solution. We further derived the global optimal solution analytically for the empirical VBMF method, where hyperparameters are also optimized based on data samples. Since no hand-tuning parameter remains in empirical VBMF, our analytic-form solution is practically useful and computationally highly efficient. Numerical experiments showed that the proposed approach is promising.

We discussed the possibility that our analytic solution can be used as a building block of novel algorithms for more general problems. Tackling such possible extensions and conducting performance analysis of those methods are our future work.

Acknowledgments

The authors thank anonymous reviewers for their helpful comments, which significantly improved the paper. Shinichi Nakajima and Masashi Sugiyama thank the support from Grant-in-Aid for Scientific Research on Innovative Areas: Prediction and Decision Making, 23120004. S. Derin Babacan was supported by a Beckman Postdoctoral Fellowship. Ryota Tomioka was supported by Grant-in-Aid for Young Scientists (B) 22700138.

Appendix A. Proof of Theorem 1

In the same way as in the analysis for the SimpleVB approximation (see the proof of Lemma 10 in Nakajima and Sugiyama, 2011), we can show that any minimizer of the free energy (20) is a stationary point. Therefore, Equations (8)–(11) hold for any solution.

We consider the following three cases:

Case 1 When no pair of diagonal entries of $C_A C_B$ coincide.

Case 2 When all diagonal entries of $C_A C_B$ coincide.

Case 3 When (not all but) some pairs of diagonal entries of $C_A C_B$ coincide.

In the following, we prove that, in Case 1, Σ_A and Σ_B are diagonal for any solution $(\widehat{A}, \widehat{B}, \Sigma_A, \Sigma_B)$, and that, in other cases, any solution has its *equivalent* solution with diagonal Σ_A and Σ_B.

Our proof relies on a technique related to the following proposition:

Proposition 6 *(Ruhe, 1970) Let $\lambda_h(\Phi), \lambda_h(\Psi)$ be the h-th largest eigenvalues of positive-definite symmetric matrices $\Phi, \Psi \in \mathbb{R}^{H\times H}$, respectively. Then, it holds that*

$$tr\{\Phi^{-1}\Psi\} \geq \sum_{h=1}^{H} \frac{\lambda_h(\Psi)}{\lambda_h(\Phi)}.$$

We use the following lemma (its proof is given in Appendix D.1):

Lemma 7 *Let $\Gamma, \Omega, \Phi \in \mathbb{R}^{H\times H}$ be a non-degenerate diagonal matrix, an orthogonal matrix, and a symmetric matrix, respectively. Let $\{\Lambda^{(k)}, \Lambda'^{(k)} \in \mathbb{R}^{H\times H}; k = 1, \ldots, K\}$ be arbitrary diagonal matrices. If*

$$G(\Omega) = tr\left\{\Gamma\Omega\Phi\Omega^\top + \sum_{k=1}^{K} \Lambda^{(k)}\Omega\Lambda'^{(k)}\Omega^\top\right\} \tag{38}$$

is minimized (as a function of Ω, given $\Gamma, \Phi, \{\Lambda^{(k)}, \Lambda'^{(k)}\}$) when $\Omega = I_H$, then Φ is diagonal. Here, K can be any natural number including $K = 0$ (when only the first term exists).

A.1 Proof for Case 1

Here, we consider the case when $c_{a_h} c_{b_h} > c_{a_{h'}} c_{b_{h'}}$ for any pair $h < h'$. We will show that any minimizer has diagonal covariances in this case.

Assume that $(A^*, B^*, \Sigma_A^*, \Sigma_B^*)$ is a minimizer of the free energy (20), and consider the following variation from it with respect to an arbitrary $H \times H$ orthogonal matrix Ω:

$$\widehat{A} = A^* C_A^{-1/2} \Omega^\top C_A^{1/2}, \tag{39}$$

$$\widehat{B} = B^* C_A^{1/2} \Omega^\top C_A^{-1/2}, \tag{40}$$

$$\Sigma_A = C_A^{1/2} \Omega C_A^{-1/2} \Sigma_A^* C_A^{-1/2} \Omega^\top C_A^{1/2}, \tag{41}$$

$$\Sigma_B = C_A^{-1/2} \Omega C_A^{1/2} \Sigma_B^* C_A^{1/2} \Omega^\top C_A^{-1/2}. \tag{42}$$

Note that this variation does not change $\widehat{B}\widehat{A}^\top$, and that $(\widehat{A},\widehat{B},\Sigma_A,\Sigma_B) = (A^*,B^*,\Sigma_A^*,\Sigma_B^*)$ holds if $\Omega = I_H$. Then, the free energy (20) can be written as a function of Ω:

$$F^{\mathrm{VB}}(\Omega) = \frac{1}{2}\mathrm{tr}\left\{C_A^{-1}C_B^{-1}\Omega C_A^{1/2}\left(B^{*\top}B^* + L\Sigma_B^*\right)C_A^{1/2}\Omega^\top\right\} + \mathrm{const.} \tag{43}$$

(the terms except the second term in the curly braces in Equation (20) are constant).

We define

$$\Phi = C_A^{1/2}\left(B^{*\top}B^* + L\Sigma_B^*\right)C_A^{1/2},$$

and rewrite Equation (43) as

$$F^{\mathrm{VB}}(\Omega) = \frac{1}{2}\mathrm{tr}\left\{C_A^{-1}C_B^{-1}\Omega\Phi\Omega^\top\right\} + \mathrm{const.} \tag{44}$$

The assumption that $(A^*,B^*,\Sigma_A^*,\Sigma_B^*)$ is a minimizer requires that Equation (44) is minimized when $\Omega = I_H$. Then, Lemma 7 (for $K = 0$) implies that Φ is diagonal.[5] Therefore,

$$C_A^{-1/2}\Phi C_A^{-1/2}(=\Phi C_A^{-1}) = B^{*\top}B^* + L\Sigma_B^*$$

is also diagonal. Consequently, Equation (10) implies that Σ_A^* is diagonal.

Next, consider the following variation with respect to an arbitrary $H \times H$ orthogonal matrix Ω',

$$\begin{aligned}
\widehat{A} &= A^* C_B^{1/2}\Omega'^\top C_B^{-1/2},\\
\widehat{B} &= B^* C_B^{-1/2}\Omega'^\top C_B^{1/2},\\
\Sigma_A &= C_B^{-1/2}\Omega' C_B^{1/2}\Sigma_A^* C_B^{1/2}\Omega'^\top C_B^{-1/2},\\
\Sigma_B &= C_B^{1/2}\Omega' C_B^{-1/2}\Sigma_B^* C_B^{-1/2}\Omega'^\top C_B^{1/2}.
\end{aligned}$$

Then, the free energy as a function of Ω' is given by

$$F^{\mathrm{VB}}(\Omega') = \frac{1}{2}\mathrm{tr}\left\{C_A^{-1}C_B^{-1}\Omega' C_B^{1/2}\left(A^{*\top}A^* + M\Sigma_A^*\right)C_B^{1/2}\Omega'^\top\right\} + \mathrm{const.}$$

From this, we can similarly prove that Σ_B^* is also diagonal, which completes the proof for Case 1. ■

A.2 Proof for Case 2

Here, we consider the case when $C_A C_B = cI_H$ for some positive $c \in \mathbb{R}$. In this case, there exist solutions with non-diagonal covariances. However, any of them belongs to an *equivalent* class involving a solution with diagonal covariances.

We can easily show that the free energy (20) is invariant of Ω under the transformation (39)–(42). This arbitrariness forms an *equivalent* class of solutions. Since there exists Ω that diagonalizes any given Σ_A^* through Equation (41), each *equivalent* class involves a solution with diagonal Σ_A. In the following, we will prove that any solution with diagonal Σ_A has diagonal Σ_B.

5. Proposition 6 implies that the diagonal entries of Φ are arranged in non-increasing order.

Assume that $(A^*, B^*, \Sigma_A^*, \Sigma_B^*)$ is a solution with diagonal Σ_A^*, and consider the following variation from it with respect to an arbitrary $H \times H$ orthogonal matrix Ω:

$$\begin{aligned}
\widehat{A} &= A^* C_A^{-1/2} \Gamma^{-1/2} \Omega^\top \Gamma^{1/2} C_A^{1/2}, \\
\widehat{B} &= B^* C_A^{1/2} \Gamma^{1/2} \Omega^\top \Gamma^{-1/2} C_A^{-1/2}, \\
\Sigma_A &= C_A^{1/2} \Gamma^{1/2} \Omega \Gamma^{-1/2} C_A^{-1/2} \Sigma_A^* C_A^{-1/2} \Gamma^{-1/2} \Omega^\top \Gamma^{1/2} C_A^{1/2}, \\
\Sigma_B &= C_A^{-1/2} \Gamma^{-1/2} \Omega \Gamma^{1/2} C_A^{1/2} \Sigma_B^* C_A^{1/2} \Gamma^{1/2} \Omega^\top \Gamma^{-1/2} C_A^{-1/2}.
\end{aligned}$$

Here, $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_H)$ is an arbitrary non-degenerate ($\gamma_h \neq \gamma_{h'}$ for $h \neq h'$) positive-definite diagonal matrix. Then, the free energy can be written as a function of Ω:

$$\begin{aligned}
F^{\mathrm{VB}}(\Omega) = \frac{1}{2} \mathrm{tr} \Big\{ & \Gamma \Omega \Gamma^{-1/2} C_A^{-1/2} \left(A^{*\top} A^* + M \Sigma_A^* \right) C_A^{-1/2} \Gamma^{-1/2} \Omega^\top \\
& + c^{-1} \Gamma^{-1} \Omega \Gamma^{1/2} C_A^{1/2} \left(B^{*\top} B^* + L \Sigma_B^* \right) C_A^{1/2} \Gamma^{1/2} \Omega^\top \Big\}. \qquad (45)
\end{aligned}$$

We define

$$\begin{aligned}
\Phi_A &= \Gamma^{-1/2} C_A^{-1/2} \left(A^{*\top} A^* + M \Sigma_A^* \right) C_A^{-1/2} \Gamma^{-1/2}, \\
\Phi_B &= c^{-1} \Gamma^{1/2} C_A^{1/2} \left(B^{*\top} B^* + L \Sigma_B^* \right) C_A^{1/2} \Gamma^{1/2},
\end{aligned}$$

and rewrite Equation (45) as

$$F^{\mathrm{VB}}(\Omega) = \frac{1}{2} \mathrm{tr} \left\{ \Gamma \Omega \Phi_A \Omega^\top + \Gamma^{-1} \Omega \Phi_B \Omega^\top \right\}. \qquad (46)$$

Since Σ_A^* is diagonal, Equation (10) implies that Φ_B is diagonal. The assumption that $(A^*, B^*, \Sigma_A^*, \Sigma_B^*)$ is a solution requires that Equation (46) is minimized when $\Omega = I_H$. Accordingly, Lemma 7 implies that Φ_A is diagonal. Consequently, Equation (11) implies that Σ_B^* is diagonal.

Thus, we have proved that any solution has its *equivalent* solution with diagonal covariances, which completes the proof for Case 2. ■

A.3 Proof for Case 3

Finally, we consider the case when $c_{a_h} c_{b_h} = c_{a'_h} c_{b_{h'}}$ for (not all but) some pairs $h \neq h'$. First, in the same way as for Case 1, we can prove that Σ_A and Σ_B are block diagonal where the blocks correspond to the groups sharing the same $c_{a_h} c_{b_h}$. Next, we can apply the proof for Case 2 to each block, and show that any solution has its *equivalent* solution with diagonal Σ_A and Σ_B. Combining these results completes the proof of Theorem 1. ■

A.4 General Expression

In summary, for any minimizer of Equation (20), the covariances can be written in the following form:

$$\Sigma_A = C_A^{1/2} \Theta C_A^{-1/2} \Gamma_{\Sigma_A} C_A^{-1/2} \Theta^\top C_A^{1/2} (= C_B^{-1/2} \Theta C_B^{1/2} \Gamma_{\Sigma_A} C_B^{1/2} \Theta^\top C_B^{-1/2}), \qquad (47)$$

$$\Sigma_B = C_A^{-1/2} \Theta C_A^{1/2} \Gamma_{\Sigma_B} C_A^{1/2} \Theta^\top C_A^{-1/2} (= C_B^{1/2} \Theta C_B^{-1/2} \Gamma_{\Sigma_B} C_B^{-1/2} \Theta^\top C_B^{1/2}). \qquad (48)$$

Here, Γ_{Σ_A} and Γ_{Σ_B} are positive-definite diagonal matrices, and Θ is a block diagonal matrix such that the blocks correspond to the groups sharing the same $c_{a_h}c_{b_h}$, and each block consists of an orthogonal matrix. Furthermore, if there exists a solution with (Σ_A, Σ_B) written in the form of Equations (47) and (48) with a certain set of $(\Gamma_{\Sigma_A}, \Gamma_{\Sigma_B}, \Theta)$, then there also exist its *equivalent* solutions with the same $(\Gamma_{\Sigma_A}, \Gamma_{\Sigma_B})$ for *any* Θ. Focusing on the solution with $\Theta = I_H$ as the representative of each *equivalent* class, we can assume that Σ_A and Σ_B are diagonal without loss of generality.

Appendix B. Proof of Theorem 3 and Theorem 4

Combining Theorem 1 and Proposition 2, we have the following lemma:

Lemma 8 *Let γ_h (≥ 0) be the h-th largest singular value of V, and let $\boldsymbol{\omega}_{a_h}$ and $\boldsymbol{\omega}_{b_h}$ be the associated right and left singular vectors:*

$$V = \sum_{h=1}^{L} \gamma_h \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top.$$

Then, the global VB solution (under the matrix-wise independence (6)*) can be expressed as*

$$\widehat{U}^{\mathrm{VB}} \equiv \langle BA^\top \rangle_{r^{\mathrm{VB}}(A,B)} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{VB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top.$$

Let

$$\widetilde{\gamma}_h = \sqrt{\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2} + \sqrt{\left(\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM\sigma^4}}. \tag{49}$$

When

$$\gamma_h \leq \widetilde{\gamma}_h,$$

the VB solution for the h-th component is $\widehat{\gamma}_h^{\mathrm{VB}} = 0$. When

$$\gamma_h > \widetilde{\gamma}_h, \tag{50}$$

$\widehat{\gamma}_h^{\mathrm{VB}}$ is given as a positive real solution of

$$\widehat{\gamma}_h^2 + q_1(\widehat{\gamma}_h) \cdot \widehat{\gamma}_h + q_0 = 0, \tag{51}$$

where

$$q_1(\widehat{\gamma}_h) = \frac{-(M-L)^2(\gamma_h - \widehat{\gamma}_h) + (L+M)\sqrt{(M-L)^2(\gamma_h - \widehat{\gamma}_h)^2 + \frac{4\sigma^4 LM}{c_{a_h}^2 c_{b_h}^2}}}{2LM}, \tag{52}$$

$$q_0 = \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} - \left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h^2. \tag{53}$$

When Inequality (50) *holds, Equation* (51) *has only one positive real solution, which lies in*

$$0 < \widehat{\gamma}_h < \gamma_h. \tag{54}$$

To obtain an analytic-form solution, we will find the positive solution of Equation (51) for $\gamma_h > \widetilde{\gamma}_h$. Because $q_1(\widehat{\gamma}_h)$ depends on the square root of $\widehat{\gamma}_h$, Equation (51) is not polynomial. However, since it has only one non-polynomial term, we can easily convert it to a polynomial form in the following way.

Substituting Equations (52) and (53), we can rewrite Equation (51) as

$$-\frac{(M^2+L^2)}{2LM}\widehat{\gamma}_h^2+\frac{(M-L)^2\gamma_h}{2LM}\widehat{\gamma}_h+\sqrt{\xi_0}=\left(\frac{(M+L)\sqrt{\{(M-L)(\gamma_h-\widehat{\gamma}_h)\}^2+\frac{4\sigma^4ML}{c_{a_h}^2c_{b_h}^2}}}{2LM}\right)\widehat{\gamma}_h. \tag{55}$$

Squaring both sides of Equation (55) removes the square root in the right-hand side, and leads to the quartic equation (23),

$$f(\widehat{\gamma}_h) := \widehat{\gamma}_h^4+\xi_3\widehat{\gamma}_h^3+\xi_2\widehat{\gamma}_h^2+\xi_1\widehat{\gamma}_h+\xi_0=0, \tag{23}$$

where

$$\xi_3 = \frac{(L-M)^2\gamma_h}{LM}, \tag{56}$$

$$\xi_2 = -\left(\xi_3\gamma_h+\frac{(L^2+M^2)\eta_h^2}{LM}+\frac{2\sigma^4}{c_{a_h}^2c_{b_h}^2}\right), \tag{57}$$

$$\xi_1 = \xi_3\sqrt{\xi_0}, \tag{58}$$

$$\xi_0 = \left(\eta_h^2-\frac{\sigma^4}{c_{a_h}^2c_{b_h}^2}\right)^2, \tag{59}$$

$$\eta_h^2 = \left(1-\frac{\sigma^2L}{\gamma_h^2}\right)\left(1-\frac{\sigma^2M}{\gamma_h^2}\right)\gamma_h^2. \tag{60}$$

Since we derived Equation (23) from Equation (51), any solution satisfying Equation (51) satisfies Equation (23). However, the converse does not necessarily hold, because squaring both sides of Equation (55) can create solutions that do not satisfy the original equation (51). By examining the possible range of the positive solution of Equation (51), we obtain the following lemma (the proof is given in Appendix D.2):

Lemma 9 *Assume that Inequality* (50) *holds. Any positive solution of Equation* (51) *lying in the range* (54) *satisfies the quartic equation* (23)*, and lies in the following range:*

$$0 < \widehat{\gamma}_h < \xi_0^{1/4}. \tag{61}$$

Conversely, any positive solution of the quartic equation (23) *lying in the range* (61) *satisfies Equation* (51)*, and lies in the range* (54)*.*

Lemma 8 and Lemma 9 imply that finding the VB solution is achieved by finding a positive solution of the quartic equation (23) lying in the range (61). Investigating the shape of the quartic function $f(\widehat{\gamma}_h)$, defined in Equation (23), we have the following lemma (the proof is given in Appendix D.3):

Lemma 10 *Assume that Inequality* (50) *holds. The quartic equation* (23) *has two positive real solutions. The smaller one lies in the range* (61)*, and the larger one lies outside the range.*

Combining Lemma 8, Lemma 9, and Lemma 10 completes the proof of Theorem 3.

The following lemma is obtained by summarizing Lemma 11, Lemma 12, Lemma 14, Lemma 15, and Lemma 17 in Nakajima and Sugiyama (2011), and then combining with Theorem 1 in this paper:[6]

Lemma 11 *Let*

$$
\begin{aligned}
(\widehat{\eta}_h^2)^{\text{null}} &= \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}, \\
(\sigma_{a_h}^2)^{\text{null}} &= \frac{-\left((\widehat{\eta}_h^2)^{\text{null}} - \sigma^2(M-L)\right) + \sqrt{((\widehat{\eta}_h^2)^{\text{null}} - \sigma^2(M-L))^2 + 4M\sigma^2(\widehat{\eta}_h^2)^{\text{null}}}}{2M\sigma^2 c_{a_h}^{-2}}, \\
(\sigma_{b_h}^2)^{\text{null}} &= \frac{-\left((\widehat{\eta}_h^2)^{\text{null}} + \sigma^2(M-L)\right) + \sqrt{((\widehat{\eta}_h^2)^{\text{null}} + \sigma^2(M-L))^2 + 4L\sigma^2(\widehat{\eta}_h^2)^{\text{null}}}}{2L(\widehat{\gamma}_h\widehat{\delta}_h + \sigma^2 c_{b_h}^{-2})}.
\end{aligned}
$$

When $\gamma_h \leq \widetilde{\gamma}_h$, the means and the variances of the VB posterior for the h-th component are given by

$$
(\widehat{a}_h, \widehat{b}_h, (\Sigma_A)_{h,h}, (\Sigma_B)_{h,h}) = \left(\mathbf{0}, \mathbf{0}, (\sigma_{a_h}^2)^{\text{null}}, (\sigma_{b_h}^2)^{\text{null}}\right).
$$

For $\gamma_h > \widetilde{\gamma}_h$, let

$$
\begin{aligned}
\widehat{\delta}_h &= \frac{(M-L)(\gamma_h - \widehat{\gamma}_h) + \sqrt{(M-L)^2(\gamma_h - \widehat{\gamma}_h)^2 + \frac{4\sigma^4 LM}{c_{a_h}^2 c_{b_h}^2}}}{2\sigma^2 M c_{a_h}^{-2}}, \\
\sigma_{a_h}^2 &= \frac{-\left(\widehat{\eta}_h^2 - \sigma^2(M-L)\right) + \sqrt{(\widehat{\eta}_h^2 - \sigma^2(M-L))^2 + 4M\sigma^2\widehat{\eta}_h^2}}{2M(\widehat{\gamma}_h\widehat{\delta}_h^{-1} + \sigma^2 c_{a_h}^{-2})}, \\
\sigma_{b_h}^2 &= \frac{-\left(\widehat{\eta}_h^2 + \sigma^2(M-L)\right) + \sqrt{(\widehat{\eta}_h^2 + \sigma^2(M-L))^2 + 4L\sigma^2\widehat{\eta}_h^2}}{2L(\widehat{\gamma}_h\widehat{\delta}_h + \sigma^2 c_{b_h}^{-2})}.
\end{aligned}
\tag{62}
$$

When $\gamma_h > \widetilde{\gamma}_h$, the means and the variances of the VB posterior for the h-th component are given by

$$
(\widehat{a}_h, \widehat{b}_h, (\Sigma_A)_{h,h}, (\Sigma_B)_{h,h}) = (\pm\sqrt{\widehat{\gamma}_h\widehat{\delta}_h}\boldsymbol{\omega}_{a_h}, \pm\sqrt{\widehat{\gamma}_h\widehat{\delta}_h^{-1}}\boldsymbol{\omega}_{b_h}, \sigma_{a_h}^2, \sigma_{b_h}^2).
$$

Combining Theorem 3 and Lemma 11 completes the proof of Theorem 4. ■

Appendix C. Proof of Theorem 5

Summarizing Lemma 22, Lemma 23, and Lemma 24 in Nakajima and Sugiyama (2011), and then combining with Theorem 1 in this paper, we have the following lemma:

6. We also used Equation (147) in Nakajima and Sugiyama (2011), which is identical to Equation (62) in this paper.

Lemma 12 *If $\gamma_h \geq \underline{\gamma}_h$, the VB free energy* (20) *can have two local minima, i.e., $c_{a_h}c_{b_h} \to 0$ and $c_{a_h}c_{b_h} = \check{c}_{a_h}\check{c}_{b_h}$. Otherwise, $c_{a_h}c_{b_h} \to 0$ is the only local minimum.*

It was also shown in Nakajima and Sugiyama (2011) that the (scaled) free energy difference between the two local minima is given by Δ_h (the *positive* local minimum with $c_{a_h}c_{b_h} = \check{c}_{a_h}\check{c}_{b_h}$ gives lower free energy than the *null* local minimum with $c_{a_h}c_{b_h} \to 0$ if and only if $\Delta_h \leq 0$).[7] Thus, we have the following lemma:

Lemma 13 *The hyperparameter $\widehat{c}_{a_h}\widehat{c}_{b_h}$ that globally minimizes the VB free energy* (20) *is given by $\widehat{c}_{a_h}\widehat{c}_{b_h} = \check{c}_{a_h}\check{c}_{b_h}$ if $\gamma_h > \underline{\gamma}_h$ and $\Delta_h \leq 0$. Otherwise $\widehat{c}_{a_h}\widehat{c}_{b_h} \to 0$.*

Combining Lemma 13 and Theorem 3 completes the proof of Theorem 5. ■

Appendix D. Proof of Lemmas

In this appendix, we prove lemmas used in the previous appendices.

D.1 Proof of Lemma 7

Let

$$\Phi = \Omega'\Gamma'\Omega'^{\top} \tag{63}$$

be the eigenvalue decomposition of Φ. Let $\gamma, \gamma', \{\boldsymbol{\lambda}^{(k)}\}, \{\boldsymbol{\lambda}'^{(k)}\}$ be the vectors consist of the diagonal entries of $\Gamma, \Gamma', \{\Lambda^{(k)}\}, \{\Lambda'^{(k)}\}$, respectively, i.e,

$$\Gamma = \mathrm{diag}(\gamma), \qquad \Gamma' = \mathrm{diag}(\gamma'), \qquad \Lambda^{(k)} = \mathrm{diag}(\boldsymbol{\lambda}^{(k)}), \qquad \Lambda'^{(k)} = \mathrm{diag}(\boldsymbol{\lambda}'^{(k)}).$$

Then, Equation (38) can be written as

$$G(\Omega) = \mathrm{tr}\left\{\Gamma\Omega\Phi\Omega^{\top} + \sum_{k=1}^{K}\Lambda^{(k)}\Omega\Lambda'^{(k)}\Omega^{\top}\right\} = \gamma^{\top}Q\gamma' + \sum_{k=1}^{K}\boldsymbol{\lambda}^{(k)\top}R\boldsymbol{\lambda}'^{(k)}, \tag{64}$$

where

$$Q = (\Omega\Omega') * (\Omega\Omega'), \qquad R = \Omega * \Omega.$$

Here, $*$ denotes the Hadamard product.[8]

Using this expression, we will prove that Φ is diagonal if $\Omega = I_H$ minimizes Equation (64). Let us consider a bilateral perturbation $\Omega = \Delta$ such that the 2×2 matrix $\Delta_{(h,h')}$ consisting of the h-th and the h'-th columns and rows form an 2×2 orthogonal matrix

$$\Delta_{(h,h')} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

7. Equation (26) was obtained as Equation (172) in Nakajima and Sugiyama (2011).

8. Note that Q as well as R is the Hadamard square of an orthogonal matrix, which is known to be doubly stochastic (i.e., any of the columns and the rows sums up to one) (Marshall et al., 2009). Therefore, it can be seen that Q reassigns the components of γ to those of γ' when calculating the element-wise product in the first term of Equation (64). The same applies to R and $\{\boldsymbol{\lambda}^{(k)}, \boldsymbol{\lambda}'^{(k)}\}$ in the second term. Naturally, rearranging the components of γ in non-decreasing order and the components of γ' in non-increasing order minimizes $\gamma^{\top}Q\gamma'$, which proves Proposition 6 (Ruhe, 1970; Marshall et al., 2009).

and the rest entries coincide with those of the identity matrix. Then, the elements of Q become

$$Q_{i,j} = \begin{cases} (\Omega'_{h,j}\cos\theta - \Omega'_{h',j}\sin\theta)^2 & \text{if } i = h, \\ (\Omega'_{h,j}\sin\theta + \Omega'_{h',j}\cos\theta)^2 & \text{if } i = h', \\ \Omega'^2_{i,j} & \text{otherwise,} \end{cases}$$

and Equation (64) can be written as a function of θ:

$$G(\theta) = \sum_{j=1}^{H} \left\{ \gamma_h (\Omega'_{h,j}\cos\theta - \Omega'_{h',j}\sin\theta)^2 + \gamma_{h'} (\Omega'_{h,j}\sin\theta + \Omega'_{h',j}\cos\theta)^2 \right\} \gamma'_j + \sum_{k=1}^{K} \begin{pmatrix} \lambda_h^{(k)} & \lambda_{h'}^{(k)} \end{pmatrix} \begin{pmatrix} \cos^2\theta & \sin^2\theta \\ \sin^2\theta & \cos^2\theta \end{pmatrix} \begin{pmatrix} \lambda_h^{(k)} \\ \lambda_{h'}^{(k)} \end{pmatrix} + \text{const.}. \tag{65}$$

Since Equation (65) is differentiable at $\theta = 0$, our assumption that Equation (64) is minimized when $\Omega = I_H$ requires that $\theta = 0$ is a stationary point of Equation (65) for any $h \neq h'$. Therefore, it holds that

$$0 = \left.\frac{\partial G}{\partial \theta}\right|_{\theta=0} = 2(\gamma_{h'} - \gamma_h) \sum_j \Omega'_{h,j} \gamma'_j \Omega'_{h',j} = 2(\gamma_{h'} - \gamma_h)\Phi_{h,h'}. \tag{66}$$

In the last equation, we used Equation (63). Since we assume that Γ is non-degenerate ($\gamma_h \neq \gamma_{h'}$ for $h \neq h'$), Equation (66) implies that Φ is diagonal, which completes the proof of Lemma 7. ■

D.2 Proof of Lemma 9

Assume that Inequality (50) holds, i.e.,

$$\gamma_h > \widetilde{\gamma}_h. \tag{50}$$

By using Equation (60), we have

$$\eta_h^2 - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} = \left(1 - \frac{\sigma^2 L}{\gamma_h^2}\right)\left(1 - \frac{\sigma^2 M}{\gamma_h^2}\right)\gamma_h^2 - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} = \gamma_h^{-2}\left(\gamma_h^2 - \widetilde{\gamma}_h^2\right)\left(\gamma_h^2 - \acute{\gamma}_h^2\right), \tag{67}$$

where

$$\acute{\gamma}_h = \sqrt{\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2} - \sqrt{\left(\frac{(L+M)\sigma^2}{2} + \frac{\sigma^4}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM\sigma^4}}. \tag{68}$$

Comparing Equations (49) and (68) leads to

$$\widetilde{\gamma}_h > \acute{\gamma}_h,$$

and therefore, Equation (67) is positive, i.e.,

$$\eta_h^2 - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} > 0. \tag{69}$$

Combining Equations (59) and (60), and Inequality (69) leads to

$$0 < \xi_0^{1/4} < \eta_h < \gamma_h. \tag{70}$$

Combining Equations (53), (59), and (60) leads to

$$q_0 = -\sqrt{\xi_0}. \tag{71}$$

Let us first assume that we have a positive solution of Equation (51) lying in the range (54),

$$0 < \widehat{\gamma}_h < \gamma_h. \tag{54}$$

Since Equation (23) was derived from Equation (51), this solution naturally satisfies Equation (23). For the solution, Equation (52) implies that

$$q_1(\widehat{\gamma}_h) > 0.$$

Inequalities (70) and Equation (71) imply that

$$q_0 < 0.$$

Therefore, by ignoring the positive second term in the left-hand side of Equation (51), we find that the solution lies in the range (61),

$$0 < \widehat{\gamma}_h < \sqrt{-q_0} = \xi_0^{1/4}. \tag{61}$$

Here, we used Equation (71) in the last equality.

Conversely, assume that we have a positive solution of Equation (23) lying in the range (61). Since Equation (23) was derived by squaring both sides of Equation (55), the solution satisfies Equation (55) if the both sides of Equation (55) have the same sign. Clearly, the right-hand side of Equation (55) is positive. We will show that the left-hand side of Equation (55),

$$g(\widehat{\gamma}_h) = -\frac{(M^2+L^2)}{2LM}\widehat{\gamma}_h^2 + \frac{(M-L)^2\gamma_h}{2LM}\widehat{\gamma}_h + \sqrt{\xi_0},$$

is also positive.

Note that $g(\widehat{\gamma}_h)$ is strictly concave because it is a quadratic function with a negative coefficient of the quadratic term. Since we are assuming that the solution lies in the range (61), the following holds:

$$\begin{aligned} g(\widehat{\gamma}_h) &> \min\left\{ g(0), g(\xi_0^{1/4}) \right\} \\ &> \min\left\{ \sqrt{\xi_0}, \frac{(M-L)^2\gamma_h}{2LM}\xi_0^{1/4}(\gamma_h - \xi_0^{1/4}) \right\} \\ &> 0. \end{aligned}$$

We used Inequalities (70) in the last inequality. Thus, we have shown that the left-hand side, $g(\widehat{\gamma}_h)$, of Equation (55) is also positive, and therefore, the solution satisfies Equation (55). This means that the solution also satisfies its equivalent equation (51). Since Inequalities (70) imply that the range (61) is included in the range (54), the solution trivially lies in the range (54), which completes the proof of Lemma 9. ■

D.3 Proof of Lemma 10

We will investigate the shape of the quartic function (23),

$$f(\widehat{\gamma}_h) := \widehat{\gamma}_h^4 + \xi_3\widehat{\gamma}_h^3 + \xi_2\widehat{\gamma}_h^2 + \xi_1\widehat{\gamma}_h + \xi_0. \tag{23}$$

Since the coefficient of the quartic term is positive (equal to one), $f(\widehat{\gamma}_h)$ goes to infinity as $\widehat{\gamma}_h \to -\infty$ or $\widehat{\gamma}_h \to \infty$. Since Equation (59) implies that $\xi_0 > 0$, it holds that $f(0) > 0$.

By using Equation (58), we have

$$\begin{aligned} f(\xi_0^{1/4}) &= \xi_0 + \xi_3\xi_0^{3/4} + \xi_2\sqrt{\xi_0} + \xi_1\xi_0^{1/4} + \xi_0 \\ &= \xi_0 + \xi_3\xi_0^{3/4} + \xi_2\sqrt{\xi_0} + \xi_3\xi_0^{3/4} + \xi_0 \\ &= \sqrt{\xi_0}\left(2\sqrt{\xi_0} + 2\xi_3\xi_0^{1/4} + \xi_2\right). \end{aligned}$$

By using Inequalities (70) and Equation (57), this can be bounded as

$$\begin{aligned} f(\xi_0^{1/4}) &< \sqrt{\xi_0}\left(2\eta_h^2 + 2\xi_3\eta_h - \xi_3\gamma_h - \frac{(L^2+M^2)\eta_h^2}{LM}\right) \\ &= \sqrt{\xi_0}\left(2\xi_3\eta_h - \xi_3\gamma_h - \frac{(L-M)^2\eta_h^2}{LM}\right) \\ &= \frac{\sqrt{\xi_0}\xi_3}{\gamma_h}\left(2\eta_h\gamma_h - \gamma_h^2 - \eta_h^2\right) \\ &= -\frac{\sqrt{\xi_0}\xi_3}{\gamma_h}\left(\gamma_h - \eta_h\right)^2 \\ &< 0. \end{aligned}$$

Here, we used Equation (56) in the third equality, and Inequalities (70) in the last inequality.

In summary, we have the following:

$$\lim_{\widehat{\gamma}_h \to -\infty} f(\widehat{\gamma}_h) = \infty,$$

$$f(0) > 0, \tag{72}$$

$$f(\xi_0^{1/4}) < 0, \tag{73}$$

$$\lim_{\widehat{\gamma}_h \to \infty} f(\widehat{\gamma}_h) = \infty. \tag{74}$$

Furthermore, since Equation (57) implies that $\xi_2 < 0$, $f(\widehat{\gamma}_h)$ has a negative curvature at the origin, i.e., $(\partial^2 f/\partial^2\widehat{\gamma}_h)(0) < 0$. This means that $f(\widehat{\gamma}_h)$ has one inflection point each in the positive region $\widehat{\gamma}_h > 0$ and in the negative region $\widehat{\gamma}_h < 0$. The shape of the quartic function $f(\widehat{\gamma}_h)$ is shown in Figure 9. Note that the points at which $f(\widehat{\gamma}_h)$ crosses the horizontal axis are the solutions of the quartic equation (23).

Inequality (73) and Equation (74) imply that at least one solution exists in the region

$$\widehat{\gamma}_h > \xi_0^{1/4}.$$

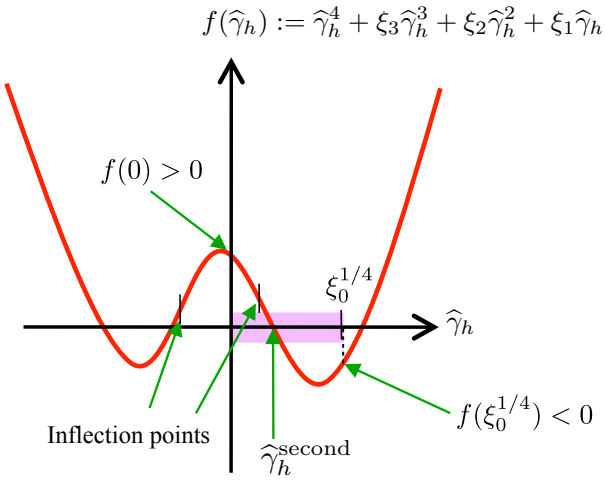


Figure 9: The shape of a quartic function $f(\widehat{\gamma}_h) := \widehat{\gamma}_h^4 + \xi_3\widehat{\gamma}_h^3 + \xi_2\widehat{\gamma}_h^2 + \xi_1\widehat{\gamma}_h + \xi_0$, where $\xi_2 < 0$, $\xi_0 (= f(0)) > 0$, and $f(\xi_0^{1/4}) < 0$. The range $0 < \widehat{\gamma}_h < \xi_0^{1/4}$, where the second largest positive real solution $\widehat{\gamma}_h^{\text{second}}$ exists, is highlighted.

Inequalities (72) and (73) imply that at least one solution exists in the region

$$0 < \widehat{\gamma}_h < \xi_0^{1/4}.$$

Since $f(\widehat{\gamma}_h)$ has only one inflection point in the positive region, it has no more solution in the positive region without contradiction with Inequality (72) (see Figure 9), which completes the proof of Lemma 10. ■

References

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, second edition, 1984.

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 21–30, San Francisco, CA, 1999. Morgan Kaufmann.

S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos. Sparse Bayesian methods for low-rank matrix estimation. *IEEE Trans. on Signal Processing*, 60(8):3964–3977, 2012.

P. F. Baldi and K. Hornik. Learning in linear neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(4):837–858, 1995.

J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48: 259–302, 1986.

C. M. Bishop. Variational principal components. In *Proc. of ICANN*, volume 1, pages 514–509, 1999.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006.

J. F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *CoRR*, abs/0912.3599, 2009.

J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970.

S. Funk. Try this at home. http://sifter.org/˜simon/journal/20061211.html, 2006.

D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

R. A. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

M. Hazewinkel, editor. *Encyclopaedia of Mathematics*. Springer, 2002.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3–4):321–377, 1936.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *JMLR*, 11:1957–2000, 2010.

W. James and C. Stein. Estimation with quadratic loss. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 361–379, Berkeley, CA., USA, 1961. University of California Press.

H. Jeffreys. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, volume 186, pages 453–461, 1946.

S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of International Conference on Machine Learning*, pages 457–464, 2009.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Y. J. Lim and T. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, 2007.

D. J. C. Mackay. Local minima, symmetry-breaking, and model pruning in variational free energy minimization. Available from `http://www.inference.phy.cam.ac.uk/mackay/minima.pdf`. 2001.

A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: Theory of Majorization and Its Applications, Second Edition*. Springer, 2009.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

S. Nakajima. Variational Bayesian algorithm for relational tensor factorization. *Under Preparation*, 2012.

S. Nakajima and M. Sugiyama. Theoretical analysis of Bayesian matrix factorization. *Journal of Machine Learning Research*, 12:2579–2644, 2011.

S. Nakajima, M. Sugiyama, and R. Tomioka. Global analytic solution for variational Bayesian matrix factorization. In *Advances in Neural Information Processing Systems 23*, pages 1759–1767, 2010.

S. Nakajima, M. Sugiyama, and S. D. Babacan. Global solution of fully-observed variational Bayesian matrix factorization is column-wise independent. In *Advances in Neural Information Processing Systems 24*, 2011.

S. Nakajima, M. Sugiyama, and S. D. Babacan. Sparse additive matrix factorization for robust PCA and its generalization. In *Proceedings of Fourth Asian Conference on Machine Learning*, 2012a.

S. Nakajima, R. Tomioka, M. Sugiyama, and S. D. Babacan. Perfect dimensionality recovery by variational Bayesian PCA. In *Advances in Neural Information Processing Systems 25*, 2012b.

A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, 2007.

G. R. Reinsel and R. P. Velu. *Multivariate Reduced-Rank Regression: Theory and Applications*. Springer, New York, 1998.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine learning*, pages 713–719, 2005.

R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. In C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection Techniques*, volume 3940 of *Lecture Notes in Computer Science*, pages 34–51, Berlin, 2006. Springer.

A. Ruhe. Perturbation bounds for means of eigenvalues and invariant subspaces. *BIT Numerical Mathematics*, 10:343–354, 1970.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264, Cambridge, MA, 2008. MIT Press.

M. Seeger and G. Bouchard. Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Proc. of AISTATS*, La Palma, Spain, 2012.

N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2005.

G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4): 551–556, 1993.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61:611–622, 1999.

R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. An efficient and general augmented Lagrangian algorithm for learning low-rank matrices. In *Proceedings of International Conference on Machine Learning*, 2010.

L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1996.

S. Watanabe. *Algebraic Geometry and Statistical Learning*. Cambridge University Press, Cambridge, UK, 2009.

K. J. Worsley, J-B. Poline, K. J. Friston, and A. C. Evanss. Characterizing the response of PET and fMRI data using multivariate linear models. *NeuroImage*, 6(4):305–319, 1997.

Journal of Machine Learning Research 14 (2013) 39-73 Submitted 12/10; Revised 2/12; Published 1/13

Ranking Forests

Stéphan Clémençon STEPHAN.CLEMENCON@TELECOM-PARISTECH.FR
Marine Depecker MARINE-DEPECKER@TELECOM-PARISTECH.FR
Institut Telecom LTCI - UMR Telecom ParisTech/CNRS No. 5141
Telecom ParisTech
46 rue Barrault, Paris, 75634, France

Nicolas Vayatis NICOLAS.VAYATIS@CMLA.ENS-CACHAN.FR
CMLA - UMR ENS Cachan/CNRS No. 8536
ENS Cachan
61, avenue du Président Wilson, Cachan, 94230, France

Editor: Tong Zhang

Abstract

The present paper examines how the aggregation and feature randomization principles underlying the algorithm RANDOM FOREST (Breiman, 2001) can be adapted to *bipartite ranking*. The approach taken here is based on nonparametric scoring and ROC curve optimization in the sense of the AUC criterion. In this problem, aggregation is used to increase the performance of scoring rules produced by ranking trees, as those developed in Clémençon and Vayatis (2009c). The present work describes the principles for building median scoring rules based on concepts from rank aggregation. Consistency results are derived for these aggregated scoring rules and an algorithm called RANKING FOREST is presented. Furthermore, various strategies for feature randomization are explored through a series of numerical experiments on artificial data sets.

Keywords: bipartite ranking, nonparametric scoring, classification data, ROC optimization, AUC criterion, tree-based ranking rules, bootstrap, bagging, rank aggregation, median ranking, feature randomization

1. Introduction

Aggregating decision rules or function estimators has now become a folk concept in machine learning and nonparametric statistics. Indeed, the idea of combining decision rules with an additional randomization ingredient brings a dramatic improvement of performance in various contexts. These ideas go back to the seminal work of Amit and Geman (1997), Breiman (1996), and Nemirovski (2000). However, in the context of the "learning-to-rank" problem, the implementation of this idea is still at a very early stage. In the present paper, we propose to take one step beyond in the program of boosting performance by aggregation and randomization for this problem. The present paper explores the particular case of learning to rank high dimensional observation vectors in presence of binary feedback. This case is also known as the bipartite ranking problem, see Freund et al. (2003), Agarwal et al. (2005), Clémençon et al. (2005). The setup of bipartite ranking is useful when considering real-life applications such as credit-risk or medical screening, spam filtering, or recommender systems. There are two major approaches to bipartite ranking: the *preference-based* approach (see Cohen et al. 1999) and the *scoring-based* approach (in the spirit of logistic regression methods, see, e.g., Hastie and Tibshirani 1990, Hilbe 2009). The idea of combining ranking

rules to learn preferences was introduced in Freund et al. (2003) with a boosting algorithm and the consistency for this type of methods was proved in Clémençon et al. (2008) by reducing the bipartite ranking problem to a classification problem over pairs of observations (see also Agarwal et al. 2005). Here, we will cast bipartite ranking in the context of nonparametric scoring and we will consider the issue of combining randomized *scoring rules*. Scoring rules are real-valued functions mapping the observation space with the real line, thus conveying an order relation between high dimensional observation vectors.

Nonparametric scoring has received an increasing attention in the machine learning literature as a part of the growing interest which affects ROC analysis. The scoring problem can be seen as a learning problem where one observes input observation vectors X in a high dimensional space $\mathcal{X}$ and receives only a binary feedback information through an output variable $Y \in \{-1,+1\}$. Whereas classification only focuses on predicting the label $\widetilde{Y}$ of a new observation $\widetilde{X}$, scoring algorithms aim at recovering an order relation on $\mathcal{X}$ in order to predict the ordering over a new sample of observation vectors $X'_1,\ldots,X'_m$ so that there as many as possible positive instances at the top of the list. From a statistical perspective, the scoring problem is more difficult than classification but easier than regression. Indeed, in classification, the goal is to learn *one* single level set of the regression function whereas, in scoring, one wants to recover the nested collection of *all* the level sets of the regression function (without necessarily knowing the corresponding levels), but not the regression function itself (see Clémençon and Vayatis 2009b). In previous work, we developed a tree-based procedure for nonparametric scoring called TREERANK, see Clémençon and Vayatis (2009c), Clémençon et al. (2010). The TREERANK algorithm and its variants produce scoring rules expressed as partitions of the input space coupled with a permutation over the cells of the partition. These scoring rules present the interesting feature that they can be stored in an oriented binary tree structure, called a *ranking tree*. Moreover, their very construction actually implements the optimization of the ROC curve which reflects the quality measure of the scoring rule for the end-user.

The use of resampling in this context was first considered in Clémençon et al. (2009). A more thorough analysis is developed throughout this paper and we show how to combine feature randomization and bootstrap aggregation techniques based on the ranking trees produced by the TREERANK algorithm in order to increase ranking performance in the sense of the ROC curve. In the classification setup, theoretical evidence has been recently provided for the aggregation of randomized classifiers in the spirit of random forests (see Biau et al. 2008). However, in the context of ROC optimization, combining scoring rules through naive aggregation does not necessarily make sense. Our approach builds on the advances in the rank aggregation problem. Rank aggregation was originally introduced in social choice theory (see Barthélémy and Montjardet 1981 and the references therein) and recently "rediscovered" in the context of internet applications (see Pennock et al. 2000). For our needs, we shall focus on *metric-based consensus methods* (see Hudry 2004 or Fagin et al. 2006, and the references therein), which provide the key to the aggregation of ranking trees. In the paper, we also discuss various aspects of feature randomization which can be incorporated at various levels in ranking trees. Also a novel ranking methodology, called RANKING FOREST, is introduced.

The article is structured as follows. Section 2 sets out the notations and shortly describes the main notions for the bipartite ranking problem. Section 3 describes the elements from the theory of rank aggregation and measures of consensus leading to the aggregation of scoring rules defined over finite partitions of the input space. The next section presents the main theoretical results of the paper

which are consistency results for scoring rules based on the aggregation of randomized piecewise constant scoring rules. Section 5 presents RANKING FOREST, a new algorithm for nonparametric scoring which implements the theoretical concepts developed so far. Section 6 presents an empirical study of the RANKING FOREST algorithm with numerical results based on simulated data. Finally, some concluding remarks are collected in Section 7. Reminders, technical details and proofs are deferred to the Appendix.

2. Probabilistic Setup for Bipartite Ranking

ROC analysis is a popular way of evaluating the capacity of a given scoring rule to discriminate between two populations, see Egan (1975). ROC curves and related performance measures such as the AUC have now become of standard use for assessing the quality of ranking methods in a bipartite framework. Throughout this section, we recall basic concepts related to bipartite ranking from the angle of ROC analysis.

Modeling the data. The probabilistic setup is the same as in standard binary classification. The random variable Y is a binary label, valued in $\{-1,+1\}$, while the random vector $X = (X^{(1)},\ldots,X^{(q)})$ models some multivariate observation for predicting Y, taking its values in a high-dimensional space $\mathcal{X} \subset \mathbb{R}^q$, $q \geq 1$. The probability measure on the underlying space is entirely described by the pair (μ,η), where μ denotes the marginal distribution of X and $\eta(x) = \mathbb{P}\{Y = +1 \mid X = x\}$, $x \in \mathcal{X}$, the posterior probability. With no restriction, here we assume that $\mathcal{X}$ coincides with the support of μ.

The scoring approach to bipartite ranking. An informal way of considering the ranking task under this model is as follows. Given a a sample of independent copies of the pair (X,Y), the goal is to learn how to order new data $X_1,\ldots,X_m$ without label feedback, so that positive instances are mostly at the top of the resulting list with large probability. A natural way of defining a total order on the multidimensional space $\mathcal{X}$ is to map it with the natural order on the real line by means of a *scoring rule*, that is, a measurable mapping $s : \mathcal{X} \to \mathbb{R}$. A preorder[1] $\preccurlyeq_s$ on $\mathcal{X}$ is then defined by: $\forall (x,x') \in \mathcal{X}^2$, $x \preccurlyeq_s x'$ if and only if $s(x) \leq s(x')$.

Measuring performance. The capacity of a candidate s to discriminate between the positive and negative populations is generally evaluated by means of its ROC curve (standing for "Receiver Operating Characteristic" curve), a widely used functional performance measure which we recall here.

Definition 1 (TRUE ROC CURVE) *Let s be a scoring rule. The true* ROC *curve of s is the "probability-probability" plot given by:*

$$t \in \mathbb{R} \mapsto \left(\mathbb{P}\{s(X) > t \mid Y = -1\}, \mathbb{P}\{s(X) > t \mid Y = 1\}\right) \in [0,1]^2 .$$

By convention, when a jump occurs in the plot of the ROC *curve, the corresponding extremities of the curve are connected by a line segment, so that the* ROC *curve of s can be viewed as the graph of a continuous mapping $\alpha \in [0,1] \mapsto \mathrm{ROC}(s,\alpha)$.*

We refer to Clémençon and Vayatis (2009c) for a list of properties of ROC curves (see the Appendix section therein). The ROC curve offers a visual tool for assessing ranking performance (see Figure 1): the closer to the left upper corner of the unit square $[0,1]^2$ the curve $\mathrm{ROC}(s,.)$, the better the scoring rule s. Therefore, the ROC curve conveys a partial order on the set of all

1. A preorder is a binary relation which is reflexive and transitive.

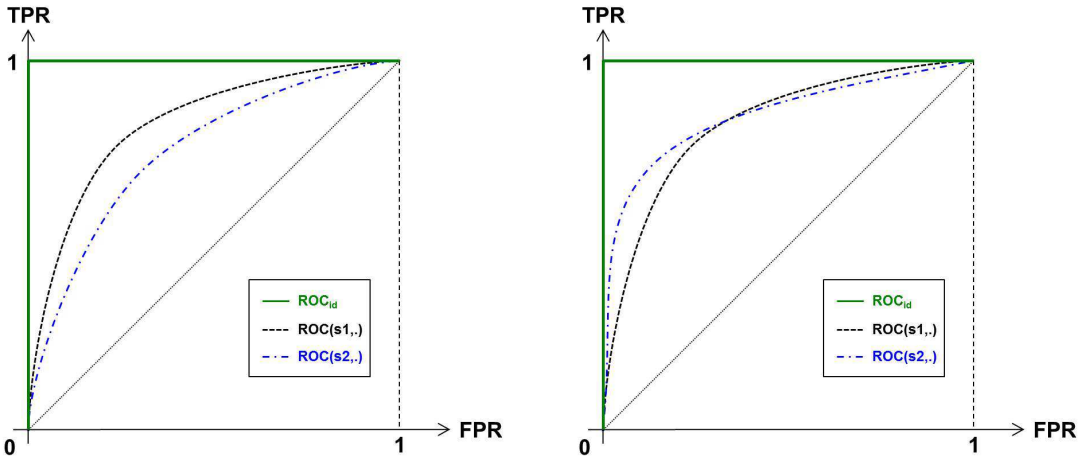


Figure 1: ROC curves.

scoring rules: for all pairs of scoring rules s_1 and s_2, we say that s_2 is more accurate than s_1 when $\mathrm{ROC}(s_1,\alpha) \leq \mathrm{ROC}(s_2,\alpha)$ for all $\alpha \in [0,1]$. By a standard Neyman-Pearson argument, one may establish that the most accurate scoring rules are increasing transforms of the regression function which is equal to the conditional probability function η up to an affine transformation.

Definition 2 (OPTIMAL SCORING RULES) *We call optimal scoring rules the elements of the set $\mathcal{S}^*$ of scoring functions s^* such that $\forall (x,x') \in \mathcal{X}^2$, $\eta(x) < \eta(x') \Rightarrow s^*(x) < s^*(x')$.*

The fact that the elements of $\mathcal{S}^*$ are optimizers of the ROC curve is shown in Clémençon and Vayatis (2009c) (see Proposition 4 therein). When, in addition, the random variable $\eta(X)$ is assumed to be continuous, then $\mathcal{S}^*$ coincides with the set of strictly increasing transforms of η. The performance of a candidate scoring rule s is often summarized by a scalar quantity called the *Area Under the* ROC *Curve* (AUC) which can be considered as a summary of the ROC curve. In the paper, we shall use the following definition of the AUC.

Definition 3 (AUC) *Let s be a scoring rule. The* AUC *is the functional defined as:*

$$\mathrm{AUC}(s) = \mathbb{P}\{s(X_1) < s(X_2) \mid (Y_1,Y_2) = (-1,+1)\} + \frac{1}{2}\mathbb{P}\{s(X_1) = s(X_2) \mid (Y_1,Y_2) = (-1,+1)\},$$

where (X_1,Y_1) and (X_2,Y_2) denote two independent copies of the pair (X,Y), for any scoring function s.

This functional provides a *total order* on the set of scoring rules and, equipped with the convention introduced in Definition 1, $\mathrm{AUC}(s)$ coincides with $\int_0^1 \mathrm{ROC}(s,\alpha)\,d\alpha$ (see, for instance, Proposition 1 in Clémençon et al. 2011). We shall denote the optimal curve and the corresponding (maximum) value for the AUC criterion by $\mathrm{ROC}^* = \mathrm{ROC}(s^*,.)$ and $\mathrm{AUC}^* = \mathrm{AUC}(s^*)$, where $s^* \in \mathcal{S}^*$. The

statistical counterparts of ROC(s,.) and AUC(s) based on sampling data $\mathcal{D}_n = \{(X_i, Y_i) : 1 \leq i \leq n\}$ are obtained by replacing the class distributions by their empirical versions in the definitions. They are denoted by $\widehat{\mathrm{ROC}}(s,.)$ and $\widehat{\mathrm{AUC}}(s)$ in the sequel.
Piecewise constant scoring rules. In the paper, we will focus on a particular subclass of scoring rules.

Definition 4 (PIECEWISE CONSTANT SCORING RULE) *A scoring rule s is piecewise constant if there exists a finite partition $\mathcal{P}$ of $\mathcal{X}$ such that for all $\mathcal{C} \in \mathcal{P}$, there exists a constant $k_{\mathcal{C}} \in \mathbb{R}$ such that $\forall x \in \mathcal{C}$, $s(x) = k_{\mathcal{C}}$.*

This definition does not provide a unique characterization of the underlying partition. The partition $\mathcal{P}$ is minimal if, for any two of its elements $\mathcal{C} \neq \mathcal{C}'$, we have $k_{\mathcal{C}} \neq k_{\mathcal{C}'}$. The scoring rule conveys an ordering on the cells of the minimal partition.

Definition 5 (RANK OF A CELL) *Let s be a scoring rule and $\mathcal{P}$ the associated minimal partition. The scoring rule induces a ranking $\preceq_s$ over the cells of the partition. For a given cell $\mathcal{C} \in \mathcal{P}$, we define its rank $\mathcal{R}_{\preceq_s}(\mathcal{C}) \in \{1, \ldots, |\mathcal{P}|\}$ as the rank affected by the ranking $\preceq_s$ over the elements of the partition. By convention, we set rank 1 to correspond to the highest score.*

The advantage of the class of piecewise constant scoring rules is that they provide finite rankings on the elements of $\mathcal{X}$ and they will be the key for applying the aggregation procedure.

3. Aggregation of Scoring Rules

In recent years, the issue of summarizing or aggregating various rankings has been a topic of growing interest in the machine-learning community. This evolution was mainly motivated by practical problems in the context of internet applications: design of meta-search engines, collaborative filtering, spam-fighting, *etc.* We refer for instance to Pennock et al. (2000), Dwork et al. (2001), Fagin et al. (2003) and Ilyas et al. (2002). Such problems have led to a variety of results, ranging from the generalization of the mathematical concepts introduced in social choice theory (see Barthélémy and Montjardet 1981 and the references therein) for defining relevant notions of *consensus* between rankings (Fagin et al., 2006), to the development of efficient procedures for computing such "consensus rankings" (Betzler et al., 2008; Mandhani and Meila, 2009; Meila et al., 2007) through the study of probabilistic models over sets of rankings (Fligner and Verducci , Eds.; Lebanon and Lafferty, 2003). Here we consider rank aggregation methods in the perspective of extending the bagging approach to ranking trees.

3.1 The Case of Piecewise Constant Scoring Rules

The ranking rules considered in this paper result from the aggregation of a collection of piecewise constant scoring rules. Since each of these scoring rules is related to a possibly different partition, we are lead to consider a collection of partitions of $\mathcal{X}$. Hence, the aggregated rule needs to be defined on the least fine subpartition of this collection of partitions.

Definition 6 (SUBPARTITION OF A COLLECTION OF PARTITIONS) *Consider a collection of B partitions of $\mathcal{X}$ denoted by $\mathcal{P}_b$, $b = 1, \ldots, B$. A subpartition of this collection is a partition $\mathcal{P}_B$ made*

of nonempty subsets $\mathcal{C} \subset \mathcal{X}$ which satisfy the following constraint : for all $\mathcal{C} \in \mathcal{P}_B$, there exists $(\mathcal{C}_1, \ldots, \mathcal{C}_B) \in \mathcal{P}_1 \times \cdots \times \mathcal{P}_B$ such that

$$\mathcal{C} \subseteq \bigcap_{b=1}^{B} \mathcal{C}_b \ .$$

We denote $\mathcal{P}_B^* = \bigcap_{b \leq B} \mathcal{P}_b$.

One may easily see that $\mathcal{P}_B^*$ is a subpartition of any of the $\mathcal{P}_b$'s, and the largest one in the sense that any partition $\mathcal{P}$ which is a subpartition of $\mathcal{P}_b$ for all $b \in \{1, \ldots, B\}$ is a subpartition of $\mathcal{P}_B^*$. The case where the partitions are obtained from a binary tree structure is of particular interest as we shall consider tree-based piecewise constant scoring rules later on. Incidentally, it should be noticed that, from a computational perspective, the underlying tree structures considerably help in getting the cells of $\mathcal{P}_B^*$ explicitly. We refer to Appendix D for further details.

Now consider a collection of piecewise constant scoring rules s_b, $b = 1, \ldots, B$, and denote their associated (minimal) partitions by $\mathcal{P}_b$. Each scoring rule s_b naturally induces a ranking (or a *pre-order*) $\preceq_b^*$ on the partition $\mathcal{P}_B^*$. Indeed, for all $(\mathcal{C}, \mathcal{C}') \in \mathcal{P}_B^{*2}$, one writes by definition $\mathcal{C} \preceq_b^* \mathcal{C}'$ (respectively, $\mathcal{C} \prec_b^* \mathcal{C}'$) if and only if $\mathcal{C}_b \preceq_b^* \mathcal{C}_b'$ (respectively, $\mathcal{C}_b \prec_b^* \mathcal{C}_b'$) where $(\mathcal{C}_b, \mathcal{C}_b') \in \mathcal{P}_b^2$ are such that $\mathcal{C} \times \mathcal{C}' \subset \mathcal{C}_b \times \mathcal{C}_b'$.

The collection of scoring rules leads to a collection of B rankings on $\mathcal{P}_B^*$. Such a collection is called a *profile* in voting theory. Now, based on this *profile*, we would like to define a "central ranking" or a *consensus*. Whereas the mean, or the median, naturally provides such a summary when considering scalar data, various meanings can be given to this notion for rankings (see Appendix B).

3.2 Probabilistic Measures of Scoring Agreement

The purpose of this subsection is to extend the concept of measures of agreement for rankings to scoring rules defined over a general space $\mathcal{X}$ which is not necessarily finite. In practice, however, we will only consider the case of piecewise constant scoring rules and we shall rely on the definition of the probabilistic Kendall tau.

Notations. We already introduced the notation $\preceq_s$ for the preorder relation over the cells of a partition $\mathcal{P}$ as induced by a piecewise scoring rule s. We shall use the 'curly' notation for the preorder relation $\preccurlyeq_s$ on $\mathcal{X}$ which is described through the following condition: $\forall \mathcal{C}, \mathcal{C}' \in \mathcal{P}$, we have $x \preccurlyeq_s x'$, $\forall x \in \mathcal{C}$, $\forall x' \in \mathcal{C}'$, if and only if $\mathcal{C} \preceq_s \mathcal{C}'$. This is also equivalent to $s(x) \leq s(x')$, $\forall x \in \mathcal{C}$, $\forall x' \in \mathcal{C}'$. We now introduce a measure of similarity for preorders on $\mathcal{X}$ induced by scoring rules s_1 and s_2.

We recall here the definition of the theoretical Kendall τ between two random variables.

Definition 7 (PROBABILISTIC KENDALL τ) *Let (Z_1, Z_2) be two random variables defined on the same probability space. The probabilistic Kendall τ is defined as*

$$\tau(Z_1, Z_2) = 1 - 2d_\tau(Z_1, Z_2) \ ,$$

with:

$$d_\tau(Z_1, Z_2) = \mathbb{P}\{(Z_1 - Z_1') \cdot (Z_2 - Z_2') < 0\} + \frac{1}{2}\mathbb{P}\{Z_1 = Z_1',\, Z_2 \neq Z_2'\} + \frac{1}{2}\mathbb{P}\{Z_1 \neq Z_1',\, Z_2 = Z_2'\}.$$

where (Z_1', Z_2') *is an independent copy of the pair* (Z_1, Z_2).

As shown by the following result, whose proof is left to the reader, the Kendall τ for the pair $(s(X), Y)$ is related to AUC(s).

Proposition 8 *We use the notation* $p = \mathbb{P}\{Y = 1\}$. *For any real-valued scoring rule s, we have:*

$$\frac{1}{2}\left(1 - \tau(s(X), Y)\right) = 2p(1-p)\left(1 - \mathrm{AUC(s)}\right) + \frac{1}{2}\mathbb{P}\{s(X) \neq s(X') \ , \ Y = Y'\} \ .$$

For given scoring rules s_1 and s_2 and considering the probabilistic Kendall tau for random variables $s_1(X)$ and $s_2(X)$, we can set: $d_X(s_1, s_2) = d_\tau(s_1(X), s_2(X))$. One may easily check that d_X defines a distance between the orderings $\preccurlyeq_{s_1}$ and $\preccurlyeq_{s_2}$ induced by s_1 and s_2 on the set $\mathcal{X}$ (which is supposed to coincide with the support of the distribution of X). The following proposition shows that the deviation between scoring rules in terms of AUC is controlled by a quantity involving the probabilistic agreement based on Kendall tau.

Proposition 9 *(*AUC AND KENDALL τ*)* *Assume* $p \in (0,1)$. *For any scoring rules* s_1 *and* s_2 *on* $\mathcal{X}$, *we have:*

$$\left|\mathrm{AUC}(s_1) - \mathrm{AUC}(s_2)\right| \leq \frac{d_X(s_1, s_2)}{2p(1-p)} = \frac{1 - \tau_X(s_1, s_2)}{4p(1-p)} \ .$$

The converse inequality does not hold in general. Indeed, scoring rules with same AUC may yield to different rankings. However, the following result guarantees that a scoring rule with a nearly optimal AUC is close to the optimal scoring rules in the sense of Kendall tau, under the additional assumption that the noise condition introduced in Clémençon et al. (2008) is fulfilled.

Proposition 10 *(*KENDALL τ AND OPTIMAL AUC*)* *Assume that the random variable* $\eta(X)$ *is continuous and that there exist* $c < \infty$ *and* $a \in (0,1)$ *such that:*

$$\forall x \in \mathcal{X}, \ \mathbb{E}\left[\left|\eta(X) - \eta(x)\right|^{-a}\right] \leq c \ . \tag{1}$$

Then, we have, for any scoring rule s and any optimal scoring rule $s^* \in \mathcal{S}^*$*:*

$$1 - \tau_X(s^*, s) \leq C \cdot \left(\mathrm{AUC}^* - \mathrm{AUC(s)}\right)^{a/(1+a)} \ ,$$

with $C = 3 \cdot c^{1/(1+a)} \cdot (2p(1-p))^{a/(1+a)}$.

Remark 11 *(*ON THE NOISE CONDITION*) As shown in previous work, the condition (1) is rather weak. Indeed, it is fulfilled for any* $a \in (0,1)$ *as soon the probability density function of* $\eta(X)$ *is bounded (see Corollary 8 in Clémençon et al. 2008).*

The next result shows the connection between the Kendall tau distance between preorders on $\mathcal{X}$ induced by piecewise constant scoring rules s_1 and s_2 and a specific notion of distance between the rankings $\preceq_{s_1}$ and $\preceq_{s_2}$ on $\mathcal{P}$.

Lemma 12 *Let* s_1, s_2, *two piecewise constant scoring rules. We have:*

$$d_X(s_1, s_2) = 2 \sum_{1 \leq k < l \leq K} \mu(C_k)\mu(C_l) \cdot U_{k,l}(\preceq_{s_1}, \preceq_{s_2}) \ , \tag{2}$$

where, for two orderings $\preceq$, $\preceq'$ on a partition of cells $\{C_k : k = 1,\ldots,K\}$, we have:

$$U_{k,l}(\preceq,\preceq') = \mathbb{I}\{(\mathcal{R}_{\preceq}(C_k) - \mathcal{R}_{\preceq}(C_l))(\mathcal{R}_{\preceq'}(C_k) - \mathcal{R}_{\preceq'}(C_l)) < 0\} + \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq}(C_k) = s_{\preceq}(C_l),\ \mathcal{R}_{\preceq'}(C_k) \neq \mathcal{R}_{\preceq'}(C_l)\} + \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq'}(C_k) = \mathcal{R}_{\preceq'}(C_l),\ \mathcal{R}_{\preceq}(C_k) \neq \mathcal{R}_{\preceq}(C_l)\}\ .$$

The proof is straightforward and thus omitted.

Notice that the term $U_{k,l}(\preceq_{s_1},\preceq_{s_2})$ involved in Equation (2) is equal to 1 when the cells C_k and C_l are not sorted in the same order by s_1 and s_2 (in absence of ties), to $1/2$ when they are tied for one ranking but not for the other, and to 0 otherwise. As a consequence, the agreement $\tau_X(s_1,s_2)$ may be viewed as a "weighted version" of the rate of concordant pairs of the cells of $\mathcal{P}$ measured by the classical Kendall τ (see the Appendix B). A statistical version of $\tau_X(s_1,s_2)$ is obtained by replacing the values of $\mu(C_k)$ by their empirical counterparts in Equation (2). We thus set:

$$\widehat{\tau}_X(s_1,s_2) = 1 - 2\widehat{d}_X(s_1,s_2), \tag{3}$$

where $\widehat{d}_X(s_1,s_2) = 2/(n(n-1))\sum_{i<j} K(X_i,X_j)$ is a U-statistic of degree 2 with symmetric kernel given by:

$$K(x,x') = \mathbb{I}\{(s_1(x)-s_1(x'))\cdot(s_2(x)-s_2(x')) < 0\} + \frac{1}{2}\mathbb{I}\{s_1(x) = s_1(x'),\ s_2(x) \neq s_2(x')\} + \frac{1}{2}\mathbb{I}\{s_1(x) \neq s_1(x'),\ s_2(x) = s_2(x')\}\ .$$

Remark 13 *Other measures of agreement between $\preccurlyeq_{s_1}$ and $\preccurlyeq_{s_2}$ could be considered alternatively. For instance the definitions previously stated can easily be extended to the Spearman correlation coefficient $\rho_X(s_1,s_2)$ (see Appendix B), that is the linear correlation coefficient between the random variables $F_{s_1}(s_1(X))$ and $F_{s_2}(s_2(X))$, where F_{s_i} denotes the cdf of $s_i(X)$, $i \in \{1,\ 2\}$.*

3.3 Median Rankings

The method for aggregating rankings we consider here relies on the so-called *median procedure*, which belongs to the family of *metric aggregation procedures* (see Barthélémy and Montjardet 1981 for further details). Let $d(.,.)$ be some metric or dissimilarity measure on the set of rankings on a finite set $\mathcal{Z}$. By definition, a *median ranking* among a profile $\Pi = \{\preceq_k:\ 1 \leq k \leq K\}$ with respect to d is any ranking $\preceq_{med}$ on $\mathcal{Z}$ that minimizes the sum $\Delta_\Pi(\preceq) \stackrel{def}{=} \sum_{k=1}^K d(\preceq,\preceq_k)$ over the set $\mathbf{R}(\mathcal{Z})$ of all rankings $\preceq$ on $\mathcal{Z}$:

$$\Delta_\Pi(\preceq_{med}) = \min_{\preceq:\ \text{ranking on } \mathcal{Z}} \Delta_\Pi(\preceq).$$

Notice that, when $\mathcal{Z}$ is of cardinality $N < \infty$, there are

$$\#\mathbf{R}(\mathcal{Z}) = \sum_{k=1}^{N} (-1)^k \sum_{m=1}^{k} (-1)^m \left(\begin{array}{c} k \\ m \end{array} \right) m^N$$

possible rankings on $\mathcal{Z}$ (that is the sum over k of the number of surjective mappings from $\{1,\ldots,N\}$ to $\{1,\ldots,k\}$) and in most cases, the computation of (metric) median rankings leads to NP-hard

combinatorial optimization problems (see Wakabayashi 1998, Hudry 2004, Hudry 2008 and the references therein). It is worth noticing that a median ranking is far from being unique in general. One may immediately check for instance that any ranking among the profile made of all rankings on $\mathcal{Z} = \{1, 2\}$ is a median in Kendall sense, that is, for the metric d_τ. From a practical perspective, acceptably good solutions can be computed in a reasonable amount of time by means of metaheuristics such as simulated annealing, genetic algorithms or tabu search (see Spall 2003). The description of these computational aspects is beyond the scope of the present paper (see Charon and Hudry 1998 or Laguna et al. 1999 for instance). We also refer to recent work in Klementiev et al. (2009).

When it comes to preorders on a set $\mathcal{X}$ of infinite cardinality, defining a notion of aggregation becomes harder. Given a pseudo-metric such as d_τ and $B \geq 1$ scoring rules $s_1, \ldots, s_B$ on $\mathcal{X}$, the existence of $\bar{s}$ in $\mathcal{S}$ such that $\sum_{b=1}^{B} d_\tau(\bar{s}, s_b) = \min_s \sum_{b=1}^{B} d_\tau(s, s_b)$ is not guaranteed in general. However, when considering piecewise constant scoring rules with corresponding finite subpartition $\mathcal{P}$ of $\mathcal{X}$, the corresponding preorders are in one-to-one correspondence with rankings on $\mathcal{P}$ and the minimum distance is thus effectively attained.

Aggregation of piecewise constant scoring rules. Consider a finite collection of piecewise constant scoring rules $\Sigma_B = \{s_1, \ldots, s_B\}$ on $\mathcal{X}$, with $B \geq 1$.

Definition 14 (TRUE MEDIAN SCORING RULE). *Let $\mathcal{S}$ be a collection of scoring rules. We call $\bar{s}_B$ a* median scoring rule *for Σ_B with respect to $\mathcal{S}$ if*

$$\bar{s}_B = \arg\min_{s \in \mathcal{S}} \Delta_B(s),$$

where $\Delta_B(s) = \sum_{b=1}^{B} d_X(s, s_b)$ for $s \in \mathcal{S}$.

The empirical median scoring rule is obtained in a similar way, but the true distance d_X is replaced by its empirical counterpart $d_{\widehat{\tau}_X}$, see Equation (3).

The ordinal approach. Metric aggregation procedures are not the only way to summarize a profile of rankings. The so-called "ordinal approach" provides a variety of alternative techniques for combining rankings (or, more generally, *preferences*), returning to the famous "Arrow's voting paradox". The ordinal approach consists of a series of duels (i.e., *pairwise comparisons*) as in Condorcet's method or successive tournaments as in the proportional voting Hare system, see Fishburn (1973). Such approaches have recently been the subject of a good deal of attention in the context of *preference learning* (also referred to as methods for *ranking by pairwise comparison*, see Hüllermeier et al. 2008 for instance).

Ranks vs. Rankings. Let $\Sigma_B = \{s_1, \ldots, s_B\}$, $B \geq 1$, be a collection of piecewise constant scoring rules and $\mathbf{X}'^{(m)} = \{X'_1, \ldots, X'_m\}$ a collection of $m \geq 1$ i.i.d. copies of the input variable X. When it comes to rank the observations X'_i "consensually", two strategies can be considered: (i) compute a "median ranking rule" based on the B rankings of the cells for the largest subpartition and use it for ranking the new data as previously described, or (ii) compute, for each scoring rule s_b, the related rank vector of the data set $\mathbf{X}'^{(m)}$, and then a "median rank vector", that is, a median ranking on the set $\mathbf{X}'^{(m)}$ (data lying in a same cell of the largest subpartition being tied). Although they are not equivalent, these two methods generally produce similar results, especially when m is large. Indeed, considering medians in the sense of probabilistic Kendall τ, it is sufficient to notice that the Kendall τ distance d_τ between rankings on $\mathbf{X}'^{(m)}$ induced by two piecewise constant rules s_1 and s_2 can be viewed as an empirical estimate of $d_X(s_1, s_2)$ based on the data set $\mathbf{X}'^{(m)}$. Now assume the

collection Σ_B is obtained from training data $\mathcal{D}_n$. The difference between (i) and (ii) is that (i) does not use the data to be ranked $\mathbf{X}'^{(m)}$ but only relies on training data $\mathcal{D}_n$. However, when both the size of the training sample $\mathcal{D}_n$ and of the test data set $\mathbf{X}'^{(m)}$ are large, the two approaches lead to the optimization of related quantities.

4. Consistency of Aggregated Scoring Rules

We now provide statistical results for the aggregated scoring rules in the spirit of random forests (Breiman, 2001). In the context of classification, consistency theorems were derived in Biau et al. (2008). Conditions for consistency of piecewise constant scoring rules have been studied in Clémençon and Vayatis (2009c) and Clémençon et al. (2011). Here, we address the issue of AUC consistency of scoring rules obtained as medians over a profile of consistent randomized scoring rules for the (probabilistic) Kendall τ distance. A *randomized scoring rule* is a random element of the form $\widehat{s}_n(\cdot, Z)$, depending on both the training sample $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ and a random variable Z, taking values over a measurable space $\mathcal{Z}$, independent of $\mathcal{D}_n$, which describes the randomization mechanism.

The AUC of a randomized scoring rule $\widehat{s}_n(\cdot, Z)$ is given by:

$$\mathrm{AUC}(\widehat{s}_n(\cdot,Z)) = \mathbb{P}\{\widehat{s}_n(X,Z) < \widehat{s}_n(X',Z) \mid (Y,Y') = (-1,+1)\} + \frac{1}{2}\mathbb{P}\{\widehat{s}_n(X,Z) = \widehat{s}_n(X',Z) \mid (Y,Y') = (-1,+1)\},$$

where the conditional probabilities are taken over the joint probability of independent copies (X,Y) and (X,Y') and Z, given the training data $\mathcal{D}_n$.

Definition 15 (AUC-CONSISTENCY) *The randomized scoring rule $\widehat{s}_n$ is said to be* AUC*-consistent (respectively, strongly* AUC*-consistent) when the convergence*

$$\mathrm{AUC}(\widehat{s}_n(\cdot,Z)) \to \mathrm{AUC}^* \text{ as } \mathrm{n} \to \infty ,$$

holds in probability (respectively, almost-surely).

Let $B \geq 1$. Given $\mathcal{D}_n$, one may draw B i.i.d. copies $Z_1, \ldots, Z_B$ of Z, yielding the collection $\widehat{\Sigma}_B$ of scoring rules $\widehat{s}_n(\cdot, Z_j)$, $1 \leq j \leq B$. Let $\mathcal{S}$ be a collection of scoring rules and suppose that $\bar{s}_B$ is a *median scoring rule* for the profile $\widehat{\Sigma}_B$ with respect to $\mathcal{S}$ in the sense of Definition 14. The next result shows that AUC-consistency is preserved for a median scoring rule of AUC-consistent randomized scoring rules.

Theorem 16 *(*CONSISTENCY AND AGGREGATION*) Set $B \geq 1$. Consider a class $\mathcal{S}$ of scoring rules. Assume that:*

- *the assumptions on the distribution of (X,Y) in Proposition 10 are fulfilled.*
- *the randomized scoring rule $\widehat{s}_n(\cdot, Z)$ is* AUC*-consistent (respectively, strongly* AUC*-consistent).*
- *for all $n, B \geq 1$, and for any sample $\mathcal{D}_n$, there exists a median scoring rule $\bar{s}_B \in \mathcal{S}$ for the collection $\{\widehat{s}_n(\cdot, Z_j), 1 \leq j \leq B\}$ with respect to $\mathcal{S}$.*

- *we have* $\mathcal{S}^* \cap \mathcal{S} \neq \emptyset$.

Then, the aggregated scoring rule $\bar{s}_B$ *is* AUC*-consistent (respectively, strongly* AUC*-consistent).*

We point out that the last assumption which states that the class $\mathcal{S}$ of candidate median scoring rules contains at least one optimal scoring function can be removed at the cost of an extra bias term in the rate bound. Consistency results are then derived by picking the median scoring rule, for each n, in a class $\mathcal{S}_n$ such that there exists a sequence $\tilde{s}_n \in \mathcal{S}_n$ which fulfills $\mathrm{AUC}(\tilde{\mathrm{s}}_\mathrm{n}) \to \mathrm{AUC}^*$ as $n \to \infty$. This remark covers the special case where $\widehat{s}_n(\cdot, Z)$ is a piecewise constant scoring rule with a range of cardinality $k_n \uparrow \infty$ and the median is taken over the set $\mathcal{S}_n$ of scoring functions with range of cardinality less than k_n^B. The bias is then of order $1/k_n^{2B}$ under mild smoothness conditions on ROC^*, as shown by Proposition 7 in Clémençon and Vayatis (2009b).

From a practical perspective, median computation is based on empirical versions of the probabilistic Kendall τ involved (see Equation (3)). The following result shows the existence of scoring rules that are asymptotically median with respect to d_X, provided that the class $\mathcal{S}$ over which the median is computed is not too complex. Here we formulate the result in terms of a VC major class of functions of finite dimension (see Dudley 1999 for instance). We first introduce the following notation, for any $s \in \mathcal{S}$:

$$\widehat{\Delta}_{B,m}(s) = \sum_{j=1}^{B} \widehat{d}_X(s, s_j) \ ,$$

where the estimate $\widehat{d}_X$ of d_X is based on $m \geq 1$ independent copies of X.

Theorem 17 *(*EMPIRICAL MEDIAN COMPUTATION*) Fix* $B \geq 1$. *Let* $\Sigma_B = \{s_1, \ldots, s_B\}$ *be a finite collection scoring rules and* $\mathcal{S}$ *a class of scoring rules which is a VC major class. We consider the empirical median scoring rule* $\widetilde{s}_m = \arg\min_{s \in \mathcal{S}} \widehat{\Delta}_{B,m}(s)$. *Then, as* $m \to \infty$, *we have*

$$\Delta_B(\widetilde{s}_m) \to \min_{s \in \mathcal{S}} \Delta_B(s) \text{ with probability one .}$$

The empirical aggregated scoring rule we consider in the next result relies on two data samples. The training sample $\mathcal{D}_n$, completed by the randomization on Z, leads to a collection of scoring rules which are instances of the randomized scoring rule. Then a sample $\mathbf{X}'^{(m)} = \{X'_1, \ldots, X'_m\}$ is used to compute the empirical median. Combining the two preceding theorems, we finally obtain the consistency result for the aggregated scoring rule.

Corollary 18 *Fix* $B \geq 1$ *and* $\mathcal{S}$ *a VC major class of scoring rules. Consider a training sample* $\mathcal{D}_n$ *of size n with i.i.d. copies of* (X, Y) *and a sample* $\mathbf{X}'^{(m)}$ *of size m with i.i.d. copies of X. We consider the collection* $\widehat{\Sigma}_B$ *of randomized scoring rules* $\widehat{s}_n(\cdot, Z_j)$ *in* $\mathcal{S}$ *built out of* $\mathcal{D}_n$ *and we introduce the empirical median of* $\widehat{\Sigma}_B$ *with respect to* $\mathcal{S}$ *obtained by using the test set* $\mathbf{X}'^{(m)}$. *We denote this fully empirical median scoring rule by* $\widehat{s}_{n,m}$. *If the assumptions of Theorem 16 are satisfied, then we have:*

$$\mathrm{AUC}(\widehat{s}_{n,m}) \xrightarrow{P} \mathrm{AUC}^* \text{ as } \mathrm{n}, \mathrm{m} \to \infty \ .$$

The results stated above can be extended to any median scoring rule based on a pseudo-metric d on the set of preorders on $\mathcal{S}$ which is equivalent to d_X, that is, such that $c_1 d_X \leq d \leq c_2 d_X$, with $0 < c_1 \leq c_2 < \infty$. Moreover, other complexity assumptions about the class $\mathcal{S}$ over which optimization is performed could be considered (see Clémençon et al. 2008). The present choice of VC major classes captures the complexity of scoring rules which will be considered in the next section (see Proposition 6 in Clémençon et al. 2011).

5. Ranking Forests

In this section, we introduce an implementation of the principles described in the previous sections for the aggregation of scoring rules. Here we focus on specific piecewise constant scoring rules based on ranking trees (Clémençon and Vayatis, 2009c; Clémençon et al., 2011). We propose various schemes for randomizing the features of these trees. We eventually describe the RANKING FOREST algorithm which extends to bipartite ranking the celebrated RANDOM FORESTS algorithm (Breiman, 1996; Amit and Geman, 1997; Breiman, 2001).

5.1 Tree-structured Scoring Rules

We consider piecewise constant scoring rules which can be represented in a left-right oriented binary tree. We recall that, in the context of classification, decision trees are very useful as they offer the possibility of interpretation for the selected classification rule. In the presence of classification data, one may entirely characterize a classification rule by means of a partition $\mathcal{P}$ of the input space $\mathcal{X}$ and a training set $\mathcal{D}_n = \{(X_i, Y_i) : \ 1 \leq i \leq n\}$ of i.i.d. copies of the pair (X, Y) through a *majority voting scheme*. Indeed, a new instance $x \in \mathcal{X}$ would receive the label corresponding to the most frequent one among the data points X_i within the cell $\mathcal{C} \in \mathcal{P}$ such that $x \in \mathcal{C}$. However, in bipartite ranking, the notion of local majority vote makes no sense since the ranking problem is of global nature. As a matter of fact, the issue is to rank the cells of the partition with respect to each other. It is assumed that ties among the ordered cells can be observed in the subsequent analysis and the usual MID-RANK convention is adopted. We refer to the Appendix A for a rigorous definition of the notion of *ranking* in the case of ties. We also point out that the term *partial ranking* is often used in this context (see Diaconis 1989, Fagin et al. 2006).

By restricting the search of candidates to the collection of piecewise constant scoring rules, the learning problem boils down here to finding a partition $\mathcal{P} = \{\mathcal{C}_k\}_{1 \leq k \leq K}$ of $\mathcal{X}$, with $1 \leq K < \infty$, together with a ranking $\preceq_{\mathcal{P}}$ of the $\mathcal{C}_k$'s (i.e., a preorder on $\mathcal{P}$), so that the ROC curve of the scoring rule given by

$$s_{\mathcal{P}, \preceq_{\mathcal{P}}}(x) = \sum_{k=1}^{K} (K - \mathcal{R}_{\preceq_{\mathcal{P}}}(\mathcal{C}_k) + 1) \cdot \mathbb{I}\{x \in \mathcal{C}_k\}$$

be as close as possible of ROC*, where $\mathcal{R}_{\preceq_{\mathcal{P}}}(\mathcal{C}_k)$ denotes the rank of $\mathcal{C}_k$, $1 \leq k \leq K$, among all cells of $\mathcal{P}$ according to $\preceq_{\mathcal{P}}$.

We now describe such scoring rules in the case where the partition arises from a tree structure. For such a partition, a ranking of the cells can be simply defined by equipping the tree with a left-right orientation. In order to describe how a ranking tree can be built so as to maximize AUC, further concepts are required. By *master ranking tree* $\mathcal{T}_D$, here we mean a complete, left-right oriented, rooted binary tree with depth $D \geq 1$. At depth $d = 0$, the entire input space $\mathcal{C}_{0,0} = \mathcal{X}$ forms its root. Every non terminal node (d, k), with $0 \leq d < D$ and $0 \leq k < 2^d$, is in correspondence with a subset $C_{d,k} \subset \mathcal{X}$, and has two siblings, each one corresponding to a subcell obtained by splitting $C_{d,k}$: the *left sibling* $C_{d+1,2k}$ is related to the leaf $(d+1, 2k)$, while the *right sibling* $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$ is related to the leaf $(d+1, 2k+1)$ in the tree structure. We point out that an asymmetry is introduced at this point as the left sibling is assumed to have a lower rank (or higher score) than the right sibling in the ranking of the partition's cells. With this convention, it is easy to use any subtree $\mathcal{T} \subset \mathcal{T}_D$ as a ranking rule. A ranking of the terminal cells naturally results from the left-right orientation of the tree, the top of the list being represented by the cell in the bottom left corner of the tree, and is

related to the scoring rule defined by: $\forall x \in \mathcal{X}$,

$$s_{\mathcal{T}}(x) = \sum_{(d,k)\text{: terminal node of } \mathcal{T}} (2^D - 2^{D-d}k) \cdot \mathbb{I}\{x \in C_{d,k}\} \ .$$

The score value $s_{\mathcal{T}}(x)$ can be computed in a top-down manner, using the underlying "heap" structure. Starting from the initial value 2^D at the root node, at each subsequent inner node (d,k), $2^{D-(d+1)}$ is subtracted to the current value of the score if x moves down to the right sibling $(d+1, 2k+1)$, whereas one leaves the score unchanged if x moves down to the left sibling. The procedure is depicted in Figure 2.

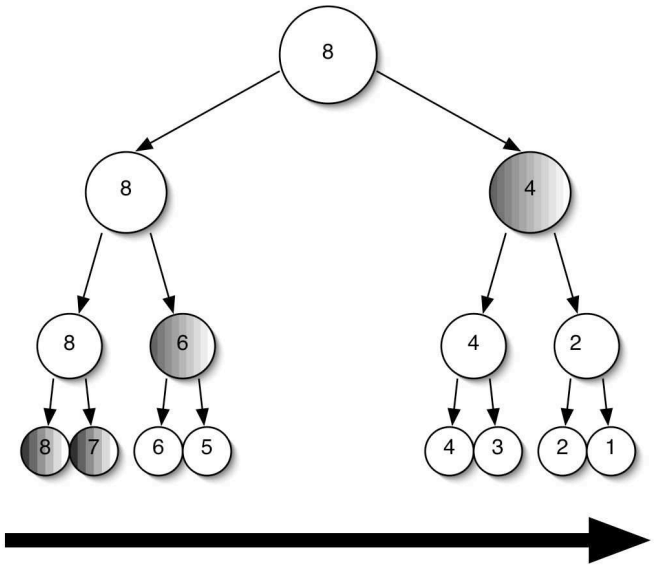


Figure 2: Ranking tree - the ranks can be read on the leaves of the tree from left (8 is the highest rank/score) to right (1 corresponds to the smallest rank/score). In case of a pruned tree (such as the one with leaves taken to be the shaded nodes), the orientation is conserved.

5.2 Feature Randomization in TREERANK

The concept of *bagging* (for **b**ootstrap **agg**regat**ing** technique) was introduced in Breiman (1996). The major novelty in the RANDOM FOREST method (Breiman, 2001) consisted in randomizing the features used for recursively splitting the nodes of the classification/regression trees involved in the committee-based prediction procedure. Our reference method for aggregating tree-based scoring rules is the TREERANK procedure (we refer to the Appendix and the papers Clémençon and Vayatis 2009c, Clémençon et al. 2011 for a full description). Beyond the specific structure of the master ranking tree, an additional ingredient in the growing stage is the splitting criterion. It turns out that a natural choice is a data-dependent and cost-sensitive classification error functional and its optimization can be performed with any binary classification method. This procedure for node splitting is called LEAFRANK. We point out that LEAFRANK implements a classifier and when this

classifier is chosen to be a decision tree, this permits an additional randomization step. We thus propose two possible feature randomization schemes F_T for TREERANK and F_L for LEAFRANK.

F_T: *At each node (d,k) of the master ranking tree $\mathcal{T}_D$, draw at random a set of $q_0 \leq q$ indexes $\{i_1,\ldots,i_{q_0}\} \subset \{1,\ldots,q\}$. Implement the* LEAFRANK *splitting procedure based on the descriptor $(X^{(i_1)},\ldots,X^{(i_{q_0})})$ to split the cell $C_{d,k}$.*

F_L: *For each node (d,k) of the master ranking tree $\mathcal{T}_D$, at each node of the cost-sensitive classification tree describing the split of the cell $C_{d,k}$ into two children, draw at random a set of $q_1 \leq q$ indexes $\{j_1,\ldots,j_{q_1}\} \subset \{1,\ldots,q\}$ and perform an axis-parallel cut using the descriptor $(X^{(j_1)},\ldots,X^{(j_{q_1})})$.*

We underline that, of course, the randomization strategy F_T can be applied to the TREERANK algorithm whatever the classification technique chosen for the splitting step. In addition, when the latter is itself a tree-based method, these randomization procedures do not exclude each other. At each node (d,k) of the ranking tree, one may first draw at random a collection $\mathcal{F}_{d,k}$ of q_0 features and then, when growing the cost-sensitive classification tree describing $C_{d,k}$'s split, divide each node based on a sub-collection of $q_1 \leq q_0$ features drawn at random among $\mathcal{F}_{d,k}$.

5.3 The RANKING FOREST Algorithm

Now that the rationale behind the RANKING FOREST procedure has been given, we describe its successive steps in detail. Based on a training sample $\mathcal{D}_n = \{(X_1,Y_1),\ldots,(X_n,Y_n)\}$, the algorithm is performed in three stages, as follows.

RANKING FOREST

1. **Parameters.** B number of bootstrap replicates, n^* bootstrap sample size, TREERANK tuning parameters (depth D and presence/absence of pruning), (F_T,F_L) feature randomization strategy, d pseudo-metric.

2. **Bootstrap profile makeup.**

   (a) (RESAMPLING STEP.) Build B independent bootstrap samples $\mathcal{D}_1^*, \ldots, \mathcal{D}_B^*$, by drawing with replacement $n^* \cdot B$ pairs among the original training sample $\mathcal{D}_n$.

   (b) (RANDOMIZED TREERANK.) For $b = 1, \ldots, B$, run TREERANK combined with the feature randomization method (F_T,F_L) based on the sample $\mathcal{D}_b^*$, yielding the ranking tree $\mathcal{T}_b^*$, related to the partition $\mathcal{P}_b^*$ of the space $\mathcal{X}$.

3. **Aggregation.** Compute the largest subpartition partition $\mathcal{P}^* = \bigcap_{b=1}^B \mathcal{P}_b^*$. Let $\preceq_b^*$ be the ranking of the cells of $\mathcal{P}^*$ induced by $\mathcal{T}_b^*$, $b = 1, \ldots, B$. Compute a median ranking $\preceq^*$ related to the bootstrap profile $\Pi^* = \{\preceq_b^*:\ 1 \leq b \leq B\}$ with respect to the metric d on $\mathbf{R}(\mathcal{P}^*)$:

$$\preceq^* = \operatorname*{arg\,min}_{\preceq \in \mathbf{R}(\mathcal{P}^*)} d_{\Pi^*}(\preceq),$$

   as well as the scoring rule $s_{\preceq^*,\mathcal{P}^*}(x)$.

Remark 19 *(*ON TUNING PARAMETERS.*) As mentioned in 3.3, aggregating ranking rules is computationally expensive. The empirical results displayed in Section 6 suggest to aggregate several dozens of randomized ranking trees of moderate, or even small, depth built from bootstrap samples of size $n^* \leq n$.*

Remark 20 *(*"PLUG-IN" BAGGING.*) As pointed out in Clémençon and Vayatis (2009c) (see Remark 6 therein), given an ordered partition $(\mathcal{P}, \mathcal{R}_{\mathcal{P}})$ of the feature space $\mathcal{X}$, a "plug-in" estimate of the (optimal scoring) function $S = H_\eta \circ \eta$ can be automatically deduced from any ordered partition (or piecewise constant scoring rule equivalently) and the data $\mathcal{D}_n$, where H_η denotes the conditional cdf of $\eta(X)$ given $Y = -1$. This scoring rule is somehow canonical in the sense that, given $Y = -1$, $H(X)$ is distributed as a uniform r.v. on $[0,1]$, with H being the conditional distribution of X. Considering a partition $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$ equipped with a ranking $\mathcal{R}_{\mathcal{P}}$, the plug-in estimate is given by*

$$\widehat{S}_{\mathcal{P},\mathcal{R}_{\mathcal{P}}}(x) = \sum_{k=1}^{K} \widehat{\alpha}(R_k) \cdot \mathbb{I}\{x \in C_k\}, \ \ x \in \mathcal{X},$$

where $R_k = \bigcup_{l:\, \mathcal{R}(k) \leq \mathcal{R}(l)} C_l$. Notice that, as a scoring rule, $\widehat{S}_{\mathcal{P},\mathcal{R}_{\mathcal{P}}}$ yields the same ranking as $s_{\mathcal{P},\mathcal{R}_{\mathcal{P}}}$, provided that $\widehat{\alpha}(C_k) > 0$ for all $k = 1, \ldots, K$. Adapting the idea proposed in Section 6.1 of Breiman (1996) in the classification context, an alternative to the rank aggregation approach proposed here naturally consists in computing the average of the piecewise-constant scoring rules $\widetilde{S}^*_{\mathcal{T}^*_b}$ thus defined by the bootstrap ranking trees and consider the rankings induced by the latter. This method we call "plug-in bagging" is however less effective in many situations, due to the inaccuracy/variability of the probability estimates involved.

Ranking stability. Let $\Theta = \mathcal{X} \times \{-1, +1\}$. From the view developed in this paper, a ranking algorithm is a function $\mathbf{S}$ that maps any data sample $\mathcal{D}_n \in \Theta^n$, $n \geq 1$, to a scoring rule $\widehat{s}_n$. In the ranking context, we will say that a learning algorithm is "stable" when the preorder on $\mathcal{X}$ it outputs is not much affected by small changes in the training set. We propose a natural way of measuring ranking (in)stability, through the computation of the following quantity:

$$\mathbf{Instab}_n(\mathbf{S}) = \mathbb{E}\left(d_X\left(\widehat{s}_n, \widehat{s}'_n\right)\right) \ , \tag{4}$$

where the expectation is taken over two independent training samples $\mathcal{D}_n$ and $\mathcal{D}'_n$, both made of n i.i.d. copies of the pair (X,Y), and $\widehat{s}_n = \mathbf{S}(\mathcal{D}_n)$, $\widehat{s}'_n = \mathbf{S}(\mathcal{D}'_n)$. Incidentally, we highlight the fact that the bootstrap stage of RANKING FOREST can be used for assessing the stability of the base ranking algorithm. Indeed, set $\widehat{s}^{(b)}_{n^*} = \mathbf{S}(\mathcal{D}^*_b)$ and $\widehat{s}^{(b')}_{n^*} = \mathbf{S}(\mathcal{D}^*_{b'})$ obtained from two bootstrap samples. Then, the quantity:

$$\widehat{\mathbf{Instab}}_n(\mathbf{S}) = \frac{2}{B(B-1)} \sum_{1 \leq b < b' \leq B} \widehat{d}_X\left(\widehat{s}^{(b)}_{n^*}, \widehat{s}^{(b')}_{n^*}\right) \ ,$$

can be possibly interpreted as a bootstrap estimate of (4).

We finally underline that the outputs of the RANKING FOREST can also be used for monitoring ranking performance, in an analogous fashion to RANDOM FOREST in the classification/regression context (see Section 3.1 in Breiman 2001 and the references therein). An *out-of-bag* estimate of the AUC criterion can be obtained by considering, for all pairs (X,Y) and (X',Y') in the original training sample, those ranking trees that are built from bootstrap samples containing neither of them, avoiding this way the use of a test data set.

6. Numerical Experiments

The purpose of this section is to measure the impact of aggregation with resampling and feature randomization on the performance of the TREERANK/LEAFRANK procedure.
Data sets. We have considered artificial data sets where class-conditional distributions of X goven $Y = \pm 1$ are gaussian in dimensions 10 and 20. Three examples are considered here:

- *RF 10* - class-conditional distributions have the same means ($\mu_+ = \mu_- = 0$) but different covariance matrices ($\Sigma_+ = \mathbf{Id}_{10}$ and $\Sigma_- = 1.023 \cdot \mathbf{Id}_{10}$); optimal AUC is $\mathrm{AUC}^* = 0.76$;
- *RF 20* - class-conditional distributions have different mean vectors ($||\mu_+ - \mu_-|| = 0.9$) and covariance matrices ($\Sigma_+ = \mathbf{Id}_{20}$ and $\Sigma_- = 1.23 \cdot \mathbf{Id}_{20}$); optimal AUC is $\mathrm{AUC}^* = 0.77$;
- *RF 10 sparse* - class-conditional distributions have a 6-dimensional marginal distribution in common, and the regression function $\eta(x)$ depends on four components of the input vector X onlyoptimal AUC is $\mathrm{AUC}^* = 0.89$.

With these data sets, the series of experiments below capture the influence on ranking performance of separability, dimension, and sparsity.
Sample sizes. In order to quantify the impact of bagging and random feature selection on the accuracy/stability of the resulting ranking rule, the algorithm has been run under various configurations for each data set on 30 independent training samples for each sample size ranging from $n = 250$ to $n = 3000$. The test sample was taken of size 3000 in all experiments.
Variants of TREERANK *and parameters.* In the intensive comparisons we have performed, we have considered the following variants:

- Plain TREERANK/LEAFRANK - in this version, all input dimensions are involved in the splitting stage; the maximum depth of the master ranking tree is 10, and the maximum depth of the ranking tree using orthogonal splits in the LEAFRANK procedure is 8 for the use case *RF 10 sparse* and also 10 for the two others.
- BAGGING RANKING TREES - the *bagging* version uses the plain TREERANK/LEAFRANK as described above with bootstrap samples of size $B = 20$, $B = 50$, and $B = 100$.
- RANKING FORESTS - the *forest* version involves additional parameters for feature randomization which can affect both the master ranking tree (F_T for TREERANK) and the splitting rule (F_L for LEAFRANK); these parameters indicate the number of dimensions randomly chosen along which the best split is chosen ; we have tried six different sets of parameters (Cases 1 to 6) where F_T takes values 3, 5, and 10 (or 20 for the data set *RF 20*), and F_L takes values 1, 3, and 5 (plus 10 for the data set *RF 20*); bootstrap samples are chosen of size $B = 1$ (single tree with feature randomization), $B = 20$, $B = 50$, and $B = 100$.

In the case of bagging and forests, aggregation is performed by taking the pointwise median value of ranks for the collection of ranking trees which have been estimated on each bootstrap sample. This choice allows for very fast evaluations of the aggregated scoring rule (see the last paragraph of Section 3.3 for a justification).
Performance. For each variant and each set of parameters and sample size, we performed 30 replications using independent training sets. These replications are used to derive performance results on a same test set. Performance is measured through a collection of indicators:

- $\overline{\text{AUC}}$ and $\widehat{\sigma}^2$ - Average AUC and standar type error are computed based on the test sample results over the 30 replications;

- ΔEnv - this indicator quantifies the accuracy of the variant through the relative improvement of the envelope on the ROC curve over the 30 replications compared to the plain TREERANK/LEAFRANK (e.g., if $\Delta\text{Env} = -30\%$ for BAGGING it means that the envelope of the ROC curve is 30% narrower than with TREERANK); the more negative the better the performance accuracy;

- $\mathbf{Instab}_\tau$ - Instability measure applied to the ranking algorithm (e.g., Ranking Forest), estimate of (4), which reproduces the quantity $\widehat{\mathbf{Instab}}_n(\mathbf{S})$ using the Kendal τ as a distance; the smaller the quantity the more stable the method;

- DCG and AVE - the *Discounted Cumulative Gain* and the *Average Precision* provide measures which are sensitive to the top ranked instances; they can both be expressed as conditional linear rank statistics (see Clémençon and Vayatis 2007 and Clémençon and Vayatis 2009a) with score-generating function given by $1/(\ln(1+x))$ (DCG) or $1/x$ (AP);

- HR@$u\%$ - the *Hit Ratio* at $u\%$ is a relative count of positive instances among a proportion u of best scored instances.

These indicators capture the most important properties as far as quality assessment for scoring rules is concerned: average and local performance, stability of the rule, accuracy of ROC performance.
Results and comments. Results are collected in a series of Tables 1, 2, 3, 4, 5, 6. We also report enveloppes on ROC curves over the series of replications of the experiments with the same parameters (see Figures 3 and 4). We study in particular the impact of mixed effects of randomization with sample size (Tables 1, 2, 3) or aggregation (Tables 4, 5, 6). Our main observations are the following:

- The sample size of the training set has a moderate impact on performance of RANKING FOREST while it helps significantly single trees in the plain TREERANK;

- In the case of small sample sizes, RANKING FOREST with little randomization (Cases 2 and 5) boost performance compared to the plain TREERANK;

- Increasing the amount of aggregation always improves performance and accuracy except in some situations in the non-sparse data sets (little randomization $F_T = d$, B large);

- BAGGING with $B = 20$ ranking trees already improves plain TREERANK dramatically;

- Randomization reveals its power in the sparse data set; when all input variables are relevant, highly randomized strategies (Cases 4 and 6) may fail to capture good scoring rules unless a large amount of ranking trees are aggregated (B above 50).

These empirical results aim at illustrating the effect of the combination of rank aggregation and random feature selection on ranking accuracy/stability. A complete and detailed empirical analysis of the merits and limitations of RANKING FOREST is beyond the scope of this paper and it will be the object of future work.

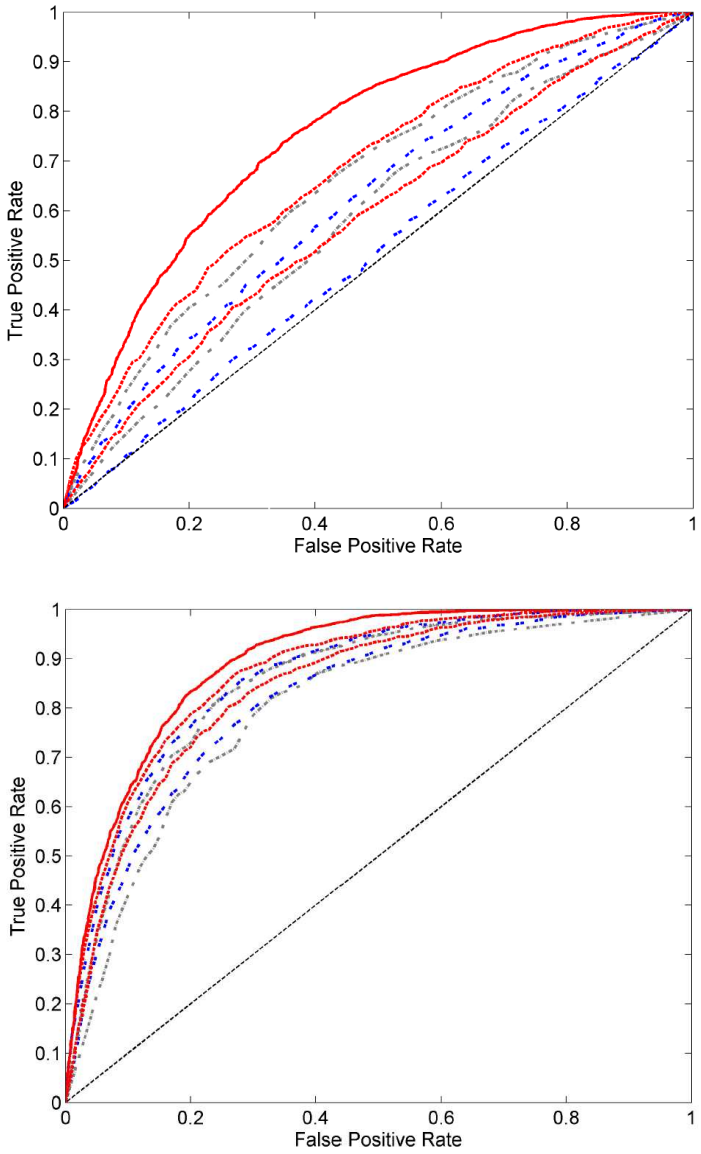


Figure 3: Comparison of envelopes on ROC curves - Results obtained with RANKING FORESTS with $B = 50$ (blue, double dashed) and 100 (red, solid, dashed). The upper display shows results on the data set *RF 10* while the lower display corresponds to the curves obtained on the data set *RF 10 sparse*. RANKING FORESTS used correspond to Case 3, training size is 2000, and optimal ROC curve is in thick red.

7. Conclusion

The major contribution of the paper was to show how to apply the principles of the RANDOM FOREST approach to the ranking/scoring task. Several ways of randomizing and aggregating ranking

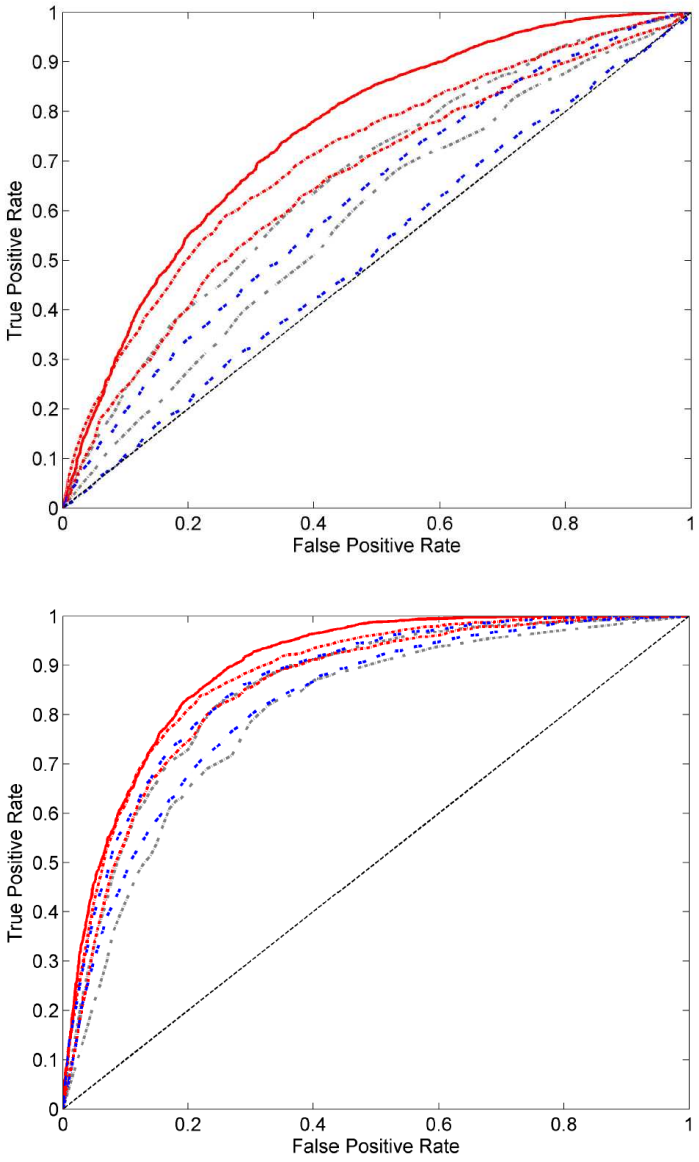


Figure 4: Comparison of envelopes on ROC curves - Results obtained with BAGGING (red, solid and dashed) and RANKING FORESTS (blue, double dashed) with $B = 50$. The upper display shows results on the data set *RF 10* while the lower display corresponds to the curves obtained on the data set *RF 10 sparse*. RANKING FORESTS used correspond to Case 3, training size is 2000, and optimal ROC curve is in thick red.

trees, such as those produced by the TREERANK algorithm, have been rigorously described. We proposed a specific notion of *stability* in the ranking setup and provided some preliminary back-

RF 10 - $\mathrm{AUC}^* = 0.756$ - dependence on aggregation										
	F_T	F_L	B	$\overline{\mathrm{AUC}}$ $(\pm\hat{\sigma})$	ΔEnv	$\mathbf{Instab}_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	-	0.628 (±0.013)	-	0.013	1.574	0.59	66%	64%
Bagging	-	-	20	0.678 (±0.010)	−25%	0.010	1.708	0.64	77%	74%
			50	0.686 (±0.008)	−29%	0.009	1.745	0.64	78%	74%
			100	0.689 (±0.009)	−29%	0.009	1.819	0.65	78%	74%
Forest Case 1	5	5	1	0.508 (±0.027)	+65%	0.016	1.563	0.50	49%	50%
			20	0.550 (±0.026)	+55%	0.015	2.059	0.53	57%	55%
			50	0.567 (±0.025)	+46%	0.015	2.210	0.55	59%	57%
			100	0.642 (±0.016)	−7%	0.011	2.288	0.61	71%	67%
Forest Case No. 2	10	5	1	0.525 (±0.025)	+68%	0.015	1.564	0.51	52%	52%
			20	0.577 (±0.024)	+22%	0.014	2.012	0.56	61%	59%
			50	0.615 (±0.020)	+21%	0.013	2.187	0.58	67%	64%
			100	0.585 (±0.025)	+34%	0.014	2.357	0.56	62%	60%
Forest Case 3	5	3	1	0.512 (±0.024)	+61%	0.016	1.564	0.50	49%	49%
			20	0.546 (±0.024)	+35%	0.015	2.047	0.53	56%	54%
			50	0.577 (±0.025)	+35%	0.014	2.215	0.56	61%	59%
			100	0.648 (±0.019)	+23%	0.011	2.294	0.61	72%	68%
Forest Case 4	3	3	1	0.512 (±0.023)	+51%	0.015	1.570	0.50	47%	49%
			20	0.537 (±0.026)	+27%	0.015	2.067	0.52	54%	53%
			50	0.563 (±0.028)	+42%	0.015	2.249	0.54	58%	57%
			100	0.595 (±0.019)	0%	0.014	2.345	0.57	64%	61%
Forest Case 5	10	3	1	0.516 (±0.029)	+95%	0.016	1.564	0.51	51%	51%
			20	0.582 (±0.022)	+32%	0.014	2.016	0.56	62%	59%
			50	0.616 (±0.022)	+11%	0.013	2.161	0.59	67%	64%
			100	0.579 (±0.023)	+30%	0.014	2.423	0.56	61%	59%
Forest Case 6	3	1	1	0.517 (±0.028)	+81%	0.016	1.567	0.51	51%	52%
			20	0.545 (±0.026)	+38%	0.015	2.075	0.53	56%	55%
			50	0.565 (±0.024)	+28%	0.015	2.224	0.55	59%	57%
			100	0.647 (±0.016)	+3%	0.011	2.306	0.61	70%	67%

Table 1: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with aggregation (B) on the data set *RF 10* with training sample size $n = 2000$.

ground theory for ranking rule aggregation. Encouraging experimental results based on artificial data have also been obtained, demonstrating how bagging combined with feature randomization may significantly enhance ranking accuracy and stability both at the same time. Truth be told, theoretical explanations for the success of RANKING FOREST in these situations are left to be found. Results obtained by Friedman and Hall (2007) or Grandvalet (2004) for the bagging approach in the classification/regression context suggest possible lines of research in this regard. At the same time, further experiments, based on real data sets in particular, will be carried out in a dedicated article in order to determine precisely the situations in which RANKING FOREST is competitive compared to alternative ranking methods.

RF 20 - $\mathrm{AUC}^* = 0.773$ - dependence on aggregation										
	F_T	F_L	B	$\overline{\mathrm{AUC}}$ ($\pm\hat{\sigma}$)	ΔEnv	$\mathbf{Instab}_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	-	0.613 ($\pm$0.013)	-	0.013	1.614	0.59	67%	64%
Bagging	-	-	20	0.691 ($\pm$0.009)	-32%	0.009	1.715	0.66	80%	75%
			50	0.699 ($\pm$0.006)	-43%	0.008	1.816	0.66	81%	76%
Forest Case 1	10	10	1	0.534 ($\pm$0.033)	$+120\%$	0.015	1.599	0.53	56%	56%
			20	0.623 ($\pm$0.028)	$+78\%$	0.013	2.017	0.60	68%	65%
			50	0.667 ($\pm$0.021)	$+33\%$	0.011	2.017	0.63	73%	70%
			100	0.726 ($\pm$0.011)	-25%	0.007	2.160	0.67	80%	77%
Forest Case 2	20	10	1	0.551 ($\pm$0.033)	$+114\%$	0.015	1.599	0.54	58%	57%
			20	0.673 ($\pm$0.019)	$+28\%$	0.011	1.989	0.64	73%	70%
			50	0.706 ($\pm$0.012)	-15%	0.009	2.104	0.66	77%	74%
			100	0.693 ($\pm$0.014)	0%	0.009	2.250	0.65	76%	73%
Forest Case 3	10	5	1	0.534 ($\pm$0.030)	$+100\%$	0.015	1.599	0.53	56%	55%
			20	0.625 ($\pm$0.025)	$+64\%$	0.013	2.077	0.60	68%	65%
			50	0.675 ($\pm$0.013)	-6%	0.011	2.179	0.64	75%	71%
			100	0.726 ($\pm$0.009)	-35%	0.007	2.171	0.67	80%	77%
Forest Case 4	5	5	1	0.516 ($\pm$0.038)	$+138\%$	0.016	1.599	0.52	53%	53%
			20	0.585 ($\pm$0.030)	$+93\%$	0.014	2.050	0.57	63%	61%
			50	0.625 ($\pm$0.026)	$+50\%$	0.013	2.217	0.60	67%	65%
			100	0.702 ($\pm$0.013)	-16%	0.009	2.247	0.66	78%	74%
Forest Case 5	20	5	1	0.547 ($\pm$0.034)	$+123\%$	0.015	1.598	0.54	58%	56%
			20	0.666 ($\pm$0.020)	$+25\%$	0.011	2.007	0.63	74%	70%
			50	0.705 ($\pm$0.011)	-23%	0.009	2.128	0.66	78%	74%
			100	0.658 ($\pm$0.021)	$+24\%$	0.011	2.329	0.62	71%	69%
Forest Case 6	5	1	1	0.510 ($\pm$0.040)	$+157\%$	0.016	1.597	0.51	52%	52%
			20	0.574 ($\pm$0.035)	$+97\%$	0.015	2.120	0.56	61%	59%
			50	0.614 ($\pm$0.027)	$+64\%$	0.014	2.238	0.59	67%	64%
			100	0.710 ($\pm$0.011)	-19%	0.009	2.261	0.66	78%	75%

Table 2: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with aggregation (B) on the data set *RF 20* with training sample size $n = 2000$.

Appendix A. Axioms for Ranking Rules

Throughout this paper, we call a *ranking* of the elements of a set $\mathcal{Z}$ any *total preorder* on $\mathcal{Z}$, that is, a binary relation $\preceq$ for which the following axioms are checked.

1. (TOTALITY) For all $(z_1, z_2) \in \mathcal{Z}^2$, either $z_1 \preceq z_2$ or else $z_2 \preceq z_1$ holds.

2. (TRANSITIVITY) For all (z_1, z_2, z_3): if $z_1 \preceq z_2$ and $z_2 \preceq z_3$, then $z_1 \preceq z_3$.

When the assertions $z_1 \preceq z_2$ and $z_2 \preceq z_1$ hold both at the same time, we write $z_1 \asymp z_2$ and $z_1 \prec z_2$ when solely the first one is true. Assuming in addition that $\mathcal{Z}$ has finite cardinality $\#\mathcal{Z} < \infty$, the rank

RF 10 sparse - $\mathrm{AUC}^* = 0.89$ - dependence on aggregation B										
	F_T	F_L	B	$\overline{\mathrm{AUC}}$ ($\pm\hat{\sigma}$)	ΔEnv	**Instab**$_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	-	0.826 (±0.007)	-	0.007	1.622	0.70	84%	83%
Bagging	-	-	20	0.865 (±0.004)	−30%	0.004	1.643	0.74	89%	88%
			50	0.867 (±0.003)	−35%	0.004	1.650	0.74	89%	88%
			100	0.868 (±0.003)	−36%	0.004	1.708	0.74	89%	88%
Forest Case 1	5	5	1	0.630 (±0.071)	+502%	0.014	1.632	0.58	66%	63%
			20	0.814 (±0.018)	+61%	0.008	1.977	0.71	86%	84%
			50	0.832 (±0.012)	+22%	0.006	2.163	0.72	88%	85%
			100	0.858 (±0.006)	−30%	0.004	2.110	0.74	90%	88%
Forest Case 2	10	5	1	0.636 (±0.083)	+588%	0.014	1.598	0.59	71%	66%
			20	0.845 (±0.010)	−12%	0.005	1.893	0.73	89%	86%
			50	0.863 (±0.005)	−43%	0.004	1.918	0.74	90%	88%
			100	0.869 (±0.003)	−51%	0.003	1.956	0.74	91%	89%
Forest Case 3	5	3	1	0.622 (±0.071)	+553%	0.014	1.607	0.57	64%	60%
			20	0.809 (±0.010)	+72%	0.008	2.060	0.71	86%	83%
			50	0.844 (±0.009)	−15%	0.005	2.089	0.73	89%	87%
			100	0.859 (±0.005)	−38%	0.004	2.133	0.74	90%	88%
Forest Case 4	3	3	1	0.580 (±0.083)	+672%	0.015	1.612	0.55	61%	59%
			20	0.772 (±0.036)	+195%	0.010	2.056	0.68	83%	79%
			50	0.829 (±0.015)	+39%	0.007	2.211	0.72	88%	85%
			100	0.849 (±0.008)	−10%	0.005	2.253	0.73	90%	87%
Forest Case 5	10	3	1	0.661 (±0.069)	+480%	0.014	1.602	0.60	69%	66%
			20	0.840 (±0.010)	−9%	0.006	1.926	0.73	88%	86%
			50	0.863 (±0.005)	−41%	0.004	1.974	0.74	90%	88%
			100	0.868 (±0.010)	−54%	0.003	1.990	0.74	91%	89%
Forest Case 6	3	1	1	0.593 (±0.073)	+566%	0.015	1.611	0.55	63%	60%
			20	0.745 (±0.036)	+228%	0.011	2.162	0.66	79%	76%
			50	0.807 (±0.026)	+108%	0.008	2.252	0.70	86%	83%
			100	0.835 (±0.010)	−6%	0.006	2.318	0.72	88%	85%

Table 3: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with aggregation (B) on the data set *RF 10 sparse* with training sample size $n = 2000$.

of any element $z \in \mathcal{Z}$ is given by

$$\mathcal{R}_{\preceq}(z) = \sum_{z' \in \mathcal{Z}} \left\{ \mathbb{I}\{z' \prec z\} + \frac{1}{2}\mathbb{I}\{z' \asymp z\} \right\},$$

when using the standard MID-RANK convention (Kendall, 1945), that is, by assigning to tied elements the average of the ranks they cover.

Any scoring rule $s : \mathcal{Z} \to \mathbb{R}$ naturally defines a ranking $\preceq_s$ on $\mathcal{Z}$: $\forall (z_1, z_2) \in \mathcal{Z}^2$, $z_1 \preceq_s z_2$ iff $s(z_1) \leq s(z_2)$. Equipped with these notations, it is clear that $\preceq_{\mathcal{R}_{\preceq}}$ coincides with $\preceq$ for any ranking $\preceq$ on a finite set $\mathcal{Z}$.

RF 10 - $\mathrm{AUC}^* = 0.76$ - dependence on sample size										
	F_T	F_L	n	$\overline{\mathrm{AUC}}$ $(\pm\hat{\sigma})$	ΔEnv	$\mathbf{Instab}_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	250	0.573 (±0.024)	-	0.014	1.673	0.54	60%	58%
			500	0.576 (±0.018)	-	0.014	1.607	0.54	59%	58%
			1000	0.595 (±0.018)	-	0.014	1.583	0.56	62%	60%
			2000	0.628 (±0.013)	-	0.013	1.574	0.59	66%	64%
			3000	0.632 (±0.011)	-	0.013	1.560	0.59	66%	65%
Bagging	-	-	2000	0.678 (±0.010)	−25%	0.010	1.708	0.64	77%	74%
			3000	0.678 (±0.010)	−25%	0.010	1.708	0.64	77%	74%
Case 1	5	5	250	0.546 (±0.023)	−16%	0.015	2.034	0.53	56%	54%
			500	0.544 (±0.028)	+40%	0.015	2.032	0.53	56%	55%
			1000	0.547 (±0.026)	+10%	0.015	2.009	0.53	57%	55%
			2000	0.550 (±0.026)	+55%	0.015	2.059	0.53	57%	55%
			3000	0.549 (±0.019)	+33%	0.015	2.034	0.53	55%	55%
Case 2	10	5	250	0.571 (±0.025)	+11%	0.015	1.990	0.55	60%	58%
			500	0.571 (±0.030)	+34%	0.015	1.984	0.55	60%	59%
			1000	0.578 (±0.028)	+41%	0.014	1.999	0.56	60%	59%
			2000	0.577 (±0.024)	+22%	0.014	2.012	0.56	61%	59%
			3000	0.585 (±0.028)	+76%	0.014	1.998	0.56	62%	60%
Case 3	5	3	250	0.546 (±0.031)	+7%	0.015	2.049	0.53	56%	55%
			500	0.556 (±0.029)	+40%	0.015	1.993	0.54	58%	57%
			1000	0.563 (±0.023)	+8%	0.015	2.024	0.54	58%	57%
			2000	0.546 (±0.024)	+35%	0.015	2.047	0.53	56%	54%
			3000	0.549 (±0.019)	+30%	0.015	2.026	0.53	56%	55%
Case 4	3	3	250	0.546 (±0.023)	+15%	0.015	2.090	0.53	55%	55%
			500	0.536 (±0.028)	+36%	0.015	2.071	0.52	54%	53%
			1000	0.540 (±0.027)	+15%	0.015	2.075	0.53	55%	54%
			2000	0.537 (±0.026)	+27%	0.015	2.067	0.52	54%	53%
			3000	0.536 (±0.022)	+55%	0.015	2.063	0.52	54%	54%
Case 5	10	3	250	0.588 (±0.027)	+5%	0.014	1.984	0.56	62%	60%
			500	0.570 (±0.030)	+65%	0.015	1.970	0.55	59%	58%
			1000	0.587 (±0.023)	+16%	0.014	1.971	0.56	63%	60%
			2000	0.582 (±0.022)	+32%	0.014	2.016	0.56	62%	59%
			3000	0.587 (±0.026)	+83%	0.014	1.991	0.57	63%	60%
Case 6	3	1	250	0.546 (±0.028)	+8%	0.015	2.085	0.53	56%	55%
			500	0.543 (±0.024)	+11%	0.015	2.077	0.53	55%	54%
			1000	0.549 (±0.026)	+13%	0.015	2.066	0.53	56%	55%
			2000	0.545 (±0.026)	+38%	0.015	2.075	0.53	56%	55%
			3000	0.546 (±0.026)	+71%	0.015	2.065	0.53	56%	55%

Table 4: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with sample size (n) on the data set *RF 10* for $B = 20$.

Appendix B. Agreement Between Rankings

The most widely used approach to the *rank aggregation* issue relies on the concept of *measure of agreement* between rankings which uses *pseudo-metrics*. Since the seminal contribution of Kemeny

RF 20 - $\mathrm{AUC}^* = 0.77$ - dependence on sample size										
	F_T	F_L	n	$\overline{\mathrm{AUC}}$ ($\pm\tilde{\sigma}$)	ΔEnv	$\mathbf{Instab}_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	250	0.561 ($\pm$0.019)	-	0.014	1.742	0.55	57%	58%
			500	0.579 ($\pm$0.018)	-	0.014	1.666	0.56	60%	59%
			1000	0.593 ($\pm$0.014)	-	0.014	1.626	0.57	63%	62%
			2000	0.613 ($\pm$0.013)	-	0.013	1.614	0.59	67%	65%
			3000	0.621 ($\pm$0.013)	-	0.013	1.597	0.59	67%	65%
Bagging	-	-	2000	0.691 ($\pm$0.009)	-32%	0.009	1.715	0.66	80%	75%
			3000	0.691 ($\pm$0.009)	-32%	0.009	1.715	0.66	80%	75%
Case 1	10	10	250	0.612 ($\pm$0.026)	$+25\%$	0.014	2.019	0.59	67%	64%
			500	0.630 ($\pm$0.029)	$+41\%$	0.013	2.018	0.61	69%	66%
			1000	0.628 ($\pm$0.025)	$+44\%$	0.013	2.024	0.60	68%	66%
			2000	0.623 ($\pm$0.028)	$+78\%$	0.013	2.017	0.60	68%	65%
			3000	0.636 ($\pm$0.029)	$+54\%$	0.012	2.012	0.61	67%	65%
Case 2	20	10	250	0.646 ($\pm$0.027)	$+27\%$	0.012	1.964	0.62	71%	68%
			500	0.660 ($\pm$0.018)	$+6\%$	0.012	1.945	0.63	72%	69%
			1000	0.666 ($\pm$0.019)	$+23\%$	0.011	1.984	0.63	73%	70%
			2000	0.673 ($\pm$0.019)	$+28\%$	0.011	1.989	0.64	73%	70%
			3000	0.665 ($\pm$0.017)	$+17\%$	0.011	1.997	0.63	73%	70%
Case 3	10	5	250	0.610 ($\pm$0.030)	$+69\%$	0.014	2.039	0.59	66%	63%
			500	0.617 ($\pm$0.033)	$+56\%$	0.013	2.027	0.59	66%	64%
			1000	0.621 ($\pm$0.024)	$+44\%$	0.013	2.035	0.60	67%	65%
			2000	0.625 ($\pm$0.025)	$+64\%$	0.013	2.077	0.60	68%	65%
			3000	0.631 ($\pm$0.025)	$+55\%$	0.013	2.039	0.61	69%	66%
Case 4	5	5	250	0.568 ($\pm$0.036)	$+82\%$	0.015	2.088	0.56	61%	59%
			500	0.579 ($\pm$0.018)	$+47\%$	0.014	2.064	0.58	63%	61%
			1000	0.585 ($\pm$0.041)	$+155\%$	0.014	2.060	0.57	63%	61%
			2000	0.585 ($\pm$0.030)	$+93\%$	0.014	2.050	0.57	63%	61%
			3000	0.585 ($\pm$0.030)	$+88\%$	0.014	2.052	0.57	63%	61%
Case 5	20	5	250	0.631 ($\pm$0.018)	-4%	0.013	1.962	0.61	69%	67%
			500	0.658 ($\pm$0.021)	$+4\%$	0.012	1.941	0.62	72%	69%
			1000	0.659 ($\pm$0.022)	$+25\%$	0.012	1.988	0.63	72%	69%
			2000	0.666 ($\pm$0.020)	$+25\%$	0.011	2.007	0.63	74%	70%
			3000	0.670 ($\pm$0.021)	$+46\%$	0.011	1.978	0.63	73%	70%
Case 6	5	1	250	0.561 ($\pm$0.033)	$+57\%$	0.015	2.099	0.55	59%	57%
			500	0.570 ($\pm$0.028)	$+32\%$	0.015	2.061	0.56	61%	59%
			1000	0.571 ($\pm$0.031)	$+119\%$	0.015	2.066	0.56	60%	59%
			2000	0.574 ($\pm$0.035)	$+97\%$	0.015	2.120	0.56	61%	59%
			3000	0.570 ($\pm$0.032)	$+88\%$	0.015	2.053	0.56	61%	60%

Table 5: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with sample size (n) on the data set *RF 10* for $B = 20$.

(1959), numerous ways of measuring agreement have been proposed in the literature. Here we re-

RF 10 sparse - $\text{AUC}^* = 0.89$ - dependence on sample size										
	F_T	F_L	n	$\overline{\text{AUC}}$ ($\pm\bar{\sigma}$)	ΔEnv	$\textbf{Instab}_\tau$	DCG	AVE	HR @10%	HR @20%
TreeRank	-	-	250	0.749 (±0.022)	-	0.010	1.739	0.63	74%	74%
			500	0.771 (±0.015)	-	0.008	1.662	0.65	76%	76%
			1000	0.806 (±0.009)	-	0.008	1.637	0.68	80%	80%
			2000	0.827 (±0.007)	-	0.007	1.622	0.70	84%	83%
			3000	0.836 (±0.006)	-	0.007	1.602	0.70	85%	84%
Bagging	-	-	2000	0.865 (±0.004)	−30%	0.004	1.643	0.74	89%	88%
			3000	0.865 (±0.004)	−30%	0.004	1.643	0.74	89%	88%
Case 1	5	5	250	0.808 (±0.020)	−28%	0.008	2.010	0.71	87%	36%
			500	0.814 (±0.024)	+32%	0.008	1.958	0.71	86%	83%
			1000	0.862 (±0.005)	−49%	0.004	1.701	0.74	89%	88%
			2000	0.814 (±0.018)	+61%	0.008	1.977	0.71	86%	84%
			3000	0.870 (±0.005)	−19%	0.004	1.670	0.74	90%	88%
Case 2	10	5	250	0.835 (±0.012)	−57%	0.006	1.869	0.72	89%	86%
			500	0.841 (±0.011)	−36%	0.006	1.839	0.73	89%	86%
			1000	0.845 (±0.009)	−30%	0.006	1.853	0.73	90%	86%
			2000	0.845 (±0.010)	−12%	0.005	1.893	0.73	89%	86%
			3000	0.848 (±0.011)	+12%	0.006	1.851	0.73	89%	86%
Case 3	5	3	250	0.795 (±0.027)	−13%	0.009	2.014	0.70	86%	82%
			500	0.810 (±0.023)	+17%	0.008	1.984	0.71	86%	83%
			1000	0.811 (±0.020)	+40%	0.008	1.966	0.71	86%	83%
			2000	0.809 (±0.020)	+72%	0.008	2.060	0.71	86%	83%
			3000	0.809 (±0.023)	+110%	0.008	1.979	0.70	86%	83%
Case 4	3	3	250	0.764 (±0.042)	+27%	0.010	2.114	0.68	82%	78%
			500	0.773 (±0.038)	+115%	0.010	2.068	0.68	83%	79%
			1000	0.780 (±0.031)	+105%	0.009	2.063	0.69	83%	80%
			2000	0.772 (±0.036)	+195%	0.010	2.056	0.68	83%	79%
			3000	0.783 (±0.036)	+280%	0.009	2.044	0.69	83%	80%
Case 5	10	3	250	0.828 (±0.016)	−48%	0.007	1.931	0.72	87%	85%
			500	0.836 (±0.014)	−21%	0.006	1.883	0.72	88%	86%
			1000	0.841 (±0.012)	−9%	0.006	1.876	0.73	89%	86%
			2000	0.840 (±0.010)	+9%	0.006	1.926	0.73	88%	86%
			3000	0.843 (±0.008)	+5%	0.006	1.893	0.73	89%	86%
Case 6	3	1	250	0.724 (±0.049)	+32%	0.012	2.149	0.65	77%	74%
			500	0.757 (±0.035)	+76%	0.011	2.085	0.67	81%	78%
			1000	0.742 (±0.045)	+198%	0.011	2.096	0.66	79%	76%
			2000	0.745 (±0.036)	+228%	0.011	2.162	0.66	79%	76%
			3000	0.728 (±0.049)	+350%	0.012	2.079	0.65	78%	75%

Table 6: Comparison of TREERANK/LEAFRANK and BAGGING with RANKING FORESTS - Impact of randomization (F_T, F_L) and resampling with sample size (n) on the data set *RF 10 sparse* for $B = 20$.

view three popular choices, originally introduced in the context of *nonparametric statistical testing* (see Fagin et al. 2003 for instance).

Let $\preceq$ and $\preceq'$ be two rankings on a finite set $\mathcal{Z} = \{z_1, \ldots, z_K\}$. The notation $\mathcal{R}_{\preceq}(z)$ is used for the rank of the element z according to the ranking $\preceq$.
Kendall τ. Consider the quantity $d_\tau(\preceq, \preceq')$, obtained by summing up all the terms

$$U_{i,j}(\preceq, \preceq') = \mathbb{I}\{(\mathcal{R}_{\preceq}(z_i) - \mathcal{R}_{\preceq}(z_j))(\mathcal{R}_{\preceq'}(z_i) - \mathcal{R}_{\preceq'}(z_j)) < 0\} + \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq}(z_i) = s_{\preceq}(z_j),\ \mathcal{R}_{\preceq'}(z_i) \neq \mathcal{R}_{\preceq'}(z_j)\} + \frac{1}{2}\mathbb{I}\{\mathcal{R}_{\preceq'}(z_i) = \mathcal{R}_{\preceq'}(z_j),\ \mathcal{R}_{\preceq}(z_i) \neq \mathcal{R}_{\preceq}(z_j)\}$$

over all pairs (z_i, z_j) such that $1 \leq i < j \leq K$. It counts, among the $K(K-1)$ pairs of $\mathcal{Z}$'s elements, how many are "discording", assigning the weight $1/2$ when two elements are tied in one ranking but not in the other. The Kendall τ is obtained by renormalizing this distance:

$$\tau(\preceq, \preceq') = 1 - \frac{4}{K(K-1)} d_\tau(\preceq, \preceq').$$

Large values of $\tau(\preceq, \preceq')$ indicate agreement (or similarity) between $\preceq$ and $\preceq'$: it ranges from -1 (full disagreement) to 1 (full agreement). It is worth noticing that it can be computed in $O((K \log K)/\log\log K)$ time (see Bansal and Fernandez-Baca 2009).
Spearman footrule. Another natural distance between rankings is defined by considering the l_1-metric between the corresponding rank vectors:

$$d_1(\preceq,\ \preceq') = \sum_{i=1}^{K} |\mathcal{R}_{\preceq}(z_i) - \mathcal{R}_{\preceq'}(z_i)|.$$

The affine transformation given by

$$F(\preceq, \preceq') = 1 - \frac{3}{K^2 - 1} d_1(\preceq,\ \preceq').$$

is known as the Spearman footrule measure of agreement and takes its values in $[-1, +1]$.
Spearman rank-order correlation. Considering instead the l_2-metric

$$d_2(\preceq,\ \preceq') = \sum_{i=1}^{K} \left(\mathcal{R}_{\preceq}(z_i) - \mathcal{R}_{\preceq'}(z_i)\right)^2$$

leads to the Spearman ρ coefficient:

$$\rho(\preceq,\ \preceq') = 1 - \frac{6}{K(K^2-1)} d_2(\preceq,\ \preceq').$$

Remark 21 *(*EQUIVALENCE*.) It should be noticed that these three measures of agreement are equivalent in the sense that:*

$$\begin{aligned} c_1\left(1 - \rho(\preceq,\ \preceq')\right) \leq\ & (1 - F(\preceq,\ \preceq'))^2 \leq\ c_2\left(1 - \rho(\preceq,\ \preceq')\right), \\ c_3\left(1 - \tau(\preceq,\ \preceq')\right) \leq\ & 1 - F(\preceq,\ \preceq') \leq\ c_4\left(1 - \tau(\preceq,\ \preceq')\right), \end{aligned}$$

with $c_2 = K^2/(2(K^2-1)) = Kc_1$ *and* $c_4 = 3K/(2(K+1)) = 2c_3$*; see Theorem 13 in Fagin et al. (2006).*

We point out that many fashions of measuring agreement or distance between rankings have been considered in the literature, see Mielke and Berry (2001). Well-known alternatives to the measures recalled above are the Cayley/Kemeny distance (Kemeny, 1959) and variants for top k-lists (Fagin et al., 2006), in order to focus on the "best instances" (see Clémençon and Vayatis 2007). Many other distances between rankings could naturally be deduced through suitable extensions of *word metrics* on the symmetric groups on finite sets (see Howie 2000 or Deza and Deza 2009).

Appendix C. The TREERANK Algorithm

Here we briefly review the TREERANK method, on which the procedure we call RANKING FOREST crucially relies. One may refer to Clémençon and Vayatis (2009c) and Clémençon et al. (2011) for further details as well as rigorous statistical foundations for the algorithm. As for most tree-based techniques, a greedy top-down recursive partitioning stage based on a training sample $\mathcal{D}_n = \{(X_i, Y_i) : 1 \leq i \leq n\}$ is followed by a pruning procedure, where children of a same parent node are recursively merged until an estimate of the AUC performance criterion is maximized. A package for R statistical software (see http://www.r-project.com) implementing TREERANK is available at http://treerank.sourceforge.net (see Baskiotis et al. 2009).

C.1 Growing Stage

The goal is here to grow a master ranking tree of large depth $D \geq 1$ with empirical AUC as large as possible. In order to describe this first stage, we introduce the following quantities. Let $\mathcal{C} \subset \mathcal{X}$, consider the empirical rate of negative (respectively, positive) instances lying in the region $\mathcal{C}$:

$$\widehat{\alpha}(\mathcal{C}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X_i \in \mathcal{C},\, Y_i = -1\} \text{ and } \widehat{\beta}(\mathcal{C}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X_i \in \mathcal{C},\, Y_i = +1\},$$

as well as $n(\mathcal{C}) = n(\widehat{\alpha}(\mathcal{C}) + \widehat{\beta}(\mathcal{C}))$ the number of data falling in $\mathcal{C}$.

One starts from the trivial partition $\mathcal{P}_0 = \{\mathcal{X}\}$ at root node $(0,0)$ (we set $C_{0,0} = \mathcal{X}$) and proceeds recursively as follows. A tree-structured scoring rule $s(x)$ described by an oriented tree, with outer leaves forming a partition $\mathcal{P}$ of the input space, is refined by splitting a cell $\mathcal{C} \in \mathcal{P}$ into two sub-cells: $\mathcal{C}'$ denoting the left child and $\mathcal{C}'' = \mathcal{C} \setminus \mathcal{C}'$ the right one. Let $s'(x)$ be the scoring rule thus obtained. From the perspective of AUC maximization, one seeks a subregion $\mathcal{C}'$ maximizing the gain $\Delta_{\widehat{\text{AUC}}}(\mathcal{C}, \mathcal{C}')$ in terms of empirical AUC induced by the split, which may be written as:

$$\widehat{\text{AUC}}(s') - \widehat{\text{AUC}}(s) = \frac{1}{2}\{\widehat{\alpha}(\mathcal{C})\widehat{\beta}(\mathcal{C}') - \widehat{\beta}(\mathcal{C})\widehat{\alpha}(\mathcal{C}')\}.$$

Therefore, taking the rate of positive instances within the cell $\mathcal{C}$, $\widehat{p}(\mathcal{C}) = \widehat{\alpha}(\mathcal{C}) \cdot n/n(\mathcal{C})$ namely, as cost for the type I error (i.e., predicting label $+1$ when $Y = -1$) and $1 - \widehat{p}(\mathcal{C})$ as cost for the type II error, the quantity $1 - \Delta_{\widehat{\text{AUC}}}(\mathcal{C}, \mathcal{C}')$ may be viewed as the *cost-sensitive empirical misclassification error* of the classifier $C(X) = 2 \cdot \mathbb{I}\{X \in \mathcal{C}'\} - 1$ on $\mathcal{C}$ up to a multiplicative factor, $4\widehat{p}(\mathcal{C})(1 - \widehat{p}(\mathcal{C}))$ precisely. Hence, once the local cost $\widehat{p}(\mathcal{C})$ is computed, any binary classification method can be straightforwardly adapted in order to perform the splitting step. Here, splits are obtained using empirical-cost sensitive versions of the standard CART algorithm with axis-parallel splits, this one-step procedure for AUC maximization being called LEAFRANK in Clémençon et al. (2011). As depicted by Figure 5, the growing stage appears as a recursive implementation of a cost-sensitive

CART procedure with a cost updated at each node of the ranking tree, equal to the local rate of positive instances within the node to split, see Section 3 of Clémençon et al. (2011).

C.2 Pruning Stage

The way the master ranking tree $\mathcal{T}_D$ obtained at the end of the growing stage is pruned is entirely similar to the one described in Breiman et al. (1984), the sole difference lying in the fact that here, for any $\lambda > 0$, one seeks a subtree $\mathcal{T} \subset \mathcal{T}_D$ that maximizes the penalized empirical AUC

$$\widehat{\mathrm{AUC}}(s_{\mathcal{T}}) - \lambda \cdot |\mathcal{T}|,$$

where $|\mathcal{T}|$ denotes the number of terminal leaves of $\mathcal{T}$, the constant being next picked using N-fold cross validation.

The fact that alternative complexity-based penalization procedures, inspired from recent non-parametric model selection methods and leading to the concept of *structural* AUC *maximization*, can be successfully used for pruning ranking trees has also been pointed up in Section 4.2 of Clémençon et al. (2011). However, the resampling-based technique previously mentioned is preferred to such pruning schemes in practice, insofar as it does not require, in contrast, to specify any tuning constant. Following in the footsteps of Arlot (2009) in the classification setup, estimation of the ideal penalty through bootstrap methods could arise as the answer to this issue. This question is beyond the scope of the present paper but will soon be tackled.

C.3 Some Practical Considerations

Like other types of decision trees, ranking trees (based on perpendicular splits) have a number of crucial advantages. Concerning interpretability first, it should be noticed that they produce ranking rules that can be easily visualized through the binary tree graphic, see Figure 5, the rank/score of an instance $x \in \mathcal{X}$ being obtained through checking of a nested combination of simple rules of the form "$X^{(k)} \geq t$" or "$X^{(k)} < t$". In addition, ranking trees can adapt straightforwardly to situations where some data are missing and/or some predictor variables are categorical and some monitoring tools helping to evaluate the relative importance of each predictor variable $X^{(k)}$ or to depict the partial dependence of the prediction rule on a subset of input variables are readily available. These facets are described in section 5 of Clémençon et al. (2011). From a computational perspective now, we also underline that the tree structure makes the computation of consensus rankings much more tractable, we refer to Appendix D for further details.

Appendix D. On Computing the Largest Subpartition

We now briefly explain how to make crucial use of the fact that the partitions of $\mathcal{X}$ we consider here are tree-structured to compute the largest subpartition they induce. Let $\mathcal{P}_1 = \{\mathcal{C}_k^{(1)}\}_{1 \leq k \leq K_1}$ and $\mathcal{P}_2 = \{\mathcal{C}_k^{(2)}\}_{1 \leq k \leq K_2}$ be two partitions of $\mathcal{X}$, related to (ranking) trees $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively. For any $k \in \{1, \ldots, K_1\}$, the collection of subsets of the form $\mathcal{C}_k^{(1)} \cap \mathcal{C}_l^{(2)}$, $1 \leq l \leq K_2$, can be obtained by extending the $\mathcal{T}_1$ tree structure the following way. At the $\mathcal{T}_1$'s terminal leave defining the cell $\mathcal{C}_k^{(1)}$, add a subtree corresponding to $\mathcal{T}_2$ with root $\mathcal{C}_k^{(1)}$: the terminal nodes of the resulting subtree, starting at the global root $\mathcal{X}$, correspond to the desired collection of subsets (notice that some of these can be empty), see Figure 6 below. Of course, this scheme can be iterated in order to recover

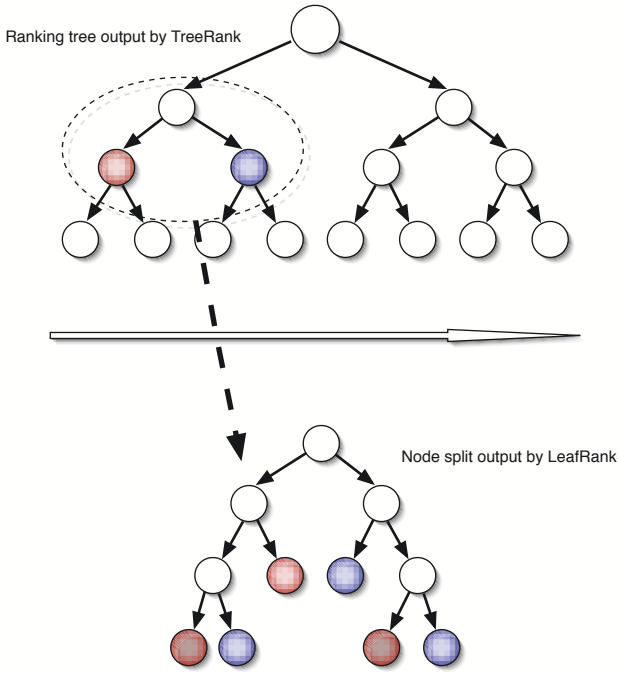


Figure 5: The TREERANK algorithm as a recursive implementation of cost-sensitive CART.

all the cells of the subpartition induced by $B > 2$ tree-structured partitions. For obvious reasons of computational nature, one should start with the most complex tree and bind progressively less and less complex trees as one goes along.

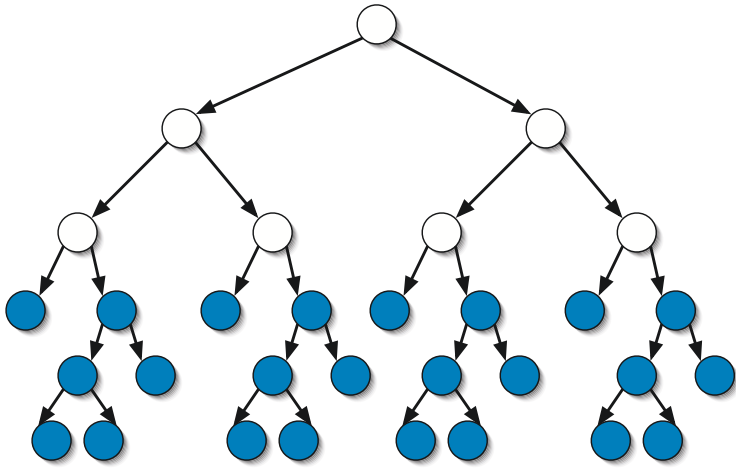


Figure 6: Characterizing the largest subpartition induced by tree-structured partitions.

Appendix E. Proofs

This section contains the proofs of the theoretical results presented in the core of the paper.

E.1 Proof of Proposition 9

Recall that $\tau_X(s_1,s_2) = 1 - 2d_X(s_1,s_2)$, where $d_X(s_1,s_2)$ is given by:

$$\mathbb{P}\{(s_1(X)-s_1(X'))\cdot(s_2(X)-s_2(X'))<0\} + \frac{1}{2}\mathbb{P}\{s_1(X)=s_1(x'),\ s_2(X)\neq s_2(X')\} + \frac{1}{2}\mathbb{P}\{s_1(X)\neq s_1(x'),\ s_2(X)=s_2(X')\}.$$

Observe first that, for all s, AUC(s) may be written as:

$$\mathbb{P}\{(s(X)-s(X'))\cdot(Y-Y')>0\}/(2p(1-p)) + \mathbb{P}\{s(X)=s(X'),\ Y\neq Y'\}/(4p(1-p)).$$

Notice also that, using Jensen's inequality, one easily obtain that $2p(1-p)|\mathrm{AUC}(s_1)-\mathrm{AUC}(s_2)|$ is bounded by the expectation of the random variable

$$\mathbb{I}\{(s_1(X)-s_1(X'))\cdot(s_2(X)-s_2(X'))>0\} + \frac{1}{2}\mathbb{I}\{s_1(X)=s_1(X')\}\cdot\mathbb{I}\{s_2(X)\neq s_2(X')\} + \frac{1}{2}\mathbb{I}\{s_1(X)\neq s_1(X')\}\cdot\mathbb{I}\{s_2(X)=s_2(X')\},$$

which is equal to $d_X(s_1,s_2) = (1-\tau_X(s_1,s_2))/2$.

E.2 Proof of Proposition 10

Recall first that, for all $s\in\mathcal{S}$, the AUC deficit $2p(1-p)\{\mathrm{AUC}^*-\mathrm{AUC}(s)\}$ may be written as

$$\mathbb{E}\left[|\eta(X)-\eta(X')|\cdot\mathbb{I}\{(X,X')\in\Gamma_s\}\right] + \mathbb{P}\{s(X)=s(X'),\ (Y,Y')=(-1,+1)\},$$

with

$$\Gamma_s = \{(x,x')\in\mathcal{X}^2:\ (s(x)-s(x'))\cdot(\eta(x)-\eta(x'))<0\},$$

refer to Example 1 in Clémençon et al. (2008) for instance. Now, Hölder inequality combined with noise condition (1) shows that $\mathbb{P}\{(X,X')\in\Gamma_s\}$ is bounded by

$$\left(\mathbb{E}\left[|\eta(X)-\eta(X')|\cdot\mathbb{I}\{(X,X')\in\Gamma_s\}\right]\right)^{a/(1+a)}\times c^{1/(1+a)}.$$

Therefore, we have for all $s^*\in\mathcal{S}^*$:

$$d_X(\preccurlyeq_s,\preccurlyeq_{s^*}) = \mathbb{P}\{(X,X')\in\Gamma_s\} + \frac{1}{2}\mathbb{P}\{s(X)=s(X')\}.$$

Notice that $p(1-p)\mathbb{P}\{s(X)=s(X')\mid(Y,Y')=(-1,+1)\}$ can be rewritten as

$$\mathbb{E}[\mathbb{I}\{s(X)=s(X')\}\cdot\eta(X')(1-\eta(X))] = \frac{1}{2}\mathbb{E}[\mathbb{I}\{s(X)=s(X')\}\cdot(\eta(X')+\eta(X)-2\eta(X)\eta(X'))],$$

which term can be easily shown to be larger than $\frac{1}{2}\mathbb{E}[\mathbb{I}\{s(X)=s(X')\}\cdot|\eta(X')-\eta(X)|]$. Using the same argument as above, we obtain that $\mathbb{P}\{s(X)=s(X')\}$ is bounded by

$$\left(\mathbb{E}\left[|\eta(X)-\eta(X')|\cdot\mathbb{I}\{s(X)=s(X')\}\right]\right)^{a/(1+a)}\times c^{1/(1+a)}.$$

Combined withe the bound previously established, this leads to the desired result.

E.3 Proof of Theorem 16

By virtue of Proposition 9, we have:

$$\mathrm{AUC}^* - \mathrm{AUC}(\bar{s}_B) \leq \frac{d_X(s^*, \bar{s}_B)}{2p(1-p)},$$

for any $s^* \in \mathcal{S}^*$. Using now triangular inequality applied to the distance d_X between preorders on $\mathcal{X}$, one gets

$$d_X(s^*, \bar{s}_B) \leq d_X(s^*, \widehat{s}_n(.,Z_j)) + d_X(\widehat{s}_n(.,Z_j), \bar{s}_B),$$

for all $j \in \{1, \ldots, B\}$. Averaging then over j and using the fact that, if one chooses s^* in $\mathcal{S}$,

$$\sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), \bar{s}_B) \leq \sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), s^*),$$

one obtains that

$$d_X(s^*, \bar{s}_B) \leq \frac{2}{B} \sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), s^*).$$

The desired result finally follows from Proposition 10 combined with the consistency assumption of the randomized scoring rule.

Remark 22 *Observe that, in the case where $\mathcal{S}$ is allowed to depend on n and one only assumes the existence of $\tilde{s}_n^* \in \mathcal{S}_n$ such that $\mathrm{AUC}(\tilde{s}_n^*) \to \mathrm{AUC}^*$ as $n \to \infty$ (relaxing thus the assumption $\mathcal{S} \cap \mathcal{S}^* \neq \emptyset$), the argument above leads to*

$$\mathrm{AUC}^* - \mathrm{AUC}(\bar{s}_B) \leq \frac{1}{2p(1-p)} \left\{ \frac{2}{B} \sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), s^*) + d_X(\tilde{s}_n^*, s^*) \right\}.$$

which shows that AUC *consistency of the median still holds true.*

E.4 Proof of Theorem 17

Observe that we have:

$$\begin{aligned} \Delta_B(\tilde{s}_m) - \min_{s \in \mathcal{S}} \Delta_B(s) &\leq 2 \cdot \sup_{s \in \mathcal{S}} |\widehat{\Delta}_{B,m}(s) - \Delta_B(s)| \\ &\leq 2 \sum_{j=1}^{B} \sup_{s \in \mathcal{S}} |\widehat{d}_X(s, s_j) - d_X(s, s_j)|. \end{aligned}$$

Now, it results from the strong Law of Large Numbers for U-processes stated in Corollary 5.2.3 in de la Pena and Giné (1999) that $\sup_{s \in \mathcal{S}} |\widehat{d}_X(s, s_j) - d_X(s, s_j)| \to 0$ as $N \to \infty$, for all $j = 1, \ldots, B$. The convergence rate $O_{\mathbb{P}}(m^{-1/2})$ follows from the Central Limit Theorem for U-processes given in Theorem 5.3.7 in de la Pena and Giné (1999).

E.5 Proof of Corollary 18

Reproducing the argument of Theorem 16, one gets:

$$d_X(s^*, \hat{s}_{n,m}) \leq \frac{1}{B}\sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), s^*) + \frac{1}{B}\sum_{j=1}^{B} d_X(\widehat{s}_n(.,Z_j), \hat{s}_{n,m}).$$

As in Theorem 17's proof, we also have:

$$\frac{1}{B}\sum_{j=1}^{B}\{d_X(\widehat{s}_n(.,Z_j), \widehat{s}_{n,m}) \quad - \quad d_X(\widehat{s}_n(.,Z_j), \bar{s}_B)\} \quad \leq \quad 2 \quad \cdot \quad \sup_{(s,s')\in\mathcal{S}^2} |\widehat{d}_X(s,s') \quad - \quad d_X(s,s')|.$$

Using again Corollary 5.2.3 in de la Pena and Giné (1999), we obtain that the term on the right hand side of the bound above vanishes as $m \to \infty$. Now the desired result immediately follows from Theorem 16.

References

S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the ROC curve. *J. Mach. Learn. Res.*, 6:393–425, 2005.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1587, 1997.

S. Arlot. Model selection by resampling techniques. *Electronic Journal of Statistics*, 3:557–624, 2009.

M.S. Bansal and D. Fernandez-Baca. Computing distances between partial rankings. *Information Processing Letters*, 109:238–241, 2009.

J.P. Barthélémy and B. Montjardet. The median procedure in cluster analysis and social choice theory. *Mathematical Social Sciences*, 1:235–267, 1981.

N. Baskiotis, S. Clémençon, M. Depecker, and N. Vayatis. R-implementation of the TreeRank algorithm. *Submitted for publication*, 2009.

N. Betzler, M.R. Fellows, J. Guo, R. Niedermeier, and F.A. Rosamond. Computing kemeny rankings, parameterized by the average kt-distance. In *Proceedings of the 2nd International Workshop on Computational Social Choice*, 2008.

G. Biau, L. Devroye, and G. Lugosi. Consistency of Random Forests. *J. Mach. Learn. Res.*, 9: 2039–2057, 2008.

L. Breiman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

I. Charon and O. Hudry. Lamarckian genetic algorithms applied to the aggregation of preferences. *Annals of Operations Research*, 80:281–297, 1998.

S. Clémençon and N. Vayatis. Ranking the best instances. *Journal of Machine Learning Research*, 8:2671–2699, 2007.

S. Clémençon and N. Vayatis. Empirical performance maximization based on linear rank statistics. In *NIPS*, volume 3559 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009a.

S. Clémençon and N. Vayatis. Overlaying classifiers: a practical approach to optimal scoring. *To appear in Constructive Approximation*, 2009b.

S. Clémençon and N. Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, 2009c.

S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *Proceedings of COLT*, 2005.

S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874, 2008.

S. Clémençon, M. Depecker, and N. Vayatis. Bagging ranking trees. *Proceedings of ICMLA'09*, pages 658–663, 2009.

S. Clémençon, M. Depecker, and N. Vayatis. AUC-optimization and the two-sample problem. In *Proceedings of NIPS'09*, 2010.

S. Clémençon, M. Depecker, and N. Vayatis. Adaptive partitioning schemes for bipartite ranking. *Machine Learning*, 83(1):31–69, 2011.

W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

V. de la Pena and E. Giné. *Decoupling: from Dependence to Independence*. Springer, 1999.

M.M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

P. Diaconis. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, 17(3):949–979, 1989.

R.M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, 1999.

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the 10th International WWW Conference*, pages 613–622, 2001.

J.P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *Proceedings of the 12-th WWW Conference*, pages 366–375, 2003.

R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing partial rankings. *SIAM J. Discrete Mathematics*, 20(3):628–648, 2006.

P. Fishburn. *The Theory of Social Choice*. University Press, Princeton, 1973.

M.A. Fligner and J.S. Verducci (Eds.). *Probability Models and Statistical Analyses for Ranking Data*. Springer, 1993.

Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

J. Friedman and P. Hall. On bagging and non-linear estimation. *Journal of Statistical Planning and Inference*, 137(3):669–683, 2007.

Y. Grandvalet. Bagging equalizes influence. *Machine Learning*, 55:251–270, 2004.

T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall/CRC, 1990.

J.M. Hilbe. *Logistic Regression Models*. Chapman and Hall/CRC, 2009.

J. Howie. Hyperbolic groups. *In Groups and Applications, edited by V. Metaftsis, Ekdoseis Ziti, Thessaloniki*, pages 137–160, 2000.

O. Hudry. Computation of median orders: complexity results. *Annales du LAMSADE: Vol. 3. Proceedings of the Workshop on Computer Science and Decision Theory, DIMACS*, 163:179–214, 2004.

O. Hudry. NP-hardness results for the aggregation of linear orders into median orders. *Ann. Oper. Res.*, 163:63–88, 2008.

E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

I. Ilyas, W. Aref, and A. Elmagarmid. Joining ranked inputs in practice. In *Proceedings of the 28th International Conference on Very Large Databases*, pages 950–961, 2002.

J. G. Kemeny. Mathematics without numbers. *Daedalus*, (88):571–591, 1959.

M.G. Kendall. The treatment of ties in ranking problems. *Biometrika*, (33):239–251, 1945.

A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised rank aggregation with domain-specific expertise. In *IJCAI'09: Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1101–1106, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

M. Laguna, R. Marti, and V. Campos. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers and Operations Research*, 26(12):1217–1230, 1999.

G. Lebanon and J. Lafferty. Conditional models on the ranking poset. In *Proceedings of NIPS'03*, 2003.

B. Mandhani and M. Meila. Tractable search for learning exponential models of rankings. In *Proceedings of AISTATS, Vol. 5 of JMLR:W&CP 5*, 2009.

M. Meila, K. Phadnis, A. Patterson, and J. Bilmes. Consensus ranking under the exponential model. In *Conference on Artificial Intelligence (UAI)*, pages 729–734, 2007.

P.W. Mielke and K.J. Berry. *Permutation Methods*. Springer, 2001.

A. Nemirovski. *Lectures on Probability Theory and Statistics, Ecole d'ete de Probabilities de Saint-Flour XXVIII - 1998*, chapter Topics in Non-Parametric Statistics. Number 1738 in Lecture Notes in Mathematics. Springer, 2000.

D.M. Pennock, E. Horvitz, and C.L. Giles. Social choice theory and recommender systems: analysis of the foundations of collaborative filtering. In *National Conference on Artificial Intelligence*, pages 729–734, 2000.

J.C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, 2003.

Y. Wakabayashi. The complexity of computing medians of relations. *Resenhas*, 3(3):323–349, 1998.

Journal of Machine Learning Research 14 (2013) 75-109 Submitted 2/12; Revised 9/12; Published 1/13

Nested Expectation Propagation for Gaussian Process Classification with a Multinomial Probit Likelihood

Jaakko Riihimäki JAAKKO.RIIHIMAKI@AALTO.FI
Pasi Jylänki PASI.JYLANKI@AALTO.FI
Aki Vehtari AKI.VEHTARI@AALTO.FI
Department of Biomedical Engineering and Computational Science
Aalto University School of Science
P.O. Box 12200
FI-00076 Aalto
Finland

Editor: Neil Lawrence

Abstract

This paper considers probabilistic multinomial probit classification using Gaussian process (GP) priors. Challenges with multiclass GP classification are the integration over the non-Gaussian posterior distribution, and the increase of the number of unknown latent variables as the number of target classes grows. Expectation propagation (EP) has proven to be a very accurate method for approximate inference but the existing EP approaches for the multinomial probit GP classification rely on numerical quadratures, or independence assumptions between the latent values associated with different classes, to facilitate the computations. In this paper we propose a novel nested EP approach which does not require numerical quadratures, and approximates accurately all between-class posterior dependencies of the latent values, but still scales linearly in the number of classes. The predictive accuracy of the nested EP approach is compared to Laplace, variational Bayes, and Markov chain Monte Carlo (MCMC) approximations with various benchmark data sets. In the experiments nested EP was the most consistent method compared to MCMC sampling, but in terms of classification accuracy the differences between all the methods were small from a practical point of view.

Keywords: Gaussian process, multiclass classification, multinomial probit, approximate inference, expectation propagation

1. Introduction

Gaussian process (GP) priors enable flexible model specification for Bayesian classification. In multiclass GP classification, the posterior inference is challenging because each target class increases the number of unknown latent variables by the number of observations n. Typically, independent GP priors are set for the latent values for each class and this is assumed throughout this paper. Since all the latent values depend on each other through the likelihood, they become a posteriori dependent, which can rapidly lead to computationally unfavorable scaling as the number of classes c grows. A cubic scaling in c is prohibitive, and from a practical point of view, a desired complexity is $O(cn^3)$ which is typical for the most existing approaches for multiclass GP classification. The cubic scaling with respect to the number of data points is standard for full GP priors, and to reduce this n^3 complexity, sparse approximations can be used, but these are not considered in this paper.

As an additional challenge, the posterior inference is analytically intractable because the likelihood term related to each observation is non-Gaussian and depends on multiple latent values (one for each class).

A Markov chain Monte Carlo (MCMC) approach for multiclass GP classification with a softmax likelihood (also called a multinomial logistic likelihood) was described by Neal (1998). Sampling of the latent values with the softmax model is challenging because the dimensionality is often high and standard methods such as the Metropolis-Hastings and Hamiltonian Monte Carlo algorithms require tuning of the step size parameters. Later Girolami and Rogers (2006) proposed an alternative approach based on the multinomial probit likelihood which can be augmented with auxiliary latent variables. This enables a convenient Gibbs sampling framework in which the latent function values are conditionally independent between classes and normally distributed. If the hyperparameters are sampled, one MCMC iteration scales as $O(cn^3)$ which can become computationally expensive for large n because thousands of posterior draws may be required to obtain uncorrelated posterior samples, and strong dependency between the hyperparameters and latent values can cause slow mixing of the chains.

To speed up the inference, Williams and Barber (1998) used the Laplace approximation (LA) to approximate the non-Gaussian posterior distribution of the latent function values with a tractable Gaussian distribution. Conveniently the LA approximation with the softmax likelihood leads to an efficient representation of the approximative posterior covariance scaling as $O((c+1)n^3)$, which facilitates considerably the predictions and gradient-based type-II maximum a posteriori (MAP) estimation of the covariance function hyperparameters. Later Girolami and Rogers (2006) proposed a factorized variational Bayes approximation (VB) for the augmented multinomial probit model. Assuming the latent values and the auxiliary variables a posteriori independent, a computationally efficient posterior approximation scheme is obtained. If the latent processes related to each class share the same fixed hyperparameters, VB requires only one $O(n^3)$ matrix inversion per iteration step compared to LA in which $c+1$ such inverses are required in each iteration. Recently, Chai (2012) proposed an alternative variational bounding approximation for the multinomial logistic likelihood, which results in $O(c^3n^3)$ base scaling. To reduce the computational complexity, sparse approximations were determined by active inducing set selection.

Expectation propagation (EP) is the method of choice in binary GP classification where it has been found very accurate with a reasonable computational cost (Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008). Two types of EP approximations have been considered for the multiclass setting; the first assuming the latent values from different classes a posteriori independent (IEP) and the second assuming them fully correlated (Seeger and Jordan, 2004; Seeger et al., 2006; Girolami and Zhong, 2007). Incorporating the full posterior couplings requires evaluating the non-analytical moments of c-dimensional tilted distributions which Girolami and Zhong (2007) approximated with Laplace's method resulting in an approximation scheme known as Laplace propagation described by Smola et al. (2004). Earlier Seeger and Jordan (2004) proposed an alternative approach where the full posterior dependencies were approximated by enforcing a similar structure for the posterior covariance as in LA using the softmax likelihood. This enables a posterior representation scaling as $O((c+1)n^3)$ but the proposed implementation requires a c-dimensional numerical quadrature and double-loop optimization to obtain a restricted-form site covariance approximation for each likelihood term (Seeger and Jordan, 2004).[1] To reduce the computational demand of EP,

1. Seeger and Jordan (2004) achieve also a linear scaling in the number of training points but we omit sparse approaches here.

factorized posterior approximations were proposed by both Seeger et al. (2006) and Girolami and Zhong (2007). Both approaches omit the between-class posterior dependencies of the latent values which results in a posterior representation scaling as $O(cn^3)$. The approaches rely on numerical two-dimensional quadratures for evaluating the moments of the tilted distributions with the main difference being that Seeger et al. (2006) used fewer two-dimensional quadratures for computational speed-up.

A different EP approach for the multiclass setting was described by Kim and Ghahramani (2006) who adopted the threshold function as an observation model. Each threshold likelihood term factorizes into $c-1$ terms dependent on only two latent values. This property can be used to transform the inference onto an equivalent non-redundant model which includes $n(c-1)$ unknown latent values with a Gaussian prior and a likelihood consisting of $n(c-1)$ factorizing terms. It follows that standard EP methodology for binary GP classification (Rasmussen and Williams, 2006) can be applied for posterior inference but a straightforward implementation results in a posterior representation scaling as $O((c-1)^3n^3)$ and means to improve the scaling are not discussed by Kim and Ghahramani (2006). Contrary to the usual EP approach of maximizing the marginal likelihood approximation, Kim and Ghahramani (2006) determined the hyperparameters by maximizing a lower bound on the log marginal likelihood in a similar way as is done in the expectation maximization (EM) algorithm. Recently Hernández-Lobato et al. (2011) introduced a robust generalization of the multiclass GP classifier with a threshold likelihood by incorporating n additional binary indicator variables for modeling possible labeling errors. Efficiently scaling EP inference is obtained by making the IEP assumption.

In this paper, we focus on the multinomial probit model and describe an efficient quadrature-free nested EP approach for multiclass GP classification that scales as $O((c+1)n^3)$. The proposed EP method takes into account all the posterior covariances between the latent variables, and the posterior computations scale as efficiently as in the LA approximation. We validate the proposed nested EP algorithm with several experiments. First, we compare the nested EP algorithm to various quadrature-based EP methods with respect to the approximate marginal distributions of the latent values and class probabilities with fixed hyperparameter values, and show that nested EP achieves similar accuracy in a computationally efficient manner. Using the nested EP algorithm, we study visually the utility of the full EP approximation over IEP, and compare their convergence properties. Second, we compare nested EP and IEP to other Gaussian approximations (LA and VB). We visualize the accuracy of the approximate marginal distributions with respect to MCMC, illustrate the suitability of the respective marginal likelihood approximations for type-II MAP estimation of the covariance function hyperparameters, and discuss their computational complexities. Finally, we compare the predictive performance of all these methods with estimation of the hyperparameters using several real-world data sets. Since LA is known to be fast, we also test whether the predictive probability estimates of LA can be further improved using Laplace's method as described by Tierney and Kadane (1986).

2. Gaussian Processes for Multiclass Classification

We consider a classification problem consisting of d-dimensional input vectors $\mathbf{x}_i$ associated with target class labels $y_i \in \{1,\ldots,c\}$, where $c > 2$, for $i = 1,\ldots,n$. All the class labels are collected in the $n \times 1$ target vector $\mathbf{y}$, and all the covariate vectors are collected in the matrix $X = [\mathbf{x}_1,\ldots,\mathbf{x}_n]^T$ of size $n \times d$. Given the latent function values $\mathbf{f}_i = \left[f_i^1, f_i^2, \ldots, f_i^c\right]^T = \mathbf{f}(\mathbf{x}_i)$ at the observed input

locations $\mathbf{x}_i$, the observations y_i are assumed independently and identically distributed as defined by the observation model $p(y_i|\mathbf{f}_i)$. The latent vectors related to all the observations are denoted by $\mathbf{f} = \left[f_1^1,\ldots,f_n^1,f_1^2,\ldots,f_n^2,\ldots,f_1^c,\ldots,f_n^c\right]^T$.

Our goal is to predict the class membership for a new input vector $\mathbf{x}_*$ given the observed data $\mathcal{D} = \{X,\mathbf{y}\}$, which is why we need to make some assumptions on the unknown function $f(\mathbf{x})$. We set a priori independent zero-mean Gaussian process priors on the latent values related to each class, which is the usual assumption in multiclass GP classification (see, for example, Williams and Barber, 1998; Seeger and Jordan, 2004; Rasmussen and Williams, 2006; Girolami and Zhong, 2007). This specification results in the following zero-mean Gaussian prior for $\mathbf{f}$:

$$p(\mathbf{f}|X) = \mathcal{N}(\mathbf{f}|\mathbf{0},K),$$

where K is a $cn \times cn$ block-diagonal covariance matrix with matrices $K^1,K^2,\ldots,K^c$ (each of size $n \times n$) on its diagonal. Element $K^k_{i,j}$ of the k'th covariance matrix defines the prior covariance between the function values f_i^k and f_j^k, which is defined by the covariance function $\kappa(\mathbf{x}_i,\mathbf{x}_j)$, that is, $K^k_{i,j} = \kappa(\mathbf{x}_i,\mathbf{x}_j) = \mathrm{Cov}\left[f_i^k,f_j^k\right]$ for the latent values related to class k. A common choice for the covariance function is the squared exponential

$$\kappa_{\mathrm{se}}(\mathbf{x}_i,\mathbf{x}_j|\theta) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{k=1}^{d} l_k^{-2}(x_{i,k}-x_{j,k})^2\right),$$

where $x_{i,k}$ is the k'th component of $\mathbf{x}_i$, and $\theta = \{\sigma^2,l_1,\ldots,l_d\}$ collects the hyperparameters governing the smoothness properties of latent functions. The magnitude parameter σ^2 controls the overall variance of the unknown function values, and the lengthscale parameters $l_1,\ldots,l_d$ control the smoothness of the latent function by defining how fast the correlation decreases in each input dimension. The framework allows separate covariance functions or hyperparameters for different classes but throughout this work, for simplicity, we use the squared exponential covariance function with the same θ for all classes.

In this paper, we consider two different observation models: the softmax model

$$p(y_i|\mathbf{f}_i) = \frac{\exp(f_i^{y_i})}{\sum_{j=1}^{c}\exp(f_i^j)}, \tag{1}$$

and the multinomial probit model

$$p(y_i|\mathbf{f}_i) = \mathrm{E}_{p(u_i)}\left\{\prod_{j=1,j\neq y_i}^{c}\Phi(u_i+f_i^{y_i}-f_i^j)\right\}, \tag{2}$$

where the auxiliary variable u_i is distributed as $p(u_i) = \mathcal{N}(u_i|0,1)$, and $\Phi(x)$ denotes the cumulative density function of the standard normal distribution. The softmax and multinomial probit models are multiclass generalizations of the logistic and the probit models respectively.

By applying Bayes' theorem, the conditional posterior distribution of the latent values can be written as

$$p(\mathbf{f}|\mathcal{D},\theta) = \frac{1}{Z}p(\mathbf{f}|X,\theta)\prod_{i=1}^{n}p(y_i|\mathbf{f}_i), \tag{3}$$

where $Z = p(\mathbf{y}|X,\theta) = \int p(\mathbf{f}|X,\theta)\prod_{i=1}^{n} p(y_i|\mathbf{f}_i)d\mathbf{f}$ is known as the marginal likelihood of θ. Both observation models result in an analytically intractable posterior distribution and therefore approximate methods are needed for integration over the latent variables. Different approximate methods are more suitable for a particular likelihood function because of the convenience of implementation: the softmax is preferable for LA because of the efficient structure and computability of the partial derivatives (Williams and Barber, 1998), while the multinomial probit is preferable for VB, EP and Gibbs sampling because of the convenient auxiliary variable representations (Girolami and Rogers, 2006; Girolami and Zhong, 2007).

3. Approximate Inference Using Expectation Propagation

In this section, we first give a general description of EP for multiclass GP classification and review some existing approaches. Then we present a novel nested EP approach for the multinomial probit model.

3.1 Expectation Propagation for Multiclass GP Classification

Expectation propagation is an iterative algorithm for approximating integrals over functions that factor into simple terms (Minka, 2001b). Using EP the posterior distribution (3) can be approximated with

$$q_{\mathrm{EP}}(\mathbf{f}|\mathcal{D},\theta) = \frac{1}{Z_{\mathrm{EP}}} p(\mathbf{f}|X,\theta)\prod_{i=1}^{n}\tilde{t}_i(\mathbf{f}_i|\tilde{Z}_i,\tilde{\boldsymbol{\mu}}_i,\tilde{\Sigma}_i), \tag{4}$$

where $\tilde{t}_i(\mathbf{f}_i|\tilde{Z}_i,\tilde{\boldsymbol{\mu}}_i,\tilde{\Sigma}_i) = \tilde{Z}_i\mathcal{N}(\mathbf{f}_i|\tilde{\boldsymbol{\mu}}_i,\tilde{\Sigma}_i)$ are local likelihood term approximations parameterized with scalar normalization terms $\tilde{Z}_i$, $c\times 1$ site location vectors $\tilde{\boldsymbol{\mu}}_i$, and $c\times c$ site covariances $\tilde{\Sigma}_i$. In the algorithm, first the site approximations are initialized, and then each site is updated in turns. The update for the i'th site is done by first removing the site term from the marginal posterior which gives the cavity distribution

$$q_{-i}(\mathbf{f}_i) = \mathcal{N}(\mathbf{f}_i|\boldsymbol{\mu}_{-i},\Sigma_{-i}) \propto q(\mathbf{f}_i|\mathcal{D},\theta)\tilde{t}(\mathbf{f}_i)^{-1}.$$

The cavity distribution is then combined with the exact i'th likelihood term $p(y_i|\mathbf{f}_i)$ to form the non-Gaussian tilted distribution

$$\hat{p}(\mathbf{f}_i) = \hat{Z}_i^{-1} q_{-i}(\mathbf{f}_i)p(y_i|\mathbf{f}_i), \tag{5}$$

which is assumed to encompass more information about the true marginal distribution. Next a Gaussian approximation $\hat{q}(\mathbf{f}_i)$ is determined for $\hat{p}(\mathbf{f}_i)$ by minimizing the Kullback-Leibler (KL) divergence $\mathrm{KL}(\hat{p}(\mathbf{f}_i)||\hat{q}(\mathbf{f}_i))$, which for a Gaussian $\hat{q}(\mathbf{f}_i)$ is equivalent to matching the first and second moments of $\hat{q}(\mathbf{f}_i)$ with the corresponding moments of $\hat{p}(\mathbf{f}_i)$. Finally, the parameters of the i'th site are updated so that the mean and covariance of $q(\mathbf{f}_i)$ are consistent with $\hat{q}(\mathbf{f}_i)$. After updating the site parameters, the posterior distribution (4) is updated. This can be done either in a sequential way, where immediately after each site update the posterior is refreshed using a rank-c update, or in a parallel way (see, for example, van Gerven et al., 2009), where the posterior is refreshed only after all the site approximations have been updated once. This procedure is repeated until convergence, that is, until all the marginal distributions $q(\mathbf{f}_i)$ are consistent with $\hat{p}(\mathbf{f}_i)$.

In binary GP classification, determining the moments of the tilted distribution requires solving only one-dimensional integrals, and assuming the probit likelihood function, these univariate integrals can be computed efficiently without numerical quadratures. In the multiclass setting, the problem is how to evaluate the multi-dimensional integrals which are required to determine the moments of the tilted distributions (5). Girolami and Zhong (2007) approximated these moments using the Laplace approximation which results in an algorithm called Laplace propagation (Smola et al., 2004). The problem with the LA approach is that the mean is replaced with the mode of the distribution and the covariance with the inverse Hessian of the log density at the mode. Because of the skewness of the tilted distribution caused by the likelihood function, the LA method can lead to inaccurate mean and covariance estimates in which case the resulting posterior approximation does not correspond to the full EP solution. Seeger and Jordan (2004) estimated the tilted moments using multi-dimensional quadratures, but this becomes computationally demanding when c increases, and to achieve a posterior representation scaling linearly in c, they do an additional optimization step to obtain a constrained site precision matrix for each likelihood term approximation.

Computations can be facilitated by using the IEP approximation where explicit between-class posterior dependencies are omitted. This simplification enables posterior computations scaling linearly in c. The existing approaches for the multinomial probit rely on multiple numerical quadratures for each site update; the implementation of Girolami and Zhong (2007) requires a total of $2c+1$ two-dimensional numerical quadratures for each likelihood term, whereas Seeger et al. (2006) described an alternative approach where only two two-dimensional and $2c-1$ one-dimensional quadratures are needed. Later, we will demonstrate that compared to the full EP approximation, IEP underestimates the uncertainty on the latent values and in practice it may require more iterations than full EP for convergence especially if the hyperparameter setting results in strong between-class posterior couplings.

3.2 Efficiently Scaling Quadrature-Free Implementation

In this section, we present a novel nested EP approach for multinomial probit classification that does not require numerical quadratures or sampling for estimation of the tilted moments and predictive probabilities. The method also leads simultaneously to low-rank site approximations which retain all posterior couplings but results in linear computational scaling with respect to the number of target classes c. Using the proposed nested EP approach a quadrature-free IEP approximation can also be formed with similar computational complexity as the full EP approximation.

3.2.1 QUADRATURE-FREE NESTED EXPECTATION PROPAGATION

Here we use the multinomial probit as the likelihood function because its product form consisting of cumulative Gaussian factors is computationally more suitable for EP than the sum of exponential terms in the softmax likelihood. Given the mean $\boldsymbol{\mu}_{-i}$ and the covariance Σ_{-i} of the cavity distribution, we need to determine the normalization factor $\hat{Z}_i$, mean vector $\hat{\boldsymbol{\mu}}_i$, and covariance matrix $\hat{\Sigma}_i$ of the tilted distribution

$$\hat{p}(\mathbf{f}_i) = \hat{Z}_i^{-1} \mathcal{N}(\mathbf{f}_i|\boldsymbol{\mu}_{-i}, \Sigma_{-i}) \int \mathcal{N}(u_i|0,1) \left(\prod_{j=1, j\neq y_i}^{c} \Phi(u_i + f_i^{y_i} - f_i^j) \right) du_i, \tag{6}$$

which requires solving non-analytical $(c+1)$-dimensional integrals over $\mathbf{f}_i$ and u_i. Instead of quadrature methods (Seeger and Jordan, 2004; Seeger et al., 2006; Girolami and Zhong, 2007), we use EP

to approximate these integrals. At first, this approach may seem computationally very demanding since individual EP approximations are required for each of the n sites. However, it turns out that these inner EP approximations can be updated incrementally between the outer EP loops. This scheme also leads naturally to an efficiently scaling representation for the site precisions $\tilde{\Sigma}_i^{-1}$.

To form a computationally efficient EP algorithm for approximating the tilted moments, it is helpful to consider the joint distribution of $\mathbf{f}_i$ and the auxiliary variable u_i arising from (6). Defining $\mathbf{w}_i = [\mathbf{f}_i^T, u_i]^T$ and removing the marginalization over u_i results in the following augmented tilted distribution:

$$\hat{p}(\mathbf{w}_i) = \hat{Z}_i^{-1} \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathrm{w}_i}, \Sigma_{\mathrm{w}_i}) \prod_{j=1, j\neq y_i}^{c} \Phi(\mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j}), \tag{7}$$

where $\boldsymbol{\mu}_{\mathrm{w}_i} = [\boldsymbol{\mu}_{-i}^T, 0]^T$ and Σ_{w_i} is a block-diagonal matrix formed from Σ_{-i} and 1. Denoting the j'th unit vector of the c-dimensional standard basis by $\mathbf{e}_j$, the auxiliary vectors $\tilde{\mathbf{b}}_{i,j}$ can be written as $\tilde{\mathbf{b}}_{i,j} = [(\mathbf{e}_{y_i} - \mathbf{e}_j)^T, 1]^T$. The normalization term $\hat{Z}_i$ is the same for $\hat{p}(\mathbf{f}_i)$ and $\hat{p}(\mathbf{w}_i)$, and it is defined by $\hat{Z}_i = \int \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathrm{w}_i}, \Sigma_{\mathrm{w}_i}) \prod_{j\neq y_i} \Phi(\mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j}) d\mathbf{w}_i$. The other quantities of interest, $\hat{\boldsymbol{\mu}}_i$ and $\hat{\Sigma}_i$, are equal to the marginal mean and covariance of the first c components of $\mathbf{w}_i$ with respect to $\hat{p}(\mathbf{w}_i)$.

The augmented distribution (7) is of similar functional form as the posterior distribution resulting from a linear binary classifier with a multivariate Gaussian prior on the weights $\mathbf{w}_i$ and a probit likelihood function. Therefore, the moments of (7) can be approximated with EP similarly as in linear classification (see, for example, Qi et al., 2004) or by applying the general EP formulation for latent Gaussian models described by Cseke and Heskes (2011, Appendix C). For clarity, we have summarized a computationally efficient implementation of the algorithm in Appendix A. The augmented tilted distribution (7) is approximated with

$$\hat{q}(\mathbf{w}_i) = Z_{\hat{q}_i}^{-1} \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathrm{w}_i}, \Sigma_{\mathrm{w}_i}) \prod_{j=1, j\neq y_i}^{c} \tilde{Z}_{\hat{q}_{i,j}} \mathcal{N}(\mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j} | \tilde{\alpha}_{i,j}^{-1} \tilde{\beta}_{i,j}, \tilde{\alpha}_{i,j}^{-1}) d\mathbf{w}_i, \tag{8}$$

where the cumulative Gaussian functions are approximated with scaled Gaussian site functions and the normalization constant $\hat{Z}_i$ is approximated with $Z_{\hat{q}_i}$. From now on the site parameters of $\hat{q}(\mathbf{w}_i)$ in their natural exponential form are denoted by $\tilde{\boldsymbol{\alpha}}_i = [\tilde{\alpha}_{i,j}]_{j\neq y_i}^T$ and $\tilde{\boldsymbol{\beta}}_i = [\tilde{\beta}_{i,j}]_{j\neq y_i}^T$.

Note that the probit terms in Equation (7) depend on the unknown latents $\mathbf{f}_i$ only through the linear transformation $\mathbf{g}_i = \tilde{B}_i^T \mathbf{w}_i$, where $\tilde{B}_i = [\tilde{\mathbf{b}}_{i,j}]_{j\neq y_i}$, that is $g_i^j = f_i^{y_i} - f_i^j + u_i$. This relation implies that the likelihood of $\mathbf{f}_i$ increases as the latent value associated with the correct class y_i increases compared to the latents associated with the other classes. Integration over the auxiliary variable u_i results from the conic truncation of the latent variable representation of the multinomial probit model (see, for example, Girolami and Rogers, 2006). This relationship between $\mathbf{w}_i$ and $\mathbf{g}_i$ has two important computational consequences. First, the fully-coupled nested EP solution can be computed by propagating scalar moments of g_i^j which requires solving only one-dimensional integrals because each probit factor in the augmented tilted distribution depends only on the scalar g_i^j (see Appendix A and references therein). Second, it can be shown that the exact mean and covariance of $\mathbf{w}_i \sim \hat{p}(\mathbf{w}_i)$ can be solved from the respective moments of $\mathbf{g}_i$ whose distribution is obtained by $\mathbf{g}_i = \tilde{B}_i^T \mathbf{w}_i$ on Equation (7). Because the dimension of $\mathbf{g}_i$ is $c-1$ we can form computationally cheaper quadrature-based estimates of the tilted moments as described in Section 3.3. We will also use the approximate marginal moments of $\mathbf{g}_i$ to visualize differences in the predictive accuracy of EP and IEP approximations in Section 5.2.

3.2.2 EFFICIENTLY SCALING REPRESENTATION

In this section we show that the approximation (8) leads to matrix computations scaling as $O((c+1)n^3)$ in the evaluation of the moments of the approximate posterior (4). The idea is to show that the site precision matrix $\tilde{\Sigma}_i^{-1}$ resulting from the EP update step with $\hat{\Sigma}_i$ derived from (8) has a similar structure with the Hessian matrix of $\log p(y_i|\mathbf{f}_i)$ in the Laplace approximation (Williams and Barber, 1998; Seeger and Jordan, 2004; Rasmussen and Williams, 2006).

The approximate marginal covariance of $\mathbf{f}_i$ derived from (8) is given by

$$\hat{\Sigma}_i = H^T \left(\Sigma_{\mathrm{w}_i}^{-1} + \tilde{B}_i \tilde{T}_i \tilde{B}_i^T\right)^{-1} H, \tag{9}$$

where the matrix $\tilde{T}_i = \mathrm{diag}(\tilde{\boldsymbol{\alpha}}_i)$ is diagonal,[2] and $H^T = \begin{bmatrix} I_c & \mathbf{0} \end{bmatrix}$ picks up the desired components of $\mathbf{w}_i$, that is, $\mathbf{f}_i = H^T \mathbf{w}_i$. Using the matrix inversion lemma and denoting $B_i = H^T \tilde{B}_i = \mathbf{e}_{y_i} \mathbf{1}^T - E_{-y_i}$, where $E_{-y_i} = [\mathbf{e}_j]_{j \neq y_i}$ and $\mathbf{1}$ is a $(c-1) \times 1$ vector of ones, we can write the tilted covariance as

$$\begin{aligned} \hat{\Sigma}_i &= \Sigma_{-i} - \Sigma_{-i} B_i (\tilde{T}_i^{-1} + \mathbf{1}\mathbf{1}^T + B_i^T \Sigma_{-i} B_i)^{-1} B_i^T \Sigma_{-i} \\ &= (\Sigma_{-i}^{-1} + B_i (\tilde{T}_i^{-1} + \mathbf{1}\mathbf{1}^T)^{-1} B_i^T)^{-1}. \end{aligned} \tag{10}$$

Because in the moment matching step of the EP algorithm the site precision matrix is updated as $\tilde{\Sigma}_i^{-1} = \hat{\Sigma}_i^{-1} - \Sigma_{-i}^{-1}$, we can write

$$\tilde{\Sigma}_i^{-1} = B_i (\tilde{T}_i^{-1} + \mathbf{1}\mathbf{1}^T)^{-1} B_i^T = B_i (\tilde{T}_i - \tilde{\boldsymbol{\alpha}}_i (1 + \mathbf{1}^T \tilde{\boldsymbol{\alpha}}_i)^{-1} \tilde{\boldsymbol{\alpha}}_i^T) B_i^T. \tag{11}$$

Since B_i is a $c \times (c-1)$ matrix, we see that $\tilde{\Sigma}_i^{-1}$ is of rank $c-1$ and therefore a straightforward implementation based on (11) would result into $O((c-1)^3 n^3)$ scaling in the posterior update. However, a more efficient representation can be obtained by simplifying (11) further. Writing $B_i = -A_i E_{-y_i}$, where $A_i = [I_c - \mathbf{e}_{y_i} \mathbf{1}_c^T]$ and $\mathbf{1}_c$ is a $c \times 1$ vector of ones, we get

$$\tilde{\Sigma}_i^{-1} = A_i \left(E_{-y_i} \tilde{T}_i E_{-y_i}^T - \boldsymbol{\pi}_i (\mathbf{1}_c^T \boldsymbol{\pi}_i)^{-1} \boldsymbol{\pi}_i^T\right) A_i^T,$$

where we have defined $\boldsymbol{\pi}_i = E_{-y_i} \tilde{\boldsymbol{\alpha}}_i + \mathbf{e}_{y_i}$ and used $B_i \tilde{\boldsymbol{\alpha}}_i = -A_i \boldsymbol{\pi}_i$. Since $A_i \mathbf{e}_{y_i} = \mathbf{0}$ we can add $\mathbf{e}_{y_i} \mathbf{e}_{y_i}^T$ to the first term inside the brackets to obtain

$$\tilde{\Sigma}_i^{-1} = A_i \Pi_i A_i^T = \Pi_i, \quad \text{where} \quad \Pi_i = \mathrm{diag}(\boldsymbol{\pi}_i) - (\mathbf{1}_c^T \boldsymbol{\pi}_i)^{-1} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^T. \tag{12}$$

The second equality can be explained as follows. Matrix Π_i is of similar form with the precision contribution of the i'th likelihood term, $W_i = -\nabla^2_{\mathbf{f}_i} \log p(y_i|\mathbf{f}_i)$, in the Laplace algorithm (Williams and Barber, 1998), and it has one eigenvector, $\mathbf{1}_c$, with zero eigenvalue: $\Pi_i \mathbf{1}_c = \mathbf{0}$. It follows that $A_i \Pi_i = (I_c - \mathbf{e}_{y_i} \mathbf{1}_c^T) \Pi_i = \Pi_i - \mathbf{e}_{y_i} \mathbf{0}^T = \Pi_i$ and therefore $\tilde{\Sigma}_i^{-1} = \Pi_i$. Matrix Π_i is also precisely of the same form as the a priori constrained site precision block that Seeger and Jordan (2004) determined by double-loop optimization of $\mathrm{KL}(\hat{q}(\mathbf{f}_i)||q(\mathbf{f}_i))$.

In a similar fashion, we can determine a simple formula for the natural location parameter $\tilde{\boldsymbol{\nu}}_i = \tilde{\Sigma}_i^{-1} \tilde{\boldsymbol{\mu}}_i$ as a function of $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$. The marginal mean of $\mathbf{f}_i$ with respect to $\hat{q}(\mathbf{w}_i)$ is given by

$$\hat{\boldsymbol{\mu}}_i = H_i^T \left(\Sigma_{\mathrm{w}_i}^{-1} + \tilde{B}_i \tilde{T}_i \tilde{B}_i^T\right)^{-1} \left(\Sigma_{\mathrm{w}_i}^{-1} \boldsymbol{\mu}_{\mathrm{w}_i} + \tilde{B}_i \tilde{\boldsymbol{\beta}}_i\right), \tag{13}$$

2. We use the following notation: $\mathrm{diag}(\mathbf{a})$ with a vector argument is a square matrix with $\mathbf{a}$ on the main diagonal, and $\mathrm{diag}(A)$ with a matrix argument is a column vector containing the diagonal elements of the matrix A.

which we can write using the matrix inversion lemma as

$$\hat{\boldsymbol{\mu}}_i = \hat{\Sigma}_i \Sigma_{-i}^{-1} \boldsymbol{\mu}_{-i} + \Sigma_{-i} B_i (\tilde{T}_i^{-1} + \mathbf{1}\mathbf{1}^T + B_i^T \Sigma_{-i} B_i)^{-1} \tilde{T}_i^{-1} \tilde{\boldsymbol{\beta}}_i. \tag{14}$$

Using the update formula $\tilde{\boldsymbol{\nu}}_i = \hat{\Sigma}_i^{-1} \hat{\boldsymbol{\mu}}_i - \Sigma_{-i}^{-1} \boldsymbol{\mu}_{-i}$ resulting from the EP moment matching step and simplifying further with the matrix inversion lemma, the site location $\tilde{\boldsymbol{\nu}}_i$ can be written as

$$\tilde{\boldsymbol{\nu}}_i = B_i \left(\tilde{\boldsymbol{\beta}}_i - \tilde{\boldsymbol{\alpha}}_i a_i \right) = a_i \boldsymbol{\pi}_i - E_{-y_i} \tilde{\boldsymbol{\beta}}_i, \tag{15}$$

where $a_i = (\mathbf{1}^T \tilde{\boldsymbol{\beta}}_i)/(\mathbf{1}_c^T \boldsymbol{\pi}_i)$. The site precision vector $\tilde{\boldsymbol{\nu}}_i$ is orthogonal with $\mathbf{1}_c$, that is, $\mathbf{1}_c^T \tilde{\boldsymbol{\nu}}_i = 0$, which is congruent with (12). Note that with results (12) and (15), the mean and covariance of the approximate posterior (4) can be evaluated using only $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$. It follows that the posterior (predictive) means and covariances as well as the marginal likelihood can be evaluated with similar computational complexity as with the Laplace approximation (Williams and Barber, 1998; Rasmussen and Williams, 2006). For clarity the main components are summarized in Appendix B. The IEP approximation in our implementation is formed by matching the i'th marginal covariance with $\mathrm{diag}(\mathrm{diag}(\hat{\Sigma}_i))$, and the corresponding mean with $\hat{\boldsymbol{\mu}}_i$.

3.2.3 EFFICIENT IMPLEMENTATION

Approximating the tilted moments using inner EP for each site may appear too slow for larger problems because typically several iterations are required to achieve convergence. However, the number of inner-loop iterations can be reduced by storing the site parameters $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$ after each inner EP run and continuing from the previous values in the next run. This framework where the inner site parameters $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$ are updated iteratively instead of $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\Sigma}_i$, can be justified by writing the posterior approximation (4) using the approximative site terms from (8):

$$q(\mathbf{f}|\mathcal{D},\theta) \propto p(\mathbf{f}|X,\theta) \prod_{i=1}^{n} \int \mathcal{N}(u_i|0,1) \prod_{j=1, j\neq y_i}^{c} \tilde{Z}_{\hat{q}_i,j} \mathcal{N}(u_i + f_i^{y_i} - f_i^j | \tilde{\alpha}_{i,j}^{-1} \tilde{\beta}_{i,j}, \tilde{\alpha}_{i,j}^{-1}) du_i. \tag{16}$$

Calculating the Gaussian integral over u_i leads to the same results for $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\Sigma}_i$ as derived earlier (Equations 12 and 15). Apart from the integration over the auxiliary variables u_i, Equation (16) resembles an EP approximation where $n(c-1)$ probit terms of the form $\Phi(u_i + f_i^{y_i} - f_i^j)$ are approximated with Gaussian site functions. In accordance with the standard EP framework we form the cavity distribution $q_{-i}(\mathbf{f}_i)$ by removing $c-1$ sites from (16) and subsequently refine $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$ using the mean and covariance of the tilted distribution (6). If we alternatively expand only the i'th site approximation with respect to u_i and write the corresponding marginal approximation as

$$q(\mathbf{f}_i|\mathcal{D},\theta) \propto q_{-i}(\mathbf{f}_i) \int \mathcal{N}(u_i|0,1) \prod_{j=1, j\neq y_i}^{c} \tilde{Z}_{\hat{q}_i,j} \mathcal{N}(u_i + f_i^{y_i} - f_i^j | \tilde{\alpha}_{i,j}^{-1} \tilde{\beta}_{i,j}, \tilde{\alpha}_{i,j}^{-1}) du_i, \tag{17}$$

we can consider updating only one of the approximative terms in (17) at a time. This is equivalent to starting the inner EP iterations with the values of $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$ from the previous outer-loop iteration instead of a zero initialization which is customary to standard EP implementations. In our experiments, only one inner-loop iteration per site was found sufficient for convergence with comparable number of outer-loop iterations, which results in significant computational savings in the tilted moment evaluations.

The previous interpretation of the algorithm is also useful for defining damping (Minka and Lafferty, 2002), which is commonly used to improve the numerical stability and convergence of EP. In damping the site parameters in their natural exponential forms are updated to a convex combination of the old and new values. Damping cannot be directly applied on the site precision matrix $\Pi_i = \tilde{\Sigma}_i^{-1}$ because the constrained form of the site precision (12) is lost. Instead we damp the updates on $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$ which preserves the desired structure. This can be justified with the same arguments as in the previous paragraph where we considered updating only one of the approximative terms in (17) at a time. Convergence of the nested EP algorithm with full posterior couplings using this scheme is illustrated with different damping levels in Section 5.4.

3.3 Quadrature-Based Full EP Implementation

A challenge in forming the fully-coupled EP approximation using numerical quadratures is how to obtain a site precision structure, which results in efficiently scaling posterior computations. Seeger and Jordan (2004) used c-dimensional Gauss-Hermite rules and determined a similar site precision matrix as in Equation (12) by optimizing $\mathrm{KL}(\hat{p}_i(\mathbf{f}_i)||q(\mathbf{f}_i))$. In this section, we use the ideas from Section 3.2 to form a simpler fully-coupled EP algorithm that uses similar approximate site precision structures determined directly using $(c-1)$-dimensional quadratures instead of separate optimizations.

We use the previously defined transformation $\mathbf{g}_i = \tilde{B}_i^T \mathbf{w}_i$, where $\mathbf{w}_i \sim \hat{p}(\mathbf{w}_i)$, and denote the tilted mean vector and covariance matrix of $\mathbf{w}_i$ with $\hat{\boldsymbol{\mu}}_{\mathrm{w}_i}$ and $\hat{\Sigma}_{\mathrm{w}_i}$. Analogously, we denote the corresponding moments of $\mathbf{g}_i$ resulting from the transformation with $\hat{\boldsymbol{\mu}}_{\mathrm{g}_i}$ and $\hat{\Sigma}_{\mathrm{g}_i}$. Making the transformation on (7) and differentiating twice with respect to $\boldsymbol{\mu}_{\mathrm{w}_i}$, it can be shown that the following relation holds between the exact covariance matrices of the random vectors $\mathbf{w}_i$ and $\mathbf{g}_i$:

$$\hat{\Sigma}_{\mathrm{g}_i} = \tilde{B}_i^T \hat{\Sigma}_{\mathrm{w}_i} \tilde{B}_i = \tilde{B}_i^T (\Sigma_{\mathrm{w}_i}^{-1} + \tilde{B}_i \Lambda_i \tilde{B}_i^T)^{-1} \tilde{B}_i, \tag{18}$$

where Σ_{w_i} is the cavity covariance of $\mathbf{w}_i$. Solving Λ_i from (18) gives

$$\Lambda_i = \hat{\Sigma}_{\mathrm{g}_i}^{-1} - \Sigma_{\mathrm{g}_i}^{-1}, \tag{19}$$

where $\Sigma_{\mathrm{g}_i} = B_i^T \Sigma_{-i} B_i + \mathbf{1}\mathbf{1}^T$, and $\hat{\Sigma}_{\mathrm{g}_i}$ can be estimated with a $(c-1)$-dimensional quadrature rule. The marginal tilted covariance of $\mathbf{f}_i$ can be computed from $\hat{\Sigma}_{\mathrm{w}_i}$ similarly as in Equations (9) and (10), and the corresponding site precision matrix $\tilde{\Sigma}_i^{-1}$ can be computed as in Equation (11) with Λ_i now in place of $\tilde{T}_i$. This gives the following site precision structure

$$\tilde{\Sigma}_i^{-1} = B_i (\Lambda_i^{-1} + \mathbf{1}\mathbf{1}^T)^{-1} B_i^T,$$

which depends only on Λ_i. The form of the site precision is similar to nested EP, except that now Λ_i is a full matrix, which would result in the unfavorable $O((c-1)^3 n^3)$ posterior scaling. Therefore, we approximate Λ_i with its diagonal to get the same structure as in Equation (12), where now $\tilde{\Lambda}_i = \mathrm{diag}(\mathrm{diag}(\Lambda_i))$ is used instead of $\tilde{T}_i$. This results in posterior computations scaling linearly in c similarly as with the full nested EP approach.

To estimate the site location parameter $\tilde{\boldsymbol{\nu}}_i$ using quadratures, we proceed in the same way as for the site precision. Making the transformation on (7) and differentiating once with respect to $\boldsymbol{\mu}_{\mathrm{w}_i}$, it can be shown that the tilted means of $\mathbf{w}_i$ and $\mathbf{g}_i$ are related according to

$$\hat{\boldsymbol{\mu}}_{\mathrm{g}_i} = \tilde{B}_i^T \hat{\boldsymbol{\mu}}_{\mathrm{w}_i} = \tilde{B}_i^T (\Sigma_{\mathrm{w}_i}^{-1} + \tilde{B}_i \Lambda_i \tilde{B}_i^T)^{-1} (\Sigma_{\mathrm{w}_i}^{-1} \boldsymbol{\mu}_{\mathrm{w}_i} + \tilde{B}_i \boldsymbol{\xi}_i), \tag{20}$$

where $\hat{\boldsymbol{\mu}}_{\mathrm{w}_i}$ has similar form as in Equation (13). The vector $\boldsymbol{\xi}_i$ corresponds to $\tilde{\boldsymbol{\beta}}_i$ in nested EP, and we can solve it from (20), which results in

$$\boldsymbol{\xi}_i = \hat{\Sigma}_{\mathrm{g}_i}^{-1} \hat{\boldsymbol{\mu}}_{\mathrm{g}_i} - \Sigma_{\mathrm{g}_i}^{-1} \boldsymbol{\mu}_{\mathrm{g}_i}, \tag{21}$$

where $\boldsymbol{\mu}_{\mathrm{g}_i} = B_i^T \boldsymbol{\mu}_{-i}$, and $\hat{\boldsymbol{\mu}}_{\mathrm{g}_i}$ can be estimated using a $(c-1)$-dimensional quadrature. If Λ_i is approximated with its diagonal, we have to substitute $\hat{\Sigma}_{\mathrm{g}_i}^{-1} = \tilde{\Lambda}_i + \Sigma_{\mathrm{g}_i}^{-1}$ in Equation (21), which results from the diagonal approximation of Λ_i made in Equation (19). In the same way as in Equations (13)-(15), we get the following expression for the site location

$$\tilde{\boldsymbol{\nu}}_i = B_i(\boldsymbol{\xi}_i - \tilde{\Lambda}_i \mathbf{1}(1 + \mathbf{1}^T \tilde{\Lambda}_i \mathbf{1})^{-1} \mathbf{1}^T \boldsymbol{\xi}_i),$$

which depends only on $\boldsymbol{\xi}_i$ and $\tilde{\Lambda}_i$. This site location structure is similar to nested EP (15) with $\boldsymbol{\xi}_i$ in place of $\tilde{\boldsymbol{\beta}}_i$. Using these results, a quadrature-based full EP algorithm can be implemented in the same way as the outer-loop of nested EP. Later in Section 5.1, we validate this approximate $(c-1)$-dimensional quadrature approach by comparing the tilted moments to those of a more expensive straightforward $(c+1)$-dimensional full quadrature solution.

4. Other Approximations for Bayesian Inference

In this section we discuss all the other approximations considered in this paper for multiclass GP classification. First we give a short description of the LA method. Then we show how it can be improved upon by computing corrections to the marginal predictive densities using Laplace's method as described by Tierney and Kadane (1986). Finally, we briefly summarize the MCMC and VB approximations.

4.1 Laplace Approximation

In the Laplace approximation a second order Taylor expansion of $\log p(\mathbf{f}|\mathcal{D},\theta)$ is made around the posterior mode $\hat{\mathbf{f}}$ which can be determined using Newton's method as described by Williams and Barber (1998) and Rasmussen and Williams (2006). This results in the posterior approximation

$$q_{\mathrm{LA}}(\mathbf{f}|\mathcal{D},\theta) = \mathcal{N}\left(\mathbf{f}|\hat{\mathbf{f}}, (K^{-1} + W)^{-1}\right),$$

where $W = -\nabla^2_{\mathbf{f}} \log p(\mathbf{y}|\mathbf{f})|_{\mathbf{f}=\hat{\mathbf{f}}}$ and in which $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n p(y_i|\mathbf{f}_i)$. With the softmax likelihood (1), the submatrix of W related to each observation will have a similar structure with Π_i in (12), which enables efficient posterior computations that scale linearly in c as already discussed in the case of EP.

4.1.1 IMPROVING MARGINAL POSTERIOR DISTRIBUTIONS

In Gaussian process classification, the LA and EP methods can be used to efficiently form a multivariate Gaussian approximation for the posterior distribution of the latent values. Recently, motivated by the earlier ideas of Tierney and Kadane (1986), two methods have been proposed for improving the marginal posterior distributions in latent Gaussian models; one based on subsequent use of Laplace's method (Rue et al., 2009), and one based on EP (Cseke and Heskes, 2011). Because in classification the focus is not on the predictive distributions of the latent values but on the predictive probabilities related to a test input $\mathbf{x}_*$, applying these methods would require additional

numerical integration over the improved posterior approximation of the corresponding latent value $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$. In the multiclass setting integration over a multi-dimensional space is required which becomes computationally demanding to perform, for example, in a grid if c is large. To avoid this integration, we test computing the corrections directly for the predictive class probabilities following another approach presented by Tierney and Kadane (1986). A related idea for approximating the predictive distribution of linear model coefficients directly with a deterministic approximation has been discussed by Snelson and Ghahramani (2005).

The posterior mean of a smooth and positive function $h(f)$ is given by

$$\mathrm{E}[h(f)] = \frac{\int h(f)p(y|f)p(f)df}{\int p(y|f)p(f)df}, \tag{22}$$

where $p(y|f)$ is the likelihood function and $p(f)$ is the prior distribution. Tierney and Kadane (1986) proposed to approximate both integrals in (22) separately with Laplace's method. This approach can be readily applied for approximating the posterior predictive probabilities $p(y_*|\mathbf{x}_*)$ of class memberships $y_* \in \{1,...,c\}$ which are given by

$$p(y_*|\mathbf{x}_*, \mathcal{D}) = \frac{1}{Z}\iint p(y_*|\mathbf{f}_*)p(\mathbf{f}_*|\mathbf{f}, \mathbf{x}_*, X)p(\mathbf{f}|X)p(\mathbf{y}|\mathbf{f})d\mathbf{f}d\mathbf{f}_*, \tag{23}$$

where $Z = \iint p(\mathbf{f}_*|\mathbf{f}, \mathbf{x}_*, X)p(\mathbf{f}|X)p(\mathbf{y}|\mathbf{f})d\mathbf{f}d\mathbf{f}_* = \int p(\mathbf{f}|X)p(\mathbf{y}|\mathbf{f})d\mathbf{f}$ is the marginal likelihood. With a fixed class label y_* the integrals can be approximated by a straightforward application of either LA or EP, which is already done for the marginal likelihood Z in the standard approximations. The LA method can be used for smooth and positive functions such as the softmax whereas EP is applicable for a wider range of models.

The integral on the right side of (23) is equivalent to the marginal likelihood resulting from a classification problem with one additional training point y_*. To compute the predictive probabilities for all classes, we evaluate this extended marginal likelihood consisting of $n+1$ observations with y_* fixed to one of the c possible class labels at a time. This is computationally demanding because several marginal likelihood evaluations are required for each test input. Additional modifications, for example, initializing the latent values to their predictive mean implied by standard LA, could be done to speed up the computations. Since further approximations can only be expected to reduce the accuracy of the predictions, we do not consider them in this paper, and focus only on the naive implementation due to its ease of use. Since LA is known to be fast, we test the goodness of the improved predictive probability estimates using only LA, and refer to the method as LA-TKP as an extension to the naming used by Cseke and Heskes (2011).

4.2 Markov Chain Monte Carlo

Because MCMC estimates become exact in the limit of infinite sample size, we use MCMC as a gold standard for measuring the performance of the other approximations. Depending on the likelihood, we use two different sampling techniques; scaled Metropolis-Hastings sampling for the softmax function, and Gibbs sampling for the multinomial probit function.

4.2.1 SCALED METROPOLIS-HASTINGS SAMPLING FOR SOFTMAX

To obtain samples from the posterior with the softmax likelihood, the following two steps are alternated. Given the hyperparameter values, the latent values are drawn from the conditional posterior

$p(\mathbf{f}|\mathcal{D},\theta)$ using the scaled Metropolis-Hastings sampling (Neal, 1998). Then, the hyperparameters can be drawn from the conditional posterior $p(\theta|\mathbf{f},\mathcal{D})$, for example, using the Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 1996).

4.2.2 GIBBS SAMPLING FOR MULTINOMIAL PROBIT

Girolami and Rogers (2006) described how to draw samples from the joint posterior using the Gibbs sampler. The multinomial probit likelihood (2) can be written in the form

$$p(y_i|\mathbf{f}_i) = \int \psi(v_i^{y_i} > v_i^k \forall k \neq y_i) \prod_{j=1}^{c} \mathcal{N}(v_i^j|f_i^j,1) d\mathbf{v}_i, \tag{24}$$

where $\mathbf{v}_i = [v_i^1,...,v_i^c]^T$ is a vector of auxiliary variables, and ψ is the indicator function whose value is one if the argument is true and zero otherwise. Gibbs sampling can then be employed by drawing samples alternately for all i from $p(\mathbf{v}_i|\mathbf{f}_i,y_i)$ which is a conic truncation of the multivariate Gaussian distribution, and from $p(\mathbf{f}|\mathbf{v},\theta)$ which is a multivariate Gaussian distribution. Given $\mathbf{v}$ and $\mathbf{f}$ the hyperparameters can be drawn, for example, using HMC.

4.3 Factorized Variational Approximation

A computationally convenient variational Bayesian approximation for $p(\mathbf{f}|\mathcal{D},\theta)$ can be formed by employing the auxiliary variable representation (24) of the multinomial probit likelihood. As shown by Girolami and Rogers (2006), assuming $\mathbf{f}$ a posteriori independent of $\mathbf{v}$ (which contains all $\mathbf{v}_i$) leads to the following approximation

$$q_{\mathrm{VB}}(\mathbf{v},\mathbf{f}|\mathcal{D},\theta) = q(\mathbf{v})q(\mathbf{f}) = \prod_{i=1}^{n} q(\mathbf{v}_i) \prod_{j=1}^{c} q(\mathbf{f}^j),$$

where the latent values associated with the j'th class, $\mathbf{f}^j$, are independent. The posterior approximation $q(\mathbf{f}^j)$ will be a multivariate Gaussian distribution, and $q(\mathbf{v}_i)$ a conic truncation of the multivariate Gaussian distribution (Girolami and Rogers, 2006). Given the hyperparameters, the parameters of $q(\mathbf{v})$ and $q(\mathbf{f})$ can be determined iteratively by maximizing a variational lower bound on the marginal likelihood. Each iteration step requires determining the expectations of $\mathbf{v}_i$ with respect to $q(\mathbf{v}_i)$ which can be obtained by either one-dimensional numerical quadratures or sampling methods. In our implementation, the hyperparameters θ are determined by maximizing the variational lower bound with fixed $q(\mathbf{v})$ and $q(\mathbf{f})$ similarly as in the maximization step of the EM algorithm.

5. Experiments

This section is divided into five parts. In Section 5.1 we compare nested EP to quadrature-based EP in cost and quality. In Section 5.2, we illustrate the differences of the nested EP and IEP approximations in a simple synthetic classification problem. In Section 5.3, we compare visually the quality of the approximate marginal distributions of $\mathbf{f}$, the marginal likelihood approximations and the predictive class probabilities between EP, IEP, VB and LA using a three-class real-world data set. In Section 5.4, we discuss the computational complexities of the different approximate methods, and in Section 5.5, we evaluate them in terms of predictive performance with estimation of the hyperparameters using several benchmark data sets.

5.1 Comparing Nested EP to Numerical Quadrature

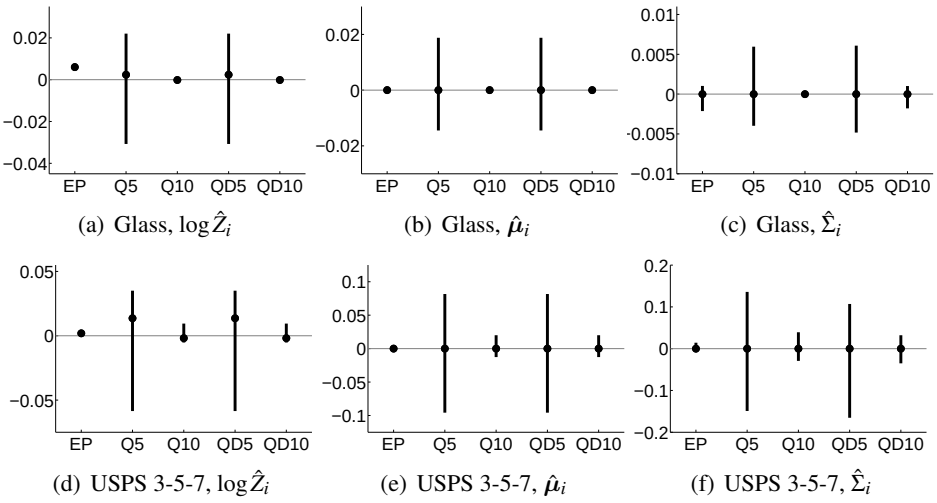
In this section we first validate the accuracy of the inner EP approximation and the full quadrature method described in Section 3.3 for estimation of the tilted moments. Then we compare the accuracy and numerical cost of the nested EP approximation to several quadrature-based EP implementations. In the comparisons, we use two different types of classification data: the Glass data set from the UCI Machine Learning Repository (Frank and Asuncion, 2010), and the USPS 3 vs. 5 vs. 7 data set from the US Postal Service (USPS) database (Hull, 1994). The USPS 3 vs. 5 vs. 7 data set is defined as a three class sub-problem from the USPS repartitioned handwritten digits data by considering classification of 3's vs. 5's vs. 7's.[3] The Glass data has six ($c = 6$) target classes but only a small number of observations ($n = 214$), whereas the USPS 3 vs. 5 vs. 7 data has only three ($c = 3$) target classes but a larger number of training points ($n = 1157$). See also Table 3.

In the first experiment, we examine the tilted moments after two parallel EP outer-loop iterations when the parameters of the cavity distributions are clearly different from their initialized values for all site terms. We fixed the hyperparameters of the squared exponential covariance function to $\log(\sigma^2) = 1$ and $\log(l) = 1$, where the small value of the magnitude parameter leads to a close-to-Gaussian posterior as will be discussed more in Section 5.3. The main reason for not using more difficult hyperparameter values (larger magnitude) in this experiment, was that we had stability problems in the actual EP algorithm using quadratures. Stability could be improved by increasing the number of quadrature points, but this became computationally too expensive with a larger number of classes.

As a baseline approximation, we compute the normalization factors $\hat{Z}_i$, the mean vectors $\hat{\boldsymbol{\mu}}_i$, and the covariance matrices $\hat{\Sigma}_i$ of the tilted distribution (6) for all $i = 1,\ldots,n$ using a $(c+1)$-dimensional Gauss-Hermite product rule with ten quadrature points in each dimension. We call this quadrature method QF10. This provides us a reference by which inner EP and the following four $(c-1)$-dimensional Gauss-Hermite quadrature methods (see Section 3.3 for further details) are assessed: Q5 using five and Q10 using ten quadrature points in each dimension with the full matrix Λ_i, and QD5 using five and QD10 using ten quadrature points in each dimension with the diagonal approximation $\tilde{\Lambda}_i$. Note that implementing an EP algorithm using QF10, Q5 or Q10, we would lose the linear posterior scaling in c. Figure 1 shows the pairwise differences of $\log \hat{Z}_i$, and all the entries of $\hat{\boldsymbol{\mu}}_i$ and $\hat{\Sigma}_i$ with respect to QF10. The mean values and the 95% intervals of the differences are illustrated. The normalization, mean and covariance are well calibrated for all the quadrature methods. Inner EP matches the mean and covariance accurately, but there is a small bias in the normalization term, probably due to the skewed tilted distributions. Variations of the pairwise differences are small with inner EP and the $(c-1)$-dimensional quadratures as long as there are enough quadrature points. Because QD10 agrees well with QF10, from now on, we use it to compute the tilted moments in the full quadrature solution, and refer to this algorithm as QEP.

In the second experiment, we compare the nested and quadrature EP algorithms in accuracy and computational cost. We use Gibbs sampling as a reference method by which nested EP and IEP, QEP, and quadrature-based IEP (QIEP) are measured. Both nested EP algorithms are implemented incrementally, so that only one inner-loop iteration per site is done at each outer-loop iteration step, which results in computational savings (see Section 3.2.3). For QIEP we use the implementation proposed by Seeger et al. (2006) with ten quadrature points for integration over the latent value from each class. We compare the absolute differences of class probabilities and latent means and

3. We use the same data partition as discussed by Rasmussen and Williams (2006).



(a) Glass, $\log \hat{Z}_i$ (b) Glass, $\hat{\boldsymbol{\mu}}_i$ (c) Glass, $\hat{\Sigma}_i$

(d) USPS 3-5-7, $\log \hat{Z}_i$ (e) USPS 3-5-7, $\hat{\boldsymbol{\mu}}_i$ (f) USPS 3-5-7, $\hat{\Sigma}_i$

Figure 1: A comparison of tilted moments after two parallel EP outer-loop iterations using the Glass and USPS 3 vs. 5 vs. 7 data sets. Using a $(c+1)$-dimensional Gauss-Hermite product rule with ten quadrature points in each dimension (QF10) as a baseline result, we compare inner EP and the following $(c-1)$-dimensional Gauss-Hermite quadrature methods: five- and ten-dimensional product rules with full Λ_i (Q5 and Q10) and diagonal $\tilde{\Lambda}_i$ (QD5 and QD10). The mean values and the 95% intervals of the pairwise differences of $\hat{Z}_i$, and all the entries of $\hat{\boldsymbol{\mu}}_i$ and $\hat{\Sigma}_i$ with respect to QF10 are shown. See the text for further details.

variances using the Glass and USPS 3 vs. 5 vs. 7 data sets with the same fixed hyperparameters as earlier. We split the Glass data set randomly into training and test parts, and use the predefined training and test parts for the USPS 3 vs. 5 vs. 7 data set. Table 1 reports the mean and maximum values of the element-wise differences with respect to Gibbs sampling after 30 outer-loop iterations of EP. Table 1 shows also the relative CPU times for training. From the table it can be seen that the differences in accuracy between the methods are small. For the Glass data the fully-coupled EP algorithms give slightly more accurate estimates for the mean and variances of the latents than the IEP algorithms do, but the class probabilities are in practice the same across all the methods. The main observation with the Glass data is that the CPU times of EP, IEP and QIEP are similar for practical purposes, but QEP is clearly slower due to the unfavorable scaling in c. We acknowledge that the performance differences in the relative CPU time are approximate and depend much on the implementation, but to reduce these effects the same outer-loop implementation was used for both nested and quadrature EP with the same fixed number of iterations. It is also worth to notice that QEP and EP have practically the same CPU times with the USPS 3 vs. 5 vs. 7 data where the number of target classes c is only three, but both of them are slower than the IEP algorithms due to larger n and the additional $n \times n$ inversion needed in the posterior update with the fully-coupled solutions (see also Section 5.4). We conclude that because the accuracy of nested and quadrature EP are similar, and we experienced some stability problems with the quadrature solutions in more difficult hyperparameter settings (larger values for the magnitude hyperparameter) and a

Glass		Training				Test			
		EP	QEP	IEP	QIEP	EP	QEP	IEP	QIEP
Latent means	mean	0.048	0.047	0.058	0.067	0.043	0.043	0.055	0.058
	max	0.181	0.186	0.190	0.217	0.176	0.177	0.193	0.181
Latent variances	mean	0.084	0.085	0.325	0.325	0.097	0.097	0.314	0.314
	max	0.553	0.560	0.656	0.667	0.571	0.572	0.643	0.655
Class probabilities	mean	0.004	0.004	0.004	0.004	0.003	0.004	0.004	0.004
	max	0.025	0.026	0.021	0.021	0.021	0.022	0.024	0.024
Relative CPU time		1.245	132.300	1.000	1.175				

USPS 3 vs. 5 vs. 7		Training				Test			
		EP	QEP	IEP	QIEP	EP	QEP	IEP	QIEP
Latent means	mean	0.065	0.065	0.065	0.065	0.006	0.006	0.006	0.006
	max	0.266	0.288	0.266	0.266	0.203	0.202	0.199	0.199
Latent variances	mean	0.167	0.167	0.167	0.167	0.215	0.215	0.216	0.215
	max	0.724	0.724	0.723	0.731	1.021	1.021	1.021	1.021
Class probabilities	mean	0.008	0.008	0.013	0.013	0.001	0.001	0.001	0.001
	max	0.040	0.043	0.059	0.058	0.022	0.024	0.035	0.035
Relative CPU time		2.542	2.602	1.000	1.001				

Table 1: A comparison of nested and quadrature-based EP in terms of accuracy and cost using the Glass and USPS 3 vs. 5 vs. 7 data sets. The table shows the element-wise mean and maximum absolute differences of the latent means and variances and the class probabilities for EP, QEP, IEP, and QIEP with respect to Gibbs sampling. See the text for further details.

small number of quadrature points (for example less than ten), from now on, we use nested EP and IEP implementations due to their stability and good computational scaling.

5.2 Illustrative Comparison of EP and IEP with Synthetic Data

In this section, we study the properties of the proposed nested EP and IEP approximations in a synthetic three-class classification problem with scalar inputs shown in Figure 2. The symbols x (class 1), + (class 2), and o (class 3) indicate the positions of $n = 15$ training inputs generated from three normal distributions with means -1, 2, and 3, and standard deviations 1, 0.5, and 0.5, respectively. The left-most observations from class 1 can be better separated from the others but the observations from classes 2 and 3 overlap more in the input space. We fixed the hyperparameters of the squared exponential covariance function at the corresponding MCMC means: $\log(\sigma^2) = 4.62$ and $\log(l) = 0.26$.

Figure 2(a) shows the predictive probabilities of all tree classes estimated with EP, IEP and MCMC as a function of the input x. At the class boundaries, the methods give similar predictions but elsewhere MCMC is the most confident while IEP seems more conservative. The performance of EP is somewhere between MCMC and IEP, although the differences are small. To explain why

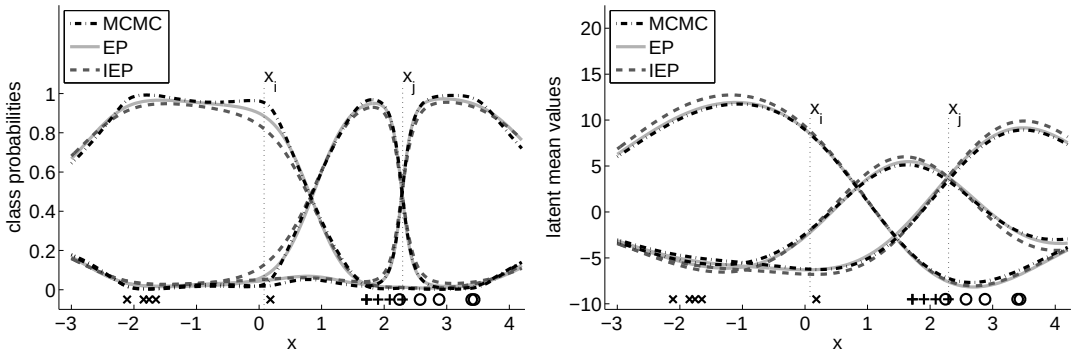


Figure 2: A synthetic one-dimensional example of a three class classification problem, where the MCMC, EP and IEP approximations are compared. The symbols x (class 1), + (class 2), and o (class 3) in the bottom of the plots indicate the positions of $n = 15$ observations. Plot (a) shows the predicted class probabilities, and (b) shows the predicted latent mean values for all three classes. The symbols x_i and x_j indicate two example positions, where the marginal distributions between the latent function values are illustrated in Figures 3 and 4. See the text for explanation.

the predictions differ, we look at the quality of the approximations made for the underlying $\mathbf{f}$. Figure 2(b) shows the approximated latent mean values which are similar at all input locations.

To illustrate the approximate posterior uncertainties of $\mathbf{f}$, we visualize two exemplary marginal distributions at locations x_i and x_j marked in Figure 2. The MCMC samples of f_i^1 and f_i^2 (the latents associated with classes 1 and 2 related to x_i) together with a smoothed density estimate are shown in Figure 3(a). The marginal distribution is non-Gaussian, and the latent values are more likely larger for class 1 than for class 2 indicating a larger predictive probability for class 1. The corresponding EP and IEP approximations are shown in Figures 3(b)-(c). EP captures the shape of the true marginal posterior distribution better than IEP. To illustrate the effect of these differences on the predictive probabilities, we show the unnormalized tilted distributions

$$\hat{p}(\mathbf{g}_i|\mathcal{D},x_i) = q(\mathbf{g}_i|\mathcal{D},x_i) \prod_{k=1,k\neq y_i}^{c} \Phi(g_i^k), \tag{25}$$

where the random vector $\mathbf{g}_i$ is defined in Section 3.2.1, and $q(\mathbf{g}_i|\mathcal{D},x_i)$ is the approximate marginal obtained from $q(\mathbf{f}_i|\mathcal{D},x_i)$ by a linear transformation. Note that the marginal predictive probability for class label y_i with the multinomial probit model (2) can be obtained by appropriately forming the transformation B_i and calculating the integral over $\mathbf{g}_i$ in (25). Figures 3(d)-(f) show the contours of the different approximations of $\hat{p}(\mathbf{g}_i|\mathcal{D},x_i)$ for $k \in \{2,3\}$, which for MCMC are obtained using a smoothed estimate of $q(\mathbf{g}_i|\mathcal{D},x_i)$ determined from transformed samples. The distributions are heavily skewed by the probit factors elsewhere than the upper-right quadrant. Compared to the MCMC estimate, IEP places more probability mass to the other quadrants, and therefore underestimates the predictive probability for class 1 more than EP. The approximate predictive probabilities are 0.95 for MCMC, 0.88 for EP, and 0.82 for IEP.

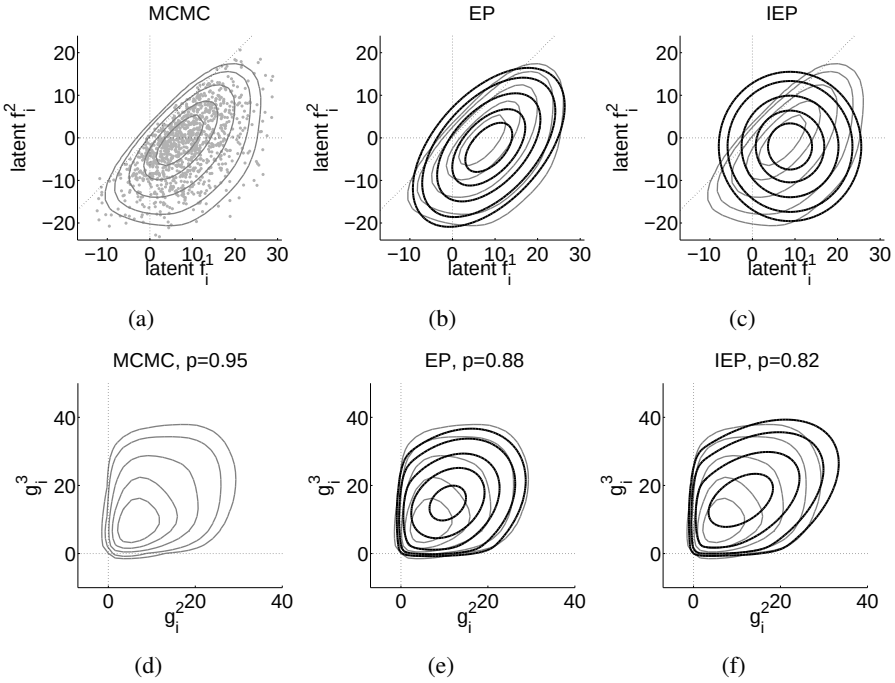


Figure 3: An example of a non-Gaussian marginal posterior distribution for the latent values related to the input x_i in the synthetic example shown in Figure 2. The first row shows the distribution for the latents f_i^1 and f_i^2. Plot (a) shows a scatter-plot of MCMC samples drawn from the posterior and estimated density contour levels which correspond to the areas that include approximately 95%, 90% 75%, 50%, and 25% of the probability mass. Plots (b) and (c) show the equivalent contour levels of the EP and IEP approximations (bold black lines) and the contour levels of the MCMC approximation (gray lines) for comparison. Plots (d)-(f) show contours of $\hat{p}(\mathbf{g}_i|\mathcal{D},x_i)$ for g_i^2 and g_i^3. The probability for class 1 is obtained by calculating the integral over $\mathbf{g}_i$, which results in approximately 0.95 for MCMC, 0.88 for EP, and 0.82 for IEP. See the text for explanation.

The second location x_j is near the class boundary, where all the methods give similar predictive probabilities, although the latent approximations can differ notably as shown in Figures 4(a)-(c), which visualize the marginal approximations for f_j^2 and f_j^3. EP is consistent with the MCMC estimate but due to the independence constraint IEP underestimates the uncertainty of this close-to-Gaussian but non-isotropic marginal distribution. Although Figures 4(d)-(f) show that IEP is more inaccurate than EP, the integral over the tilted distribution of $\mathbf{g}_j$ is in practice the same, since equal amount of probability mass is distributed on both sides of the diagonal in Figure 4(c). The predictive probability for class 2 is approximately 0.47 for all the methods.

5.3 Approximate Marginal Densities with Digit Classification Data

In this section, we compare the predictive performances and marginal likelihood approximations of EP, IEP, VB and LA using the USPS 3 vs. 5 vs. 7 data set, which consists of 1157 training points

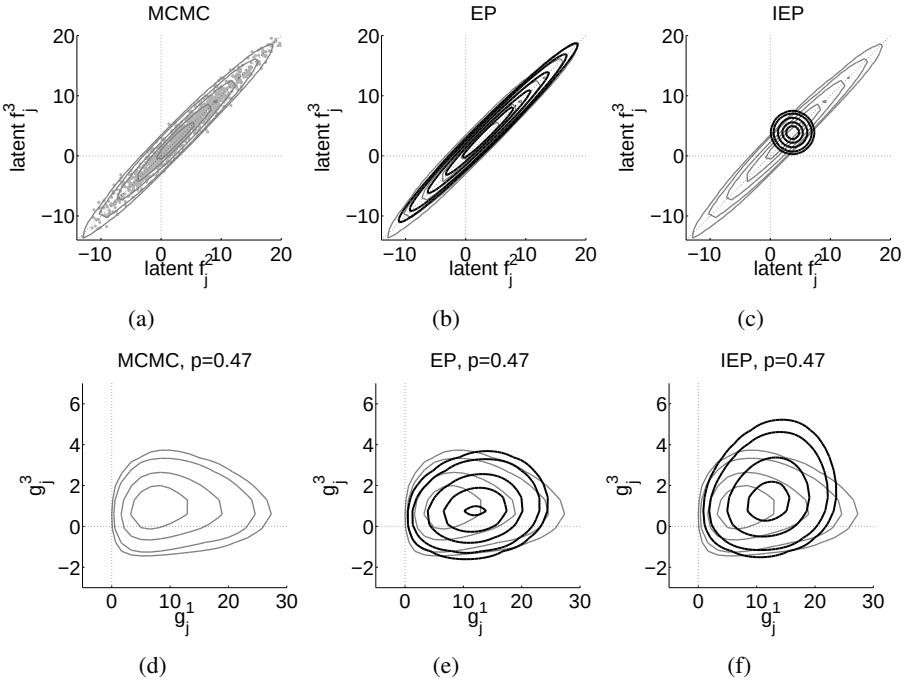


Figure 4: An example of a close-to-Gaussian but non-isotropic marginal posterior distribution for the latent values related to the input x_j in the synthetic example shown in Figure 2. The first row shows the distribution for the latents f_j^2 and f_j^3. Plot (a) shows a scatter-plot of MCMC samples drawn from the posterior and estimated density contour levels which correspond to the areas that include approximately 95%, 90% 75%, 50%, and 25% of the probability mass. Plots (b) and (c) show the equivalent contour levels of the EP and IEP approximations (bold black lines) and the contour levels of the MCMC approximation (gray lines) for comparison. Plots (d)-(f) show contours of $\hat{p}(\mathbf{g}_j|\mathcal{D},x_j)$ for g_j^1 and g_j^3. The probability for class 2 is obtained by calculating the integral over $\mathbf{g}_j$, which results in approximately 0.47 for all the methods. See the text for explanation.

and 1175 test points with 256 covariates. We fixed the hyperparameter values at $\log(\sigma^2) = 4$ and $\log(l) = 2$ which leads to skewed non-Gaussian marginal posterior distributions as will be illustrated shortly.

Figure 5 shows the predictive probabilities of the true class labels for all the approximate methods plotted against the MCMC estimate. The first row shows the training and the second row the test cases. Overall, EP gives the most accurate estimates while IEP slightly underestimates the probabilities for the training cases but performs well for the test cases. Both VB and LA underestimate the predictive probabilities for the test cases, but LA-TKP with the marginal corrections clearly improves the estimates of the LA approximation. Note that the LA methods use a different observation model, and therefore they are compared to the scaled Metropolis-Hastings sampling with the softmax model.

Figure 6 shows an example of the latent marginal posterior distributions for one training point with the correct class label being 2. For each method, the latent pairs (f_i^1, f_i^2), (f_i^1, f_i^3), and (f_i^2, f_i^3),

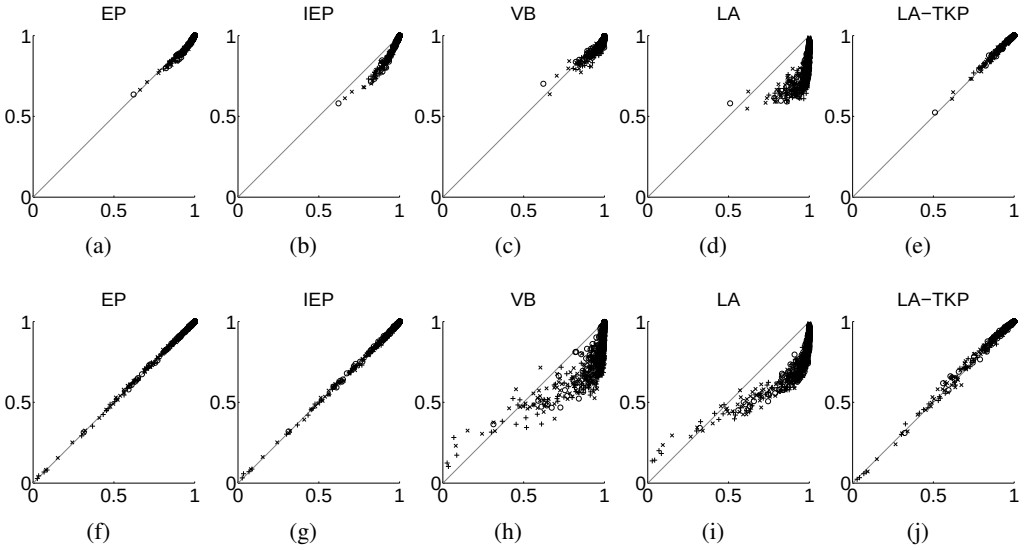


Figure 5: Class probabilities on the USPS 3 vs. 5 vs. 7 data. The MCMC estimates are shown on the x-axis and EP, IEP, VB, LA, and LA-TKP on the y-axis. The first row shows the predictive probabilities of the true class labels for the training points and the second row for the test points. The symbols (x, +, o) corresponds to the handwritten digit target classes 3, 5, and 7. The hyperparameters of the squared exponential covariance function were fixed at $\log(\sigma^2) = 4$ and $\log(l) = 2$.

are shown. The EP approximation agrees reasonably well with the MCMC samples. IEP underestimates the latent uncertainty, especially near the training inputs because of the skewing effect of the likelihood. This seems to affect more the predictive probabilities of the training points in Figure 5(b), which effect can also be seen in the previous example of Figure 2(a) further away from the decision boundary near the input x_i. Figure 6 shows that the VB method underestimates the latent uncertainty. The independence assumption of VB leads to an isotropic approximate distribution, and although the predictive probabilities for the training cases are somewhat consistent with MCMC, the predictions on the test data are less accurate (plots (c) and (h) in Figure 5). Note that the specific hyperparameter values are not optimal for VB, and these values are not supported by the marginal likelihood approximation of VB either, as will be visualized later in this section. The LA approximation captures some of the dependencies between the latent variables associated with different classes, but the joint mode of $\mathbf{f}$ is a poor estimate for the true mean, which causes inaccurate predictive probabilities (plots (d) and (i) in Figure 5). The VB mean estimate is also closer to LA than MCMC, although LA uses a different observation model.

Kuss and Rasmussen (2005) and Nickisch and Rasmussen (2008) discussed how a large value of the magnitude hyperparameter σ^2 can lead to a skewed posterior distribution in binary classification. In the multiclass setting, similar behavior can be seen in the marginal distributions as illustrated in Figures 3 and 6. A large σ^2 leads to a more widely distributed prior which in turn is skewed more strongly by the likelihood where it disagrees with the target class. In the previous comparison, the hyperparameter values were chosen to produce non-Gaussian marginal posterior distributions

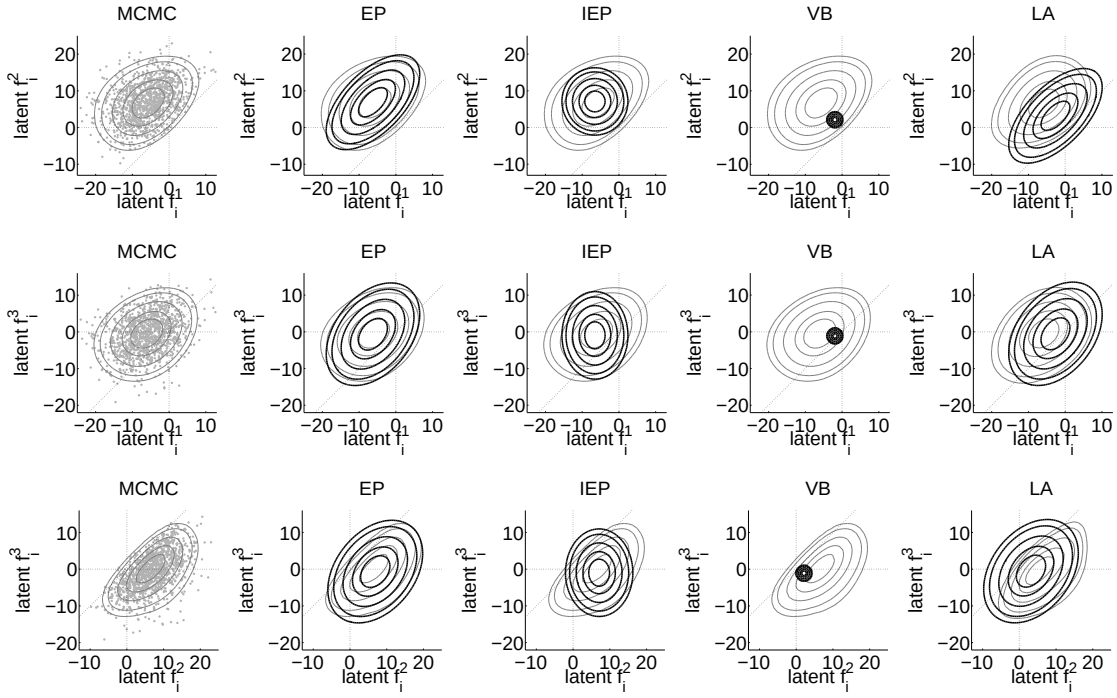


Figure 6: Marginal posterior distributions for one training point with the true class label being 2 on the USPS 3 vs. 5 vs. 7 data. Each row corresponds to one of the latent pairs (f_i^1, f_i^2), (f_i^1, f_i^3), and (f_i^2, f_i^3). The first column shows a scatter-plot of MCMC samples drawn from the posterior and estimated density contour levels which correspond to the areas that include approximately 95%, 90% 75%, 50%, and 25% of the probability mass. The rest of the columns show the equivalent contour levels of the EP, IEP, VB, and LA approximations (bold black lines) and the contour levels of the MCMC approximation (gray lines) for comparison. Note that the last column visualizes a different marginal distribution because LA uses the softmax likelihood. The hyperparameters of the squared exponential covariance function were fixed at $\log(\sigma^2) = 4$ and $\log(l) = 2$ to obtain a non-Gaussian posterior distribution.

for demonstration purposes. However, usually the hyperparameters are estimated by maximizing the marginal likelihood. Kuss and Rasmussen (2005) and Nickisch and Rasmussen (2008) studied the suitability of the marginal likelihood approximations for selecting hyperparameters in binary classification. They compared the calibration of predictive performance and the marginal likelihood estimates on a grid of hyperparameter values. In the following, we extend these comparisons to multiple classes with the USPS data set, for which similar considerations were done by Rasmussen and Williams (2006) with the LA method.

The upper row of Figure 7 shows the log marginal likelihood approximations for EP, IEP, and LA, and the lower bound on evidence for VB as a function of the log-lengthscale $\log(l)$ and log-magnitude $\log(\sigma^2)$ using the USPS 3 vs. 5 vs. 7 data. The middle row shows the log predictive densities evaluated on the test set, and the bottom row shows the corresponding classification accu-

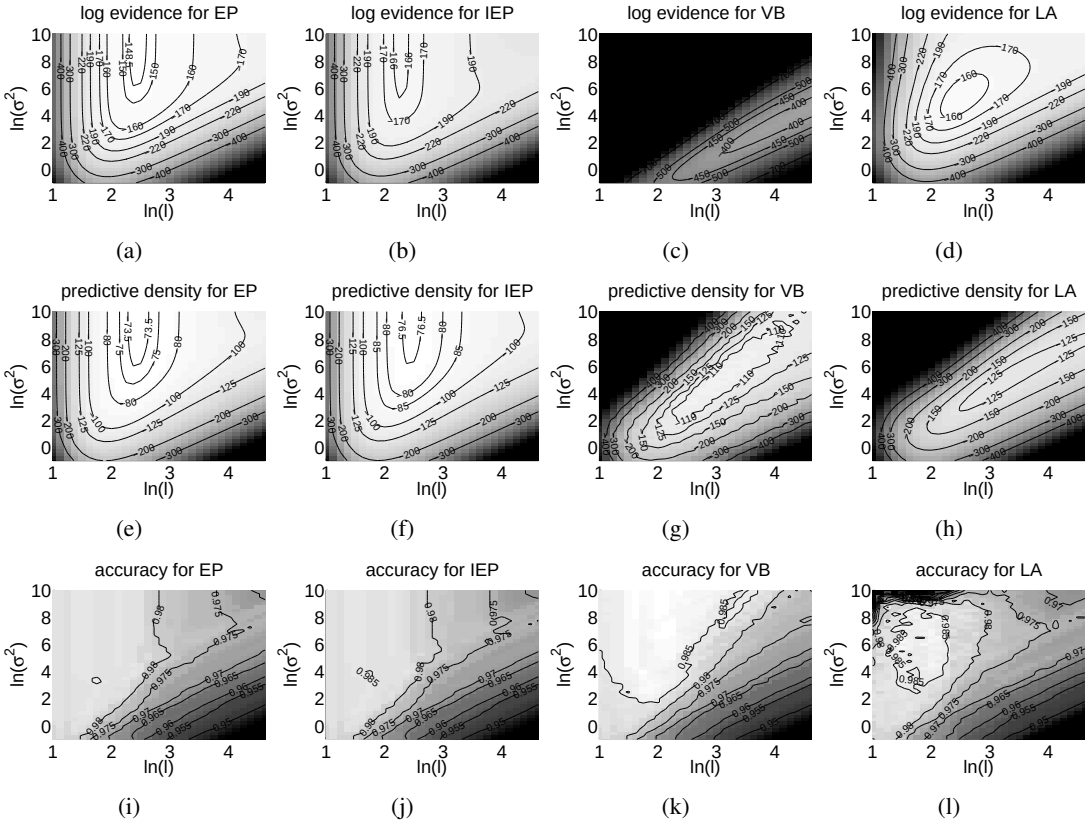


Figure 7: Marginal likelihood approximations and predictive performances as a function of the log-lengthscale $\log(l)$ and log-magnitude $\log(\sigma^2)$ for EP, IEP, VB, and LA on the USPS 3 vs. 5 vs. 7 data. The first row shows the log marginal likelihood approximations, the second row shows the log predictive densities in a test set, and the third row shows the classification accuracies in a test set.

racies. The marginal likelihood approximations and predictive densities for EP and IEP appear to be similar, but the maximum contour of the log marginal likelihood for IEP (the contour labeled with -166 in plot (b) of Figure 7) does not coincide with the maximum contour of the predictive density (the contour labeled with -76.5 in plot (f) of Figure 7), which is why a small bias can occur if the approximate marginal likelihood is used for selecting the hyperparameter values. With EP there is a good agreement between the maximum values in plots (a) and (e), and overall, the log predictive densities are higher than with the other approximations. The log predictive densities of VB and LA are small where $\log(\sigma^2)$ is large (regions where $q(\mathbf{f}|\mathcal{D},\theta)$ is likely to be non-Gaussian), but on the other hand, also the marginal likelihood approximations favor the areas of smaller $\log(\sigma^2)$ values.

There is a reasonable agreement with the marginal likelihood approximations and classification accuracies with EP and IEP, although the maximum accuracies are slightly lower than with VB and LA. The maximum accuracies are very high with VB, but the region of the highest accuracy does not agree with the region of the highest estimate of the marginal likelihood. With LA the marginal like-

lihood estimate is better calibrated in terms of classification accuracy, but the performance worsens when the posterior distribution is skewed with large values of $\log(\sigma^2)$.

5.4 Computational Complexity and Convergence

In this section we consider the computational complexities of the approximate methods for one iteration with fixed hyperparameter values. Note that the following discussion is only approximate, and that the practical efficiency of the algorithms depends much on implementations and the choices of convergence criteria.

Table 2 summarizes the approximate scaling of the number of computations as a function of n and c. EP and IEP refer to the fully-coupled and class-independent approximations, respectively, determined with the proposed nested EP algorithm. QEP refers to the quadrature-based fully-coupled solution using the diagonal approximation $\tilde{\Lambda}_i$ (see Section 3.3), QIEP refers to the quadrature-based class-independent approximation proposed by Seeger et al. (2006), and MCMC refers to Gibbs sampling with the multinomial probit model. The column Posterior complexity of Table 2 describes the overall scaling of the mean and covariance calculations related to the approximate conditional posterior of $\mathbf{f}$. The base computational cost resulting from the full GP prior scales as $O(n^3)$ due to the $n \times n$ matrix inversion (in practice computed using Cholesky decomposition), which is required c times for IEP, QIEP and MCMC, and one additional time for EP, QEP and LA due to incorporation of the between-class correlations. If the same prior covariance structure is used for all classes, VB has the lowest cost, because only one matrix inversion is required per iteration.

The column Likelihood complexity of Table 2 approximates the scaling of the number of calculations that are required besides the posterior mean and covariance evaluations (mainly likelihood related computations for one iteration). For both EP and IEP, this column describes the scaling of the computations needed for the tilted moment approximations done with the inner EP algorithm. For QEP the column summarizes the cost associated with a $(c-1)$-dimensional quadrature rule (denoted by n_{q}^{c-1}) required for the tilted moment evaluations, and for QIEP the cost of one- and two-dimensional quadratures (denoted by n_{q} and n_{q}^2 respectively) required under the independence assumption. For LA the column shows the number of calculations required for evaluating the first and second order derivatives of the softmax likelihood. Each VB iteration requires evaluating the expectations of the auxiliary variables either by a quadrature or sampling, and the cost of one such operation is denoted by n_{q} (for example, the number of quadrature design points). Gibbs sampling with the multinomial probit likelihood requires drawing from the conic truncation of a c-dimensional normal distribution for each observation, and the cost of one draw is denoted by n_{s}. QEP scales inefficiently in c, and is therefore limited to cases with a moderate number of target classes. The QIEP solution can be implemented efficiently because the same function evaluations can be used in all of the $2c-1$ one-dimensional quadratures and the number of two-dimensional quadratures does not depend on c. The cubic scaling in $c-1$ of the tilted moment evaluations in the nested EP and IEP algorithms can be alleviated by reducing the number of inner-loop iterations n_{in} as discussed in Section 3.2.3.

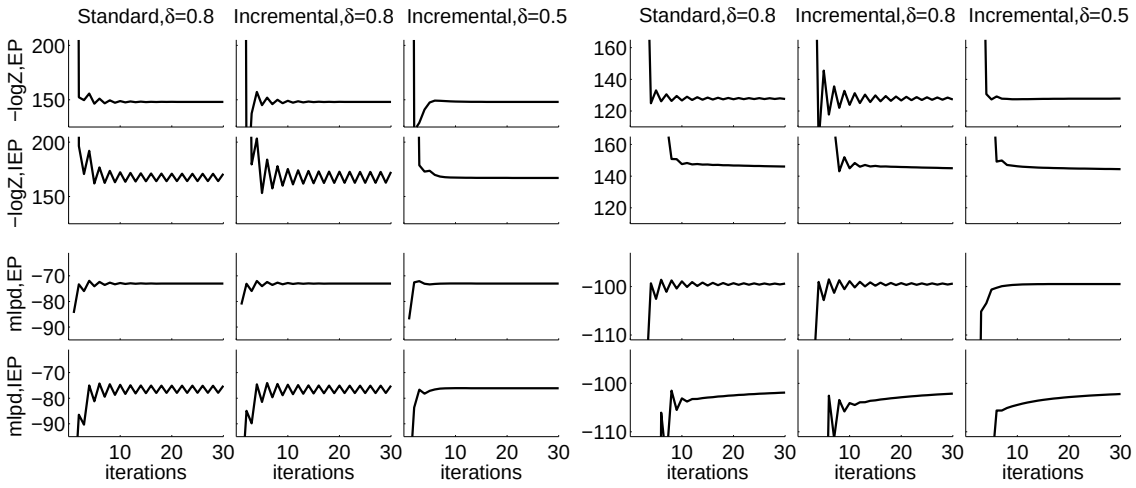
Using the USPS 3 vs. 5 vs. 7 data set, we measured the CPU time required for the posterior inference on $\mathbf{f}$ given nine different preselected hyperparameter values from the grid of Figure 7. With our implementations, LA was the fastest, and EP and VB were about three times more expensive than LA. Because of the efficient scaling (Table 2), VB should be much faster, and probably closer to the running time of LA. One reason for the slow performance may be our implementation based

Algorithm	Posterior complexity	Likelihood complexity
EP	$(c+1)n^3$	$nn_{\text{in}}(c-1)^3$
QEP	$(c+1)n^3$	nn_{q}^{c-1}
IEP	cn^3	$nn_{\text{in}}(c-1)^3$
QIEP	cn^3	$n((2c-1)n_{\text{q}}+2n_{\text{q}}^2)$
VB	n^3	$n(c-1)2n_{\text{q}}$
LA	$(c+1)n^3$	nc
MCMC (Gibbs sampling)	cn^3	ncn_{s}

Table 2: Approximate computational complexities of the various methods as a function of n and c for one iteration with fixed hyperparameters. The column Posterior complexity summarizes the scaling of the mean and covariance calculations related to the approximate conditional posterior of $\mathbf{f}$. The column Likelihood complexity approximates the scaling of the number of calculations required for additional likelihood related computations. The parameter n_{in} refers to the number of inner EP iterations in nested EP, n_{q}^c to the cost of a c-dimensional numerical quadrature, and n_{s} to the cost of sampling from a conic truncation of a c-variate Gaussian distribution.

on importance sampling steps, which may result in slower convergence due to fluctuations. The MCMC and LA-TKP approaches were overall very slow compared to LA. One iteration of MCMC is relatively cheap, but in our experiments thousands of posterior samples were required to obtain chains of sufficiently uncorrelated samples which is why MCMC was over hundred times slower than LA. LA-TKP requires roughly $c+1$ times the CPU time of LA for computing the predictions for each test input. Therefore, the computational cost of LA-TKP becomes quickly prohibitive as the number of test points increases.

In the CPU time comparisons across the range of hyperparameter values producing variety of skewed and non-isotropic posterior distributions, fully-coupled nested EP converged in fewer outer-loop iterations than nested IEP if the same convergence criteria were used. Figure 8 illustrates the difference in convergence with the USPS 3 vs. 5 vs. 7 and Glass data sets. We fixed the hyperparameters at $\log(\sigma^2) = 8$ and $\log(l) = 2.5$ which results in good predictive performances on the independent test data set with both methods for the USPS 3 vs. 5 vs. 7 data (see Figure 7). For both methods, the negative log marginal likelihood approximation $-\log Z_{\text{EP}}$ and the mean log predictive density (mlpd) in the test data set are shown after each iteration. Note that the converged EP approximation satisfying the moment matching conditions between $\hat{p}(\mathbf{f}_i)$ and $q(\mathbf{f}_i)$ corresponds to stationary points of an objective function similar to $-\log Z_{\text{EP}}$ (Minka, 2001b; Opper and Winther, 2005). The convergence is illustrated with a small amount of damping (damping factor $\delta = 0.8$) and with a larger amount of damping ($\delta = 0.5$). With the fully-coupled nested EP algorithm the damping is applied on the inner-EP site parameters $\tilde{\boldsymbol{\alpha}}_i$ and $\tilde{\boldsymbol{\beta}}_i$, whereas with IEP the damping is applied on the natural exponential site parameters $\tilde{\boldsymbol{\nu}}_i$ and $\tilde{\Sigma}_i^{-1}$. In the columns denoted Standard in Figure 8, the inner-loops of the nested EP and IEP algorithms are run until convergence at each outer-loop iteration, whereas in the rest of the columns (Incremental) only one inner-loop iteration per site is done at each outer-loop iteration. Recall from the previous discussion and from Table 2 that the



(a) USPS 3 vs. 5 vs. 7 (b) Glass

Figure 8: A convergence comparison between EP and IEP using parallel updates in the outer EP loop with the USPS 3 vs. 5 vs. 7 data (columns 1-3), and with the Glass data (columns 4-6). The first two rows show the negative log marginal likelihood estimates $-\log Z_{\mathrm{EP}}$ as a function of iterations for two different damping factors δ, and the bottom two rows show the corresponding mean log predictive density (mlpd) evaluated using a separate test data set. In the columns denoted Standard the inner-loops of the nested EP and IEP algorithms are run until convergence at each outer-loop iteration, whereas in the rest of the columns (Incremental) only one inner-loop iteration per site is done at each outer-loop iteration. The hyperparameters of the squared exponential covariance function were fixed at $\log(\sigma^2) = 8$ and $\log(l) = 2.5$ which results in a non-Gaussian posterior distribution.

incremental updates ($n_{\mathrm{in}} = 1$) reduce the computational burden of the inner-loop of the nested EP algorithm which scales as $O(n_{\mathrm{in}}(c-1)^3)$.

From Figure 8 it can be seen that the incremental updates require more damping than standard updates, but both update schemes seem to converge into the same solution. There is a clear difference in the amplitude of oscillations between the nested EP and IEP algorithms with the same damping level but this may be partly caused by the different parameterization. Compared to fully-coupled EP, there is a slow drift in $-\log Z_{\mathrm{EP}}$ and in the mlpd score even after 20 iterations with IEP, and the drift is more visible with the Glass data. One explanation for this behavior can be that the fully-coupled Gaussian distribution is more suitable approximating family for the true posterior (Minka, 2005), which is strongly non-Gaussian because of the large magnitude hyperparameter value, and has stronger between-class posterior dependencies induced through the likelihood terms because of the relatively large lengthscale.

5.5 Predictive Performance Across Data Sets with Hyperparameter Estimation

In this section we assess the predictive performances with estimation of the hyperparameters. We compare the performances of nested EP and IEP, VB, LA, LA-TKP, and Gibbs sampling with the multinomial probit model (MCMC) on various benchmark data sets. All the methods are compared

Data Set	n_{train}	n_{test}	Classes (c)	Covariates (d)	ARD
New-thyroid	215	215 (Ten-fold CV)	3	5	yes
Teaching	151	151 (Ten-fold CV)	3	5	yes
Glass	214	214 (Ten-fold CV)	6	9	yes
Wine	178	178 (Ten-fold CV)	3	13	yes
Image segmentation	210	2100	7	18	no
USPS 3 vs. 5 vs. 7	1157	1175	3	256	no
USPS 10-class	4649	4649	10	256	no

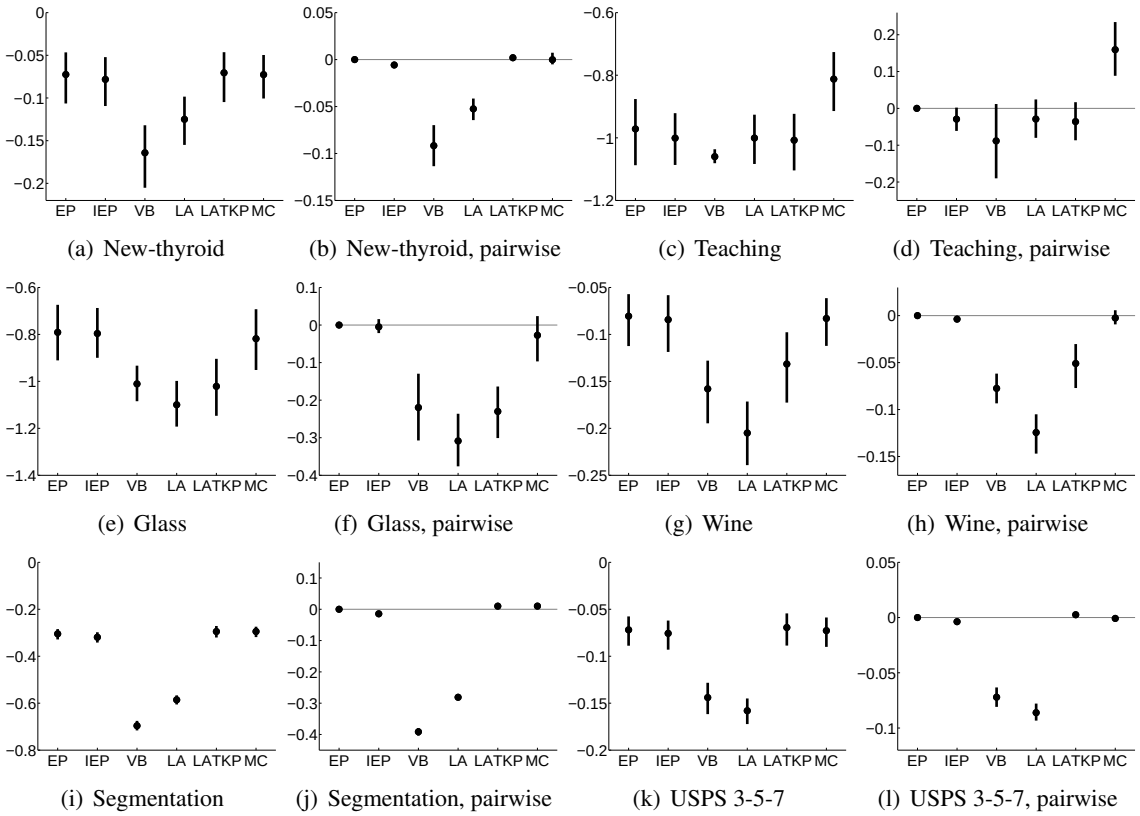
Table 3: Data sets used in the experiments.

using the USPS 3 vs. 5 vs. 7 data, and the following five UCI Machine Learning Repository (Frank and Asuncion, 2010) data sets: New-thyroid, Teaching, Glass, Wine, and Image segmentation. The comparisons are also done using the USPS 10-class data set, but only for EP, IEP, VB, and LA due to the large n. The main characteristics of the data sets are summarized in Table 3.

For all the data sets, we standardize the covariates to zero mean and unit variance, and use the squared exponential covariance function with the same hyperparameters for all classes. For the Image segmentation and USPS data sets we use a common lengthscale parameter for all dimensions. For other data sets we set individual lengthscale parameters for each input dimension (Automatic Relevance Determination, ARD, see, for example, Rasmussen and Williams, 2006). To avoid unnecessarily large hyperparameter values, we place a weakly informative prior on the lengthscale and magnitude parameters by choosing a half Student-t distribution with four degrees of freedom and variance equal to one hundred. With MCMC we sample the hyperparameters, and with the other methods, we use gradient-based type-II MAP estimation to select the hyperparameter values. The predictive performance is measured using a ten-fold cross-validation (CV) with four of the data sets, and using predetermined training and test sets with three of the data sets (see Table 3).

The first and third column in Figure 9 visualize the mean log predictive densities and their approximate 95% central credible intervals for six data sets estimated using the Bayesian bootstrap method as described by Vehtari and Lampinen (2002). To highlight the differences between the methods more clearly, we compute the pairwise differences of the log posterior predictive densities with respect to EP. The second and fourth column in Figure 9 show the mean values and the approximate 95% central credible intervals of the pairwise differences. The comparisons reveal that EP performs well when compared to MCMC; only in the Teaching and Image segmentation data sets MCMC is significantly better. IEP performs worse than EP in all the data sets except Teaching and Glass. The predictive densities of VB and LA are overall worse than EP, IEP or MCMC. LA-TKP improves the performance of LA with all the data sets except Teaching.

The first and third column in Figure 10 visualize the mean classification accuracies and their approximate 95% central credible intervals. The second and fourth column in Figure 10 show the pairwise mean differences of the classification accuracies together with the approximate 95% central credible intervals with respect to EP. Because the difference of the classification outcomes for each observation is a discrete variable with three possible values (worse, same, or better than EP), we use a multinomial model with a non-informative Dirichlet prior distribution for the comparison test. In a case where the method has exactly the same predictions as EP, a small circle is plotted at the mean value. The differences between all the methods are small. In the Teaching data set, where the overall



(a) New-thyroid (b) New-thyroid, pairwise (c) Teaching (d) Teaching, pairwise

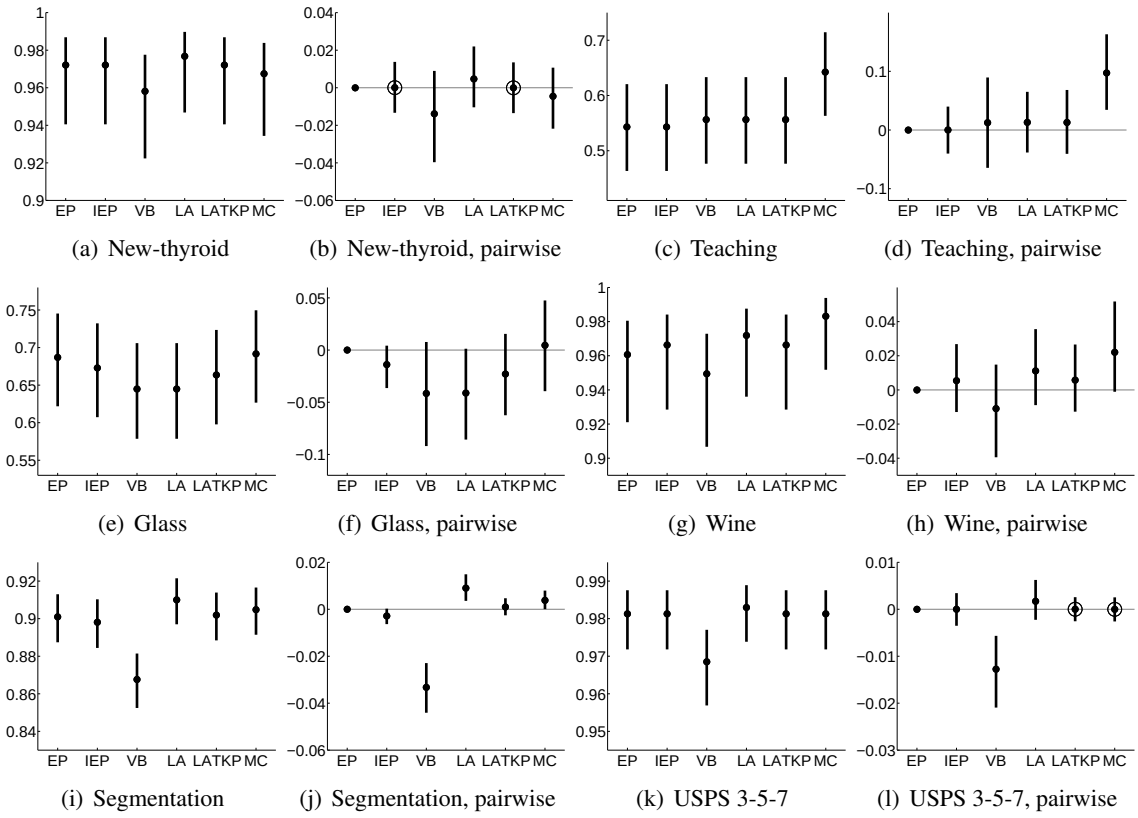
(e) Glass (f) Glass, pairwise (g) Wine (h) Wine, pairwise

(i) Segmentation (j) Segmentation, pairwise (k) USPS 3-5-7 (l) USPS 3-5-7, pairwise

Figure 9: The first and third column: The mean log predictive densities and their approximate 95% credible intervals for six data sets (see Table 3) using EP, IEP, VB, LA, LA-TKP, and MCMC with Gibbs sampling. The second and fourth column: The pairwise differences of the log predictive densities with respect to EP (mean + approximate 95% credible intervals). Values above zero indicate that a method is performing better than EP.

accuracy is the lowest, the MCMC estimate is significantly better than any other method. There is no statistically significant difference between EP and IEP; IEP performs slightly better in the Wine data set, but EP has a better accuracy in the Glass and Image segmentation data sets, which both have more than three target classes, and in which the overall classification accuracies are among the lowest. LA has a good classification accuracy, and performs better than EP in Image segmentation. A possible explanation for this is the different shape of the softmax likelihood function used by LA. If the classification accuracy is the only criterion, the LA-TKP correction seems unnecessary. VB has the lowest classification performance and is significantly worse than the other methods in the Image segmentation and USPS 3 vs. 5 vs. 7 data sets, which is probably caused by a worse estimate of the hyperparameter values.

Finally, we summarize the mlpd scores and classification accuracies of EP, IEP, VB, and LA with the USPS 10-class data set in Figure 11. Both EP approaches are significantly better than VB or LA with both measures. Considering the EP approaches, fully-coupled EP achieves a slightly better



(a) New-thyroid (b) New-thyroid, pairwise (c) Teaching (d) Teaching, pairwise

(e) Glass (f) Glass, pairwise (g) Wine (h) Wine, pairwise

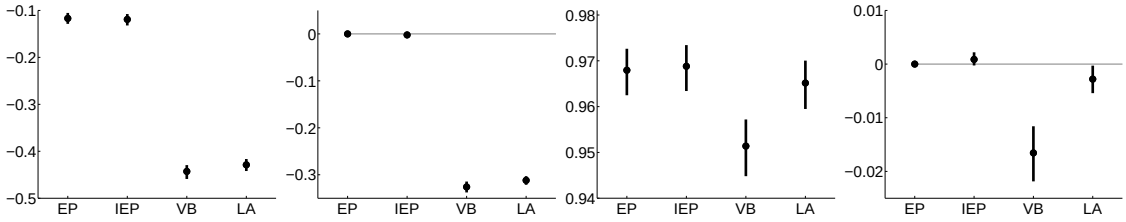
(i) Segmentation (j) Segmentation, pairwise (k) USPS 3-5-7 (l) USPS 3-5-7, pairwise

Figure 10: The first and third column: The classification accuracies and their approximate 95% credible intervals for six data sets (see Table 3) using EP, IEP, VB, LA, LA-TKP, and MCMC with Gibbs sampling. The second and fourth column: The pairwise differences of the classification accuracies with respect to EP (mean + approximate 95% credible intervals). Values above zero indicate that a method is performing better than EP. A small circle is plotted at the mean value if the predictions are exactly the same as with EP.

mlpd score, whereas IEP is slightly better in terms of classification accuracy, but the differences are not statistically significant.

6. Conclusions and Further Research

EP approaches for GP classification with the multinomial probit model have already been proposed by Seeger et al. (2006) and Girolami and Zhong (2007). In this paper, we have complemented their work with a novel quadrature-free nested EP algorithm that maintains all between-class posterior dependencies but still scales linearly in the number of classes. Our comparisons with fixed hyperparameters show that compared to quadrature-based EP algorithms, nested EP achieves similar accuracy, and its computational cost is comparable with a class-independent approximation whereas with full posterior couplings nested EP scales more efficiently. When the hyperparame-



(a) USPS (b) USPS, pairwise (c) USPS (d) USPS, pairwise

Figure 11: The mean log predictive densities (a) and classification accuracies (c) for the USPS 10-class data set (see Table 3) using EP, IEP, VB, and LA. The pairwise differences of the log predictive densities and the classification accuracies with respect to EP are shown in plots (b) and (d), respectively. In plots (b) and (d) values above zero indicate that a method is performing better than EP. In each plot, the mean values and approximate 95% central credible intervals are shown.

ters are determined by optimizing the marginal likelihood, nested EP is a consistent approximate method compared to full MCMC. In terms of predictive density, nested EP is close to MCMC, and more accurate compared to VB and LA, but if only the classification accuracy is concerned, all the approximations perform similarly. LA-TKP improves the predictive density estimates of LA but the computational cost becomes increasingly demanding if a larger number of predictions are needed.

In our comparisons the predictive accuracies of the full EP and IEP solutions obtained using the nested EP algorithm are similar for practical purposes. However, our visualizations show that the approximate marginal posterior distributions of the latent values provided by full EP are clearly more accurate, although the full nested EP solution can be calculated with similar computational burden as nested IEP. Because there is no convergence guarantee for the standard EP algorithm, it is worth to notice the differences in the convergence properties of full EP and IEP observed in our experiments. With the same hyperparameter values, nested IEP converged more slowly and required more damping than full nested EP. This can be due to slower propagation of information caused by the independence assumptions, and this behavior can get worse as the between-class posterior couplings get stronger with certain hyperparameter values. Given all these observations, we prefer full EP to IEP.

Models in which each likelihood term related to a certain observation depends on multiple latent values, such as the multinomial probit, are challenging for EP because a straightforward quadrature-based implementation may become computationally infeasible unless independence assumptions between the latent values or some other simplifications are made. In the presented nested EP approach, we have applied inner EP approximations for each likelihood term within an outer EP framework in a computationally efficient manner. This approach could be applicable also for other similar multi-latent models which involve integral representations consisting of simple factorized functions each depending on linear transformations of the latent variables. For example, one straightforward extension would be linear multinomial probit regression with Gaussian priors on the weights.

A drawback with GP classifiers is the fundamental computational scaling $O(n^3)$ resulting from the prior structure. To speed up the inference in multiclass GP classification, sparse approximations

such as the informative vector machine (IVM) have been proposed (Seeger and Jordan, 2004; Girolami and Rogers, 2006; Seeger et al., 2006). IVM uses the information provided by all observations to form an active subset which is then used to form the posterior mean and covariance approximations. The presented EP approach could be extended to IVM in a similar fashion as described by Seeger and Jordan (2004). The accurate marginal approximations of full EP could be useful in determining the relative entropy measures used as a scoring criterion to select the active set. To speed up the computations, the inner EP site parameters could be updated iteratively even for the observations not in the active set in a similar fashion as described in Section 3.2. Recently, a similar approach to IVM called predictive active set selection (PASS-GP) has been proposed by Henao and Winther (2010) to lower the computational complexity in binary GP classification. PASS-GP uses the approximate cavity and cavity predictive distributions of EP to determine a representative active set. The proposed EP approach could prove useful when extending PASS-GP to multiple classes, because it provides accurate marginal predictive density estimates.

The presented fully-coupled nested EP approach for approximate inference with Gaussian process models is implemented in the free GPstuff software package and the code will be made available at `http://becs.aalto.fi/en/research/bayes/gpstuff/`.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments to improve the manuscript. The research has been funded by the Academy of Finland (grant 218248).

Appendix A. Approximating Tilted Moments Using EP

For convenience, we summarize the inner EP algorithm for approximating the tilted moments resulting from a multinomial probit likelihood. Essentially the same algorithm was presented by Minka (2001a) for classification with the Bayes point machine and later by Qi et al. (2004) for the binary probit classifier. To facilitate a computationally efficient implementation, the following algorithm description is written with an emphasis to reduce the number of vector and matrix operations in a similar fashion as in the general EP formulation presented by Cseke and Heskes (2011, Appendix C).

We want to approximate the normalization, mean and covariance of the tilted distribution

$$\hat{p}(\mathbf{w}_i) = \hat{Z}_i^{-1} \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathrm{w}_i}, \Sigma_{\mathrm{w}_i}) \prod_{j=1, j\neq y_i}^{c} \Phi(\mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j}).$$

This is done using the EP algorithm which results in the following Gaussian approximation

$$\hat{q}(\mathbf{w}_i) = Z_{\hat{q}_i}^{-1} \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\mathrm{w}_i}, \Sigma_{\mathrm{w}_i}) \prod_{j=1, j\neq y_i}^{c} \tilde{Z}_j \mathcal{N}(\mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j} | \tilde{\beta}_{i,j} \tilde{\alpha}_{i,j}^{-1}, \tilde{\alpha}_{i,j}^{-1}) = \mathcal{N}(\mathbf{w}_i | \boldsymbol{\mu}_{\hat{q}_i}, \Sigma_{\hat{q}_i}),$$

where we have used the natural parameters $\tilde{\alpha}_{i,j}$ (precision) and $\tilde{\beta}_{i,j}$ (location) for the site approximations. The index i denotes the i'th observation, and to clarify the notation below, we leave out this index from the inner EP terms. In the first outer-loop, the site parameters $\tilde{\alpha}$ and $\tilde{\beta}$ are initialized to zero, $\boldsymbol{\mu}_{\hat{q}_i}$ to $\boldsymbol{\mu}_{\mathrm{w}_i}$, and $\Sigma_{\hat{q}_i}$ to Σ_{w_i}. After the first outer-loop, these parameters are initialized to their

last values from the previous outer-loop iteration for speed-up. The following steps are repeated for all $j = \{1,\ldots,c | j \neq y_i\}$ until convergence.

1. Cavity evaluations:

$$\begin{aligned} v_{-j} &= (v_j^{-1} - \tilde{\alpha}_j)^{-1}, \\ m_{-j} &= v_{-j}(v_j^{-1} m_j - \tilde{\beta}_j), \end{aligned}$$

   where scalars $v_j = \tilde{\mathbf{b}}_{i,j}^T \Sigma_{\hat{q}_i} \tilde{\mathbf{b}}_{i,j}$ and $m_j = \tilde{\mathbf{b}}_{i,j}^T \boldsymbol{\mu}_{\hat{q}_i}$ correspond to the marginal distribution of latent $g_i^j = \mathbf{w}_i^T \tilde{\mathbf{b}}_{i,j}$.

2. Tilted moments for g_i^j:

$$\begin{aligned} \hat{Z}_j &= \Phi(z_j), \\ \hat{m}_j &= \rho_j v_{-j} + m_{-j}, \\ \hat{v}_j &= v_{-j} - v_{-j}^2 \gamma_j, \end{aligned}$$

   where $z_j = m_{-j}(1+v_{-j})^{-1/2}$, $\rho_j = \frac{\mathcal{N}(z_j|0,1)}{\Phi(z_j)}(1+v_{-j})^{-1/2}$ and $\gamma_j = \rho_j^2 + z_j \rho_j (1+v_{-j})^{-1/2}$.

3. Site updates with damping:

$$\begin{aligned} \Delta\tilde{\alpha}_j &= \delta(\hat{v}_j^{-1} - v_j^{-1}), \\ \Delta\tilde{\beta}_j &= \delta(\hat{v}_j^{-1}\hat{m}_j - v_j^{-1} m_j), \end{aligned}$$

   where $\delta \in (0,1]$ is the damping factor.

4. Rank-1 covariance update:

$$\begin{aligned} \Sigma_{\hat{q}_i}^{\text{new}} &= \Sigma_{\hat{q}_i} - \boldsymbol{\phi}_j (1 + \Delta\tilde{\alpha}_j v_j)^{-1} \Delta\tilde{\alpha}_j \boldsymbol{\phi}_j^T, \\ \boldsymbol{\mu}_{\hat{q}_i}^{\text{new}} &= \boldsymbol{\mu}_{\hat{q}_i} + \boldsymbol{\phi}_j (1 + \Delta\tilde{\alpha}_j v_j)^{-1} (\Delta\tilde{\beta}_j - \Delta\tilde{\alpha}_j m_j), \end{aligned}$$

   where $\boldsymbol{\phi}_j = \Sigma_{\hat{q}_i} \tilde{\mathbf{b}}_{i,j}$.

Alternatively, the rank-1 updates of step 4 could be replaced by only one parallel covariance update after each sweep over the sites indexed by j.

After convergence, we evaluate the normalization $Z_{\hat{q}_i}$ of the tilted distribution as

$$\begin{aligned} \log Z_{\hat{q}_i} &= \frac{1}{2}\boldsymbol{\mu}_{\hat{q}_i}^T \Sigma_{\hat{q}_i}^{-1} \boldsymbol{\mu}_{\hat{q}_i} + \frac{1}{2}\log|\Sigma_{\hat{q}_i}| - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{w}_i}^T \Sigma_{\mathrm{w}_i}^{-1} \boldsymbol{\mu}_{\mathrm{w}_i} - \frac{1}{2}\log|\Sigma_{\mathrm{w}_i}| + \sum_{j=1, j\neq y_i}^{c} \log \hat{Z}_j \\ &\quad + \frac{1}{2}\sum_{j=1, j\neq y_i}^{c} \left(m_{-j}^2 v_{-j}^{-1} + \log v_{-j} \right) - \frac{1}{2}\sum_{j=1, j\neq y_i}^{c} \left(m_j^2 v_j^{-1} + \log v_j \right). \end{aligned}$$

Appendix B. Details of Posterior Computations

The site covariance can be written as $\tilde{T} = D - DR(R^T DR)^{-1}R^T D$, where D is a $cn \times cn$ diagonal matrix $D = \text{diag}\left[\pi_1^1, \ldots, \pi_n^1, \pi_1^2, \ldots, \pi_n^2, \ldots, \pi_1^c, \ldots, \pi_n^c\right]^T$, and R is a $cn \times n$ matrix consisting of identity matrices I_n stacked c times vertically. To compute predictions related to a test point $\mathbf{x}_*$, we need to first evaluate the mean and covariance of $\mathbf{f}_* = \left[f_*^1, f_*^2, \ldots, f_*^c\right]^T$ as

$$\begin{aligned} \mathrm{E}[\mathbf{f}_*] &= K_*\tilde{\boldsymbol{\nu}} - K_* M K\tilde{\boldsymbol{\nu}}, \quad (26) \\ \mathrm{Cov}[\mathbf{f}_*] &= K_{*,*} - K_* M K_*^T, \quad (27) \end{aligned}$$

where $\tilde{\boldsymbol{\nu}}$ contains all $\tilde{\boldsymbol{\nu}}_i$ in the same order with $\mathbf{f}$, $M = \tilde{T}(I_{cn} + K\tilde{T})^{-1}$, K_* is a $c \times cn$ covariance matrix between the test point and the training points, and $K_{*,*}$ is a $c \times c$ covariance matrix for the test point. The matrix M in Equations (26) and (27) can be evaluated using

$$M = B - BRP^{-1}R^T B,$$

where $B = D^{1/2}A^{-1}D^{1/2}$, $P = R^T BR$, and $A = I_{cn} + D^{1/2}KD^{1/2}$. To evaluate expressions involving M, we compute the Cholesky decompositions of P and the c diagonal blocks of A, which results in the scaling $O((c+1)n^3)$. The predictive mean and covariance can be computed using the block-diagonal structure of B and the sparse structure of K_*. Given $\mathrm{E}[\mathbf{f}_*]$ and $\mathrm{Cov}[\mathbf{f}_*]$, the integration over the posterior uncertainty of $\mathbf{f}_*$ required to compute the predictive class probabilities, is equivalent to the tilted moment evaluation, and can be approximated using the algorithm described in Appendix A.

To compute the marginal mean $\boldsymbol{\mu}_i$ (a vector of length c) and covariance Σ_i (a matrix of size $c \times c$) of the training latent $\mathbf{f}_i$ for all i during the posterior update step in the outer-loop iteration, we replace K_* and $K_{*,*}$ with K in Equations (26) and (27), and compute only the required blocks of the full posterior covariance matrix. After convergence of the outer EP algorithm, the marginal likelihood approximation of EP can be computed as

$$\begin{aligned} \log Z_{\mathrm{EP}} &= \frac{1}{2}\tilde{\boldsymbol{\nu}}^T\boldsymbol{\mu} - \frac{1}{2}\log|I_{cn} + K\tilde{T}| + \sum_{i=1}^{n}\log Z_{\hat{q}_i} + \frac{1}{2}\sum_{i=1}^{n}\left(\boldsymbol{\mu}_{-i}^T\Sigma_{-i}^{-1}\boldsymbol{\mu}_{-i} + \log|\Sigma_{-i}|\right) \\ &\quad - \frac{1}{2}\sum_{i=1}^{n}\left(\boldsymbol{\mu}_i^T\Sigma_i^{-1}\boldsymbol{\mu}_i + \log|\Sigma_i|\right), \quad (28) \end{aligned}$$

where $\boldsymbol{\mu}_{-i}$ and Σ_{-i} are the i'th cavity mean and covariance, and the posterior mean $\boldsymbol{\mu}$ contains all $\boldsymbol{\mu}_i$. The normalization terms $Z_{\hat{q}_i}$ are obtained from the inner EP algorithm described in Appendix A. Finally, the determinant term in (28) can be evaluated as

$$|I_{cn} + K\tilde{T}| = |A||R^T DR|^{-1}|P|.$$

The gradients of the log marginal likelihood with respect to θ can be obtained by calculating only the explicit derivatives of the first two terms of (28). The implicit derivatives with respect to the site parameters and cavity parameters (in their natural exponential forms) of the outer EP cancel each other out in the convergence (Opper and Winther, 2005; Seeger, 2005). Since the likelihood does not depend on any hyperparameters, the explicit derivatives of $\log Z_{\hat{q}_i}$ are zero. Also the implicit derivatives of $\log Z_{\hat{q}_i}$ with respect to the inner EP parameters cancel out because these terms are formed as marginal likelihood approximations with the inner EP, which also satisfies the same cancellation property of the EP algorithm.

References

Kian Ming A. Chai. Variational multinomial logit Gaussian process. *Journal of Machine Learning Research*, 13:1745–1808, 2012.

Botond Cseke and Tom Heskes. Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research*, 12:417–454, 2011.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Andrew Frank and Arthur Asuncion. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2010. URL `http://archive.ics.uci.edu/ml`.

Mark Girolami and Simon Rogers. Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, 18:1790–1817, 2006.

Mark Girolami and Mingjun Zhong. Data integration for classification problems employing Gaussian process priors. In *Advances in Neural Information Processing Systems 19*, pages 465–472. The MIT Press, 2007.

Ricardo Henao and Ole Winther. PASS-GP: Predictive active set selection for Gaussian processes. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 148–153, 2010.

Daniel Hernández-Lobato, José M. Hernández-Lobato, and Pierre Dupont. Robust multi-class Gaussian process classification. In *Advances in Neural Information Processing Systems 24*, pages 280–288, 2011.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian Gaussian process classification with the EM-EP algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1948–1959, 2006.

Malte Kuss and Carl E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

Thomas P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001a.

Thomas P. Minka. Expectation Propagation for approximative Bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann, San Francisco, CA, 2001b.

Thomas P. Minka. Divergence measures and message passing. Technical report, Microsoft Research, Cambridge, 2005.

Thomas P. Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359. Morgan Kaufmann, San Francisco, CA, 2002.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

Radford M. Neal. Regression and classification using Gaussian process priors (with discussion). In *Bayesian Statistics 6*, pages 475–501. Oxford University Press, 1998.

Hannes Nickisch and Carl E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

Manfred Opper and Ole Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005.

Yuan Qi, Thomas P. Minka, Rosalind W. Picard, and Zoubin Ghahramani. Predictive automatic relevance determination by expectation propagation. In *Proceedings of the 21st International Conference on Machine Learning*, pages 671–678, 2004.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society (Series B)*, 71(2):319–392, 2009.

Matthias Seeger. Expectation propagation for exponential families. Technical report, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2005.

Matthias Seeger and Michael Jordan. Sparse Gaussian process classification with multiple classes. Technical report, University of California, Berkeley, CA, 2004.

Matthias Seeger, Neil Lawrence, and Ralf Herbrich. Efficient nonparametric Bayesian modelling with sparse Gaussian process approximations. Technical report, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2006.

Alexander Smola, Vishy Vishwanathan, and Eleazar Eskin. Laplace propagation. In *Advances in Neural Information Processing Systems 16*. The MIT Press, 2004.

Edward Snelson and Zoubin Ghahramani. Compact approximations to Bayesian predictive distributions. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 840–847, 2005.

Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

Marcel van Gerven, Botond Cseke, Robert Oostenveld, and Tom Heskes. Bayesian source localization with the multivariate Laplace prior. In *Advances in Neural Information Processing Systems 22*, pages 1901–1909, 2009.

Aki Vehtari and Jouko Lampinen. Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14(10):2439–2468, 2002.

Christopher K. I. Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

Journal of Machine Learning Research 14 (2013) 111-152 Submitted 10/11; Revised 8/12; Published 1/13

Pairwise Likelihood Ratios for Estimation of Non-Gaussian Structural Equation Models

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Dept of Computer Science and HIIT
Dept of Mathematics and Statistics
University of Helsinki
Helsinki, Finland

Stephen M. Smith STEVE@FMRIB.OX.AC.UK
FMRIB (Oxford University Centre for Functional MRI of the Brain)
Nuffield Dept of Clinical Neurosciences
University of Oxford
Oxford, UK

Editor: Peter Spirtes

Abstract

We present new measures of the causal direction, or direction of effect, between two non-Gaussian random variables. They are based on the likelihood ratio under the linear non-Gaussian acyclic model (LiNGAM). We also develop simple first-order approximations of the likelihood ratio and analyze them based on related cumulant-based measures, which can be shown to find the correct causal directions. We show how to apply these measures to estimate LiNGAM for more than two variables, and even in the case of more variables than observations. We further extend the method to cyclic and nonlinear models. The proposed framework is statistically at least as good as existing ones in the cases of few data points or noisy data, and it is computationally and conceptually very simple. Results on simulated fMRI data indicate that the method may be useful in neuroimaging where the number of time points is typically quite small.

Keywords: structural equation model, Bayesian network, non-Gaussianity, causality, independent component analysis

1. Introduction

Estimating structural equation models (SEMs), or linear Bayesian networks is a challenging problem with many applications in bioinformatics, neuroinformatics, and econometrics. If the data is Gaussian, the problem is fundamentally ill-posed. Recently, it has been shown that using the non-Gaussianity of the data, such models can be identifiable (Shimizu et al., 2006). This led to the Linear Non-Gaussian Acyclic Model, or LiNGAM.

The original method for estimating LiNGAM was based on first applying independent component analysis (ICA) to the data and then deducing the network connections from the results of ICA. However, we believe that it may be possible to develop better methods for estimating LiNGAM directly, without resorting to ICA algorithms.

A framework called DirectLiNGAM was, in fact, proposed by Shimizu et al. (2011) to provide an alternative to the ICA-based estimation. DirectLiNGAM was shown to give promising results especially in the case where the number of observed data points is small compared to the dimension of the data. It can also have algorithmic advantages because it does not need gradient-based iterative methods. An essential ingredient in DirectLiNGAM is a measure of the causal direction between two variables.

An alternative approach to estimating SEMs is to first estimate which variables have connections and then estimate the direction of the connection. While a rigorous justification for such an approach may be missing, this is intuitively appealing especially in the case where the amount of data is limited. Determining the directions of the connections can be performed by considering each connection separately, which requires, again, analysis of the causal direction between two variables. Such an approach was found to work best by Smith et al. (2011) which considered causal analysis of simulated functional magnetic resonance imaging (fMRI) data, where the number of time points is typically small. A closely related approach was proposed by Hoyer et al. (2008), in which the PC algorithm was used to estimate the existence of connections, followed by a scoring of directions by an approximate likelihood of the LiNGAM model; see also Ramsey et al. (2011).

Thus, we see that measuring pairwise causal directions is a central problem in the theory of LiNGAM and related models. In fact, analyzing the causal direction between two non-Gaussian random variables (with no time structure) is an important problem in its own right, and was considered in the literature before the advent of LiNGAM (Dodge and Rousson, 2001).

In this paper, we develop new measures of causal direction between two non-Gaussian random variables, and apply them to the estimation of LiNGAM. The approach uses the ratio of the likelihoods of the models corresponding to the two directions of causal influence. A likelihood ratio is likely to provide a statistically powerful method because of the general optimality properties of likelihood. We further propose first-order approximations of the likelihood ratio which are easy to compute and have simple intuitive interpretations. They are also closely related to higher-order cumulants and may be more resistant to noise. The framework is also simple to extend to cyclic or nonlinear models.

The paper is structured as follows. The measures of causal directions are derived in Section 2. In Section 3 we show how to apply them to estimating the model with more than two variables. The extension to cyclic models is proposed in Section 4 and an extension to a nonlinear model in Section 5. We report simulations with comparisons to other methods in Section 6, experiments on simulated brain imaging data in Section 7, and results on a publicly available benchmark data set in Section 8. Section 9 concludes the paper. Preliminary results were published by Hyvärinen (2010).

2. Finding Causal Direction Between Two Variables

In this section, we present our main contribution: new measures of causal direction between two random variables.

This section is structured as follows: We first define the problem in Section 2.1. We derive the likelihood ratio in Section 2.2. We propose a general-purpose approximation for the likelihood ratio in Section 2.3. The connection to mutual information is explained in Section 2.4. We derive a particularly simple approximation for the likelihood ratio in Section 2.5, and propose an instance for the case of sparse, symmetric densities. A theoretical analysis of the approximation based on cumulants is given in Section 2.6. We give intuitive interpretations of the approximations in Sec-

tion 2.7, and discuss their noise-tolerance in Section 2.8. Finally, we show how to use the likelihood ratio approximations in the case of skewed variables in Section 2.9.

For the benefit of the reader, we have further created Table 3 in the Conclusion on page 150 that lists the main new measures proposed in this paper.

2.1 Problem Definition

Denote the two observed random variables by x and y. Assume they are non-Gaussian, as well as standardized to zero mean and unit variance. Our goal is to distinguish between two causal models. The first one we denote by $x \to y$ and define as

$$y = \rho x + d$$

where the disturbance d is independent of x, and the regression coefficient is denoted by ρ. The second model is denoted by $y \to x$ and defined as

$$x = \rho y + e$$

where the disturbance e is independent of y. The parameter ρ is the same in the two models because it is equal to the correlation coefficient. Note that these models belong to the LiNGAM family (Shimizu et al., 2006) with two variables. In the following, we assume that x, y follow one of these two models.

Note that in contrast to Dodge and Rousson (2001) or Dodge and Yadegari (2010), we do not assume that d or e are normal, or have zero cumulants. We make no assumptions on their distributions. It is not even necessary to assume that they are non-Gaussian; it is enough that x and y are non-Gaussian. (This is related to the identifiability theorem in ICA which says that one of the latent variables can be non-Gaussian, see Comon, 1994).

2.2 Likelihood Ratio

An attractive way of deciding between the two models is to compute their likelihoods and their ratio. Consider a sample $(x_1, y_1), \ldots, (x_T, y_T)$ of data. The likelihood of the LiNGAM in which $x \to y$ was given by Hyvärinen et al. (2010) as

$$\log L(x \to y) = \left[\sum_t G_x(x_t) + G_d\left(\frac{y_t - \rho x_t}{\sqrt{1-\rho^2}}\right) \right] - T \log(1-\rho^2)$$

where $G_x(u) = \log p_x(u)$, and G_d is the standardized log-pdf of the residual when regressing y on x. The last term here is a normalization term due to the use of standardized log-pdf G_d. From this we can compute the likelihood ratio, which we normalize by $\frac{1}{T}$ for convenience:

$$\begin{aligned} R = \frac{1}{T}\log L(x \to y) - \frac{1}{T}\log L(y \to x) \\ = \frac{1}{T}\sum_t G_x(x_t) + G_d\left(\frac{y_t - \rho x_t}{\sqrt{1-\rho^2}}\right) - G_y(y_t) - G_e\left(\frac{x_t - \rho y_t}{\sqrt{1-\rho^2}}\right). \quad (1) \end{aligned}$$

We can thus compute R and decide based on it what the causal direction is. If R is positive, we conclude $x \to y$, and if it is negative, we conclude $y \to x$.

To use (1) in practice, we need to choose the G's and estimate ρ. The statistically optimal way of estimating ρ would be to maximize the likelihood, but in practice it may be better to estimate it simply by the conventional least-squares solution to the linear regression problem. Nevertheless, maximization of likelihood might be more robust against outliers, because log-likelihood functions often grow more slowly than the squaring function when moving away from the origin.

Choosing the four log-pdf's G_x, G_y, G_d, G_e could, in principle, be done by modelling the relevant log-pdf's by parametric (Karvanen and Koivunen, 2002) or non-parametric (Pham and Garrat, 1997) methods, which will be discussed in more detail below. However, for small sample sizes such modelling can be very difficult. In the following, we provide various parametric approximations.

2.3 Maximum Entropy Approximations of Likelihood Ratio

The likelihood ratio has a simple information-theoretic interpretation which also means we can use well-known entropy approximations for its practical computation in the case where we do not want to postulate functional forms for the G's.

If we take the asymptotic limit of the likelihood ratio, we obtain

$$R \longrightarrow -H(x) - H(\hat{d}/\sigma_d) + H(y) + H(\hat{e}/\sigma_e) \tag{2}$$

where we denote differential entropy by H, the estimated residuals by $\hat{d} = y - \rho x, \hat{e} = x - \rho y$, and the variances of the estimated residuals by σ_d^2, σ_e^2.

Thus, we can approximate the likelihood ratio using any general, possibly non-parametric, approximations of differential entropy. For example, we can use the maximum entropy approximations by Hyvärinen (1998) which are computationally simple. In fact, we only need to approximate one-dimensional differential entropies, which is much simpler than approximating two-dimensional entropies.

One version of the approximations by Hyvärinen (1998) is given by

$$\hat{H}(u) = H(\nu) - k_1[E\{\log\cosh u\} - \gamma]^2 - k_2[E\{u\exp(-u^2/2)\}]^2 \tag{3}$$

where $H(\nu) = \frac{1}{2}(1 + \log 2\pi)$ is the entropy of the standardized Gaussian distribution, and the other constants are numerically evaluated as

$$\begin{aligned} k_1 &\approx 79.047, \\ k_2 &\approx 7.4129, \\ \gamma &\approx 0.37457. \end{aligned}$$

This approximation is valid for standardized variables; in fact, all the variables in (2) are standardized. The intuitive idea in this approximation is that since the Gaussian distribution has maximum entropy among all distributions of unit variance, entropy can be approximated by a measure of non-Gaussianity which is subtracted from $H(\nu)$. The sum of the second and third terms on the right-hand side of (3) is a measure of non-Gaussianity (ignoring their negative signs) since the terms are the squared differences of certain statistics from the corresponding values obtained for a Gaussian distribution. In fact, γ is defined as the expectation of $\log\cosh$ for a standardized Gaussian distribution, so the second term on the right-hand side is zero for a Gaussian distribution, just like the skewness-like statistic measured by the last term.

The expression in (2) also readily gives a simple intuitive interpretation of the estimation of causal direction. The (negative) entropies can all be interpreted as measures of non-Gaussianity, since the variables are standardized. Thus, in (2) we essentially compute the sum of the non-Gaussianities of the explaining variable and the resulting residual, and compare them for the two directions. The directions which leads to maximum non-Gaussianity is chosen.[1]

2.4 Connection to Mutual Information

It is also possible to give an information-theoretic interpretation which connects the likelihood ratios to independence measures.

A widely-used independence measure is mutual information, defined for two variables x, y as

$$I(x,y) = H(x) + H(y) - H(x,y)$$

where H denotes differential entropy. For a linear transformation

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix},$$

we have the entropy transformation formula

$$H(u,v) = H(x,y) + \log|\det \mathbf{A}|.$$

On the other hand, the transformation from x, y to x, d has unit determinant, since

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix} \begin{pmatrix} x \\ d \end{pmatrix}.$$

Thus, we have

$$H(x,d) = H(x,y)$$

and likewise for $H(y,e)$. We can now consider the mutual information of the regressors and the residuals in the two models, and in particular, compute the difference of the mutual informations to see which one is smaller. In fact, the difference of the mutual informations is asymptotically equal to the likelihood ratio R since

$$\begin{aligned} I(x,d) - I(y,e) &= H(x) + H(d) - H(x,d) - (H(y) + H(e) - H(y,e)) \\ &= H(x) + H(d) - H(y) - H(e) = H(x) + H(\frac{d}{\sigma_d}) - H(y) - H(\frac{e}{\sigma_e}) - \log\sigma_d + \log\sigma_e \\ &= H(x) + H(\frac{d}{\sigma_d}) - H(y) - H(\frac{e}{\sigma_e}) \end{aligned}$$

where the joint entropies $H(x,e)$ and $H(y,d)$ as well as the variances of the residuals (which are equal) cancel. Thus, our criterion is equivalent to evaluating the independence of x vs. d and y vs. e using mutual information, and choosing the direction in which the regressor is more independent of the residual.

Again, these developments show the important practical advantage that we only need to evaluate one-dimensional entropies, although the definition of mutual information contains a two-dimensional entropy.

1. Note that this is not the same as the simple heuristic approach in which we only compute the non-Gaussianities of the actual variables x, y and assume that direction must be from the more non-Gaussian variable to the less non-Gaussian one.

2.5 First-Order Approximation of Likelihood Ratio

Next we develop some simple approximations of the likelihood ratio. Our goal is to find causality measures which are simpler (conceptually and possibly also computationally) than the likelihood ratio or its general approximation given above.

Let us make a first-order approximation

$$G(\frac{y-\rho x}{\sqrt{1-\rho^2}}) = G(y) - \rho\, x\, g(y) + O(\rho^2)$$

where g is the derivative of G, and likewise for the regression in the other direction. Then, we get the approximation $\tilde{R}$:

$$R \approx \tilde{R} = \frac{1}{T}\sum_t G(x_t) + G(y_t) - \rho x_t g(y_t) - G(y_t) - G(x_t) + \rho y_t g(x_t) = \frac{\rho}{T}\sum_t -x_t g(y_t) + g(x_t) y_t .$$

Pham and Garrat (1997) proposed a method for estimating the derivatives of log-pdf's of random variables. Their method could be directly used for estimating g. However, since our main goal here is to find methods which work for small sample sizes, we try to avoid such estimation of the g's which has potentially a very large number of parameters. Instead, here we assume that we have some prior knowledge on the distributions of the variables in the model. In fact, a result well-known in the theory of ICA is that it does not matter very much how we choose the log-pdf's in the model as long as they are roughly of the right kind (Hyvärinen et al., 2001). This claim is partly justified by the cumulant-based analysis and simulations below.

In particular, very good empirical results are usually obtained by modelling any sparse (i.e., super-Gaussian, or positively kurtotic), symmetric densities by either the logistic density

$$G(u) = -2\log\cosh(\frac{\pi}{2\sqrt{3}}u) + \text{const.} \tag{4}$$

or the Laplacian density

$$G(u) = -\sqrt{2}|u| + \text{const.}$$

where the additive constants are immaterial. The Laplacian density is not very often used in ICA because its derivative is discontinuous at zero which leads to problems in maximization of the ICA likelihood. However, here we do not have such a problem so we can use the Laplacian density as well.

Thus, if we approximate all the log-pdf's by (4), we get the "non-linear correlation"

$$\tilde{R}_{\text{sparse}} = \rho\hat{E}\{x\tanh(y) - \tanh(x)\, y\} \tag{5}$$

where we have omitted the constant $\frac{\pi}{2\sqrt{3}}$ which is close to one, as well as a multiplicative scaling constant. Here, $\hat{E}$ means the sample average. This is the quantity we would use to determine the causal direction. Under $x \to y$, this is positive, and under $y \to x$, it is negative.

2.6 Cumulant-Based Approach

To get further insight into the likelihood ratio approximation in (5), we consider a cumulant-based approach which can be analyzed exactly. The theory of ICA has shown that cumulant-based approaches can shed light into the convergence properties of likelihood-based approaches. However,

cumulant-based methods tend to be very sensitive to outliers, so their utility is mainly in the theoretical analysis; for analysing real data, the measure in (5) is preferred.

Here, an approach based on fourth-order cumulants is possible by defining

$$\tilde{R}_{c4}(x,y) = \rho\hat{E}\{x^3y - xy^3\} \tag{6}$$

where the idea is that the third-order monomial analyzes the main nonlinearity in the nonlinear correlation. In fact, we can approximate tanh by a Taylor expansion

$$\tanh(u) = u - \frac{1}{3}u^3 + O(u^5). \tag{7}$$

Then, first-order terms are immaterial because they produce terms like $\hat{E}\{xy - xy\}$ which cancel out, and the third-order terms can be assumed to determine the qualitative behaviour of the nonlinearity.

Our main results of the cumulant-based approach is the following:

Theorem 1 *If the causal direction is $x \to y$, we have*

$$\tilde{R}_{c4} = kurt(x)(\rho^2 - \rho^4) \tag{8}$$

where $kurt(x) = E\{x^4\} - 3$ is the kurtosis of x. If the causal direction is the opposite, we have

$$\tilde{R}_{c4} = kurt(y)(\rho^4 - \rho^2). \tag{9}$$

Proof Consider the fourth-order cumulant

$$C(x,y) = \text{cum}(x,x,x,y) = E\{x^3y\} - 3E\{xy\}$$

where we assume the two variables are standardized. We have $\text{kurt}(x) = C(x,x) = \text{cum}(x,x,x,x)$. The nonlinear correlation can be expressed using this cumulant as

$$\tilde{R}_{c4} = \rho[C(x,y) - C(y,x)]$$

since the linear correlation terms cancel out. We use next two well-known properties of cumulants. First, the linearity property says that for any two random variables v, w and constants a, b we have

$$\text{cum}(v,v,v,av+bw) = a\,\text{cum}(v,v,v,v) + b\,\text{cum}(v,v,v,w)$$

and second, $\text{cum}(v,w,x,y) = 0$ if any of the variables v,w,x,y is statistically independent of the others. Thus, assuming the causal direction is $x \to y$, that is, $y = \rho x + d$ with x and d independent, we have

$$\begin{aligned}\tilde{R}_{c4} &= \rho[\text{cum}(x,x,x,\rho x+d) - \text{cum}(x,\rho x+d,\rho x+d,\rho x+d)] \\ &= \rho[\rho\text{cum}(x,x,x,x) + \text{cum}(x,x,x,d) \\ &\quad - \rho^3\text{cum}(x,x,x,x) - 3\rho^2\text{cum}(x,x,x,d) - 3\rho\text{cum}(x,x,d,d) - \text{cum}(x,d,d,d) \\ &= \rho[\rho\text{kurt}(x) - \rho^3\text{kurt}(x)] = \text{kurt}(x)[\rho^2 - \rho^4]\end{aligned}$$

which proves (8). The proof of (9) is completely symmetric: exchanging the roles of x and y will simply change the sign of the nonlinear correlation, and the kurtosis will be taken of y. ■

The regression coefficient ρ is always smaller than one in absolute value, and thus $\rho^2 - \rho^4 > 0$. Assuming that the relevant kurtosis is positive, which is very often the case for real data, the sign of $\tilde{R}_{c4}$ can be used to determine the causal direction in the same way as in the case of the likelihood approximation $\tilde{R}$ in (5). Thus, the cumulant-based approach allowed us to prove rigorously that a nonlinear correlation of the form (6) can be used to infer the causal direction, since it takes opposite signs under the two models. Note that this nonlinear correlation has exactly the same algebraic form as the likelihood ratio approximation (5); only the nonlinear scalar function is different. In particular, this analysis shows that the exact form of the nonlinearity is not important: the cubic nonlinearity is valid for all distributions of positive kurtosis.

If the relevant kurtosis is negative, a simple change of sign is needed. In general, we should thus multiply $\tilde{R}_{c4}$ by the sign of the kurtosis to obtain

$$\tilde{R}'_{c4}(x,y) = \text{sign}(\text{kurt}(x))\rho\hat{E}\{x^3y - xy^3\}.$$

Here, we get the complication that we have to choose whether we use the sign of the kurtosis of x or y. Usually, however, the signs would be the same, and we might have prior information on their sign, which is in most applications positive.[2]

Related cumulant-based measure were proposed by Dodge and Rousson (2001) and Dodge and Yadegari (2010). Their fourth-order measures used the ratio of marginal kurtoses, as opposed to the cross-cumulants we use here. They further assumed the disturbances to be Gaussian (or at least to have zero cumulants), which makes their measures less general than ours. In fact, their method relies on the fact that kurtosis is decreased by adding a Gaussian disturbance, but if the disturbance is much more kurtotic than the regressor, the opposite can be the case.

2.7 Intuitive Interpretations

Next, we provide some intuitive interpretations of the obtained first-order approximations of the likelihood ratio.

2.7.1 GRAPHICAL INTERPRETATION

The cumulants and nonlinear correlations have a simple intuitive intepretation. Let us consider the cumulant first. The expectations $E\{x^3y\}$ or $E\{xy^3\}$ are basically measuring points where both x and y have large values, but in contrast to ordinary correlation, they are strongly emphasizing large values of the variable which is raised to the third power.

Assume the data follows $x \to y$, and that both variables are sparse (super-Gaussian). Then, both variables simultaneously have large values mainly in the cases where x takes a large value, making y large as well. Now, due to regression towards the mean, that is, $|\rho| < 1$, the value of x is typically larger than the value of y. Thus, $E\{x^3y\} > E\{xy^3\}$. This is why $E\{x^3y\} - E\{xy^3\} > 0$ under $x \to y$. The idea is illustrated in Figure 1.

2. In the general case where the (real) kurtoses of x and y are allowed to have different signs, we need to compute two quantities: $\tilde{R}'_{c4}(x,y) = \text{sign}(\text{kurt}(x))\rho\hat{E}\{x^3y - xy^3\}$ and $\tilde{R}'_{c4}(y,x) = \text{sign}(\text{kurt}(y))\rho\hat{E}\{y^3x - yx^3\}$. According to the analysis above, the former quantity is positive if $x \to y$, and the latter quantity is positive if $y \to x$. However, in practice, this does not lead to a simple decision rule since due to finite sample size, or violations of the model, it could be that both of these quantities are positive, or none of them. In such cases, the decision rule should be defined so as to indicate that the causal direction could not be decided.

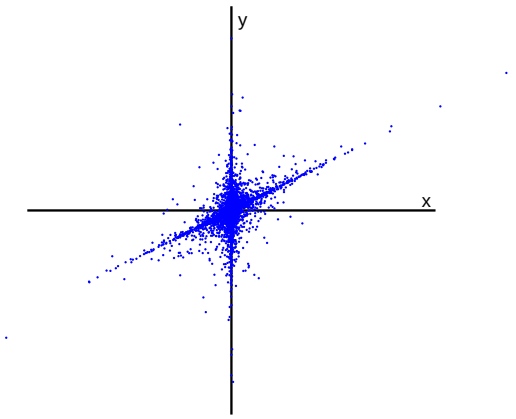


Figure 1: Intuitive illustration of the nonlinear correlations. Here, $x \to y$ and the variables are very sparse. The nonlinear correlation $E\{x^3y\}$ is larger than $E\{xy^3\}$ because when both variables are simultaneously large (the "arm" of the distribution on the right and the left), x attains larger values than y due to regression towards the mean.

This interpretation is valid for the tanh-based nonlinear correlation as well, because we can use the function $h(u) = u - \tanh(u)$ instead of tanh to measure the same correlations but with opposite sign. In fact, we have

$$\tilde{R}_{\text{sparse}} = \rho\hat{E}\{h(x)y - xh(y)\}$$

because the linear terms cancel each other. The function h is a soft thresholding function, and thus has the same effect of emphasizing large values as the third power. Thus the same logic applies for h and the third power.

2.7.2 INTERPRETATION AS IMPLICATION

Even if the data is not assumed to follow any particular model, the nonlinear correlation could be interpreted as a logical implication. In general, if the existence of event A implies the existence of event B, but there is no implication in the other direction, a causal influence from A to B might be inferred. Since $A \Rightarrow B$ is equivalent to $\neg B \Rightarrow \neg A$, there has to be some clear distinction between the events and their negations for this interpretation to be meaningful. We assume here that the events are rare, that is, have small probabilities.

Now, let us consider the events A, defined as "x takes a very large value" and B, defined as "y takes a relatively large value of the same sign as x". Notice that because the regression coefficient is smaller than one, we cannot require y to take particularly large values. It is assumed here that the thresholds for deciding when a value is large are chosen so that both of these events are rare.

To investigate implication, we can consider how to refute it. To refute $A \Rightarrow B$, we should consider cases where x takes a very large value but y takes a value of the opposite sign. This can be measured by $Ex^3(-y)$ where x^3 looks at large values of x and the minus sign changes this into a measure of how much large values of x coexist with y's of opposite sign.

Thus, $Ex^3y - Exy^3$ can be seen as measuring of how much evidence we have to refute $y \Rightarrow x$ (latter term) minus the evidence to refute $x \Rightarrow y$ (negative of first term). If it is large, we accept the implication $x \Rightarrow y$ together with its causal interpretation.

It might be argued that the connection between causality and implication could also plausibly be defined in the opposite direction: If A implies B as defined above, then B causes A. However, we shall now argue that the interpretation we gave above follows naturally from the definition of a SEM with two variables. Assume $x \to y$ and $\rho > 0$. If x is very large, y is likely to be large and of the same sign, since it is not very probable that d would cancel out the effect of ax. Thus, we have $A \Rightarrow B$ when x causes y under the SEM framework.

2.8 Noise-Tolerance of the Nonlinear Correlations

An interesting point to note is that the cumulant in (6) is, in principle, immune to additive measurement noise. Assume that instead of the real x, y, we observe noisy versions $\tilde{x} = x + n_1$ and $\tilde{y} = y + n_2$ where the noise variables are independent of each other and x and y. By the basic properties of cumulants (see proof of Theorem 1), the nonlinear correlations are not affected by the noise at all in the limit of infinite sample size. Thus, our method in not biased by noise. This is in stark contrast to ICA algorithms which are strongly affected by additive noise; thus ICA-based LiNGAM (Shimizu et al., 2006) would not yield consistent estimators in the presence of noise.

To be more precise, we have

$$\begin{aligned} E\{\tilde{x}^3\tilde{y}\} - E\{\tilde{x}\tilde{y}^3\} &= \mathrm{cum}(\tilde{x},\tilde{x},\tilde{x},\tilde{y}) - \mathrm{cum}(\tilde{x},\tilde{y},\tilde{y},\tilde{y}) \\ &= \mathrm{cum}(x,x,x,y) - \mathrm{cum}(x,y,y,y) = E\{x^3y\} - E\{xy^3\} \end{aligned}$$

due to the independence of n_1 and n_2 of the other variables and each other.

On the other hand, the estimation of ρ is strongly affected by the noise. This implies that $\tilde{R}_{c4}$ is not immune to noise. Nevertheless, measurement noise would only decrease the absolute value of ρ and not change its sign. Thus, the sign of $\tilde{R}_{c4}$ is not affected by additive measurement noise in the limit of infinite sample. This applies for both Gaussian and non-Gaussian noise.

The fact that the ρ is only a multiplicative scaling in the nonlinear correlations (6) or (5) must be contrasted with its role in the likelihood ratio (1) where its effect is more complicated. Thus, when ρ is underestimated due to measurement noise, it may have a stronger effect on the likelihood ratio, while its effect on the nonlinear correlations is likely to be weaker. While this logic is quite approximative, simulations below seem to support it.

On the other hand, the standardization of the variables is also affected by noise, in particular if the noise variances are not equal. As long as the noise variances are equal, the error in standardization will affect the measures by a multiplicative constant only, effectively making the cumulants smaller. Thus, the noise-tolerance of the cumulants may be useful in practice only if the variances of the noise variables are equal.

2.9 Skewed Variables

Above, the likelihood ratio approximations and cumulants were developed for sparse, typically symmetrically-distributed variables. Here, we consider the extension to skewed variables. Again, the underlying motivations is that if we know the distributions are skewed, we can use this prior knowledge to obtain particularly simple measures of causal direction. The cumulant-based analysis

is mainly for theoretical interest due to the sensitivity of cumulants to outliers; we provide a more robust nonlinearity for analysing real data.

2.9.1 CUMULANT-BASED APPROACH

The cumulant-based approach allows for a very simple extension of the framework to skewed variables. As a simple analogue to (6), we can define a third-order cumulant-based statistic as follows

$$\tilde{R}_{c3}(x,y) = \rho\hat{E}\{x^2y - xy^2\}. \tag{10}$$

The justification for this definition is in the following theorem, which is the analogue of Theorem 1:

Theorem 2 *If the causal direction is $x \to y$, we have*

$$\tilde{R}_{c3} = skew(x)(\rho^2 - \rho^3) \tag{11}$$

and if the causal direction is the opposite, we have

$$\tilde{R}_{c3} = skew(y)(\rho^3 - \rho^2). \tag{12}$$

Proof Consider the third-order cumulant

$$C(x,y) = \mathrm{cum}(x,x,y) = Ex^2y$$

where we assume the two variables are standardized. We have $\mathrm{skew}(x) = C(x,x) = \mathrm{cum}(x,x,x)$. The nonlinear correlation can be expressed using this cumulant as

$$\tilde{R}_{c3} = \rho[C(x,y) - C(y,x)].$$

Assuming the causal direction is $x \to y$, we have

$$\begin{aligned}\tilde{R}_{c3} &= \rho[\mathrm{cum}(x,x,\rho x+d) - \mathrm{cum}(x,\rho x+d,\rho x+d)] \\ &= \rho[\rho\,\mathrm{cum}(x,x,x) + \mathrm{cum}(x,x,d) - \rho^2\mathrm{cum}(x,x,x) - 2\rho\,\mathrm{cum}(x,x,d) - \mathrm{cum}(x,d,d)] \\ &= \rho[\rho\,\mathrm{skew}(x) - \rho^2\mathrm{skew}(x)] = \mathrm{skew}(x)[\rho^2 - \rho^3]\end{aligned}$$

which proves (11). The proof of (12) is again completely symmetric. ■

To use the measure (10) in practice, we have to take into account the fact that we cannot assume, in general, the skewnesses of the variables to have some particular sign. In some applications this is possible: For example, in resting-state fMRI data it might be safe to assume that the skewnesses are all positive because it is much more common that the signals obtain large values due to activation than due to inhibition (however, this point needs to be confirmed by empirical investigations of fMRI data).

In the general case, we propose that before computing these nonlinear correlations, the signs of the variables are first chosen so that the skewnesses are all positive. This can be accomplished simply by multiplying the variables by the signs of their skewnesses to get a new variable x^*

$$x^* = \mathrm{sign}(\mathrm{skew}(x))\,x \tag{13}$$

and the same for y (this transformation has to be done before computing ρ). Now, we have a situation similar to the previous measures: Under $x \to y$, $\tilde{R}'_{c3}(x,y) > 0$. This is because again, $|\rho| < 1$, and therefore $\rho^2 - \rho^3 > 0$ regardless of the sign of the coefficient. Likewise, for $y \to x$, $\tilde{R}'_{c3}(y,x) < 0$.

Our measure is related to the directionality measure proposed by Dodge and Rousson (2001), which in our notation would be:

$$\tilde{R}_{DR}(x,y) = [\hat{E}\{x^2 y\}]^2 - [E\{xy^2\}]^2 \tag{14}$$

which has the advantage of of being particularly simple, and does not require the skewnesses to be of any particular sign. However, our measure is more closely related to likelihood ratios which may give it some advantage in terms of statistical performance, as will be seen in the simulations below.

2.9.2 ROBUST, LIKELIHOOD-BASED APPROACH

The skewed case might also be approached by defining a skewed log-pdf and using the methods in previous sections. Unfortunately, in the theory of ICA, general-purpose skewed densities can hardly be found, and thus it is not clear how to define such densities and how generally they would be applicable. Nevertheless, a likelihood-based approach is likely to be more robust against outliers than the cumulant-based one (unless the model pdf has very light tails) which is why we develop one here.

We propose the following nonlinearity:

$$g_{\text{skew}}(x) = \log\cosh(\max(x,0)) \tag{15}$$

which can be justified as follows. Consider the following family of pdf's, defined using the derivative of the log-pdf

$$(\log p)'(x) = g_{\text{skew}}(x) - \beta x - \alpha \tag{16}$$

where β and α are parameters. Let us take α and β so that we get a standardized pdf with zero mean and unit variance. Numerical calculations show that this is obtained by values which are approximately $\alpha_0 = 0.188$ and $\beta_0 = 1.32$. The ensuing pdf is illustrated in Figure 2.

Further numerical calculations show that the higher-order cumulants of the standardized pdf are both positive: Skewness is approximately 0.37 and kurtosis 0.47.

Now, we can add any linear function and/or constant to $(\log p)'$ without changing the value of the approximative likelihood ratio in (5). In particular, using the true derivative of log-pdf in (16) is equivalent to using the algebraically simpler g_{skew}.

Thus, we obtain the following approximation for the likelihood ratio:

$$\tilde{R}_{skrb}(x,y) = \rho\hat{E}\{g_{\text{skew}}(x)y - xg_{\text{skew}}(y)\} \tag{17}$$

with g_{skew} defined in (15). Again, this applies for positively skewed variables only. If the skewnesses are not known a priori, they can be made positive by (13).

3. Estimating a Network with More Than Two Variables

In this section, we consider the general case of more than two variables. We present two approaches: First, we use the pairwise analysis in a DirectLiNGAM framework, and second, we present a two-stage method where the possible connections in a sparse graph are first pruned using covariance information.

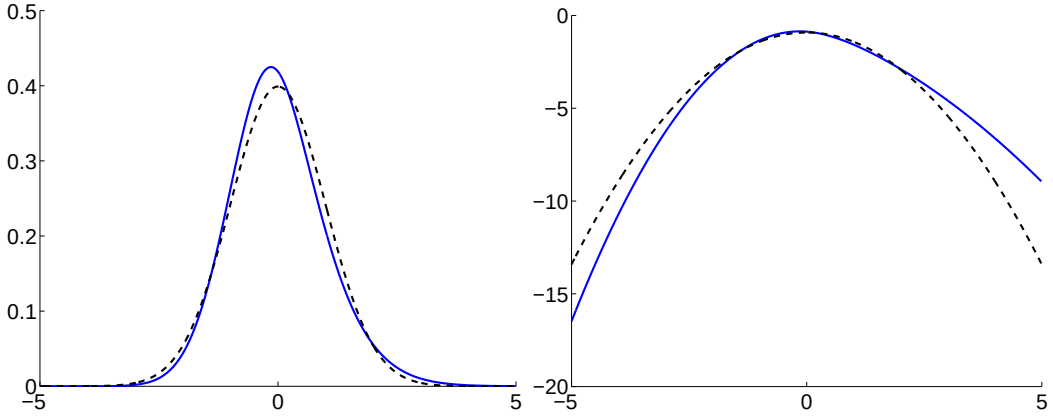


Figure 2: The pdf for robust modelling of skewed densities. Left: the pdf corresponding to the derivative of log-pdf in (16) is plotted (solid curve) with α and β chosen so that the density is standardized. For comparison, the Gaussian density of the same mean and variance is plotted as well (dashed). Right: the logarithms of the same density functions.

3.1 Model Definition

Denote by $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ the vector of observed variables. The linear non-Gaussian acyclic model (LiNGAM) proposed by Shimizu et al. (2006) can be expressed as

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}$$

where $\mathbf{e}$ is the vector of disturbances, and $\mathbf{B}$ is the matrix that describes the influences of the x_i on each other; the diagonal of $\mathbf{B}$ is defined to be zero.

It was shown by Shimizu et al. (2006) that the model is identifiable under the following assumptions: a) the e_i are non-Gaussian, b) the e_i are mutually independent, and c) the matrix $\mathbf{B}$ corresponds to a directed acyclic graph (DAG). It is well-known that the DAG property is equivalent to an existence of an ordering of the variables x_i (not necessarily unique) in which there are only connections "forward" in the ordering; if the variables are re-ordered according to the causal ordering, the matrix $\mathbf{B}$ has all zeros above the diagonal.

3.2 Using Pairwise Measures in the DirectLiNGAM Framework

The first way to use the pairwise analysis developed above to estimate LiNGAM which has more than two variables is to use the DirectLiNGAM framework (Shimizu et al., 2011).

3.2.1 FINDING ROOT OF GRAPH

In the DirectLiNGAM approach, we first compute the likelihood ratios of all different pairs of variables, and store the log-likelihood ratio for x_i and x_j as the (i, j)-th entry of a matrix $\mathbf{M}$. Alternatively, we can use the likelihood ratio approximations which can be all subsumed under the algebraic form

$$\mathbf{M} = \mathbf{C} \odot E\{\mathbf{x}g(\mathbf{x})^T - g(\mathbf{x})\mathbf{x}^T\} \tag{18}$$

where $\odot$ is element-wise multiplication. The nonlinearity g is typically chosen so that it is $g(u) = \tanh(u)$ for symmetric sparse data and $g(u) = -u^2$ or the function in (15) for skewed data. $\mathbf{C}$ is the covariance matrix of the data; since the data is assumed standardized $\mathbf{C}$ equals the matrix of correlation coefficients.

Now, for the variables x_i which have no parents, all entries in the i-th row of $\mathbf{M}$ are non-negative, neglecting random errors. (Note that there is no reason why there would be only one such "root" variable.) This was shown to be exactly true for the cumulant-based approaches $g(u) = -u^3$ and $g(u) = -u^2$ (assuming that the kurtoses or skewnesses, respectively, are positive) and is true as a first-order approximation based on (7) for $g(u) = \tanh(u)$. The reverse also holds if we assume faithfulness.[3]

Thus, we first find the row, say with index i^*, which is most likely to have all non-negative entries (the actual estimation procedure is considered below). Then, we regress ("deflate") the variable x_{i^*} out of all the other variables (Shimizu et al., 2011). We iterate this procedure by computing $\mathbf{M}$ again for the deflated $\mathbf{x}$. By locating the row which is most likely to have only non-negative entries in the newly computed $\mathbf{M}$, we thus find a variable which has no parents except for possibly the first variable found in the previous step. Repeating this, we find variables which are next in the partial order given by the DAG. Thus in the end we have the causal ordering of the variables.

After such estimation of the causal ordering, estimating the coefficients b_{ij} is easy by just ordinary least-squares estimation (Shimizu et al., 2006).

Alternatively, we could use a simple approximation which is very simple and computationally efficient. Instead of carrying out deflation by regression as described above, we simply remove the entries of the rows and columns corresponding to the already "found" variables in the matrix $\mathbf{M}$, and iterate the procedure. Thus, we obtain the causal ordering directly from a single matrix of nonlinear correlations, without any deflation. This is an approximation with no rigorous justification (because when removing the root we should also remove its effect on all the entries of $\mathbf{M}$) and it is likely to be inconsistent. However, in simulations reported below it works quite well. It has the benefit of being computationally extremely simple, and it gives a simple conceptual link between causal ordering and the nonlinear correlations and cumulants.

3.2.2 AGGREGATING PAIRWISE MEASURES

To use the method just described we have to solve the problem of aggregating the pairwise measures. We need to find the row which is most likely to be all non-negative up to random errors. Obviously, we could just take the sums of the entries in each row and locate the maximum sum but this is not likely to be optimal. So, we next develop a more principled way of aggregation.

Consider the $m_{ij}, j = 1,\dots,n$ for a fixed i, which are the estimates of pairwise likelihood ratios or some approximations. Assume they are independent and have Gaussian distributions $N(\mu_{ij}, \sigma^2)$, where the variances are assumed to be equal for simplicity. The variance σ^2 is the estimation error due to finite sample, and the μ_{ij} are the true values. The posterior of μ_{ij} given m_{ij} is then Gaussian

3. For a variable x_0 with no parents, any other variable is of the form $x_j = ax_0 + d$ where a expresses the total effect coming from x_0, and d is a sum of the inputs from other external influences, which are, by definition, independent of x_0. Thus, the pairwise model holds with a d independent of x_0 and the pairwise measure is non-negative. On the other hand, consider x_i which does have parents. Now, go backwards in the graph until you find a node x_0 which has no parents (in a DAG, such a variable is guaranteed to exist). According to the logic just given, we have $x_i = ax_0 + d$, again with an independent d. By faithfulness, $a \neq 0$. Since changing the direction simply changes the sign of our measures, there will be a negative entry in the i-th row, and it has to be non-zero.

with mean m_{ij} and variance σ. Thus, the posterior log-probability that all of the $\mu_{ij}, j = 1,\ldots,n$ are positive can be calculated as

$$\log \prod_j P(\mu_{ij} > 0 | m_{ij}) = \log \prod_j P(\frac{\mu_{ij} - m_{ij}}{\sigma} > -\frac{m_{ij}}{\sigma} | m_{ij}) = \sum_j \log \Phi(\frac{m_{ij}}{\sigma}) \tag{19}$$

where Φ is the cumulative distribution function of the standardized Gaussian distribution. Estimating σ is possible but we prefer to assume it is very small and make the following approximation:

$$\log \Phi(\frac{m_{ij}}{\sigma}) \approx -\frac{1}{2\sigma^2} \min(0, m_{ij})^2$$

which can be seen to be quite accurate by a simple numerical comparison, and avoids numerical problems in computing the logarithm of Φ for large negative values. Now, σ is simply a multiplicative scaling constant which can be ignored when comparing estimates of the log-probabilities in (19).

Thus, we propose the following way of aggregating the pairwise likelihood ratios. Compute for each row of $\mathbf{M}$

$$m_i = -\sum_j \min(0, [\mathbf{M}]_{ij})^2$$

which, intuitively speaking, punishes violations of the positivity. The index i^* with maximum m_i is thus taken as the estimate of a variable with no parents, that is, a first variable in the causal ordering.

3.3 Two-Stage Approach to Estimating a Sparse Model

If the matrix $\mathbf{B}$ is known to be sparse, we can use a two-stage method in which we first estimate the connections in an undirected sense, and then find their directions using our pairwise method. This two-stage method is interesting from the viewpoint of clearly dividing the estimating problem into two parts.

We first find undirected connections by using any known method for estimating a Gaussian undirected model (Spirtes et al., 1993). In the simplest case, this can be based on the inverse covariance matrix, or the precision matrix. As is well-known in the theory of Gaussian graphical models, there is an intimate connection between the non-zero entries in the precision matrix and the existence of connections in the SEM—although the connection is not quite simple, especially for directed graphs. In contrast, the direction of a connection cannot be easily determined from the covariances, and is often unidentifiable, which was of course the original motivation for introducing non-Gaussian models (Shimizu et al., 2006). Nevertheless, as a first approximation, we can prune the set of candidate connections using the inverse covariance matrix, and apply our pairwise analysis only on those connections which this covariance-based analysis indicates to be present.

In an estimated inverse covariance matrix, there are of course no exact zeros. Thus, we use bootstrapping to test if each entry is non-zero. That is, we draw bootstrap samples of the data, and compute the inverse covariance for each such sample. The ratio of the mean and the standard deviation of the bootstrap estimates of any given entry is then compared with the relevant quantile of a standardized Gaussian distribution.[4] The test is made separately for each non-diagonal entry of the inverse covariance matrix.

4. In the simulations below, we also tried methods for sparse estimation of the inverse covariance matrix. However, we found that this simple testing procedure works by far the best. The sparse inverse estimation methods are, in fact, developed for the case of a very large number of variables, and thus may not be useful in our simulations where we typically have 5-10 variables only.

Depending on the goal of the analysis, it may or may not be necessary to do corrections for multiple testing. If we do such corrections, we can actually claim that the connections found are statistically significant. However, this is obtained at the cost of a large number of false negatives. On the other hand, if we simply consider the existence of the connections as another set of parameters to estimate, it may be more advantageous not to make such corrections to reduce the overall error rate. In fact, a false negative (setting an existing connection to zero) could be considered quite a serious error in this context, so we prefer to use a rather large α. In the simulations below, we thus set the false positive rate $\alpha = 0.01$ with no correction for multiple testing. Such corrections will of course be needed if our aim were to claim that a particular connection exists, but if our goal here is mainly the inference of the causal ordering, some false positives should not matter since they are likely to correspond to small values of the coefficients anyway.

Then, for each of those significantly non-zero connections, we determine the direction of causality using our pairwise tests. There is no need to do any kind of deflation anymore. If we want to convert the obtained estimates into a total ordering of the variables, we input those connections which were not pruned to the ordering method presented by Shimizu et al. (2006).

4. Estimating Cyclic Models

An important generalization of the DAG framework would be to estimate cyclic models. Here, we assume the following well-known generative model for the data. First, the external influences arrive in the system at time $t = 0$

$$\mathbf{x}(0) = \mathbf{e}$$

where $\mathbf{x}(t)$ is value a hypothetical dynamic system at time point t. Then, at subsequent time steps, the external influence is completemented by feedback as

$$\mathbf{x}(t+1) = \mathbf{e} + \mathbf{B}\mathbf{x}(t)$$

where the matrix $\mathbf{B}$ has zero diagonal, which means we do not allow self-loops. Assuming that $\mathbf{B}$ is stable in the sense that its largest eigenvalue is smaller than one in absolute value, we have in the limit

$$\mathbf{x} = \sum_{k \geq 0} \mathbf{B}^k \mathbf{e} = (\mathbf{I} - \mathbf{B})^{-1} \mathbf{e}$$

and thus

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e} \tag{20}$$

where $\mathbf{B}$ is now allowed to be cyclic. This gives a simple interpretation of a model of the form (20) in the case where $\mathbf{B}$ is allowed to be cyclic. As above, the e_i are assumed independent and non-Gaussian.

In fact, estimation of such a model by ICA is possible if $\mathbf{B}$ is small enough, namely if all its entries are smaller than one in absolute value. Then, it is possible to estimate the model even by ICA, since after estimating ICA, we can find the right permutation of the components based on putting the largest entries of each row in the diagonal. Thus, the model is identifiable under these assumptions. This is shown in detail in the following Theorem:

Theorem 3 *Assume that the data follows the cyclic LiNGAM model in (21) with no self-loops. Assume further that all the entries in the matrix* $\mathbf{B}$ *have absolute values smaller than one, and that*

the dominant eigenvalue of $\mathbf{B}$ *is smaller than one in absolute value. Then, the model is uniquely identifiable, that is, the matrix* $\mathbf{B}$ *can be estimated from the data without any ambiguity.*

Proof: The data actually follows the ICA model as

$$\mathbf{x} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{e}. \tag{21}$$

The ICA model is known to be identifiable up to a) the ordering of the components and b) a scalar multiplier for each of the components (Comon, 1994). The unidentifiability of the scalar multiplier disappears here because by definition, the diagonal of the inverse of the mixing matrix has all ones due to the diagonal matrix in (21). Thus, it was shown by Shimizu et al. (2006) that this implies the idenitifiability of the LiNGAM model if we can solve the indeterminacy of the permutation. Acyclicity was used for this purpose by Shimizu et al. (2006). Here, we use the assumption of absolute values smaller than one. In fact, consider the estimate of the inverse of the mixing matrix. Normalize it by dividing each row by its maximum element. Then, it equals $\mathbf{I} - \mathbf{B}$ up to a random permutation of the rows. Due to our assumption of $\mathbf{B}$, all non-diagonal entries in this matrix are smaller than one in absolute value. Thus, the original (correct) permutation of the rows can be found by locating on each row i the unique entry which is equal to one. Denoting its column index by $j(i)$, the original matrix is given by permuting the rows of the matrix to the ordering given by $j(i)$, that is, the ordering which puts the ones in the diagonal. □

This also suggests that we can estimate the model using the sparse graphs idea above. We prune the inverse covariance matrix to find where there are (probably) connections, and then find the directions of the connections using our pairwise measures. Using pairwise connections makes sense if we further assume that there are no pairwise loops, that is, connections $x_i \to x_j$ and $x_j \to x_i$ are not both non-zero. The main justification for this approach is that since the connections are weak, one can assume that the cyclicity has little effect on local pairwise measures. However, an exact convergence of such a method to the right parameter values does not seem possible to show in general.

5. Estimation in Case of Nonlinear Relations

In this Section, we generalize our method to a nonlinear model.

5.1 Definition of Nonlinear Model

Another interesting extension of the linear causal models is obtained by considering nonlinearities instead of non-Gaussianities (Hoyer et al., 2009). We define the two models as follows. The first one, $x \to y$, is given by

$$y = f(x) + d$$

where f is a nonlinear function, not necessarily invertible or even differentiable. The disturbance d is again independent of x. Both x and y are standardized to unit variance. The second model is denoted by $y \to x$ and defined as

$$x = g(y) + e$$

where g is another nonlinear function, and e is a disturbance variable. Other approaches to inferring the causal direction with nonlinear relations were introduced by Zhang and Hyvärinen (2009), Daniušis et al. (2010) and Mooij et al. (2010).

5.2 Likelihood Ratio for Nonlinear Model

The likelihood of the model $x \to y$ can be obtained as the sum of the log-prior of the variable x and the log-likelihood of the residual:

$$\log p(x,y) = \log p_x(x) + \log p_d(y - f(x)) = G_x(x) + G_d(\frac{\hat{d}}{\sigma_d}) - \log \sigma_d$$

where we denote, like above, the variance of the standardized residual by σ_d^2, the log-pdf of the standardized residual by G_d, and the log-pdf of x by G_x. Thus, like in the linear case, we obtain

$$R \quad = \quad \left[\frac{1}{T}\sum_t G_x(x_t) + G_d(\frac{y_t - f(x_t)}{\sigma_d}) - G_y(y_t) - G_e(\frac{x_t - g(y_t)}{\sigma_e})\right] \quad - \quad \log \sigma_d \quad + \quad \log \sigma_e. \quad (22)$$

An important difference to the linear case is that the variances of the residuals need not be equal, $\sigma_d \neq \sigma_e$, so they do not cancel. In an information-theoretic formulation, we obtain asymptotically

$$R \longrightarrow -H(x) - H(\hat{d}/\sigma_d) + H(y) + H(\hat{e}/\sigma_e) - \log \sigma_d + \log \sigma_e. \quad (23)$$

We can approximate R using the same maximum entropy approximations (Hyvärinen, 1998) as in the linear case in Section 2.3. The only difference is that we need to add the log-variances of the residuals to the expression. Thus, an important advantage of our approach is that we do not need any measures of independence per se; estimation of one-dimensional differential entropies is sufficient.

On the other hand, it may be advantageous to adapt the approximation to the nonlinear case. First, it does not seem useful to consider the prior non-Gaussianities of the variables, since a nonlinear mixing can change non-Gaussianities in completely unpredictable ways. This is unlike in the case of ICA, where a linear mixing decreases non-Gaussianity. Second, we can assume that the residuals tend to be sparse, and model them as Laplacian. This has the further advantage of making the measure more robust to outliers.

Now, for a Laplacian variable, the scale parameter σ is most naturally estimated as the mean absolute deviation (MAD), which is the maximum likelihood estimate. If we plug this estimate in the likelihood ratio, and omit the priors on x and y, we have

$$R = \left[\frac{1}{T}\sum_t -\frac{\sqrt{2}|y_t - f(x_t)|}{\hat{\sigma}_d}) + \frac{\sqrt{2}|x_t - g(y_t)|}{\hat{\sigma}_e}\right] - \log \hat{\sigma}_d + \log \hat{\sigma}_e$$

$$= -\sqrt{2}\frac{\hat{\sigma}_d}{\hat{\sigma}_d} + \sqrt{2}\frac{\hat{\sigma}_e}{\hat{\sigma}_e} - \log \hat{\sigma}_d + \log \hat{\sigma}_e$$

which gives finally the following objective

$$\tilde{R}_{\text{mad}} = -\log \hat{E}\{|\hat{d}|\} + \log \hat{E}\{|\hat{e}|\} \quad (24)$$

where $\hat{E}$ denotes the sample average, and thus $\hat{E}\{|.|\}$ denotes the MAD. In other words, we have an objective which simply compares the mean absolute deviations in the two cases.

The likelihood ratio depends on the estimated nonlinearities f, g. The estimation of f and g can be done with classic least-squares estimation methods independently of any developments in this paper. A large number of non-parametric methods have been developed in the literature, see, for example, Hoyer et al. (2009) for an example.

5.3 Connection to Independence-Based Nonlinear Methods

In fact, our method has a close connection to the independence-based method by Hoyer et al. (2009), generalizing the connection shown in Section 2.4. Using basic information-theoretic properties, we have under $x \to y$

$$H(x,y) = H(x) + H(y|x) = H(x) + H(y - f(x)|x) = H(x) + H(d|x) = H(x,d)$$

and likewise, this is equal to for $H(y,e)$. Now, just like in the linear case, we can consider the difference between the mutual informations of the regressors and residuals in the two directions, and obtain

$$I(x,d) - I(y,e) = H(x) + H(d) - H(y) - H(e) = H(x) + H(\frac{d}{\sigma_d}) - H(y) - H(\frac{e}{\sigma_e}) + \log\sigma_d - \log\sigma_e$$

where two terms equal to $h(x,y)$ cancel. Here, we see that asymptotically, our objective derived from the likelihood ratio is equal to the difference of the two mutual informations (with sign reversed). Its sign tells which mutual information is larger, and in particular, in which direction the residual of the regression is more independent. Thus, using the likelihood ratio is equivalent to using mutual information as independence measure in the methods by Hoyer et al. (2009).

The developments given above thus show that when comparing independencies of the residuals like Hoyer et al. (2009), it is not necessary to explicitly estimate mutual information; estimation of one-dimensional entropies leads to an equivalent result.

6. Simulations

We conducted simulations comparing the different methods proposed in this paper, as well as previously proposed LiNGAM estimation methods. In all the simulations, we emphasize difficult conditions. In most of the simulations, this means the case where the number of observations is small; the exception being the simulations with added measurement noise. We also take weakly non-Gaussian disturbances according to the logistic distribution in Equation (4), with the same aim of simulating difficult conditions.

The methods were compared with three previously published methods:

- LiNGAM estimated using ICA, as proposed in the original paper introducing LiNGAM by Shimizu et al. (2006).[5]
- DirectLiNGAM, specifically the kernel-based version proposed by Shimizu et al. (2011).[6]
- In case of skewed data, we used the measure proposed by Dodge and Rousson (2001), given in Equation (14).

The LiNGAM methods were implemented using the software found on the authors' web sites.

We computed different performance indices for the methods. For acyclic models, we computed

5. Since basic FastICA, which is an integral part of the method, has convergence problems with the basic tanh nonlinearity in the case of a small sample size, we used the stabilized version by Hyvärinen (1999) obtained in the standard FastICA package by the option "stabilize". For skewed data, we used the skewness as a measure of non-Gaussianity.
6. We did not include the earliest version of DirectLiNGAM proposed by Shimizu et al. (2009) in the comparison because in later simulations by Sogawa et al. (2010); Hyvärinen (2010), its performance was found clearly inferior to that of the kernel-based version of DirectLiNGAM.

1. The Spearman rank-correlation coefficient between the causal ordering given by the method and the true ordering.

2. The percentage of connections for which a method correctly estimated the direction, considering only connections existing in the data-generating process. Here, the point was to look at the abilities of the methods to find the directions locally, and thus the global ordering given by the method was *not* used (except for DirectLiNGAM which essentially only computes a global ordering and derives local ordering from that). For the ICA-based LiNGAM, we computed the measure $\text{sign}(|b_{ij}| - |b_{ji}|)$ and used it in the same way as the signs of the pairwise measures.

3. The percentage of data sets for which a method correctly estimated the first variable in the causal ordering, that is, the variable with no parents.

For cyclic models, the comparison was based on the second measure only, since the other two are not well-defined. Furthermore, we computed the CPU time needed for the computations.

Unless otherwise mentioned, the connection matrices were generated completely randomly, giving a fully connected DAG. The non-zero coefficients in the acyclic **B** had a uniform distribution in the union of the intervals $[-0.6,-0.2]$ and $[0.2,0.6]$.

6.1 Simulation 1: Sparse Influences

In the first, basic simulation, sample size and data dimension were varied so that there were in total four different scenarios:

1. $n = 5$, $T = 100$, fully connected DAG

2. $n = 2$, $T = 100$, fully connected DAG

3. $n = 5$, $T = 200$, fully connected DAG

4. $n = 5$, $T = 400$, fully connected DAG

The disturbances had logistic distributions, with standard deviations equal to one. 2,000 data sets were generated for each scenario; however, for DirectLiNGAM and ICA-LiNGAM only 1,000 were used due to excessive computational demands.

To estimate the model, we used the following methods proposed above. First, the maximum entropy approximation to the likelihood ratio in (3) was used in DirectLiNGAM with deflation. Second, the LR first-order approximation matrix (18) was used in DirectLiNGAM with the nonlinearity $g(u) = \tanh(u)$ and with deflation. Third, the nonlinear correlations in (18) were used to estimate the causal ordering without any deflation, simply by locating the minimum of the row sums of that matrix, removing the corresponding rows and columns, and so on, as described at the end of section 3.

See Figure 3 for the results. Typically, the tanh-based likelihood ratio approximation ("tanh") with deflation was the best. The method without deflation ("nodf") gives, by definition, the same result for the total causal directions correct and first variable found, but looking at the rank-correlations, we see that it is typically the second best. The maximum entropy approximation is usually the third best. ICA-based LiNGAM is usually fourth but when there is more data, it can have very good

performance. The (kernel-based) DirectLiNGAM ("kdir") is typically last, although not necessarily worse than ICA-based LiNGAM.

Regarding computational load, the methods proposed here are one to two orders of magnitude faster than the others.

6.2 Simulation 2: Sparse Influences with Noise

In the second simulation, we tested the noise-tolerance of the algorithms. The data dimension was varied from $n = 2$ to $n = 8$ and fully connected DAGs were used as above. The sample size was set to $T = 10,000$, which means we are now analyzing the statistical consistency[7] of the method only and neglecting random effects by taking a very large sample size. The noise was Gaussian and had unit variance. The performance indices and algorithms are as in the first simulation. The results are shown in Figure 4. We can see that the tanh-based approximation is clearly the best, as predicted by our cumulant-based analysis. ICA-based LiNGAM, the maximum entropy approximations, and especially kernel-based DirectLiNGAM seem to be more sensitive to noise.

6.3 Simulation 3: Skewed Influences

In the third simulation, we tested the performance of the methods with skewed data. We used the nonlinear correlation based on the third order cumulant ("skew"), introduced in Section 2.9, as well as the robust measure in Equation (15), denoted by "skw2".

We used two different skewed distributions for the disturbances. In both cases, the data was obtained from a Gaussian mixture. One of the Gaussian distributions in the mixture had zero mean and unit variance, while the other had mean equal to three and unit variance. The two distributions we generated were distinguished by the amount of data points drawn from the two Gaussians. In the first case ("pdf 1"), the "outlying" distribution with mean three generated 20% of the data, while in the second case ("pdf 2"), it generated only 5%. Thus, pdf 2 was quite sparse whereas pdf 1 was not. We would then expect sparsity-based methods to work well with pdf 2 but not very well with pdf 1. The data dimension were to $n = 2, n = 5$ and sample sizes $T = 100, 200$, respectively. DAGs were generated to be fully connected.

The results are shown in Figure 5. We see that all the methods have very similar performance, except the Dodge-Rousson measure which was somewhat worse. However, the computational loads are very different, our two likelihood ratio approximations being faster than the earlier LiNGAM methods by at least an order of magnitude.

6.4 Simulation 4: Skewed Influences with Noise

We further conducted a simulation with observational noise added to the skewed data. Again, we fixed the sample size to $T = 10,000$ and the noise variance to two (larger than above since these methods seem to be more tolerant to Gaussian noise), while the dimension and the skew data distribution were varied. We used only the skewed and sparse pdf 2. The results are in Figure 6. Here, we start seeing clear differences in the statistical performances of the methods. In line with our theoretical analysis, the skewness cumulant-based method is the most resistant to noise. The robust skewed LR approximation in Section 2.9.2 is second.

7. That is, convergence in the limit of infinite sample.

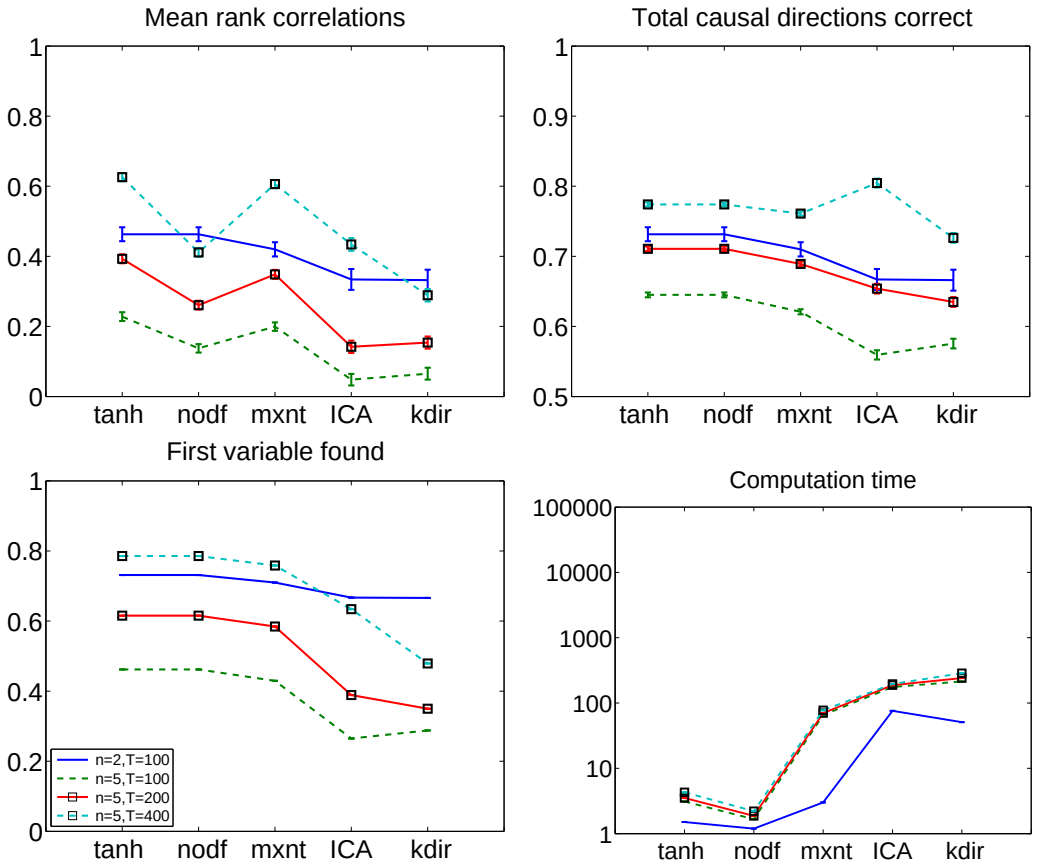


Figure 3: Simulation 1. Results of basic simulation with sparse, non-skewed data without noise. Top left: Mean of rank-correlation coefficients between the estimated causal ordering and the true ordering. The error bars are standard errors of the mean. Top right: The proportion of (really existing) connections for which the method estimated the direction correctly (chance level is 50%). Bottom left: The proportion of data sets for which the method estimated the first variable in the causal ordering correctly, that is, the variable with no parents. Bottom right: Computation times of one run of the different algorithms in milliseconds; note the logarithmic scale. Different colours are different data-generating scenarios. The algorithms used are as follows:
"tanh": LR approximations in (18) based on tanh nonlinearity, combined with deflation in DirectLiNGAM;
"nodf": no deflation in likelihood ratio approximations, that is, ordering based on the LR approximation matrix in (18) without any recomputation of the matrix;
"mxnt": maximum entropy approximation in (3) for likelihood ratios;
"ICA": LiNGAM estimated by ICA;
"kdir": kernel-based DirectLiNGAM.

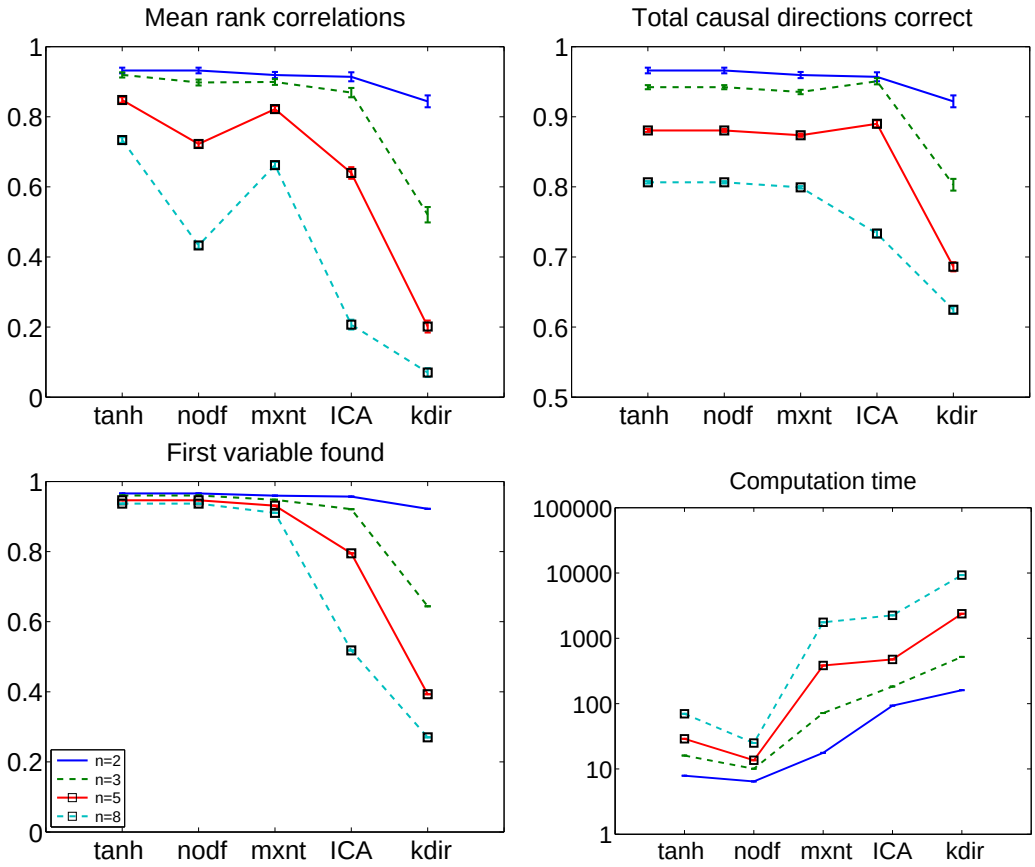


Figure 4: Simulation 2, with noise. Legend as in Figure 3, and with $T = 10,000$. The noise standard deviations were all equal to one.

6.5 Simulations 5 and 6: Two-Stage Approach and Sparse Graphs

Next we investigated the utility of the two-stage approach of Section 3.3. We generated sparse graphs only. The graphs were based on a simple "serial" structure $x_1 \to x_2 \to \ldots \to x_n$ with a random connection strength in the same range as above. We further added 0, 1, or 2 connections in random locations in the graph (preserving the DAG structure), the number of connections having equal probabilities for the three values. The data sizes were 500, 900, 900, 900 and the number of variables 5, 9, 15, 20, respectively. We used higher dimensions than above because otherwise the networks could not be very sparse. In the testing for the existence of connections, we set the false-positive rate to $\alpha = 0.01$ without correction for multiple testing, as motivated above.

In Simulation 5, we used sparse, non-skewed (logistic) influences, and in Simulation 6, skewed influences as in Simulation 3. To add more realism to the simulations, we also added noise to the data. The noise standard deviations were 0.2 in Simulation 5 and 0.6 in Simulation 6.

The results for Simulation 5 are shown in Figure 7. We can see that the two-stage method has a performance which compares quite favourably with the other methods: ICA-LiNGAM and DirectLiNGAM perform quite badly with these combinations of sample size and dimension. Note that

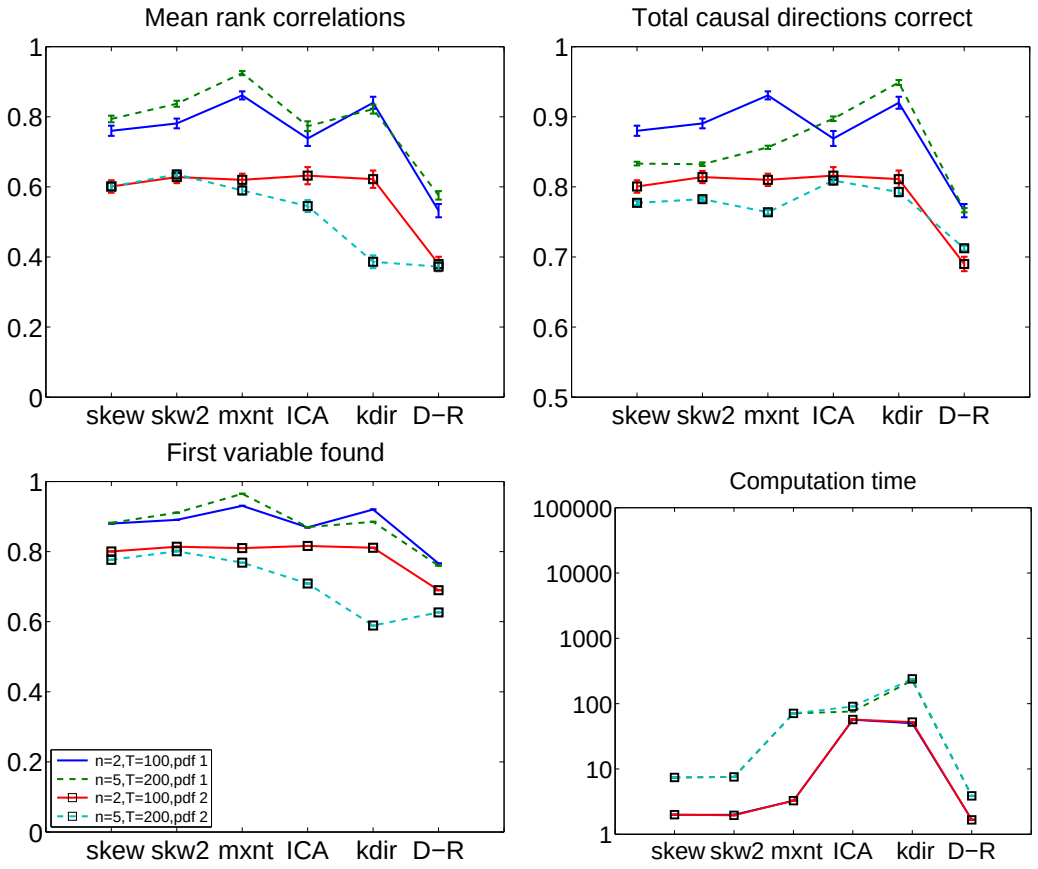


Figure 5: Simulation 3, with skewed data. Legend as in Figure 3, with the following new algorithms:
"skew": cumulant-based LR approximation in (10), combined with deflation in DirectLiNGAM;
"skw2": the robust LR approximation proposed in Section 2.9.2; and
"D-R": the measure by Dodge and Rousson (2001).

for "total causal directions correct", the two-stage method has, by definition, the same performance as "tanh" and "nodf". In fact, if our interest in only to discover the directions without bothering to estimate which variables are connected, or we are given perfect prior knowledge on which variables are connected, there is in fact no need to do the pruning in the first stage of the method.

So, the utility of the new method ("icth") is mainly seen in the mean rank correlations plot: There is a modest improvement. The point here is that knowledge of which variables are connected improves the estimation of the causal ordering (DAG structure) by showing which directionalities should be used when pooling their information together, and which directionalities should be discarded (because the variables are not connected at all).

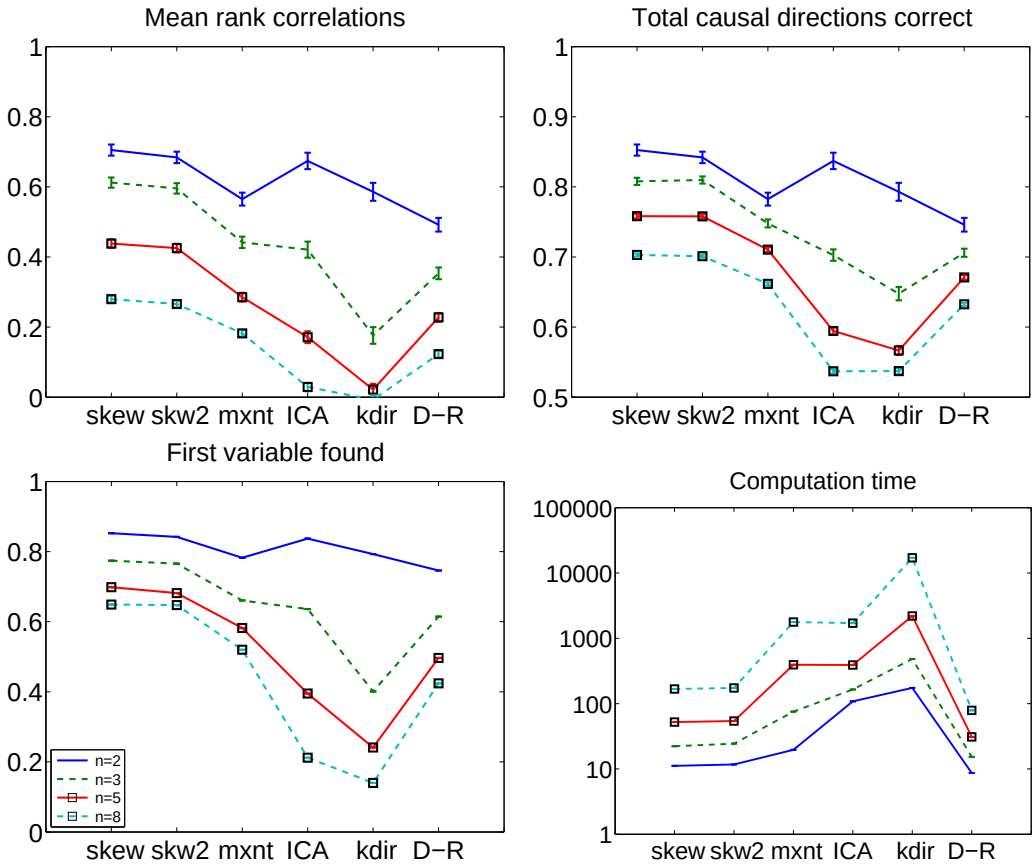


Figure 6: Simulation 4, with skewed data with noise. Legend as in Figure 5.

Interestingly, all the methods proposed in this paper are clearly superior to the methods proposed earlier (ICA-based LiNGAM and kernel-based DirectLiNGAM). Thus, the main utility of the present framework may indeed be in estimating directionality in sparse networks.

We carried out the same simulation for skewed influences using the skewed pdf 1. Results are in Figure 8. When looking at methods using the same causality measure ("skew" vs. "icsk", and "skw2" vs. "ics2"), we see that the pruning methods are better in terms of the mean rank correlations. However, the maximum entropy method without pruning is actually the best.

6.6 Overview of Simulations 1–6

To provide a succinct overview of the simulations reported above, we averaged the performance indices over the different scenarios (taking into account only scenarios in which the algorithm took part). Furthermore, we divided the simulations into three groups: basic data (simulations 1 and 2), skewed data (simulations 3 and 4) and sparse connections either with sparse data (simulation 5) or skewed data (simulation 6). We further averaged the performance indices inside these groups.

The results are shown in Figure 9.

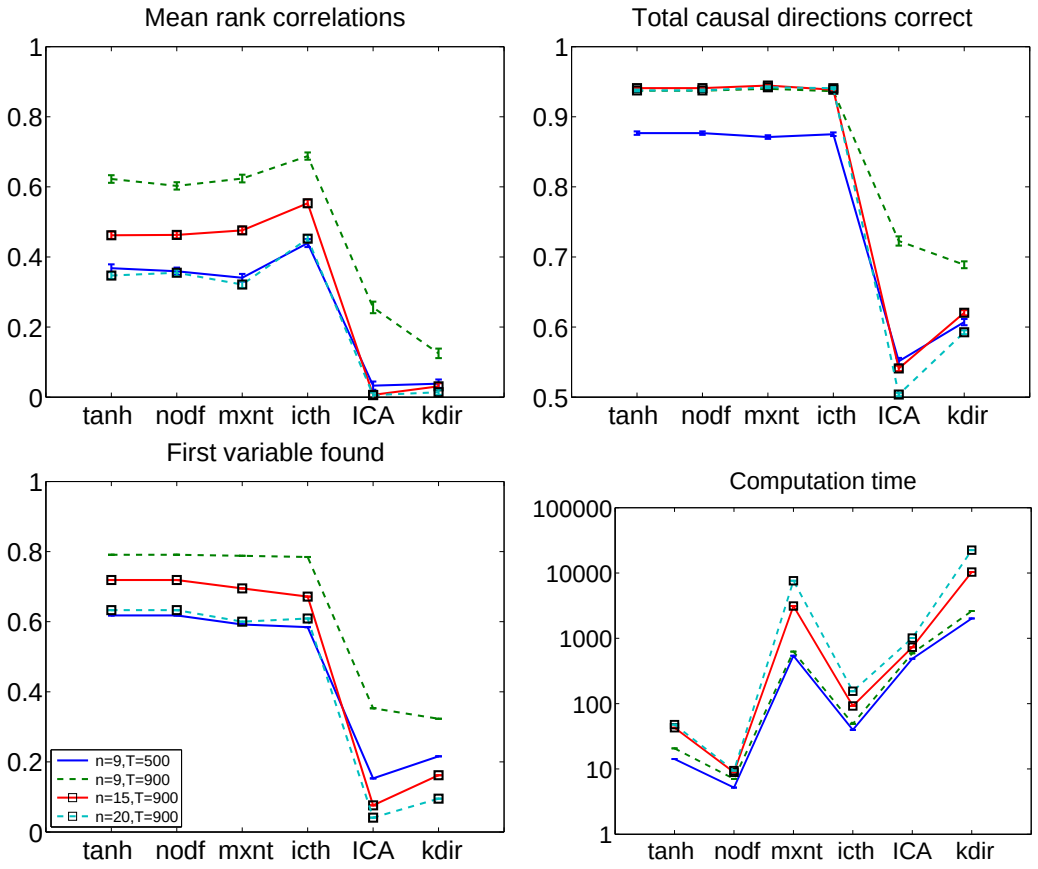


Figure 7: Simulation 5, with the two-stage pruning method and only sparse graphs. Legend as in Figure 3, but now including the new algorithm "icth" which prunes the graph based on inverse covariance and then estimates the directions using the same method as "tanh". (Note that only "icth" uses information on the pruned inverse covariance, other methods are as in Simulation 1.)

6.7 Simulation 7: More Variables than Observations

Next, we considered the case where there are more variables than observations, or at least the number of variables is equal to the number of observations. We considered four scenarios, with n ranging from 100 to 200 and T ranging from 100 to 400. In preliminary simulations, it turned out that the problem was too difficult for logistic disturbances, so we used Laplacian disturbances here.

We only attempted to estimate the first two variables and not the whole causal ordering. The very first variables in the causal ordering can be considered to be the exogenous ones and thus finding them is of special interest (Sogawa et al., 2011). We only used three of the new proposed methods because none of implementations of the existing LiNGAM methods was such that it could readily be used for this case.

The results are shown in Figure 10. While the performance of the methods is not very good, it is very much above chance level (which would be 0.01 or less for finding the first variable). It is

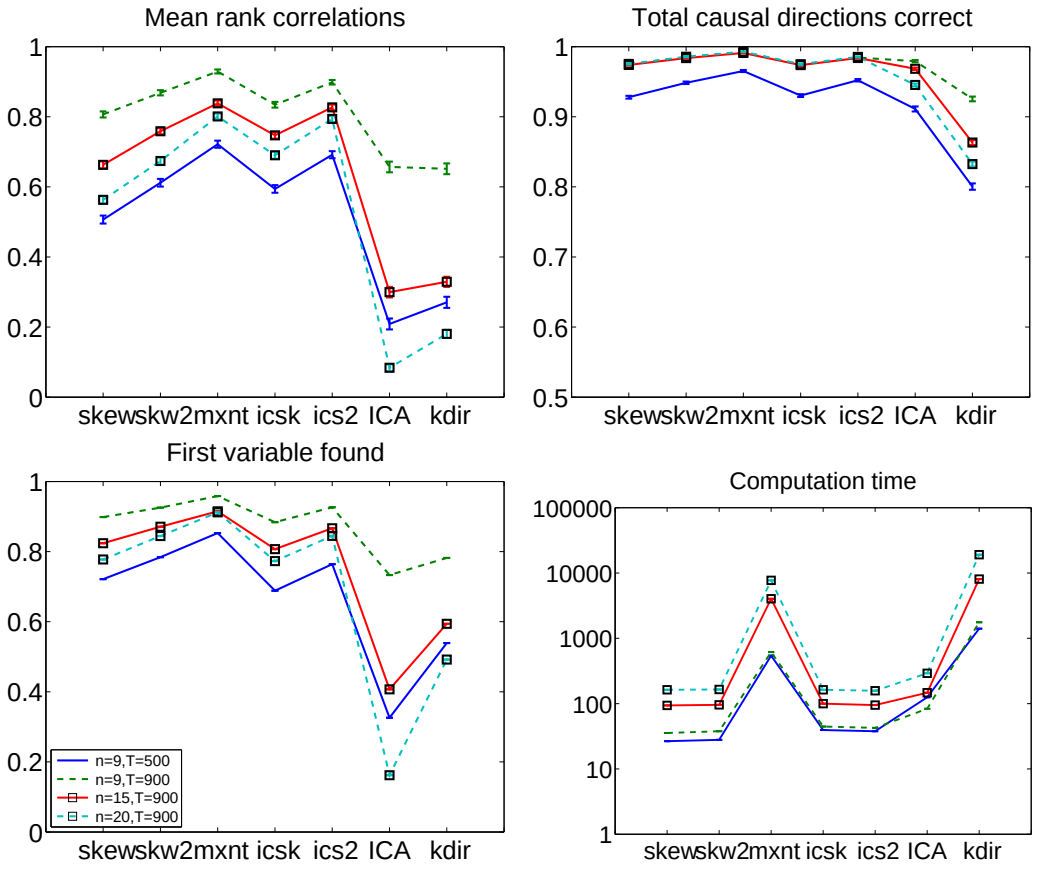


Figure 8: Simulation 6, with skewed data, the two-stage pruning method and only sparse graphs. Legend as in Figure 5, but now including the new algorithm "icsk" which prunes the graph based on inverse covariance and estimates the directions based on the skewness cumulant, and "ics2" which uses the robust skewness measure.

interesting to note that here the first-order approximation of likelihood is more than 100 times faster than the maximum entropy approximation.

6.8 Simulation 8: Cyclic Graphs

To test the new framework in the case of cyclic graphs, we created cyclic graphs by a simple ring structure: $x_1 \to x_2, \ldots, x_n \to x_1$. Further connections (0, 1, or 2) were added in random locations as in Simulation 5 above. Such data were created according to the generating model in Section 4. We further added noise with standard deviation 0.2. The dimensions of the data and the sample sizes were as in Simulations 5 and 6. The influences had logistic distributions.

The only methods we compared were ICA-based LiNGAM and our two-stage pruning methods, since the DirectLiNGAM methods cannot be used in the cyclic case. The results are shown in Figure 11. The first observation is that both methods performed relatively well, obtaining 70%-90% percent of the directions right. Our new method is slightly better than ICA-based LiNGAM.

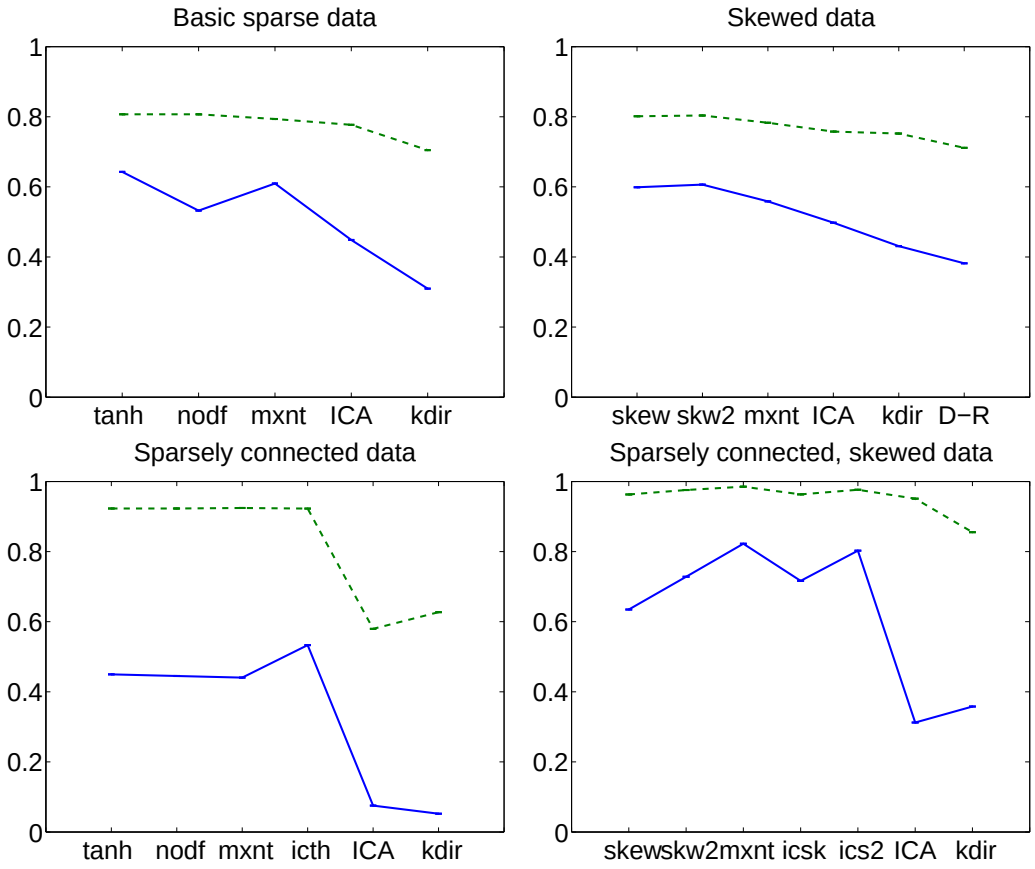


Figure 9: Overview of Simulations 1–6. Median correlations (blue, solid) and average directions correct (green, dashed) are plotted averaged over different scenarios and similar simulations.

It should be emphasized here that our method assumes that there are no self-loops, so there is no indeterminacy in the results, as shown in Section 4.

6.9 Simulation 9: Nonlinear Relations

Finally, we performed simulations on the nonlinear model. We generated data from a model

$$x_2 = \alpha \text{sign}(x_1)|x_1|^\gamma + d \tag{25}$$

where both x_1 and d were standardized Gaussian. The exponents γ were given values 0.5 and 2, and the parameter α was randomly drawn between 0.5 and 1.5. The sample sizes were either $T = 200$ or $T = 500$.

We then fitted the nonlinearity of the same functional form (25), that is, using the parameters α and γ, to the data with a least-squares fit, and estimated the causal direction using the criterion in (22), or the criterion in (24). (Thus, we did not use a nonparametric model of the nonlinearity. See Section 8 for estimation with non-parametric nonlinearities.) For comparison, we used the methods

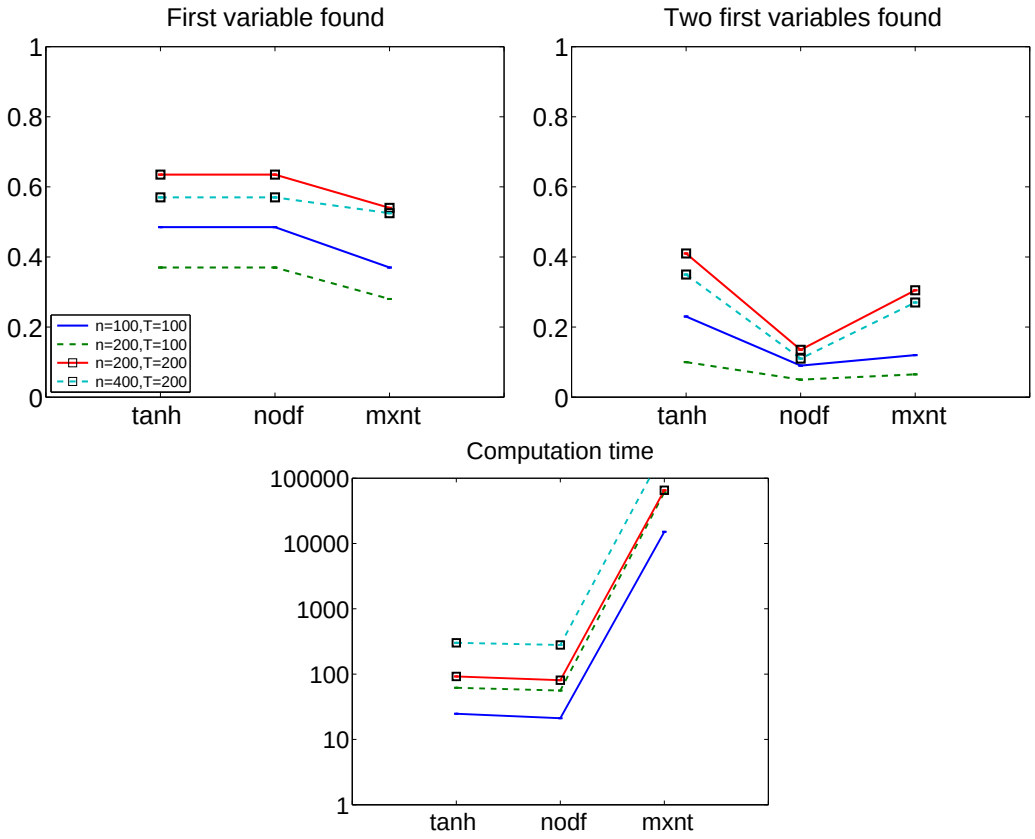


Figure 10: Simulation 7, with more variables than observations. Legend as in Figure 3. Rank correlations and causal directions correct are omitted because we only computed the first two variables for lack of computation time.

tanh and maxent introduced above in a purely linear way (i.e., *not* fitting the nonlinear function above, but just a linear function exactly as in previous simulations), to see if linear methods are able to cope with this data.

Furthermore, we used the criterion of the original method by Hoyer et al. (2009), based on the HSIC independence test by Gretton et al. (2008) of x (resp. y) and the residual in the regression of y on x (resp. of x on y). This was implemented by code provided by A. Gretton,[8] using the default setting for the kernel width.

The results are shown in Figure 12. Our likelihood ratio methods both performed relatively well, although the independence-based method by Hoyer et al. (2009) was arguably better than our maximum entropy method. However, the HSIC-based method was 10-100 times slower due to the use of kernel methods. The linear methods did not perform well at all.

8. Downloaded from `http://www.gatsby.ucl.ac.uk/~gretton/indepTestFiles/indep.htm`.

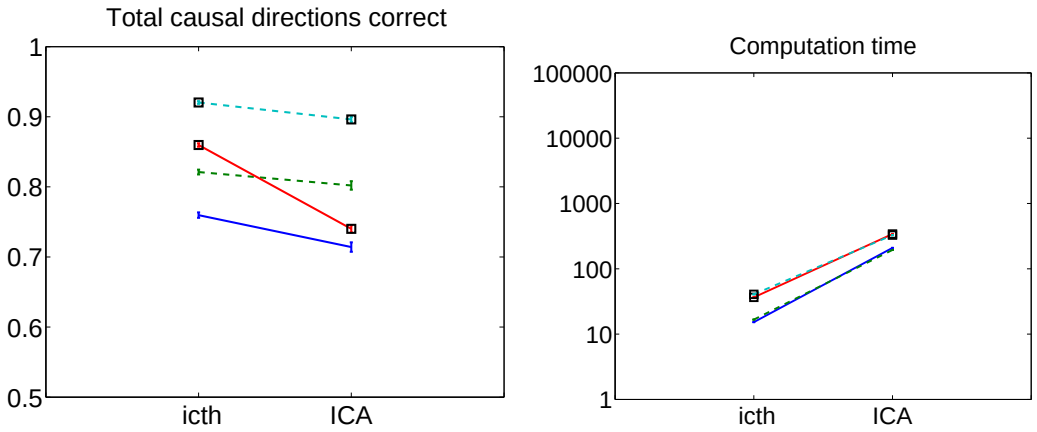


Figure 11: Simulation 8, with cyclic sparse graphs. Legend (sample sizes and dimensions) as in Figure 7.

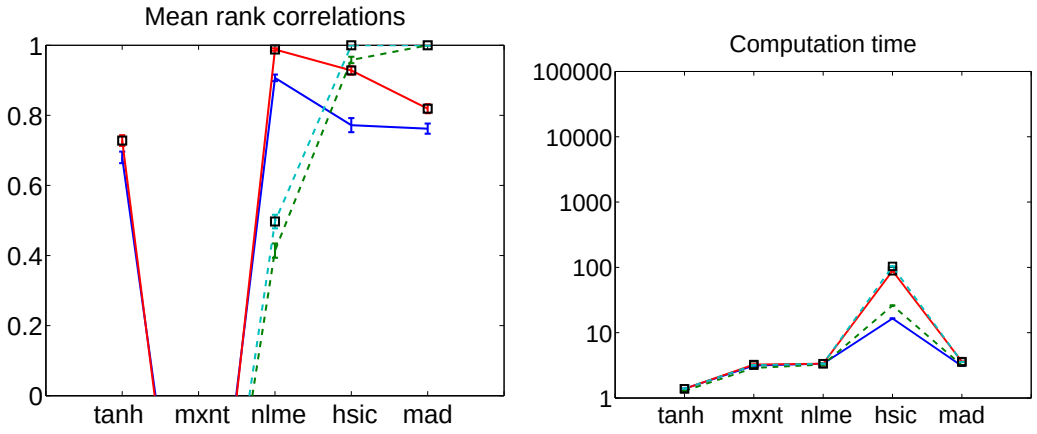


Figure 12: Simulation 9, with nonlinear model. The new algorithms are "nlme", the proposed likelihood ratio method extended to the nonlinear case using maximum entropy approximation in (22); "mad", a simplified and robustified approximation of the likelihood ratio in (24); "hsic", the original nonlinear method using independence (Hoyer et al., 2009). Blue: $\gamma = 0.5, T = 200$, Green: $\gamma = 2, T = 200$, Red: $\gamma = 0.5, T = 500$, Cyan: $\gamma = 2, T = 500$.

6.10 Simulation 10: Misspecified Disturbances

We further performed simulations in which the model is misspecified. First, we considered Simulation 1 with the following change: the disturbances generating the data had Laplacian distributions. Everything else was identical to Simulation 1, including the assumed log-pdf's and nonlinearities. Thus, the distribution of the disturbances was not exactly known, and was misspecified in the estimation. We also added the basic skewness method in the set of algorithms.

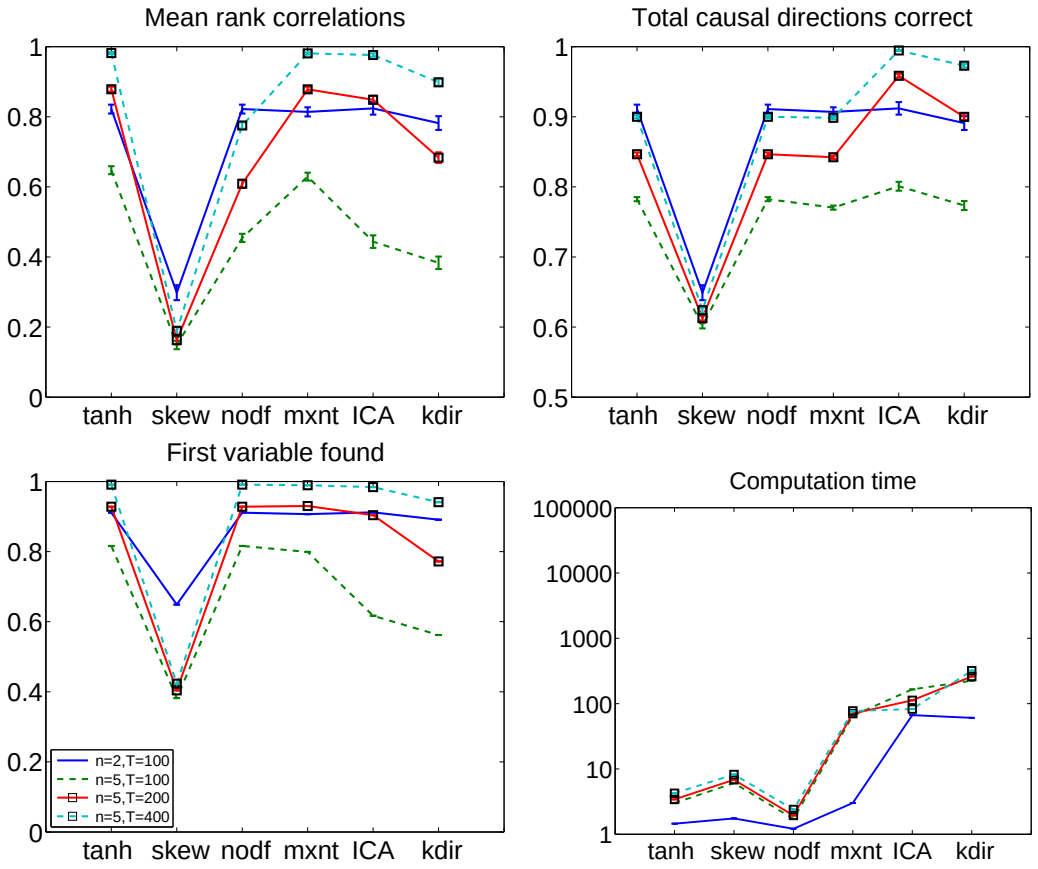


Figure 13: Simulation 10. Like Simulation 1 but with Laplacian disturbances used in generating the data, and the "skew" method added.

The results are in Figure 13. We see that the performance of most methods is actually better. This was expected in light of the theory of ICA, where it is well-known that if the actual data is more non-Gaussian than assumed in the estimation method, this is not a problem for most methods, and only increases the performance of the method compared to the case of less non-Gaussian data. In fact, the reason why we used the logistic distribution in generating data in many of the simulations above was in order to make the problem more difficult. On the other hand, the reason for using the logistic distribution in the algorithms is that it is widely used in ICA and has been empirically found to work well, partly due to the fact that its log-pdf is smooth, unlike many other super-Gaussian log-pdf's including the Laplacian.

Of course, if the non-Gaussianity is completely misspecified in the estimation method, estimation with fixed nonlinearities will inevitably fail. This is why the skewness method was hardly above chance level.

6.11 Simulation 11: Latent Variables

We conducted a further simulation to gain some insight into the robustness of the different methods to the existence of latent variables. We first created data $\mathbf{x}_0$ as in Simulation 1, with $n = 4, T = 500$. Then, we added a latent variable to the data as

$$\mathbf{x} = \mathbf{x}_0 + \alpha \mathbf{b} \tilde{s}$$

where $\tilde{s}$ is a latent variable with a standardized logistic distribution, $\mathbf{b}$ is a weight vector with elements drawn from a standardized Gaussian distribution, and α is the general strength of the latent variable, which took the values $[0, 0.25, 0.5, 1]$ in the different scenarios. (The value of $\alpha = 0$ effectively means no latent variables and is provided for comparison.) The latent variable $\tilde{s}$ violates the assumption of LiNGAM of having only one (independent) external input for each variable x_i.

The results are in Figure 14. Basically, we see that the latent variable deteriorates the performance of all the algorithms quite uniformly. It does not seem that any of the algorithms would be more resistant, or more sensitive, to latent variables than the others.

Recently, the framework presented here was generalized to a model including Gaussian latent variables by Chen and Chan (2012).

7. Experiments on Simulated fMRI Data

Since causal discovery experiments on real data are very difficult to validate, we use here brain imaging data which has been simulated using state-of-the-art biophysical models (Smith et al., 2011).

7.1 Simulation of fMRI Data

The simulations are described in detail by Smith et al. (2011); here we give a short summary. Networks of varied complexity were used to simulate fMRI timeseries. The simulations were based upon the dynamic causal modelling (DCM) forward model (Friston et al., 2003). DCM uses the nonlinear balloon model (Buxton et al., 1998) for the vascular dynamics, that is, the connection between the neural activities and the measured signal, sitting above a simple neural network model of the neural dynamics. Estimating causality from fMRI data is particularly challenging as the signal-to-noise ratio is relatively poor, fMRI timeseries are fairly Gaussian, and the number of timepoints is generally in the low hundreds.

We defined a number of nodes, which corresponded to brain regions. First, we generated the external inputs to the nodes, u_i, which are not quite the same as the external influences in the SEM, although related. They were binary (activity is "up" or "down") and generated using a Poisson process that controls the likelihood of switching the state. Neural noise of standard deviation 1/20 of the difference in height between the two states was added. The mean durations of the states were 2.5s (up) and 10s (down), with the asymmetry representing longer average "rest" than "firing" durations.

The neural activities z_i were then simulated using the DCM neural network model, as defined by

$$\dot{\mathbf{z}} = \sigma \mathbf{A} \mathbf{z} + \mathbf{M} \mathbf{u}$$

where $\mathbf{A}$ is the matrix defining network dynamics and $\mathbf{M}$ contains the weights controlling how the external inputs feed into the network (often just the identity matrix). The off-diagonal terms

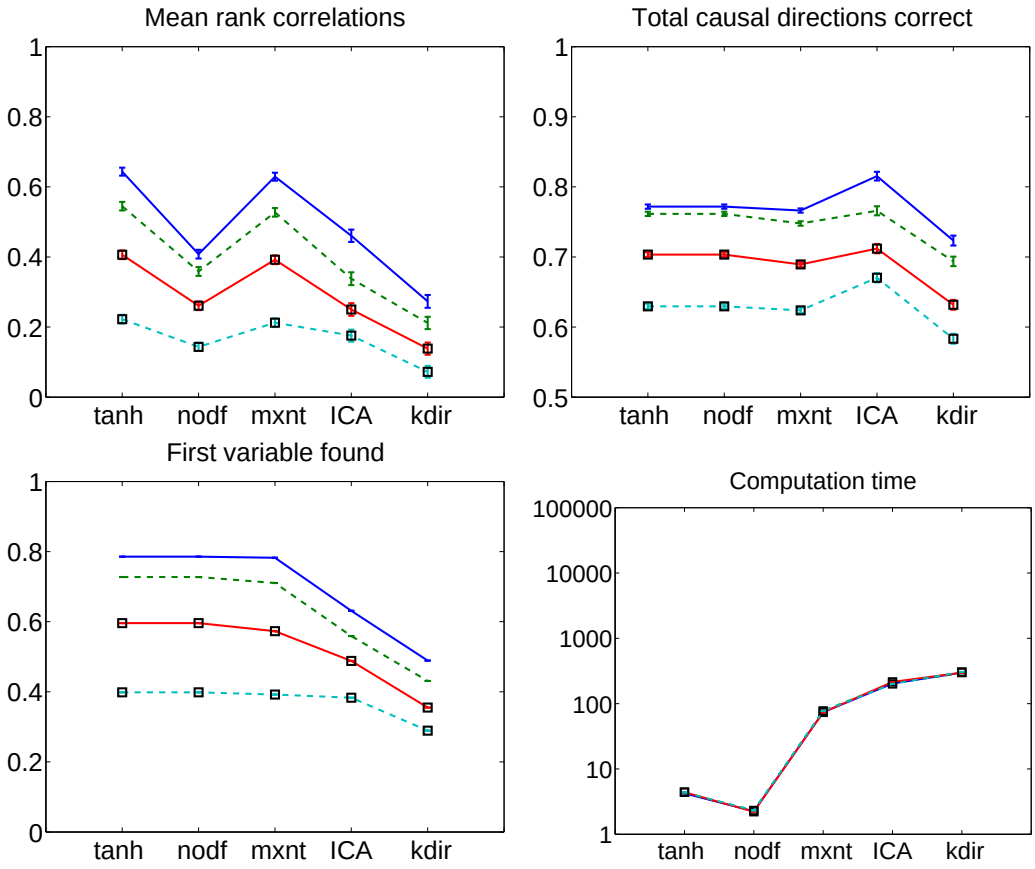


Figure 14: Simulation 11. Like Simulation 1, with $n = 4, T = 500$, but with a latent variable added. The four scenarios (curves) correspond to different strengths of the latent variable, starting with zero strength in blue curve.

in $\mathbf{A}$ determine the network connections between nodes, and the diagonal elements are all set to -1, to model within-node temporal decay; thus σ controls both the within-node (neural) temporal inertia/smoothing and the neural lag between nodes.

A central problem in fMRI is that the measured signal does not directly correspond to $\mathbf{z}$. To simulate this, each node's neural timeseries z_i was fed through the nonlinear balloon model for vascular dynamics responding to changing neural demand. The balloon model parameters were in general set according to the prior means in DCM. However, it is known that the haemodynamic processes vary across brain areas and subjects, resulting in different lags between the neural processes and the BOLD data, with variations of up to at least 1s (Handwerker et al., 2004; Chang et al., 2008). We therefore added randomness into the balloon model parameters at each node, resulting in variations in HRF (haemodynamic response function) delay of standard deviation 0.5s. Finally, thermal white (measurement) noise of standard deviation 0.1–1% (of mean signal level) was added.

Thus, we obtained the measured fMRI signals. They were sampled with a sampling interval of 3s (in most simulations), corresponding to a typical time of repetition (TR) in brain imaging literature.

The simulations comprised 50 separate realisations (or "subjects"), all using the same simulation parameters, except for having independently generated external inputs and different HRF parameters at each node (as described above); furthermore, the connection strengths were slightly perturbated for each subject. Each "subject's" data was a 10-minute fMRI session (200 timepoints) in many of the simulations.

For a summary of the specifications for the 28 simulations see Table 1.

Sim	n	length (mins)	TR (s)	noise (%)	HRF std (s)	other factors
1	5	10	3.00	1.0	0.5	
2	10	10	3.00	1.0	0.5	
3	15	10	3.00	1.0	0.5	
4	50	10	3.00	1.0	0.5	
5	5	60	3.00	1.0	0.5	
6	10	60	3.00	1.0	0.5	
7	5	250	3.00	1.0	0.5	
8	5	10	3.00	1.0	0.5	shared inputs
9	5	250	3.00	1.0	0.5	shared inputs
10	5	10	3.00	1.0	0.5	global mean confound
11	10	10	3.00	1.0	0.5	timeseries mixed with each other
12	10	10	3.00	1.0	0.5	new random timeseries mixed in
13	5	10	3.00	1.0	0.5	backwards connections
14	5	10	3.00	1.0	0.5	cyclic connections
15	5	10	3.00	0.1	0.5	stronger connections
16	5	10	3.00	1.0	0.5	more connections
17	10	10	3.00	0.1	0.5	
18	5	10	3.00	1.0	0.0	
19	5	10	0.25	0.1	0.5	neural lag=100ms
20	5	10	0.25	0.1	0.0	neural lag=100ms
21	5	10	3.00	1.0	0.5	2-group test
22	5	10	3.00	0.1	0.5	nonstationary connection strengths
23	5	10	3.00	0.1	0.5	stationary connection strengths
24	5	10	3.00	0.1	0.5	only one strong external input
25	5	5	3.00	1.0	0.5	
26	5	2.5	3.00	1.0	0.5	
27	5	2.5	3.00	0.1	0.5	
28	5	5	3.00	0.1	0.5	

Table 1: Summary of the 28 fMRI simulations' specifications (from Smith et al., 2011)

7.2 Estimation Methods for Simulated fMRI Data

We used pairwise measures with three different nonlinearities: tanh, skewness, and the robust measure of skewness. We did not estimate the existence of connections at all since that is not the main topic of the paper: We only looked at the estimated directionalities for those connections which really existed in the simulated data.

Since the skewness of the data was mainly positive, we used this prior information of positive skewness skewness-based measures. In other words, we skipped the skewness correction in (13).

For comparison, we used two methods by Patel et al. (2006) which were the most successful of the many methods tested by Smith et al. (2011), as well as basic ICA-based LiNGAM (Shimizu et al., 2006) which was applied on the whole data (not pairwise).

7.3 Results on Simulated fMRI Data

The goal is thus to recover the directionalities defined by the non-zero entries of the neural dynamics matrix **A** by estimating the directionalities given by **B** in our SEM. We evaluated the results using the same measures as Smith et al. (2011) to allow for direct comparison.

Our evaluations looked at the distribution of correctly estimated connections over the 50 simulated subjects. We concentrate here on evaluating methods for single subject (single session) data sets, and only utilise multiple subjects' data sets in order to characterise variability of results across multiple random instantiations of the same underlying network simulation. This is in contrast to the approach by Ramsey et al. (2011) who estimated the network over random subsets of 10 subjects, which is an easier task, at least if the subjects are not very different.

The raw connection strengths b_{ij} were converted into z-scores in order to make the plots more qualitatively interpretable, as the connection strengths are then more comparable across the different methods. The conversion from raw connection strengths to z-scores was achieved by using a null distribution of connection strengths, obtained by feeding in truly null timeseries data into each of the estimation methods. The null data was created by testing for connections between timeseries from *different* subjects' data sets, which have no causal connections between them (i.e., we randomly shuffled the subject labels for each node in the network). See Smith et al. (2011) for details. To specifically look at estimated directionalities, we use the higher of the two directions' measures to be the estimated connection strength.

The results are shown in Figs 15-16. The distributions are over all 50 simulated "subjects" and over all correct network edges; higher is better. Note, however, that this plot does not take into account the false positives, that is, the values estimated in the network matrix that should be empty, and concentrates exclusively on the estimation of causal directions. The plots, known as "violin plots", are simply (vertically-oriented) smoothed histograms, reflected in the vertical axis for better visualisation.

We see that the pairwise methods perform much better than Patel's measures or ICA-based LiNGAM on all the simulations. (The comparison to ICA-based LiNGAM may not be entirely fair since it estimates more than just directionalities.) In fact, our methods perform extremely well in most simulations. In all the simulations, the pairwise measures are the best, although in two cases the performances of all methods are so close to chance level that any comparison is difficult. The results are not very good in the following cases:

- Simulation 13 which has backwards connections (i.e., both $x \to y$ and $y \to x$) which is not surprising since it is against the basic philosophy of our modelling. However, the performance is

Caption for Figs. 15 and 16 on pages 147 and 148: The z-scores of the different measures used to determine the directionality, computed over subjects and connections, are shown as violin plots (i.e., histograms rotated to be horizontal and made symmetric). If the directions are found completely correctly, the violin plots are concentrated at the top. The blue dots show the the percentage of correctly estimated directions. First, we have three pairwise methods, and for comparison, two methods by Patel, as well as ICA-based LiNGAM. Each panel is one simulation.

---

clearly better than chance, which shows that our method is able to find the dominant direction some of the time.

- Simulation 22 which has nonstationary connection strengths, which violates another basic assumption of the model.
- Simulation 24 in which one of the inputs is strongly dominant. This is presumably because the effective signal-to-noise ratio is too poor for many of the connections.
- Simulations 25-27 in which the number of data points is smaller (recording length is shorter), performance being close to chance level for all methods.

Among the pairwise measures, there is no clear winner. However, the robust skewness measure is most often the best (in those cases where a clear difference can be seen), and never much worse then the other two.

8. Nonlinear Causal Discovery on Real Data

Finally, we applied our nonlinear methods on the Tübingen-UCI cause-effect data set[9] which consists of real measurements in which the true direction of causation in known. We used a total of 81 data sets, consisting of the subset of those data that had exactly two variables. In addition to our two new nonlinear methods, we applied the original HSIC-based methods by Hoyer et al. (2009), as well as the linear likelihood ratio with maximum negentropy approximation of Section 2.3. The relations in this data set are often quite nonlinear, and the linear methods are hardly above chance level (results not shown except for one method below), so we concentrate on the nonlinear methods here.

The nonlinear regression was performed by first fitting a least-squares regression curve using a Gaussian process as implemented in the `fit_gp` package by J. Mooij, based on code by C. E. Rasmussen and H. Nickisch. We used only the first 1,000 data points due to the excessive computational complexity of HSIC.

The results are shown in Table 2. The linear method, as well as our basic nonlinear method using maximum entropy were hardly better than chance. The method by Hoyer et al. (2009) was close to 62%. On the other hand, our simplest approximation using mean absolute deviation was 69% correct.

Presumably, one reason for the weak performance of our nonlinear method using maximum entropy approximations was that many of the data sets have strong outliers. The MAD-based objective in Equation (24) is quite robust against them (although the nonlinear regression method was not

9. Data set can be found at `http://webdav.tuebingen.mpg.de/cause-effect/`.

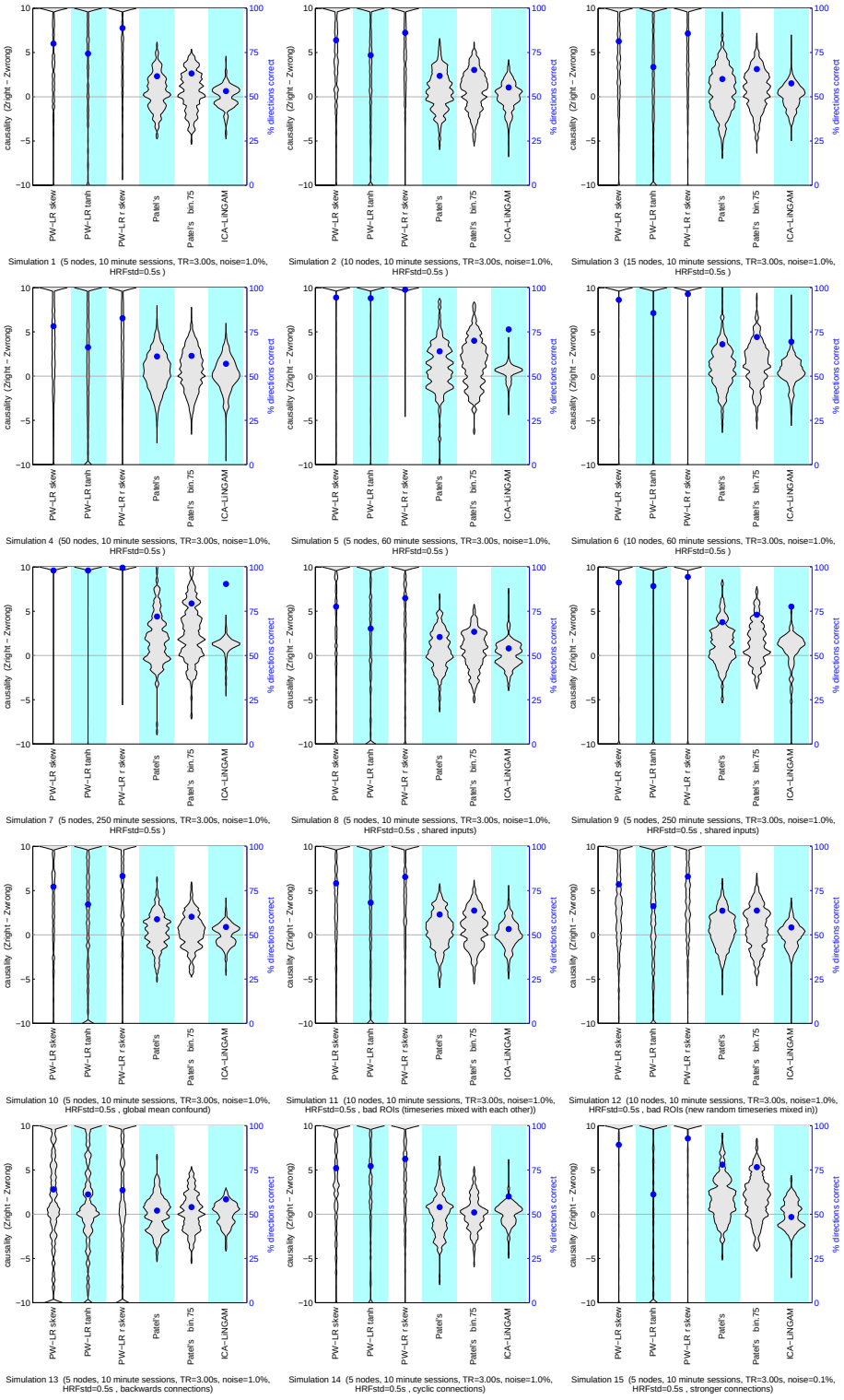


Figure 15: Results on simulated fMRI data, first half. See page 146 for caption.

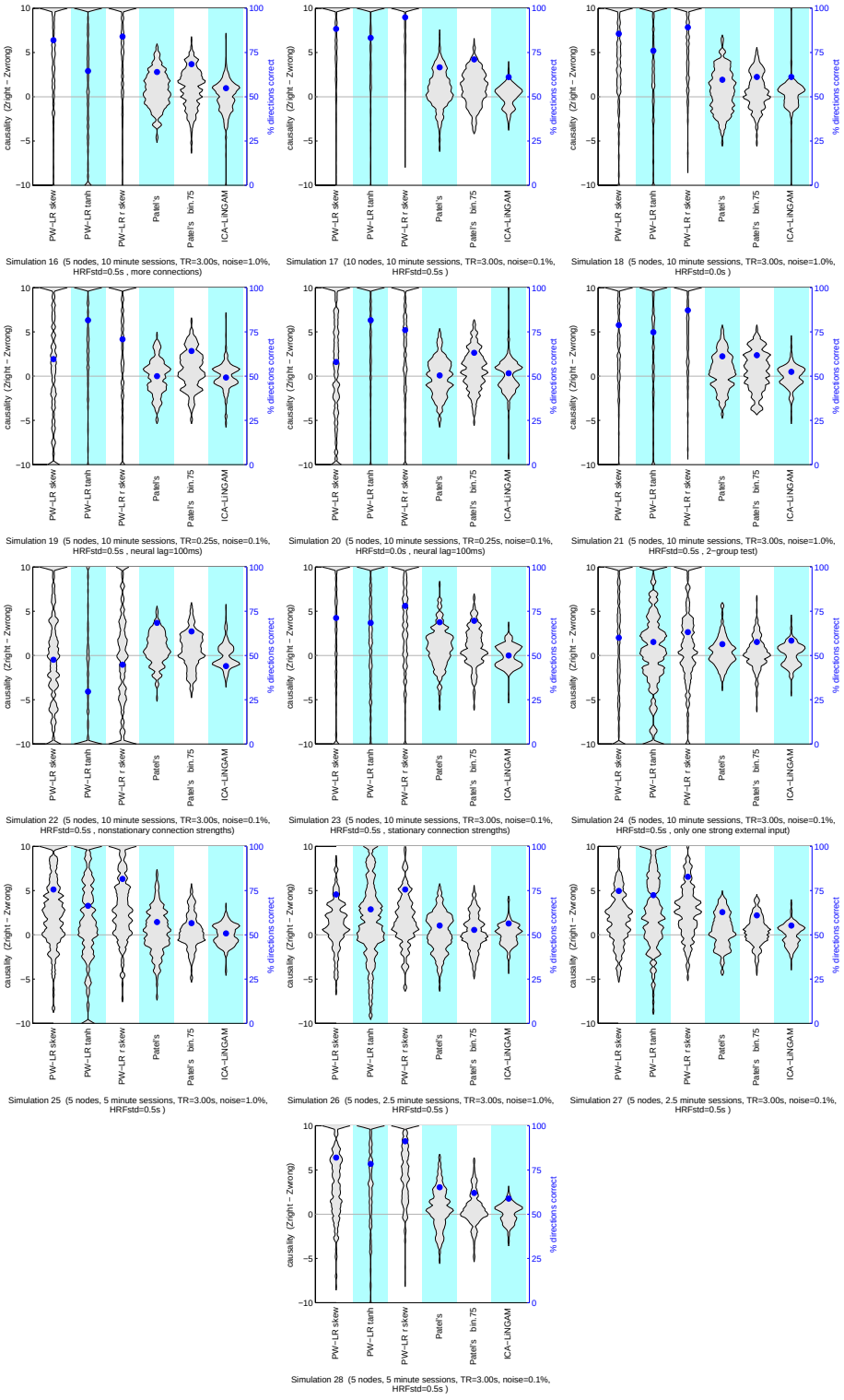


Figure 16: Results on simulated fMRI data, second half. See page 146 for caption.

Method	% correct
mxnt	50.6
hsic	61.7
nlme	53.1
mad	69.1

Table 2: Nonlinear models applied on the Tübingen-UCI data set. The algorithms are "nlme", the nonlinear likelihood ratio with maximum entropy approximation; "mad", the approximation using mean absolute deviations. For comparison: "hsic", the original nonlinear method using the HSIC measure of independence (Hoyer et al., 2009); and "mxnt", the linear method using the maximum entropy approximation.

made robust). The maximum entropy approximation might be greatly improved if the estimation of the variances used in the maximum entropy objective were made robust against outliers. Furthermore, both of our methods might be improved if the nonlinear fitting used a robust criterion instead of least squares.

9. Conclusion

We proposed very simple measures of the pairwise causal direction based on likelihood ratio tests and their approximations. We started with general measures based on entropy approximation which can accommodate different kinds of distributions, in Equations (2) and (3). Assuming that prior knowledge is available, we can develop more specific methods; for sparse variables we propose (5) and for skewed variables (17). We further showed how the measures can be extended to cyclic and nonlinear models. The different measures are recapitulated in Table 3.

We also showed how the pairwise measures can be used to estimate the whole Bayesian network in two ways. This is possible either in the DirectLiNGAM framework, or by a two-stage method based on first estimating the existence of the connections and then orienting them using the pairwise measures.

We also proposed a cumulant-based version of the nonlinear correlations. It was shown that the cumulant gives the correct pairwise direction. This shows the utility of using cumulants in theoretical analysis, and gives an intuitive interpretation of a new kind of cumulant. The cumulant-based analysis also indicated the noise-robustness of the nonlinear correlation methods, which was confirmed in the simulations. However, in practice the cumulant-based methods may suffer from sensitivity to outliers and thus their utility may be mainly in theoretical analysis.

The proposed measures seem to be particularly useful in the case where the number of data points is small compared to the dimension of the data, or the data is noisy. In such a case, the statistical performance of our methods is clearly superior to ICA-based LiNGAM and, to a lesser extent, DirectLiNGAM. The new methods are also computationally much faster than DirectLiNGAM. The importance of estimating causal networks with few data points has been recently highlighted by Smith et al. (2011) in the context of brain imaging. In fact, applied to the simulations by Smith et al. (2011), the new pairwise measures were clearly better then the methods originally tested.

Proposed measures for linear acyclic model (LiNGAM)			
Assumptions on non-Gaussianity	None	Sparse	Skewed
Equation for main new pairwise measure	(2) with (3)	(5)	(17)
Equation for previous measure by Dodge and Rousson	—	—	(14)

Proposed measures for extensions of LiNGAM		
Extension type	Cyclic case	Nonlinear case
Equation for new pairwise measure	Any LiNGAM measure	(23) with (3); or (24)

Table 3: The pairwise measures proposed in this paper recapitulated. The new cumulant-based approximations (Equations 6 and 10) have been omitted since they are mainly for theoretical analysis and not for practical use. Some of the measures by Dodge and Rousson (2001); Dodge and Yadegari (2010) would otherwise fit the "sparse" category but they assume the disturbances to be Gaussian and are thus less general. In the cyclic case, no new pairwise measures were introduced and it was merely proposed that the LiNGAM measures can be directly used even in the cyclic case. Of the measures highlighted above, the sparse-LiNGAM measure in Equation (5) was proposed in an earlier report on this work (Hyvärinen, 2010) while all others are new.

Thus, when estimating the LiNGAM model, it may be important to choose a suitable algorithm depending on data dimension, sample size, noise level, the distributions of the external influences, and other relevant factors.

Basic code for the pairwise measures is distributed on the Internet.[10]

Acknowledgments

We are grateful to Shohei Shimizu and Patrik Hoyer for deep and insightful comments on the manuscript, as well as to Christian Beckmann and Mark Woolrich for interesting discussions. We would also like to thank an anonymous referee for improving the derivation in Section 5.3. This work was supported by Academy of Finland, Computational Science Program and the Finnish Centre-of-Excellence in Algorithmic Data Analysis.

References

R.B. Buxton, E.C. Wong, and L.R. Frank. Dynamics of blood flow and oxygenation changes during brain activati on: the balloon model. *Magnetic Resonance in Medicine*, 39:855–864, 1998.

C. Chang, M.E. Thomason, and G.H. Glover. Mapping and correction of vascular hemodynamic latency in the BOLD signal. *NeuroImage*, 43:90–102, 2008.

10. Code can be found at `http://www.cs.helsinki.fi/u/ahyvarin/code/pwcausal/`.

Z. Chen and L. Chan. Causal discovery for linear non-gaussian acyclic models in the presence of latent gaussian confounders. In *Proc. Int. Conf. on Latent Variable Analysis and Signal Separation*, pages 17–24, 2012.

P. Comon. Independent component analysis—a new concept? *Signal Processing*, 36:287–314, 1994.

P. Daniušis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. In *Proc. 26th Conference on Uncertainty in Artificial Intelligence (UAI2010)*, 2010.

Y. Dodge and V. Rousson. On asymmetric properties of the correlation coefficient in the regression setting. *The American Statistician*, 55:51–54, 2001.

Y. Dodge and I. Yadegari. On direction of dependence. *Metrika*, 72:139–150, 2010.

K. J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302, 2003.

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*, volume 20. MIT Press, 2008.

D.A. Handwerker, J.M. Ollinger, and M. D'Esposito. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *NeuroImage*, 21:1639–1651, 2004.

P. O. Hoyer, A. Hyvärinen, R. Scheines, P. Spirtes, J. Ramsey, G. Lacerda, and S. Shimizu. Causal discovery of linear acyclic models with arbitrary distributions. In *Proc. 24th Conf. on Uncertainty in Artificial Intelligence (UAI2008)*, pages 282–289, Helsinki, Finland, 2008.

P. O. Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, volume 21, pages 689–696. MIT Press, 2009.

A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10, pages 273–279. MIT Press, 1998.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen. Pairwise measures of causal direction in linear non-gaussian acyclic models. In *Proc. Asian Conf. on Machine Learning, JMLR W&CP*, volume 13, pages 1–16, Tokyo, Japan, 2010.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.

A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *J. of Machine Learning Research*, 11:1709–1731, 2010.

J. Karvanen and V. Koivunen. Blind separation methods based on pearson system and its extensions. *Signal Processing*, 82(4):663–573, 2002.

J. M. Mooij, O. Stegle, D. Janzing, K. Zhang, and B. Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS*2010)*, pages 1687–1695, 2010.

R.S. Patel, F.D. Bowman, and J.K. Rilling. A Bayesian approach to determining connectivity of the human brain˙ *Human Brain Mapping*, 27:267–276, 2006.

D.-T. Pham and P. Garrat. Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. on Signal Processing*, 45(7):1712–1725, 1997.

J. D. Ramsey, S. J. Hanson, and C. Glymour. Multi-subject search correctly identifies causal connections and most causal directions in the DCM models of the Smith et al. simulation study. *NeuroImage*, 58(3):838–848, 2011.

S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *J. of Machine Learning Research*, 7:2003–2030, 2006.

S. Shimizu, A. Hyvärinen, Y. Kawahara, and T. Washio. A direct method for estimating a causal ordering in a linear non-gaussian acyclic model. In *Proc. 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, pages 506–513, Montréal, Canada, 2009.

S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvärinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen. DirectLiNGAM: A direct method for learning a linear non-gaussian structural equation model. *J. of Machine Learning Research*, 12:1225–1248, 2011.

S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fMRI. *NeuroImage*, 54:875–891, 2011.

Y. Sogawa, S. Shimizu, Y. Kawahara, and T. Washio. An experimental comparison of linear non-gaussian causal discovery methods and their variants. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN2010)*, Barcelona, Spain, 2010.

Y. Sogawa, S. Shimizu, A. Hyvärinen, T. Washio, T. Shimamura, and S. Imoto. Estimating exogenous variables in data with more variables than observations. *Neural Networks*, 24(8):875–880, 2011.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proc. 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, pages 647–655, Montréal, Canada, 2009.

Journal of Machine Learning Research 14 (2013) 153-186 Submitted 6/11; Revised 7/12; Published 1/13

Universal Consistency of Localized Versions of Regularized Kernel Methods

Robert Hable ROBERT.HABLE@UNI-BAYREUTH.DE
Department of Mathematics
University of Bayreuth
D-95440 Bayreuth, Germany

Editor: Gábor Lugosi

Abstract

In supervised learning problems, global and local learning algorithms are used. In contrast to global learning algorithms, the prediction of a local learning algorithm in a testing point is only based on training data which are close to the testing point. Every global algorithm such as support vector machines (SVM) can be localized in the following way: in every testing point, the (global) learning algorithm is not applied to the whole training data but only to the k nearest neighbors (kNN) of the testing point. In case of support vector machines, the success of such mixtures of SVM and kNN (called SVM-KNN) has been shown in extensive simulation studies and also for real data sets but only little has been known on theoretical properties so far. In the present article, it is shown how a large class of regularized kernel methods (including SVM) can be localized in order to get a universally consistent learning algorithm.

Keywords: machine learning, regularized kernel methods, localization, SVM, k-nearest neighbors, SVM-KNN

1. Introduction

In a supervised learning problem, the goal is to predict the value y of an unobserved output variable Y after observing the value x of an input variable X. A predictor is a function f which maps the observed input value x (called testing data point) to a prediction $f(x)$ of the unobserved output value y. Choosing a predictor $f = f_{D_n}$ is done on base of previously observed data $D_n = \big((x_1,y_1),\ldots,(x_n,y_n)\big)$ (called trainig data). A learning algorithm is a function $D_n \mapsto f_{D_n}$ which maps training data D_n to a predictor f_{D_n}. Among the learning algorithms commonly used in machine learning, there are local and global algorithms. The most prominent example of a local algorithm is *k-nearest neighbors* (kNN). In case of a local algorithm $D_n \mapsto f_{D_n}$, the prediction $f_{D_n}(x)$ in a testing data point x is not based on the whole training data but only on those training data points (x_i,y_i) which are close to x. In case of a global algorithm, choosing a predictor f_{D_n} is based on a global criterion—such as (penalized) empirical risk minimization—and, accordingly, the prediction $f_{D_n}(x)$ in a point x can also be based on training data points (x_i,y_i) which are not close to x. Typical examples of global algorithms are regularized kernel methods such as *support vector machines* (SVM).

Global algorithms have disadvantages if the complexity of the optimal predictor varies for different areas of the input space. For example, in one part of the the input space, an optimal predictor might be a very simple function and, in another part, it might be a highly complex and volatile func-

tion. This is a problem for global algorithms because the complexity of the selected predictor f_{D_n} is usually regularized by one or several hyperparameters which are fixed for the whole input space. One way to overcome this problem is to separate the input space into several parts in a first step and to separately use a global algorithm for each of the separated parts. For example, the input space is separated by use of decision trees and then SVMs are separately applied on the separated parts of the input space; see, for example, Bennett and Blue (1998), Wu et al. (1999), and Chang et al. (2010). Another possibility is to "localize" a global algorithm. This can be done in the following way: (1) select a few training data points which are close to the testing data point, (2) determine a predictor based on the selected training data points by use of a (global) learning algorithm, and (3) calculate the prediction in the testing data point. A number of algorithms which have been suggested in the literature can be described in this way. These algorithms only differ in the way how data points are selected in (1) and which learning algorithm is used in (2). An early investigation of such methods is Bottou and Vapnik (1992) and Vapnik and Bottou (1993). A number of recent articles apply such an approach to support vector machines (SVM). That is, SVM is used in (2), but there are differences in (1): In Zhang et al. (2006), data points are selected in the same way as for kNN. That is, the prediction in a testing point x is given by that SVM which is calculated based on the k_n training points which are nearest to x; the natural number k_n acts as a hyperparameter. In order to decide which training points are the k_n closest ones to x, a metric on the input space is needed. Zhang et al. (2006) considers different metrics. As this approach is a mixture between kNN and SVM, it is called SVM-KNN. Independently, a similar approach has been developed by E. Blanzieri and others. The main difference to Zhang et al. (2006) is that distances (for selecting the k_n nearest neighbors) are not measured in the input space but in the feature space (i.e., in the RKHS associated with the kernel of the SVM). This approach has been extensively studied in experimental comparisons in Blanzieri and Bryl (2007a), Blanzieri and Bryl (2007b), Segata and Blanzieri (2009) and Blanzieri and Melgani (2008) where the latter publication also derives a local bound on the generalization error. Another slightly different approach is developed in Cheng et al. (2007) and Cheng et al. (2010). There, data points are not selected according to a fixed number k_n of nearest neighbors as in kNN; instead, those training data points are selected which are contained in a fixed neighborhood about the testing point x. That is, not the number of testing points in the neighborhood is fixed (as in kNN), but the area of the neighborhood is fixed. In addition, it is also possible to downweight testing points depending on their distance to the testing point x.

Though all of these approaches have been extensively studied on simulated and real-world data and their success has experimentally been shown, only little is known on theoretical properties so far. In this article, it is shown that some SVM-KNN approaches are universally consistent. Though the above cited approaches only consider SVMs for classification (using the hinge loss) and linear kernels, the following theoretical investigation allows for a large class of loss functions and kernels. That is, not only SVMs but also general regularized kernel methods are considered for classification and regression as well. Here, k_n nearest neighbors are selected by use of the ordinary Euclidean metric on the input space $\mathcal{X} \subset \mathbb{R}^p$ so that this approach is closest to Zhang et al. (2006). All methods based on a kNN approach are faced with the problem of distance ties. This means that, in general, the set of the k_n nearest neighbors to a testing point x is not necessarily unique because different testing points might have the same distance to x. In case of distance ties, a number of tie-breaking strategies have been suggested in the literature; see, for example, Devroye et al. (1994, § 1). E.g. a simple tie-breaking strategy is to generate artificial additional covariates $U_1, \ldots, U_n$ i.i.d. from the uniform distribution on $[0, \varepsilon]$ for some small $\varepsilon > 0$. Then, for the new input variables $X_i' := (X_i, U_i)$,

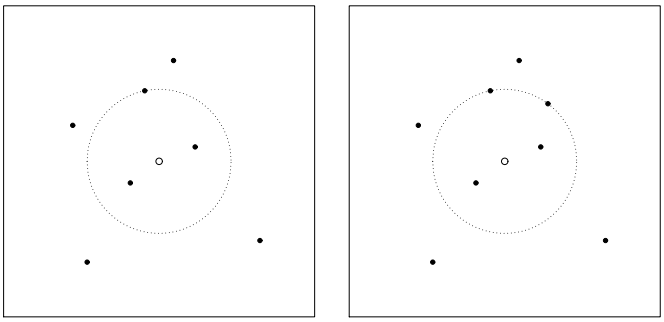


Figure 1: Neighborhood (dotted circle) determined by the k nearest neighbors of a testing point (empty point) for $k = 3$. The left figure shows a situation without distance ties at the border of the neighborhood (dotted circle). The right figure shows a situation with distance ties at the border of the neighborhood (empty point): only one of the two data points (filled points) at the border may belong to the $k = 3$ nearest neighbors; choosing between these two candidates is done by randomization here.

distance ties only occur with zero probability. The drawback of this method is that ε has to be chosen in advance and, in particular if ε is not small enough, this tie-breaking strategy changes the results even if there are no distance ties. Therefore, we use a different strategy where, in case of a distance tie, the k nearest neigbors are chosen by randomization; see Figure 1. Technically, this is done by artificially generated covariates $U_1, \ldots, U_n$ i.i.d. from the uniform distribution on $[0,1]$ where—in contrast to the simple tie-breaking strategy mentioned above—U_i is only taken into account in case of a distance tie in X_i.

It has to be pointed out that the approach of this article differs from the one in Zakai and Ritov (2009); see also Zakai (2008). There, it is shown that every consistent learning algorithm is in a sense localizable. On the one hand, this is of great theoretical importance because, roughly speaking, it says that global methods as SVMs asymptotically act like local methods. On the other hand, this also shows that any consistent method can be localized in a way so that the local version is again consistent. By a superficial inspection of these results, one might suggest that, essentially, this would already show consistency of any localized method such as SVM-KNN. However, this is not the case and these results cannot be used offhand in order to prove consistency of SVM-KNN: Firstly, the way how the methods are localized completely differ. In Zakai and Ritov (2009), localizing is not done by fixed numbers k_n of nearest neighbors (as in kNN and SVM-KNN) but by fixed sizes (radii) R_n of neighborhoods (similar as in Cheng et al. (2010)). Using fixed sizes (radii) of neighborhoods is more convenient for theoretical investigations because whether a data point x_{i_0} lies in such a neighborhood only depends on this data point; that is, variables indicating whether data points belong to such a neighborhood are i.i.d. In contrast, whether a data point x_{i_0} belongs to the k_n nearest neighbors depends on the whole sample; that is, the corresponding indicator variables are not independent and one has to work with random sets of indexes. In particular, the kNN-approach leads to random sizes of neighborhoods which depend on the testing point x while Zakai and Ritov (2009) deal with deterministic sequences of radii R_n which do not depend on the testing point x.

Secondly, due to the generality of the investigation in Zakai and Ritov (2009), it is only shown there that a (deterministic) sequence of radii R_n *exists* such that a suitably,[1] localized method is consistent. This indicates that looking for consistent localized methods may be promising; however, for practical purposes, mere existence is not enough and one also has to know how to choose such entities like R_n in order to get a consistent method. In the special case of SVM-KNN, the main result of the present article precisely specifies possible choices of all involved entities (hyperparameters etc.) which guarantee consistency.

For kNN, consistency requires that the number of selected neighbors k_n goes to infinity but not too fast for $n \to \infty$. Clearly, this will also be crucial for SVM-KNN but, now, an additional difficulty arises: the calculation of the SVM (or any other regularized kernel method) depends on a regularization parameter λ_n which determines to what extend the complexity of a predictor is penalized (in order to avoid overfitting). Consistency of SVMs is only guaranteed if λ_n converges to 0 but not too fast. Accordingly, in case of SVM-KNN, the interplay between the convergence of k_n and the convergence of λ_n is crucial. Theorem 1 below gives precise conditions on k_n and λ_n which guarantee consistency of SVM-KNN. In Therorem 1, it is assumed that k_n, $n \in \mathbb{N}$, is a predefined deterministic sequence. The regularization parameters $\lambda_n = \lambda_{D_n,x}$ are based on the training data and can, to some extend, also be chosen in a data-driven way, for example, by cross-validation. In addition, the choice of the regularization parameter is local, that is, depends on the testing point x. This enables a local regularization of the complexity of the predictor which is an important motivation for localizing a global algorithm as already stated above.

Local approaches such as SVM-KNN are computationally very efficient if the number of testing points is small. However, if the number of testing points is large, then such methods are burdened with high computational costs of the testing phase. Therefore, variants of SVM-KNN have been proposed in Cheng et al. (2007) and Segata and Blanzieri (2010). For example, in Segata and Blanzieri (2010), the computational complexity is reduced by the following modification: the SVM is not calculated on base of the k-nearest neighbors of the testing point but on base of the k-nearest neighbors of a certain training point which is close to the testing point. In this way, only a relatively small number of SVMs has to be calculated. If k is reasonable small (and fixed), then training scales as $O(n\log(n))$ and testing scales as $O(\log(n))$ in the number of training points.

The article is organized as follows: Section 2 recalls the precise mathematical definitions of kNN, regularized kernel methods (in particular, SVM) and SVM-KNN as investigated here. Section 3 contains the main result, that is, consistency of SVM-KNN, Section 4 investigates an illustrative example and Section 5 contains some concluding remarks. All proofs and auxiliary results are given in the Appendix.

2. Setup: kNN, SVM and SVM-KNN

Let $(\Omega, \mathcal{A}, Q)$ be a probability space, let $\mathcal{X}$ be an open subset of $\mathbb{R}^d$, and let $\mathcal{Y}$ be a closed subset of $\mathbb{R}$. For any (topological) space $\mathcal{W}$, its Borel-σ-algebra is denoted by $\mathfrak{B}_{\mathcal{W}}$. Let

$$X_1, \ldots, X_n : (\Omega, \mathcal{A}, Q) \longrightarrow \left(\mathcal{X}, \mathfrak{B}_{\mathcal{X}}\right) \quad \text{and} \quad Y_1, \ldots, Y_n : (\Omega, \mathcal{A}, Q) \longrightarrow \left(\mathcal{Y}, \mathfrak{B}_{\mathcal{Y}}\right)$$

be random variables such that $(X_1, Y_1), \ldots, (X_n, Y_n)$ are independent and identically distributed according to some unknown probability measure P on $\left(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}}\right)$. In order to find a prediction

1. In Zakai and Ritov (2009), localizing also involves a smoothing operation around the testing point.

$y = f(\xi)$ for a point $\xi \in \mathcal{X}$, a kNN-rule is based on the k_n nearest neighbors of ξ. The k_n nearest neighbors of $\xi \in \mathbb{R}^p$ within $x_1, \ldots, x_n \in \mathbb{R}^p$ are given by an index set $I \subset \{1, \ldots, n\}$ such that

$$\sharp(I) = k_n \qquad \text{and} \qquad \max_{i \in I} |x_i - \xi| \; < \; \min_{j \notin I} |x_j - \xi| \; . \tag{1}$$

However, in case of distance ties, some observations x_i and x_j have the same distance to ξ (i.e., $|x_i - \xi| = |x_j - \xi|$) so that the k_n nearest neighbors are not unique and an index set I as defined above does not exist. In order to break distance ties, we use randomization (see also Figure 1) as done in (Devroye et al., 1994, p. 1373f): We artificially generate data from random variables $U_1, \ldots, U_n$ which are uniformly distributed on $(0,1)$ and such that $(X_1, Y_1), \ldots, (X_n, Y_n), U_1, \ldots, U_n$ are independent. Define $Z_i := (X_i, U_i)$ for every $i \in \{1, \ldots, n\}$. That is, we observe $(Z_1, Y_1), \ldots, (Z_n, Y_n)$ now. Define

$$\mathbf{D}_n \; := \; \big((Z_1, Y_1), \ldots, (Z_n, Y_n)\big) \qquad \forall\, n \in \mathbb{N} \; .$$

We say that $z_i = (x_i, u_i)$ is (strictly) closer to $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$ than $z_j = (x_j, u_j)$ if $|x_i - \xi| < |x_j - \xi|$; and, in case of a distance tie $|x_i - \xi| = |x_j - \xi|$, we say that $z_i = (x_i, u_i)$ is (strictly) closer to $\zeta = (\xi, u)$ than $z_j = (x_j, u_j)$, if $|u_i - u| < |u_j - u|$. That is, we use some kind of a lexicographic order which guarantees that nothing changes if there are no distance ties. Note that there can also be distance ties for the u_i but these only occur with zero probability. The following is a precise definition of "nearest neighbors" which also takes into account distance ties in the x_i and the u_i. For $n \in \mathbb{N}$, let $k_n \in \{1, \ldots, n\}$. Take any $z_1 = (x_1, u_1), \ldots, z_n = (x_n, u_n), \zeta = (\xi, u) \in \mathbb{R}^p \times (0,1)$ such that there is a $\tau_n(z_1, \ldots, z_n, \zeta) = I \subset \{1, \ldots, n\}$ such that

$$\sharp(I) = k_n \, , \quad \max_{i \in I} |x_i - \xi| \leq \min_{i \notin I} |x_i - \xi| \quad \text{and} \quad \max_{j \in I \cap \mathcal{J}} |u_j - u| \; < \; \min_{j \in \mathcal{J} \setminus I} |u_j - u| \tag{2}$$

where

$$\mathcal{J} \; = \; \Big\{ j \in \{1, \ldots, n\} \Big| \; |x_j - \xi| = \max_{i \in I} |x_i - \xi| \Big\} \; . \tag{3}$$

If such a set $\tau_n(z_1, \ldots, z_n, \zeta) = I$ exists, it is unique. If it does not exist, there are also distance ties in the u_i and we arbitrarily define $\tau_n(z_1, \ldots, z_n, \zeta) := \{1, \ldots, k_n\}$ in this case. Since distance ties in the u_i occur with zero probability, the definition of $\tau_n(z_1, \ldots, z_n, \zeta)$ is meaningless in this case; it is only important to assure measurability of $\tau_n : (z_1, \ldots, z_n, \zeta) \mapsto \tau_n(z_1, \ldots, z_n, \zeta)$; see Appendix B. So, definition (2) and (3) is a modification of (1) in order to deal with distance ties in the x_i. Note that, due to the lexicographic order, the values u_i and u are only relevant in case of distance ties (at the border of the neighborhood given by the k_n nearest neighbors).

Next, define

$$I_{n,\zeta}(\omega) \; := \; \tau_n\big(Z_1(\omega), \ldots, Z_n(\omega), \zeta\big) \qquad \forall\, \omega \in \Omega \, , \quad \forall\, \zeta \in \mathbb{R}^p \times (0,1) \; . \tag{4}$$

That is, $I_{n,\zeta}$ contains the indexes of the k_n-nearest neighbors of ζ. Let $i_1 < i_2 < \ldots < i_{k_n}$ be the (ordered) elements of $I_{n,\zeta}$. Then, the vector of the k_n-nearest neighbors is

$$\mathbf{D}_{n,\zeta} \; := \; \big((Z_{i_1}, Y_{i_1}), \ldots, (Z_{i_{k_n}}, Y_{i_{k_n}})\big) \; . \tag{5}$$

The prediction of the ordinary kNN-rule in ξ is given by the mean

$$\frac{1}{k_n} \sum_{i \in I_{n,\zeta}} Y_i \; .$$

The SVM-KNN method replaces the mean by an SVM. To this end, we recall the definition of SVMs; here, the term "SVM" is used in a wide sense which covers many regularized kernel-based learning algorithms for classification and regression as well; see, for example, Steinwart and Christmann (2008) for these methods.

A measurable map $L : \mathcal{Y} \times \mathbb{R} \to [0,\infty)$ is called *loss function*. A loss function L is called *convex* loss function if it is convex in its second argument, that is, $t \mapsto L(y,t)$ is convex for every $y \in \mathcal{Y}$. The *risk* of a measurable function $f : \mathcal{X} \to \mathbb{R}$ is defined by

$$\mathcal{R}_P(f) \;=\; \int_{\mathcal{X}\times\mathcal{Y}} L\big(y, f(x)\big)\, P\big(d(x,y)\big)\,.$$

The goal is to estimate a function $f : \mathcal{X} \to \mathbb{R}$ which minimizes this risk. The estimates obtained from the method of support vector machines are elements of so-called reproducing kernel Hilbert spaces (RKHS) H. An RKHS H is a certain Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ which is generated by a *kernel* $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. See, for example, Schölkopf and Smola (2002) or Steinwart and Christmann (2008) for details about these concepts.

Let H be such an RKHS. Then, the *regularized risk* of an element $f \in H$ is defined to be

$$\mathcal{R}_{P,\lambda}(f) \;=\; \mathcal{R}_P(f) + \lambda\|f\|_H^2\,, \qquad \text{where} \quad \lambda \in (0,\infty)\,.$$

An element $f \in H$ is called a *support vector machine* (SVM) and denoted by $f_{P,\lambda}$ if it minimizes the regularized risk in H . That is,

$$\mathcal{R}_P(f_{P,\lambda}) + \lambda\|f_{P,\lambda}\|_H^2 \;=\; \inf_{f\in H}\big(\mathcal{R}_P(f) + \lambda\|f\|_H^2\big)\,. \tag{6}$$

The *empirical SVM* $f_{D_n,\lambda_{D_n}}$ is that function $f \in H$ which minimizes

$$\frac{1}{n}\sum_{i=1}^{n} L\big(y_i, f(x_i)\big) + \lambda_{D_n}\|f\|_H^2$$

in H for the data $D_n = ((x_1,y_1),\ldots,(x_n,y_n)) \in (\mathcal{X}\times\mathcal{Y})^n$ and a regularization parameter $\lambda_{D_n} \in (0,\infty)$ which is chosen in a data-driven way (e.g., by cross-validation) in applications so that it typically depends on the data. The empirical support vector machine $f_{D_n,\lambda_{D_n}}$ uniquely exists for every $\lambda_{D_n} \in (0,\infty)$ and every data-set $D_n \in (\mathcal{X}\times\mathcal{Y})^n$ if $t \mapsto L(y,t)$ is convex for every $y \in \mathcal{Y}$.

The prediction of the SVM-KNN learning algorithm in $\zeta = (\xi,u) \in \mathcal{X}\times(0,1)$ is given by $f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)$ with

$$f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}} \;=\; \arg\min_{f\in H}\left(\frac{1}{k_n}\sum_{i\in I_{n,\zeta}} L\big(Y_i, f(X_i)\big) + \Lambda_{n,\zeta}\|f\|_H^2\right) \tag{7}$$

where $\omega \mapsto \Lambda_{n,\zeta}(\omega)$ is a random regularization parameter depending on n and ζ. That is, the method calculates the empirical SVM $f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}$ for the k_n nearest neighbors (given by the index set $I_{n,\zeta}$) and uses the value $f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)$ for the prediction in ζ. The empirical SVM minimizes the regularized empirical risk where the regularization is done in order to avoid overfitting. Note that—unlike most theoretical investigations on SVMs—the regularization parameter $\Lambda_{n,\zeta}$ is random and, here, also the index set $I_{n,\zeta}$ is random, that is, a set-valued random variable. We will assume that $\mathcal{Y} \subset [-M,M]$ for

some M so that the SVM-KNN can be clipped. The clipped version of the SVM-KNN is denoted by

$$\widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi) = \begin{cases} M & f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi) > M \\ f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi) & \text{if} \quad f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi) \in [-M,M] \\ -M & f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi) < -M \end{cases} . \tag{8}$$

This means that we change the prediction to M (or $-M$) if $f_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)$ is larger (or smaller) than M (or $-M$). As we will assume that $\mathcal{Y} \subset [-M,M]$, predictions exceeding $[-M,M]$ are not sensible and, in these cases, clipping obviously improves the accuracy of our predictions.

3. Main Result

This section contains the main result, namely universal consistency of SVM-KNN where the term "SVM" is used in a broad sense. Instead of just SVMs in the original sense (i.e., classification using the hinge loss), a large class of regularized kernel methods for classification and regression as well is covered. However, as already mentioned in the introduction, not any combination of SVM and kNN is possible. In order to get consistency, the choice of the number of neighbors k_n and the data-driven local choice of the regularization parameter $\lambda = \Lambda_{n,\xi}$ needs some care. The following settings guarantee consistency of SVM-KNN. Possible choices for k_n and λ_n are, for example, $k_n = b \cdot n^{0.75}$ for $b \in (0,1]$ and $\lambda_n = a \cdot n^{-0.15}$ for $a \in (0,\infty)$, $n \in \mathbb{N}$.

Settings: Choose a sequence $k_n \in \mathbb{N}$, $n \in \mathbb{N}$, such that

$$k_1 \le k_2 \le k_3 \le \ldots \le \lim_{n\to\infty} k_n = \infty \qquad \text{and} \qquad \frac{k_n}{n} \searrow 0 \quad \text{for } n \to \infty,$$

and a sequence $\lambda_n \in (0,\infty)$, $n \in \mathbb{N}$, such that

$$\lim_{n\to\infty} \lambda_n = 0 \qquad \text{and} \qquad \lim_{n\to\infty} \lambda_n^{\frac{3}{2}} \cdot \frac{k_n}{\sqrt{n}} = \infty \tag{9}$$

and a constant $c \in (0,\infty)$, and a sequence $c_n \in [0,\infty)$ such that $\lim_{n\to\infty} c_n/\sqrt{\lambda_n} = 0$. For every $\zeta = (\xi,u) \in \mathcal{X} \times (0,1)$, define

$$\tilde{\Lambda}_{n,\zeta} = \frac{1}{k_n} \sum_{i \in I_{n,\zeta}} |X_i - \xi|^{\frac{3}{2}}$$

and choose random regularization parameters $\Lambda_{n,\zeta}$ such that

$$\mathcal{X} \times (0,1) \times \Omega \to (0,\infty), \qquad (\xi,u,\omega) = (\zeta,\omega) \mapsto \Lambda_{n,\zeta}(\omega)$$

is measurable and

$$c \cdot \max\left\{\lambda_n, \tilde{\Lambda}_{n,\zeta}\right\} \le \Lambda_{n,\zeta} \le (c + c_n) \cdot \max\left\{\lambda_n, \tilde{\Lambda}_{n,\zeta}\right\} \qquad \forall \zeta \in \mathcal{X} \times (0,1) . \tag{10}$$

Let the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be continuously differentiable, bounded, and such that its RKHS H is non-degenerated in the following sense:

$$\text{for every } x \in \mathcal{X} \text{ there is an } f \in H \text{ such that } f(x) \neq 0 . \tag{11}$$

Theorem 1 *Let $\mathcal{X} \subset \mathbb{R}^p$ be an open subset and let $\mathcal{Y} \subset [-M,M]$ be closed. Let $L : [-M,M] \times \mathbb{R} \to [0,\infty)$ be a convex loss function with the following local Lipschitz property: there are some $b_0, b_1 \in (0,\infty)$ and $q \in [0,1]$ such that, for every $a \in (0,\infty)$,*

$$\sup_{y \in [-M,M]} \bigl|L(y,t_1) - L(y,t_2)\bigr| \;\le\; |L|_{a,1} \cdot |t_1 - t_2| \qquad \forall t_1, t_2 \in [-a,a] \tag{12}$$

for $|L|_{a,1} = b_0 + b_1 a^q$. In addition, assume that there is an increasing function $\ell : [0,\infty) \to [0,\infty)$ such that $\lim_{s \to 0} \ell(s) = 0$ and

$$\sup_{t \in [-M,M]} \bigl|L(y_1,t) - L(y_2,t)\bigr| \;\le\; \ell\bigl(|y_1 - y_2|\bigr) \qquad \forall y_1, y_2 \in [-M,M]\,. \tag{13}$$

Assume that $(X_1,Y_1),\ldots,(X_n,Y_n)$ are independent and identically distributed according to some unknown probability measure P on $\bigl(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}}\bigr)$ and let $U_1,\ldots,U_n$ be uniformly distributed on $(0,1)$ such that $(X_1,Y_1),\ldots,(X_n,Y_n),U_1,\ldots,U_n$ are independent.

Then, every SVM-KNN defined by (7,8) according to the above settings and clipped at M,

$$f_{\mathbf{D}_n} : \quad \zeta = (\xi, u) \;\mapsto\; \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)$$

is risk-consistent, that is,

$$\mathcal{R}_P\bigl(f_{\mathbf{D}_n}\bigr) \xrightarrow[n\to\infty]{} \inf_{\substack{f:\mathcal{X}\to\mathbb{R} \\ \text{measurable}}} \mathcal{R}_P(f) \;=:\; \mathcal{R}_P^* \qquad \text{in probability.}$$

Essentially all commonly used loss functions satisfy assumptions (12) and (13): for example, the hinge loss and the logistic loss for classification, the ε-insensitive loss, the least squares loss, the absolute deviation loss, and the Huber loss for regression, and the pinball loss for quantile regression.

The property (11) of a nowhere degenerated RKHS H is a very weak property and replaces strong denseness properties of H which are typically needed in order to assure universal consistency of SVMs.

The settings include a data-driven local choice of the regularization parameter $\lambda = \Lambda_{n,\zeta}$. Here, "local" means that $\Lambda_{n,\zeta}$ depends on the testing point ζ. This is preferable because, in this way, it is possible to allow for different degrees of complexity on different areas of the input space. As already mentioned in the introduction, this is an important motivation for "localizing" a global algorithm. A simple rule of thumb for choosing $\Lambda_{n,\zeta}$ is to predefine a fixed $c \in (0,\infty)$ and use

$$\Lambda_{n,\zeta} \;=\; c \cdot \max\bigl\{\lambda_n, \tilde{\Lambda}_{n,\zeta}\bigr\}\,. \tag{14}$$

The deterministic λ_n prevents the regularization parameters from decreasing to 0 too fast and (9) controls the interplay between k_n and λ_n. (Recall that it is well known that classical SVMs are not consistent if the regularization parameters decrease to 0 too fast.) Note that the calculation of $\tilde{\Lambda}_{n,\zeta}$ is computationally fast as $I_{n,\zeta}$ (the index set of the k_n nearest neighbors) has to be calculated anyway. The behavior of $\tilde{\Lambda}_{n,\zeta}$ is reasonable: if the k_n nearest neighbors are relatively close to the testing point ζ, then $\tilde{\Lambda}_{n,\zeta}$ is relatively small which is favorable because this means that relatively many training points are close to ζ so that the predictor should be allowed to be relatively complex around ζ. Nevertheless, the rule of thumb suggested in (14) will not satisfactorily capture different

degrees of complexity in most cases. Then, it is possible to choose the regularization parameter on base of a (restricted) cross-validation or any other method for selecting the hyperparameter: choose a (very) small $c \in (0,\infty)$ and a (very) large $C \in (0,\infty)$, define $c_n := C\sqrt{\lambda_n}/\ln(n)$ and make sure that your selection method (e.g., cross validation) only picks a value from the interval

$$\left[c \cdot \max\left\{\lambda_n, \tilde{\Lambda}_{n,\zeta}\right\}, (c+c_n) \cdot \max\left\{\lambda_n, \tilde{\Lambda}_{n,\zeta}\right\}\right].$$

As it is assumed in Theorem 1 that $\lim_{n\to\infty} k_n/n = 0$ (i.e., the fraction of data points in the neighborhood diminishes), this SVM-KNN approach is rather a kNN-approach in which the simple (local) constant fitting is replaced by a more advanced (local) SVM fitting. That is, we follow a local modeling paradigm (see Györfi et al., 2002, § 2.1) just as done, for example, when generalizing the Nadaraya-Watson kernel estimator (constant fitting) to the local polynomial kernel estimator (polynomial fitting); for local polynomial fitting and the advantages of generalizing local constant fitting, see, for example, Fan and Gijbels (1996). In case of SVM-KNN, the advantage of generalizing constant fitting (kNN), has been demonstrated in extensive simulation studies in Zhang et al. (2006), Blanzieri and Bryl (2007a), Blanzieri and Bryl (2007b), Segata and Blanzieri (2009), and Blanzieri and Melgani (2008).

Instead, it would also be possible to assume that $\lim_{n\to\infty} k_n/n = 1$ so that the method (asymptotically) acts as an ordinary SVM. If convergence of the fraction k_n/n to 1 is fast enough, then universal consistency of such a method follows from universal consistency of SVM.

4. An Illustrative Example

It is commonly accepted in machine learning that there is no universally consistent learning algorithm which is always better than all other universally consistent learning algorithms and, for two different learning algorithms, there is always a situation in which one learning algorithm is better than the other one and there is also a situation in which it is the other way round; see, for example, (Devroye et al., 1996, § 1). The goal of this section is to illustrate where localizing SVMs provides some gain and where it does not. It has to be pointed out here that it is *not* the goal of this article or this section to empirically show the success of the SVM-KNN approach. This has previously been done; see the references cited in the introduction. The aim of this article is the proof of universal consistency and this section is only for illustrative purposes.

Let us consider the following model

$$Y_i \;=\; f_j(X_i) + \varepsilon_i\,, \qquad i \in \{1,\ldots,n\} \tag{15}$$

where, in the first scenario ($j = 1$), the regression function is given by

$$f_1(x) \;=\; 10(|x|-1)^2 \cdot \operatorname{sign}(x)\,, \qquad x \in [-1,1]$$

and, in the second scenario ($j = 2$), the regression function is given by

$$f_2(x) \;=\; 10x^2 \cdot \operatorname{sign}(x)\,, \qquad x \in [-1,1]\,.$$

As illustrated in Figure 2, the difference between f_1 and f_2 is that the parts of the functions on $(-1,0)$ and $(0,1)$ are interchanged. In both cases, $X_1,\ldots,X_n$ are i.i.d. drawn from the uniform distribution on $[-1,1]$ and $\varepsilon_1,\ldots,\varepsilon_n$ are i.i.d. drawn from $\mathcal{N}(0,\sigma^2)$ for $\sigma = 0.5$.

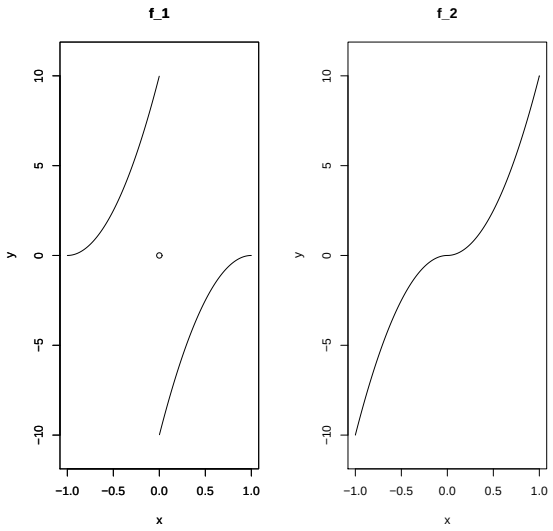


Figure 2: Graph of the regression functions $f_1(x) = 10(|x|-1)^2 \cdot \text{sign}(x)$ and $f_2(x) = 10x^2 \cdot \text{sign}(x)$ in model (15)

Classical SVMs, the localized version SVM-KNN, and classical kNN are applied to simulated data sets of size $n = 200$ for both scenarios each with 500 runs. In case of classical SVMs, the Gaussian RBF kernel $K_\gamma(x,x') = \exp(-\gamma(x-x')^2)$ and the ε-insensitive loss for $\varepsilon = 0.001$ are used. The hyperparameter γ is chosen by a five-fold cross validation among

$$0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1,\ 2,\ 5,\ 10,\ 15,\ 20,\ 30,\ \ 50,\ \ 75,\ 100,\ 150,\ 200,\ 250,\ 300,\ 350,\ 400,\ 500$$

and the regularization parameter is equal to $\lambda_n = a \cdot n^{-0.45}$ where a is chosen by a five-fold cross validation among

$$0.00001,\ 0.00005,\ 0.0001,\ 0.0005,\ 0.001,\ 0.005,\ 0.01,\ \ \ 0.05,\ 0.1,$$

The choice $\lambda_n = a \cdot n^{-0.45}$ is motivated by the fact that classical SVMs with the ε-insensitive loss are consistent if $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \lambda_n^2 n = \infty$; see (Christmann and Steinwart, 2007, Theorem 12). In case of SVM-KNN, the number of nearest neighbors is equal to $k_n = \lceil b \cdot n^{0.75} \rceil$ where the hyperparameter b is chosen by a five-fold cross validation among

```
0.15, 0.2 ,0.3 ,0.4 ,0.5 ,0.6 ,0.7 ,0.8 ,0.9 ,1 .
```

The exponent 0.75 for the definition of k_n is in accordance with the settings in Section 3. Choosing $k_n = \lceil b \cdot n^{0.75} \rceil$ would also guarantee universal consistency of classical kNN; see, for example, (Györfi et al., 2002, Theorem 6.1). For each testing point ξ, the prediction is calculated by a local SVM on the k_n nearest neighbor. For each local SVM, the polynomial kernel $K(x,x') = (x \cdot x' + 1)^3$ with degree 3 and the ε-insensitive loss for $\varepsilon = 0.001$ are used. In accordance with the settings in Section 3, the regularization parameter is equal to $\Lambda_{n,\xi} = C_{n,\xi} \max\left\{0.01 k_n^{-0.2}, \frac{1}{k_n} \sum_{i \in I_{n,\xi}} |x_i - \xi|^{1.5}\right\}$ where, for every ξ, the hyperparameter $C_{n,\xi}$ is chosen by a five-fold cross validation among

$$0.01,\ 0.1,\ 1,\ 10,\ 100,\ 1000,\ 10000,\ 100000\,.$$

Similarly to the case of SVM-KNN, the number of nearest neighbors in the classical kNN method is equal to $k_n = \lceil c \cdot n^{0.5} \rceil$ where the hyperparameter c is chosen by a five-fold cross validation among

$$0.1,\ 0.2\ ,0.3\ ,0.4\ ,0.5\ ,0.6\ ,0.7\ ,0.8\ ,0.9\ ,1\,.$$

The evaluation of the estimates is done on a test data set which consists of 1001 equidistant grid points ξ_i on $[-1,1]$. For every run $r \in \{1,\ldots,500\}$, the mean absolute error (MAE) is calculated

$$\mathrm{MAE}_{j,r}(f_\star) \;=\; \frac{1}{1001}\sum_{i=1}^{1001}\big|f_\star(x_i) - f_j(x_i)\big| \qquad \text{for } f_\star \in \big\{f_{j,r}^{\mathrm{SVM}}, f_{j,r}^{\mathrm{SVM\text{-}KNN}}, f_{j,r}^{\mathrm{kNN}}\big\}$$

where $f_{j,r}^{\mathrm{SVM}}$ denotes the SVM-estimate, $f_{j,r}^{\mathrm{SVM\text{-}KNN}}$ denotes the SVM-KNN-estimate, and $f_{j,r}^{\mathrm{kNN}}$ denotes the kNN-estimate in the r-th run of scenario j. For every scenario j and every learning algorithm, the values $\mathrm{MAE}_{j,r}(f_\star)$, $r \in \{1,\ldots,500\}$, are shown in a boxplot in Figure 3. In addition, Table 1 shows the average of $\mathrm{MAE}_{j,r}(f_\star)$ over the 500 runs:

$$\mathrm{MAE}_j(f_\star) \;=\; \frac{1}{500}\sum_{r=1}^{500}\mathrm{MAE}_{j,r}(f_\star) \qquad \text{for } f_\star \in \big\{f_{j,r}^{\mathrm{SVM}}, f_{j,r}^{\mathrm{SVM\text{-}KNN}}\big\}\,.$$

	scenario $j=1$	scenario $j=2$
SVM	0.453	0.115
SVM-KNN	0.331	0.216
kNN	0.348	0.189

Table 1: The average MAE_j of the mean absolute error over the 500 runs for classical SVMs and SVM-KNN for scenarios $j=1$ and $j=2$

It turns out that SVM-KNN is clearly better than classical SVM in scenario 1 while classical SVM is clearly better than SVM-KNN in scenario 2. In both examples, the performance of SVM-KNN is similar to that of classical kNN. Function f_2 in scenario 2 is a smooth function and classical SVMs are typically very successful for learning such smooth functions. Function f_1 in scenario 1 nearly coincides with f_2 in scenario 2 in the sense that the parts of the functions on $(-1,0)$ and $(0,1)$ are just interchanged. However, this leads to a considerable jump at $x=0$ which provides some difficulty for classical SVMs. Such jumps can be managed by classical SVMs if the hyperparameter γ and the regularization parameter λ are suitably chosen, namely, if γ is large and/or λ is small. However, such a choice increases the danger of overfitting in those parts of the input space in which the unknown regression function is a simple, smooth function. This problem is avoided by localized learners such as SVM-KNN, which is a main motivation for localizing global learning algorithms. In particular, the difference of the performance between scenario 1 and 2 is much smaller in case of SVM-KNN than in case of classical SVM. Figure 4 shows in a boxplot which values of γ are selected by the cross validation in the 500 runs for each scenario. Obviously, the jump in $x=0$ leads to large values of γ in scenario 1 compared to scenario 2. This in turn facilitates that the SVM-estimate is too volatile in those parts of the input space in which f_1 is relatively simple, for example, in the interval $[-1,-0.5]$. This tendency is exemplarily illustrated in Figure 5 which shows the estimates on the interval $[-1,0]$ of the input space in the first 9 runs of the simulation in case of scenario 1.

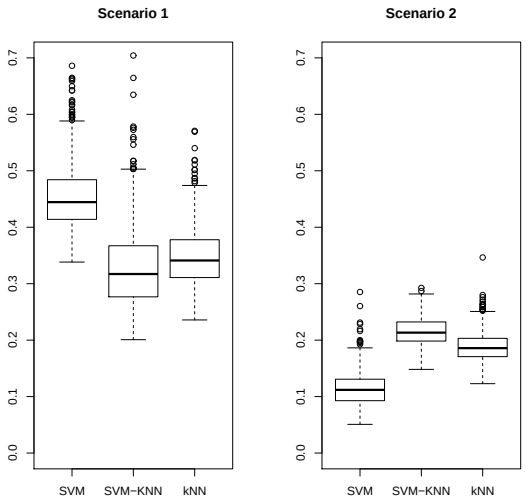


Figure 3: Boxplots of the mean absolute errors $\mathrm{MAE}_{j,r}$ in the runs $r \in \{1,\ldots,500\}$ for classical SVMs and SVM-KNN for scenarios $j = 1$ and $j = 2$

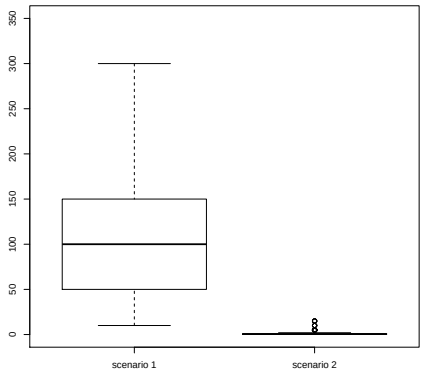


Figure 4: Values of the hyperparameter γ selected by cross validation for the classical SVM in the 500 runs for each scenario

5. Conclusions

Learning algorithms which are defined in a global manner typically can have difficulties if the complexity of the optimal predictor varies for different areas of the input space. One way to overcome this problem is to localize the learning algorithm. That is, the learning algorithm is not applied to the whole training data but only to those training data which are close to the testing point. In a num-

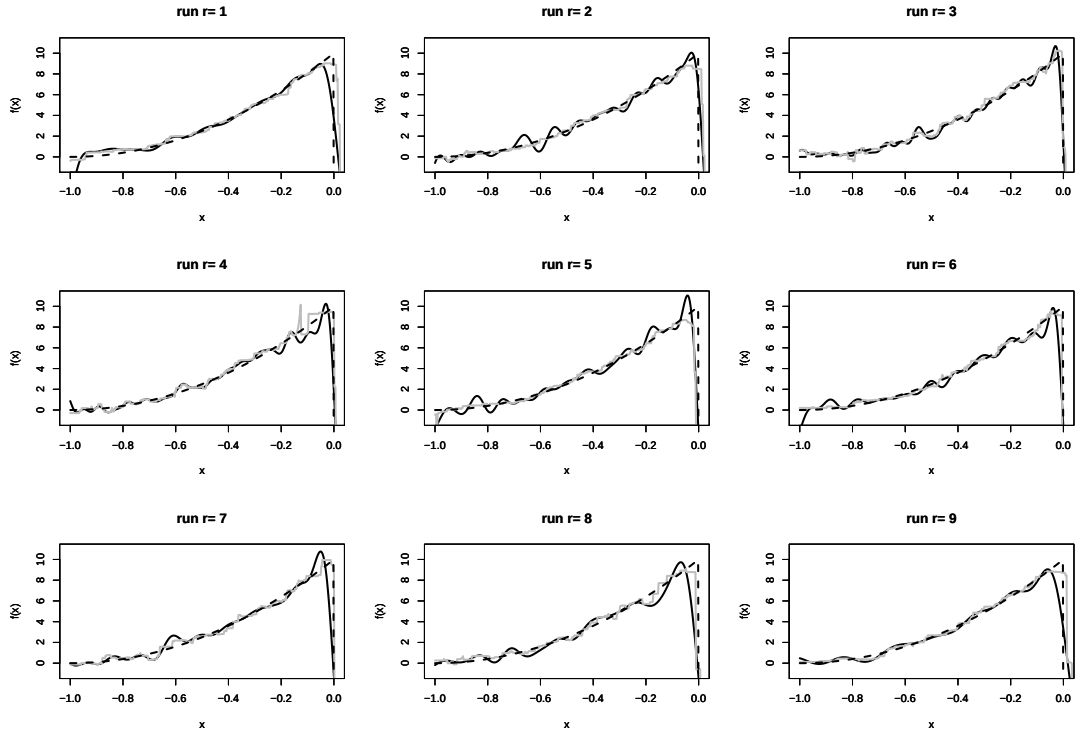


Figure 5: Estimates on the interval $[-1,0]$ in the first nine runs in scenario 1: true function f_1 (dashed black line), SVM (solid black line), SVM-KNN (solid gray line)

ber of recent articles such localizations of support vector machines have been suggested and their success has empirically been shown in extensive simulation studies and on real data sets but only little has been known on theoretical properties. In this article, it has been shown for a large class of regularized kernel methods (including SVM) that suitably localized versions (called SVM-KNN) are universally consistent.

Instead of localizing support vector machines, it would also be possible in principle to localize any other learning algorithm, for example, boosting. If this is done suitably, then localizing a learning algorithm will often lead to an algorithm which is again universally consistent. This article presents one way how this can be done in the special case of regularized kernel methods. However, it is a topic of further research if it is possible to derive a general scheme of localizing learning algorithms which, in combination with properties of the learning algorithm, always guarantees universal consistency.

Acknowledgments

I would like to thank two anonymous reviewers whose valuable comments have led to substantial improvements of the manuscript.

Appendix A. Preparations

Let $P_{\mathcal{X}}$ denote the distribution of the covariates X_i. For every $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$, there is a smallest $r_{n,\xi} \in [0,\infty]$ such that $Q\big(|X_i - \xi| \le r_{n,\xi}\big) \ge \frac{k_n}{n}$ and there is an $s_{n,\zeta} \in [0,\infty)$ such that

$$Q\Big(|X_i - \xi| < r_{n,\xi} \ \text{ or } \ \big(|X_i - \xi| = r_{n,\xi},\ |U_i - u| < s_{n,\zeta} \big) \Big) \ = \ \frac{k_n}{n} \, .$$

For every $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$, $r \in [0,\infty)$, and $s \in [0,\infty)$, define the open balls $B_r(\xi) = \{x \in \mathcal{X} \,|\, |x - \xi| < r\}$ and $B_s(u) = \{v \in (0,1) \,|\, |v - u| < s\}$, and define the boundary $\partial B_r(\xi) = \{x \in \mathcal{X} \,|\, |x - \xi| = r\}$. Define

$$B_{n,\zeta} \ = \ \Big(B_{r_{n,\xi}}(\xi) \times (0,1) \Big) \cup \Big(\partial B_{r_{n,\xi}}(\xi) \times B_{s_{n,\zeta}}(u) \Big)$$

Roughly spoken, $B_{n,\zeta}$ is a neighborhood around $\zeta = (\xi, u)$ with probability k_n/n which is in line with our tie-breaking strategy. Then,

$$P_{\mathcal{X}} \otimes \mathrm{Unif}_{(0,1)}\big(B_{n,\zeta}\big) \ = \ Q\big(Z_i \in B_{n,\zeta}\big) \ = \ \frac{k_n}{n}$$

where $\mathrm{Unif}_{(0,1)}$ denotes the uniform distribution on $(0,1)$. Let $P_{n,\zeta}$ be the conditional distribution of Z_i given $Z_i \in B_{n,\zeta}$, that is,

$$P_{n,\zeta}(B) = \frac{Q\big(Z_i \in B \cap B_{n,\zeta}\big)}{Q\big(Z_i \in B_{n,\zeta}\big)} = \frac{n}{k_n} Q\big(Z_i \in B \cap B_{n,\zeta}\big) \quad \forall B \in \mathfrak{B}_{\mathcal{X} \times (0,1)}.$$

Let $x \mapsto P(\cdot|x)$ be any regular version of the factorized conditional distribution of Y_i given $X_i = x$; see, for example, (Dudley, 2002, § 10.2). Due to independence of U_i, this coincides with the conditional distribution of Y_i given $Z_i = z$ (i.e., given $(X_i, U_i) = (x,u)$) and, accordingly, we write $P(\cdot|z) = P(\cdot|x)$. Let $Q_{Z,Y}$ denote the joint distribution of (Z_i, Y_i) and define $\mathcal{Z} := \mathcal{X} \times (0,1)$. Then, for every $\zeta \in \mathcal{Z}$, $n \in \mathbb{N}$, and every integrable $g : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$,

$$\frac{n}{k_n} \int_{\mathcal{Z} \times \mathcal{Y}} I_{B_{n,\zeta}}(z) g(z,y) \, Q_{Z,Y}\big(d(z,y)\big) \ = \ \int_{\mathcal{Z}} \int_{\mathcal{Y}} g(z,y) \, P(dy|z) P_{n,\zeta}(dz) \, . \tag{16}$$

When this does not lead to confusion, the conditional distribution of the pair of random variables (Z_i, Y_i) given $Z_i \in B_{n,\zeta}$ is also denoted by $P_{n,\zeta}$. That is, we will also write

$$\frac{n}{k_n} \int_{\mathcal{Z} \times \mathcal{Y}} I_{B_{n,\zeta}}(z) g(z,y) \, Q_{Z,Y}\big(d(z,y)\big) \ = \ \int_{\mathcal{Z} \times \mathcal{Y}} g(z,y) \, P_{n,\zeta}\big(d(z,y)\big) \, . \tag{17}$$

The following lemma is an immediate consequence of the definitions and well known facts about the support of measures, see, for example, Parthasarathy (1967, II. Theorem 2.1). It says that, for almost every $\xi \in \mathcal{X}$, the radii $r_{n,\xi}$ decrease to 0.

Lemma 2 *Define*

$$B_0 \ := \ \big\{ \xi \in \mathcal{X} \ \big| \ \nexists r \in (0,\infty) \text{ such that } P_{\mathcal{X}}(B_r(\xi)) = 0 \big\} \, .$$

Then, $P_{\mathcal{X}}(B_0) = 1$.

Furthermore, for every $\xi \in B_0$,

$$\infty \ge r_{1,\xi} \ge r_{2,\xi} \ge r_{3,\xi} \ge \ldots \ge \lim_n r_{n,\xi} = 0 \, .$$

Similarly to the definition of $I_{n,\zeta}$ and $\mathbf{D}_{n,\zeta}$ in (4) and (5), we define the modifications $I^\star_{n,\zeta}$ and $\mathbf{D}^\star_{n,\zeta}$: For every $n \in \mathbb{N}$, $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$ and $\omega \in \Omega$, define

$$I^\star_{n,\zeta}(\omega) := \big\{ i \in \{1,\ldots,n\} \,\big|\, Z_i(\omega) \in B_{n,\zeta} \big\} .$$

Fix any $n \in \mathbb{N}$, $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$ and $\omega \in \Omega$ and let $i_1 < i_2 < \ldots < i_m$ be the (ordered) elements of $I^\star_{n,\zeta}(\omega)$. Then, define

$$\mathbf{D}^\star_{n,\zeta}(\omega) \;=\; \Big(\big(Z_{i_1}(\omega), Y_{i_1}(\omega)\big), \ldots, \big(Z_{i_m}(\omega), Y_{i_m}(\omega)\big) \Big) .$$

That is, $I^\star_{n,\zeta}$ consists of all those indexes $i \in \{1,\ldots,n\}$ and $\mathbf{D}^\star_{n,\zeta}$ consists of all those data points (Z_i, Y_i) such that $Z_i \in B_{n,\zeta}$. This means: while the the sets $I_{n,\zeta}$ and $\mathbf{D}_{n,\zeta}$ consist of a *fixed number* of nearest neighbors, the sets $I^\star_{n,\zeta}$ and $\mathbf{D}^\star_{n,\zeta}$ consist of all those neighbors which lie in a *fixed neighborhood*.

As the probability that $Z_i \in B_{n,\zeta}$ is k_n/n, we expect that, for large n, the index sets $I_{n,\zeta}$ and $I^\star_{n,\zeta}$ and the vectors of data points $\mathbf{D}_{n,\zeta}$ and $\mathbf{D}^\star_{n,\zeta}$ are similar. However, working with $I^\star_{n,\zeta}$ is more comfortable because, whether $i \in I^\star_{n,\zeta}$, only depends on Z_i but, whether $i \in I_{n,\zeta}$, depends on all $Z_1, \ldots, Z_n$.

If a real-valued function f is clipped at M, then the clipped version is denoted by $\widehat{f}$, that is, $\widehat{f}(x) = f(x)$ if $-M \le f(x) \le M$, and $\widehat{f}(x) = -M$ if $f(x) < -M$, and $\widehat{f}(x) = M$ if $M < f(x)$. Note that, for every $f_1, f_2 : \mathcal{X} \to \mathbb{R}$ and $\xi \in \mathcal{X}$, it follows that $\big|\widehat{f_1}(\xi) - \widehat{f_2}(\xi)\big| \le \big|f_1(\xi) - f_2(\xi)\big|$. Furthermore, since K is bounded, every $f \in H$ fulfills $|f(\xi)| \le \|K\|_\infty \cdot \|f\|_H$; see (Steinwart and Christmann, 2008, Lemma 4.23). In combination with (12), this implies that, for every $\xi \in \mathcal{X}$ and for every $f_1, f_2 \in H$,

$$\left| \int L\big(y, \widehat{f_1}(\xi)\big) P(dy|\xi) - \int L\big(y, \widehat{f_2}(\xi)\big) P(dy|\xi) \right| \;\le\; |L|_{M,1} \cdot \|K\|_\infty \cdot \|f_1 - f_2\|_H . \tag{18}$$

Define $\|L(\cdot,0)\|_\infty = \sup_{y \in [-M,M]} \big|L(y,0)\big|$. Then, for every probability measure P_0,

$$\mathcal{R}_{P_0}(0) \;=\; \int L(y,0)\, P_0\big(d(x,y)\big) \;\le\; \|L(\cdot,0)\|_\infty \overset{(13)}{<} \infty . \tag{19}$$

The following lemma is one of the main tools; it is an application of Hoeffding's inequality and will be used several times for $V = H$ and $V = \mathbb{R}$.

Lemma 3 *Let V be a separable Hilbert space and, for every $n \in \mathbb{N}$, let $\Psi_n : \mathcal{Z} \times \mathcal{Y} \to V$ be a Borel-measurable function such that for every bounded subset $B \subset \mathcal{Z}$,*

$$\sup_{n \in \mathbb{N}} \sup_{z \in B, y \in \mathcal{Y}} \big\|\Psi_n(z,y)\big\|_H \;<\; \infty .$$

Then, for every $\zeta \in \mathcal{X}$,

$$\lambda_n^{-\frac{3}{2}} \frac{n}{k_n} \left(\frac{1}{n} \sum_{i=1}^n \Psi_n(Z_i, Y_i) I_{B_{n,\zeta}}(Z_i) - \int \Psi_n(z,y) I_{B_{n,\zeta}}(z)\, Q_{Z,Y}\big(d(z,y)\big) \right) \xrightarrow[n\to\infty]{} 0$$

in probability.

Note that the integral in Lemma 3 is an integral over a Hilbert-space-valued function and, accordingly, is a Bochner integral; see, for example, (Denkowski et al., 2003, § 3.10) for such integrals.

Proof The proof is done by an application of Hoeffding's inequality for functions with values in a separable Hilbert space. According to Lemma 2, there is an $n_0 \in \mathbb{N}$ such that $B_{n_0,\zeta}$ is bounded and $B_{n,\zeta} \subset B_{n_0,\zeta}$ for every $n \geq n_0$. Hence, there is a constant $b \in (0,\infty)$ such that, for every $n \geq n_0$,

$$\sup_{(z,y)\in\mathcal{Z}\times\mathcal{Y}} \left\|\Psi_n(z,y)I_{B_{n,\zeta}}(z)\right\|_V \leq b\,.$$

For every $n \geq n_0$ and $\tau \in (0,\infty)$, define $a_{n,\tau} := 2b\cdot\left(\sqrt{\tau n^{-1}} + \sqrt{n^{-1}} + \tau n^{-1}\right)$ and

$$A_{n,\tau} = \left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} \Psi_n(Z_i,Y_i)I_{B_{n,\zeta}}(Z_i) - \int \Psi_n I_{B_{n,\zeta}}\, dQ_{Z,Y} \right\|_V < a_{n,\tau} \right\}.$$

Then, by Hoeffding's inequality for separable Hilbert spaces (e.g., Steinwart and Christmann, 2008, Corollary 6.15),

$$Q\big(A_{n,\tau}\big) \;\geq\; 1 - e^{-\tau} \qquad \forall n \geq n_0\,, \quad \forall \tau \in (0,\infty)\,. \tag{20}$$

Define $\tau_n := \lambda_n^{\frac{3}{2}} k_n n^{-\frac{1}{2}}$ and $\varepsilon_n := \lambda_n^{-\frac{3}{2}} n k_n^{-1} a_{n,\tau_n}$ for every $n \geq n_0$. Then, for every $\omega \in A_{n,\tau}$,

$$\lambda_n^{-\frac{3}{2}} \frac{n}{k_n} \left\| \frac{1}{n}\sum_{i=1}^{n} \Psi_n(Z_i(\omega),Y_i(\omega))I_{B_{n,\zeta}}(Z_i(\omega)) - \int \Psi_n I_{B_{n,\zeta}}\, dQ_{Z,Y} \right\|_V < \varepsilon_n.$$

According to (9),

$$\varepsilon_n = \frac{n\cdot a_{n,\tau_n}}{\lambda_n^{\frac{3}{2}} k_n} = \frac{2bn}{\lambda_n^{\frac{3}{2}} k_n}\left(\sqrt{\frac{\lambda_n^{\frac{3}{2}} k_n}{\sqrt{n} n}} + \sqrt{\frac{1}{n}} + \frac{\lambda_n^{\frac{3}{2}} k_n}{\sqrt{n} n}\right) = 2b\cdot\left(\sqrt{\frac{\sqrt{n}}{\lambda_n^{\frac{3}{2}} k_n}} + \frac{\sqrt{n}}{\lambda_n^{\frac{3}{2}} k_n} + \frac{1}{\sqrt{n}}\right) \xrightarrow[n\to\infty]{} 0.$$

Hence, for every $\varepsilon > 0$, there is an $n_\varepsilon \in \mathbb{N}$ such that $\varepsilon > \varepsilon_n$ for every $n \geq n_\varepsilon$ and, therefore,

$$Q\left(\lambda_n^{-\frac{3}{2}} \frac{n}{k_n} \left\| \frac{1}{n}\sum_{i=1}^{n} \Psi_n(Z_i,Y_i)I_{B_{n,\zeta}}(Z_i) - \int \Psi_n I_{B_{n,\zeta}}\, dQ_{Z,Y} \right\|_V > \varepsilon\right) \leq Q\Big(\complement A_{n,\tau_n}\Big) \overset{(20)}{\leq} e^{-\tau_n}\,.$$

The last expression converges to 0 because $\lim_{n\to\infty}\tau_n = \infty$ due to (9), ■

Appendix B. Measurability

Measurability is an issue and needs some care because the SVM-KNN is based on a subsample which is randomly chosen. It is not possible to ignore measurability by turning over to outer probabilities here because the final step of the proof of the main theorem is based on an application of Fubini's Theorem and, therefore, heavily relies on (product) measurability.

Lemma 4

(a) The following maps are measurable with respect to the product-σ-algebra $\mathbb{B}^p \otimes \mathfrak{B}_{(0,1)} \otimes \mathcal{A}$ and the respective Borel-σ-Algebra:

(i) $\mathbb{R}^p \times (0,1) \times \Omega \to \mathbb{R}^{(p+1)k_n}, \quad (\xi,u,\omega) = (\zeta,\omega) \mapsto \mathbf{D}_{n,\zeta}(\omega)$

(ii) $\mathbb{R}^p \times (0,1) \times \Omega \to \mathbb{R}, \quad (\xi,u,\omega) = (\zeta,\omega) \mapsto R_{n,\zeta}(\omega) := \max_{i \in I_{n,\zeta}} \left|X_i(\omega) - \xi\right|.$

(iii) $\mathbb{R}^p \times (0,1) \times \Omega \to \mathbb{R}, \quad (\xi,u,\omega) = (\zeta,\omega) \mapsto \tilde{\Lambda}_{n,\zeta}(\omega).$

(b) Let $\Lambda : \mathbb{R}^p \times (0,1) \times \Omega \to (0,\infty)$ be measurable with respect to $\mathbb{B}^p \otimes \mathfrak{B}_{(0,1)} \otimes \mathcal{A}$ and the Borel-σ-Algebra. Then,

$$\mathbb{R}^p \times (0,1) \times \Omega \to \mathbb{R}, \quad (\xi,u,\omega) = (\zeta,\omega) \mapsto f_{\mathbf{D}_{n,\zeta}(\omega),\Lambda(\zeta,\omega)}(\xi)$$

is measurable with respect to $\mathbb{B}^p \otimes \mathfrak{B}_{(0,1)} \otimes \mathcal{A}$ and $\mathbb{B}$.

(c) For every $\zeta = (\xi,u) \in \mathbb{R}^p \times (0,1)$ and every $\Lambda : \Omega \to (0,\infty)$ measurable with respect to $\mathcal{A}$ and the Borel-σ-Algebra, the map

$$\Omega \to \mathbb{R}, \quad \omega \mapsto f_{\mathbf{D}^*_{n,\zeta}(\omega),\Lambda(\omega)}(\xi)$$

is measurable with respect to $\mathcal{A}$ and $\mathbb{B}$.

Proof For every $\zeta = (\xi,u) \in \mathbb{R}^p \times (0,1)$ and $\omega \in \Omega$, define $I_{n,\zeta}(\omega)$ as in Section 2. Let Ind_n denote the set of all subsets of $\{1,\ldots,n\}$ with k_n elements. First, it is shown that

$$\tilde{\tau}_n : \Omega \times \mathbb{R}^p \times (0,1) \to \mathrm{Ind}_n, \quad (\omega,\xi,u) \mapsto I_{n,(\xi,u)}(\omega)$$

is measurable with respect to $\mathcal{A} \otimes \mathbb{B}^p \otimes \mathfrak{B}_{(0,1)}$ and 2^{Ind_n}: Take any $I \in \mathrm{Ind}_n$ such that $I \neq \{1,\ldots,k_n\}$ and, for every $\mathcal{J} \subset \{1,\ldots,n\}$, define

$$\begin{aligned}
B_{\mathcal{J}}^{(1)} &:= \left\{(\omega,\xi,u) \in \Omega \times \mathbb{R}^p \times (0,1) \,\middle|\, \max_{i \in I} |X_i(\omega) - \xi| \leq \min_{\ell \notin I} |X_\ell(\omega) - \xi| \right\} \\
B_{\mathcal{J}}^{(2)} &:= \left\{(\omega,\xi,u) \in \Omega \times \mathbb{R}^p \times (0,1) \,\middle|\, |X_j(\omega) - \xi| = \max_{i \in I} |X_i(\omega) - \xi| \ \ \forall\, j \in \mathcal{J} \right\} \\
B_{\mathcal{J}}^{(3)} &:= \left\{(\omega,\xi,u) \in \Omega \times \mathbb{R}^p \times (0,1) \,\middle|\, |X_\ell(\omega) - \xi| \neq \max_{i \in I} |X_i(\omega) - \xi| \ \ \forall\, \ell \notin \mathcal{J} \right\} \\
B_{\mathcal{J}}^{(4)} &:= \left\{(\omega,\xi,u) \in \Omega \times \mathbb{R}^p \times (0,1) \,\middle|\, \max_{i \in \mathcal{J} \cap I} |U_i(\omega) - u| < \min_{j \in \mathcal{J} \setminus I} |U_j(\omega) - u| \right\}.
\end{aligned}$$

The set $B_{\mathcal{J}}^{(1)}$ says that no X_ℓ is closer to ξ than the k_n nearest neighbors. The sets $B_{\mathcal{J}}^{(2)}$ and $B_{\mathcal{J}}^{(3)}$ states that $\mathcal{J}$ specifies all those X_j which lie at the border of the neighborhood given by the nearest neighbors. The set $B_{\mathcal{J}}^{(4)}$ is concerned with all data points which lie at the border: the nearest neighbors among them have strictly smaller $|U_i - u|$ than those which do not belong to the nearest neighbors. Accordingly, the inverse image $\tilde{\tau}_n^{-1}(\{I\})$ equals

$$\tilde{\tau}_n^{-1}(\{I\}) = \bigcup_{\mathcal{J} \subset \{1,\ldots,n\}} \left(B_{\mathcal{J}}^{(1)} \cap B_{\mathcal{J}}^{(2)} \cap B_{\mathcal{J}}^{(3)} \cap B_{\mathcal{J}}^{(4)} \right).$$

Since $B_{\mathcal{J}}^{(t)}$ is measurable for every $t \in \{1,2,3,4\}$ and $\mathcal{J} \subset \{1,\dots,n\}$, this shows that $\tilde{\tau}_n^{-1}(\{I\})$ is measurable for every $I \neq \{1,\dots,k_n\}$. Hence, $\tilde{\tau}_n$ is measurable. For every $I = \{i_1,\dots,i_{k_n}\}$ such that $i_1 < i_2 < \cdots < i_{k_n}$ and every $D_n = \big((z_1,y_1),\dots,(z_n,y_n)\big) \in ((\mathbb{R}^p \times (0,1)) \times \mathbb{R})^n$ define

$$\varphi_n(I, D_n) \;=\; \big((z_{i_1},y_{i_1}),\dots,(z_{i_{k_n}},y_{i_{k_n}})\big)\,.$$

The map $\varphi_n : \mathrm{Ind}_n \times ((\mathbb{R}^p \times (0,1)) \times \mathbb{R})^n \to ((\mathbb{R}^p \times (0,1)) \times \mathbb{R})^{k_n}$ is continuous (where Ind_n is endowed with the discrete topology). Since

$$\mathbf{D}_{n,\zeta}(\omega) \;=\; \varphi_n\Big(\tilde{\tau}_n(\omega,\xi,u)\,,\mathbf{D}_n(\omega)\Big) \qquad \text{for } \zeta = (\xi,u)\,,$$

statement (i) follows from measurability of $\tilde{\tau}_n$ and φ_n. Next, (ii) follows from measurability of $(x_{i_1},\dots,x_{i_{k_n}},\xi) \mapsto \max_{j\in\{1,\dots,k_n\}} |x_{i_j} - \xi|$ and (iii) follows from

$$\tilde{\Lambda}_{n,\zeta} \;=\; \frac{1}{k_n}\sum_{i=1}^{n} |X_i - \xi|^{\frac{3}{2}} I_{[0,\infty)}(R_{n,\zeta} - X_i)\,.$$

Now, we can prove part (b) and (c): For every $I \subset \{1,2,\dots,n\}$ and every $D = \big((x_1,y_1),\dots,(x_n,y_n)\big) \in ((\mathbb{R}^p \times (0,1)) \times \mathbb{R})^n$, denote $D_I = \big((x_i,y_i)\big)_{i\in I}$. Then, it follows from Lemma 9 (a) and (Steinwart and Christmann, 2008, Lemma 4.23) that the map

$$2^{\{1,2,\dots,n\}} \times ((\mathbb{R}^p \times (0,1)) \times \mathbb{R})^n \times \mathcal{X} \;\to\; H\,, \qquad (I,D,\xi) \;\mapsto\; f_{D_I,\lambda}(\xi)$$

is continuous for every $\lambda > 0$ (where $2^{\{1,2,\dots,n\}}$ is endowed with the discrete topology). Since $\lambda \mapsto f_{D_I,\lambda}(\xi)$ is continuous for every fixed (I,D,ξ) according to (Steinwart and Christmann, 2008, Corollary 5.19 and Lemma 4.23), the map $\big((I,D,\xi),\lambda\big) \mapsto f_{D_I,\lambda}(\xi)$ is a Caratheodory function and, therefore, measurable; see, for example, Denkowski et al. (2003, Definition 2.5.18 and Theorem 2.5.22). Then, (b) follows from (a), and (c) follows from measurability of $\tilde{\tau}^*_{n,\zeta} : \omega \mapsto I^*_{n,\zeta}(\omega)$ for every fixed $\zeta = (\xi,u)$. Measurability of $\tilde{\tau}^*_{n,\zeta}$ follows from

$$\tilde{\tau}^*_{n,\zeta}{}^{-1}(I) \;=\; \bigcap_{i\in I} Z_i^{-1}(B_{n,\zeta}) \cap \bigcap_{i\notin I} Z_i^{-1}(\complement B_{n,\zeta}) \qquad \forall\, I \in 2^{\{1,2,\dots,n\}}\,.$$

■

Appendix C. Proof of Theorem 1

In the main part of the proof, it is shown that for $P_{\mathcal{X}} \otimes \mathrm{Unif}_{(0,1)}$ - almost every $\zeta = (\xi,u) \in \mathcal{X} \times (0,1)$,

$$0 \le \int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\,P(dy|\zeta) - \inf_{t\in\mathbb{R}} \int L(y,t)\,P(dy|\zeta) \;\xrightarrow[n\to\infty]{}\; 0 \tag{21}$$

in probability. Then, statement (21) implies Theorem 1 as follows:

Since, for every fixed $\zeta = (\xi,u)$, the maps

$$\omega \mapsto \int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta}(\omega),\Lambda_{n,\zeta}(\omega)}(\xi)\big)\,P(dy|\zeta) - \inf_{t\in\mathbb{R}} \int L(y,t)\,P(dy|\zeta)\,, \quad n \in \mathbb{N},$$

are uniformly bounded, convergence in probability for $P_X \otimes \mathrm{Unif}_{(0,1)}$ - almost every $\zeta = (\xi, u) \in X \times (0,1)$ implies

$$\mathbb{E}_Q \left(\int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\zeta) - \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\zeta) \right) \xrightarrow[n\to\infty]{} 0$$

for $P_X \otimes \mathrm{Unif}_{(0,1)}$ - almost every $\zeta = (\xi, u) \in X \times (0,1)$. Since the maps

$$\zeta = (\xi, u) \;\mapsto\; \mathbb{E}_Q \left(\int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\zeta) - \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\zeta) \right), \quad n \in \mathbb{N},$$

are uniformly bounded again, $P_X \otimes \mathrm{Unif}_{(0,1)}$ - almost sure convergence implies

$$\iint \mathbb{E}_Q \left(\int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) P(dy|\zeta) - \inf_{t\in\mathbb{R}} \int L(y,t) P(dy|\zeta) \right) P_X(d\xi)\, \mathrm{Unif}_{(0,1)}(du) \longrightarrow 0 \tag{22}$$

for $n \to \infty$. Note that $\zeta \mapsto \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\zeta)$ is measurable, because the assumptions on L imply continuity of $t \mapsto \int L(y,t)\, P(dy|\zeta)$, hence, $\inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\zeta) = \inf_{t\in\mathbb{Q}} \int L(y,t)\, P(dy|\zeta)$ for every $\zeta \in X \times (0,1)$. Next, recall that $f_{\mathbf{D}_n}(\zeta) = \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)$ and $P(\cdot|\xi) = P(\cdot|\zeta)$ for every $\zeta = (\xi, u)$. By a slight abuse of notation, we write

$$\mathcal{R}_P(f_{\mathbf{D}_n}) \;=\; \mathcal{R}_{P\otimes \mathrm{Unif}_{(0,1)}}(f_{\mathbf{D}_n}) \;=\; \iint L\big(y, f_{\mathbf{D}_n}(\xi,u)\big)\, P\big(d(\xi,y)\big)\, \mathrm{Unif}_{(0,1)}(du)\,.$$

Then, applying Fubini's Theorem in (22) yields

$$0 \;\le\; \mathbb{E}_Q \left(\mathcal{R}_P\big(f_{\mathbf{D}_n}\big) - \int \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\xi) P_X(d\xi) \right) \xrightarrow[n\to\infty]{} 0\,. \tag{23}$$

For every measurable $f : X \to \mathbb{R}$,

$$\int L\big(y, f(\xi)\big)\, P(dy|\xi) \;\ge\; \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\xi) \qquad \forall \xi \in X\,.$$

Hence,

$$\mathcal{R}_P^* \;\ge\; \int \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\xi) P_X(d\xi)$$

and, therefore, (23) implies

$$\mathbb{E}_Q \left(\mathcal{R}_P\big(f_{\mathbf{D}_n}\big) - \mathcal{R}_P^* \right) \xrightarrow[n\to\infty]{} 0$$

and, as $\mathcal{R}_P\big(f_{\mathbf{D}_n}\big) \ge \mathcal{R}_P^*$,

$$\mathbb{E}_Q \left| \mathcal{R}_P\big(f_{\mathbf{D}_n}\big) - \mathcal{R}_P^* \right| \xrightarrow[n\to\infty]{} 0\,.$$

In particular, this also implies

$$\mathcal{R}_P\big(f_{\mathbf{D}_n}\big) \xrightarrow[n\to\infty]{} \mathcal{R}_P^* \qquad \text{in probability.}$$

That is, it only remains to prove (21). To this end, note that, for every $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$, we have $P(\cdot|\zeta) = P(\cdot|\xi)$ and

$$
\begin{aligned}
0 \;\leq\; & \int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\zeta) \;-\; \inf_{t\in\mathbb{R}} \int L(y,t)\, P(dy|\zeta) \;\leq \\
\leq\; & \left| \int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) - \int L\big(y, \widehat{f}_{\mathbf{D}^{\star}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) \right| && (24) \\
& + \left| \int L\big(y, \widehat{f}_{\mathbf{D}^{\star}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) - \int L\big(y, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) \right| && (25) \\
& + \left| \int L\big(y, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) P(dy|\xi) - \iint L\big(y, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P(dy|x) P_{n,\zeta}(d(x,v)) \right| && (26) \\
& + \left(\iint L\big(y, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P(dy|x) P_{n,\zeta}(d(x,v)) - \inf_{t\in\mathbb{R}} \int L(y,t) P(dy|\xi) \right) \vee 0 && (27)
\end{aligned}
$$

where $a \vee 0 = \max\{a, 0\}$. Therefore, it suffices to prove convergence in probability of each of these four summands. This is done in the following four subsections but, first, we need some more preparations:

Lemma 5 *Fix any $\zeta = (\xi, u) \in B_0 \times (0,1)$ where B_0 is defined as in Lemma 2. Let $\mathbb{P}_{\mathbf{D}_{n,\zeta}}$ and $\mathbb{P}_{\mathbf{D}^{\star}_{n,\zeta}}$ denote the empirical measure corresponding to $\mathbf{D}_{n,\zeta}$ and $\mathbf{D}^{\star}_{n,\zeta}$ respectively. It follows that*

$$
\lambda_n^{-\frac{3}{2}} \frac{\big|\sharp(I^{\star}_{n,\zeta}) - k_n\big|}{k_n} \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability}, \tag{28}
$$

$$
\lambda_n^{-\frac{3}{2}} \frac{\big|\sharp(I^{\star}_{n,\zeta}) - k_n\big|}{\sharp(I^{\star}_{n,\zeta})} \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability}, \tag{29}
$$

$$
\lambda_n^{-\frac{3}{2}} \Big\| \mathbb{P}_{\mathbf{D}_{n,\zeta}} - \mathbb{P}_{\mathbf{D}^{\star}_{n,\zeta}} \Big\|_{\mathrm{TV}} \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability}, \tag{30}
$$

$$
R_{n,\zeta} \;:=\; \max_{i\in I_{n,\zeta}} |X_i - \xi| \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability}, \tag{31}
$$

and, for every $\beta \in (0,\infty)$,

$$
\lambda_n^{-\frac{3}{2}} \left| \frac{1}{k_n} \sum_{i\in I_{n,\zeta}} |X_i - \xi|^{\beta} - \int |x-\xi|^{\beta} P_{n,\zeta}(d(x,v)) \right| \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability}. \tag{32}
$$

Proof Statement (28) follows from Lemma 3 because the definitions imply

$$
\lambda_n^{-\frac{3}{2}} \frac{\big|\sharp(I^{\star}_{n,\zeta}) - k_n\big|}{k_n} \;=\; \lambda_n^{-\frac{3}{2}} \frac{n}{k_n} \left| \frac{1}{n} \sum_{i=1}^{n} I_{B_{n,\zeta}}(Z_i) - \int I_{B_{n,\zeta}}(z)\, Q_{Z,Y}\big(d(z,y)\big) \right|.
$$

In order to prove (29) note that (28) implies that $\sharp(I^\star_{n,\zeta})/k_n \to 1$ in probability and, therefore, also $k_n/\sharp(I^\star_{n,\zeta}) \to 1$ in probability. Hence, (28) implies (29) because

$$\lambda_n^{-\frac{3}{2}} \frac{\left|\sharp(I^\star_{n,\zeta}) - k_n\right|}{\sharp(I^\star_{n,\zeta})} \;=\; \lambda_n^{-\frac{3}{2}} \frac{\left|\sharp(I^\star_{n,\zeta}) - k_n\right|}{k_n} \cdot \frac{k_n}{\sharp(I^\star_{n,\zeta})} \,.$$

In order to prove (30) note that the definitions imply for almost every $\omega \in \Omega$

$$I_{n,\zeta}(\omega) \;\subset\; I^\star_{n,\zeta}(\omega) \qquad \text{or} \qquad I^\star_{n,\zeta}(\omega) \;\subset\; I_{n,\zeta}(\omega)\,. \tag{33}$$

(Only in case of distance ties in the $U_i(\omega)$, statement (33) is not true.) Therefore,

$$\sharp\big(I_{n,\zeta} \setminus I^\star_{n,\zeta}\big) \;\le\; \left|\sharp(I^\star_{n,\zeta}) - k_n\right| \qquad \text{and} \qquad \sharp\big(I^\star_{n,\zeta} \setminus I_{n,\zeta}\big) \;\le\; \left|\sharp(I^\star_{n,\zeta}) - k_n\right|. \tag{34}$$

almost surely. Then, almost surely,

$$\begin{aligned}
&\sup_{C \in \mathfrak{B}_{\mathcal{Z}\times\mathcal{Y}}} \left|\mathbb{P}_{\mathbf{D}_{n,\zeta}}(C) - \mathbb{P}_{\mathbf{D}^\star_{n,\zeta}}(C)\right| \;= \\
&\quad= \sup_C \left| \frac{1}{k_n}\Big(\sum_{i \in I_{n,\zeta} \cap I^\star_{n,\zeta}} I_C(Z_i,Y_i) + \sum_{i \in I_{n,\zeta} \setminus I^\star_{n,\zeta}} I_C(Z_i,Y_i)\Big) - \right. \\
&\qquad\qquad \left. - \frac{1}{\sharp\big(I^\star_{n,\zeta}\big)}\Big(\sum_{i \in I^\star_{n,\zeta} \cap I_{n,\zeta}} I_C(Z_i,Y_i) + \sum_{i \in I^\star_{n,\zeta} \setminus I_{n,\zeta}} I_C(Z_i,Y_i)\Big) \right| \;\le \\
&\le \sup_C \left| \frac{1}{k_n} - \frac{1}{\sharp\big(I^\star_{n,\zeta}\big)} \right| \sum_{i \in I_{n,\zeta} \cap I^\star_{n,\zeta}} I_C(Z_i,Y_i) \;+ \\
&\qquad + \frac{1}{k_n} \sup_C \sum_{i \in I_{n,\zeta} \setminus I^\star_{n,\zeta}} I_C(Z_i,Y_i) \;+\; \frac{1}{\sharp\big(I^\star_{n,\zeta}\big)} \sup_C \sum_{i \in I^\star_{n,\zeta} \setminus I_{n,\zeta}} I_C(Z_i,Y_i) \;\le \\
&\overset{(34)}{\le} \left| \frac{1}{k_n} - \frac{1}{\sharp\big(I^\star_{n,\zeta}\big)} \right| k_n + \frac{\left|\sharp(I^\star_{n,\zeta}) - k_n\right|}{k_n} + \frac{\left|\sharp(I^\star_{n,\zeta}) - k_n\right|}{\sharp(I^\star_{n,\zeta})}\,.
\end{aligned}$$

Therefore, (30) follows from (28) and (29).

In order to prove (31), fix any $\varepsilon > 0$. As $\xi \in B_0$, we have $P_X\big(B_\varepsilon(\xi)\big) > 0$ and, therefore, $P_X\big(B_\varepsilon(\xi)\big) - k_n/n > \frac{1}{2} P_X\big(B_\varepsilon(\xi)\big) > 0$ for n large enough (see Lemma 2). Then, (31) follows from

$$\begin{aligned}
Q\Big(R_{n,\zeta} > \varepsilon\Big) &= Q\Big(\sharp\big\{i \in \{1,\ldots,n\} \,\big|\, X_i \in B_\varepsilon(\xi)\big\} < k_n\Big) \;=\; Q\left(\frac{1}{n}\sum_{i=1}^n I_{B_\varepsilon(\xi)}(X_i) < \frac{k_n}{n}\right) \\
&= Q\left(P_X\big(B_\varepsilon(\xi)\big) - \frac{1}{n}\sum_{i=1}^n I_{B_\varepsilon(\xi)}(X_i) > P_X\big(B_\varepsilon(\xi)\big) - \frac{k_n}{n}\right) \;\le \\
&\le Q\left(P_X\big(B_\varepsilon(\xi)\big) - \frac{1}{n}\sum_{i=1}^n I_{B_\varepsilon(\xi)}(X_i) > \frac{1}{2} P_X\big(B_\varepsilon(\xi)\big)\right)
\end{aligned}$$

and the law of large numbers.

Now, statement (32) will be proven. An application of Lemma 3 for $\Psi_n\big((x,v),y\big) = |x-\xi|^\beta$ and (16) yield that it suffices to prove

$$\lambda_n^{-\frac{3}{2}} \left| \frac{1}{k_n} \sum_{i \in I_{n,\zeta}} |X_i - \xi|^\beta - \frac{1}{k_n}\sum_{i=1}^n |X_i - \xi|^\beta I_{B_{n,\zeta}}(Z_i) \right| \;\xrightarrow[n\to\infty]{}\; 0 \tag{35}$$

in probability in order to prove statement (32).

According to Lemma 2, there is an $n_0 \in \mathbb{N}$ such that $r_{n,\xi} \leq 1$ for every $n \geq n_0$. Then, for every $\varepsilon > 0$ and every $n \geq n_0$,

$$\begin{aligned} Q\left(\lambda_n^{-\frac{3}{2}}\left|\frac{1}{k_n}\sum_{i\in I_{n,\zeta}}|X_i-\xi|^\beta - \frac{1}{k_n}\sum_{i=1}^n |X_i-\xi|^\beta I_{B_{n,\zeta}}(Z_i)\right| > \varepsilon\right) \leq \\ \leq Q\left(\lambda_n^{-\frac{3}{2}}\left\|\mathbb{P}_{\mathbf{D}_{n,\zeta}} - \tfrac{\sharp(I^\star_{n,\zeta})}{k_n}\mathbb{P}_{\mathbf{D}^\star_{n,\zeta}}\right\|_{\mathrm{TV}} > \varepsilon,\ R_{n,\zeta}\leq 1\right) + Q\Big(R_{n,\zeta}>1\Big) \\ \leq Q\left(\lambda_n^{-\frac{3}{2}}\left\|\mathbb{P}_{\mathbf{D}_{n,\zeta}} - \mathbb{P}_{\mathbf{D}^\star_{n,\zeta}}\right\|_{\mathrm{TV}} > \frac{\varepsilon}{2}\right) + Q\left(\lambda_n^{-\frac{3}{2}}\frac{\left|\sharp(I^\star_{n,\zeta})-k_n\right|}{k_n} > \frac{\varepsilon}{2}\right) + Q\Big(R_{n,\zeta}>1\Big) \end{aligned}$$

so that (35) follows from (30), (28), and (31). ■

Lemma 6 *For every $P_{\mathcal{X}}$-integrable $h : \mathcal{X} \to \mathbb{R}$, there is a set $B_h \in \mathfrak{B}_{\mathcal{X}}$ such that $P_{\mathcal{X}}(B_h) = 1$ and*

$$\lim_{n\to\infty}\int \big|h(x)-h(\xi)\big|\,P_{n,\zeta}\big(d(x,v)\big) = 0 \qquad \forall\,\zeta=(\xi,u)\in B_h\times(0,1). \tag{36}$$

Proof Define

$$\gamma_{n,\xi} := \frac{1}{P_{\mathcal{X}}\big(B_{r_{n,\xi}}(\xi)\big)}\int_{B_{r_{n,\xi}}(\xi)}\big|h(x)-h(\xi)\big|\,P_{\mathcal{X}}(dx)$$

and, analogously, define $\bar{\gamma}_{n,\xi}$ where the open ball $B_{r_{n,\xi}}(\xi)$ is replaced by the closed ball $\overline{B}_{r_{n,\xi}}(\xi)$ around ξ with radius $r_{n,\xi}$. According to Besicovitch's Density Theorem, there is a set $B_h \in \mathfrak{B}_{\mathcal{X}}$ such that $P_{\mathcal{X}}(B_h) = 1$ and, for every $\xi \in B_h$, $\lim_{n\to\infty}\gamma_{n,\xi} = \lim_{n\to\infty}\bar{\gamma}_{n,\xi} = 0$; for $\gamma_{n,\xi}$, see, for example, (Fremlin, 2006, Theorem 472D(b)); for $\bar{\gamma}_{n,\xi}$, this follows from (Krantz and Parks, 2008, Theorem 4.3.5(2)) (exactly in the same way as Fremlin, 2006, Theorem 472D(b) follows from Fremlin, 2006, Theorem 472D(a)). Recall from Appendix A that $B_{n,\zeta} = \big(B_{r_{n,\xi}}(\xi)\times(0,1)\big)\cup\big(\partial B_{r_{n,\xi}}(\xi)\times B_{s_{n,\zeta}}(u)\big)$ and define $\alpha_{n,\zeta} := Q\big(U_i \in B_{s_{n,\zeta}}(u)\big)$, $\beta_{n,\xi} := P_{\mathcal{X}}\big(B_{r_{n,\xi}}(\xi)\big)$ and $\overline{\beta}_{n,\xi} := P_{\mathcal{X}}\big(\overline{B}_{r_{n,\xi}}(\xi)\big)$. Then,

$$\tfrac{k_n}{n} = Q\big(Z_i \in B_{n,\zeta}\big) = \beta_{n,\xi} + \alpha_{n,\zeta}\big(\overline{\beta}_{n,\xi}-\beta_{n,\xi}\big) \tag{37}$$

and

$$\begin{aligned} &\int \big|h(x)-h(\xi)\big|\,P_{n,\zeta}\big(d(x,v)\big) = \\ &= \frac{n}{k_n}\left(\int_{B_{r_{n,\xi}}(\xi)}\big|h(x)-h(\xi)\big|\,P_{\mathcal{X}}(dx) + \alpha_{n,\zeta}\int_{\partial B_{r_{n,\xi}}(\xi)}\big|h(x)-h(\xi)\big|\,P_{\mathcal{X}}(dx)\right) \\ &= \frac{n}{k_n}\Big(\beta_{n,\xi}\gamma_{n,\xi} + \alpha_{n,\zeta}\big(\overline{\beta}_{n,\xi}\bar{\gamma}_{n,\xi} - \beta_{n,\xi}\gamma_{n,\xi}\big)\Big) = \\ &= \frac{n}{k_n}\Big(\beta_{n,\xi}+\alpha_{n,\zeta}\big(\overline{\beta}_{n,\xi}-\beta_{n,\xi}\big)\Big)\bar{\gamma}_{n,\xi} + \frac{n}{k_n}(1-\alpha_{n,\zeta})\beta_{n,\xi}(\gamma_{n,\xi}-\bar{\gamma}_{n,\xi}) \leq \\ &\overset{(37)}{\leq} \bar{\gamma}_{n,\xi} + 1\cdot\big|\bar{\gamma}_{n,\xi}-\gamma_{n,\xi}\big| \xrightarrow[n\to\infty]{} 0 \end{aligned}$$

■

C.1 Convergence of the First Summand (24)

Fix any $\zeta = (\xi, u) \in B_0 \times (0,1)$ where B_0 is defined as in Lemma 2. Again let $\mathbb{P}_{\mathbf{D}_{n,\zeta}}$ and $\mathbb{P}_{\mathbf{D}^\star_{n,\zeta}}$ denote the empirical measure corresponding to $\mathbf{D}_{n,\zeta}$ and $\mathbf{D}^\star_{n,\zeta}$ respectively. It follows from (18), (19), and (51) that

$$\begin{aligned}
&\left| \int L\big(y, \widehat{f}_{\mathbf{D}_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) - \int L\big(y, \widehat{f}_{\mathbf{D}^\star_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big)\, P(dy|\xi) \right| \;\leq \\
&\qquad \leq \;\; |L|_{M,1} \|K\|_\infty^2 \Big(b_0 \Lambda_{n,\zeta}^{-1} + b_1 \|K\|_\infty^q \mathcal{R}_{P_1}(0)^{\frac{q}{2}} \Lambda_{n,\zeta}^{-\frac{q}{2}-1} \Big) \cdot \big\| \mathbb{P}_{\mathbf{D}_{n,\zeta}} - \mathbb{P}_{\mathbf{D}^\star_{n,\zeta}} \big\|_{\mathrm{TV}} \;\leq \\
&\qquad \overset{(10)}{\leq} \;\; |L|_{M,1} \|K\|_\infty^2 \Big(b_0 (c\lambda_n)^{-1} + b_1 \|K\|_\infty^q \big\|L(\cdot,0)\big\|_\infty^{\frac{q}{2}} (c\lambda_n)^{-\frac{q}{2}-1} \Big) \cdot \big\| \mathbb{P}_{\mathbf{D}_{n,\zeta}} - \mathbb{P}_{\mathbf{D}^\star_{n,\zeta}} \big\|_{\mathrm{TV}} .
\end{aligned}$$

Therefore, convergence in probability follows from (30) in Lemma 5 and $q \in [0,1]$.

C.2 Convergence of the Second Summand (25)

Fix any $\zeta = (\xi, u) \in B_0 \times (0,1)$.

Lemma 7 *For every $n \in \mathbb{N}$, define*

$$\tilde{\lambda}_{n,\zeta} \;=\; \int |x-\xi|^{\frac{3}{2}}\, P_{n,\zeta}\big(d(x,v)\big) \qquad \textit{and} \qquad \lambda_{n,\zeta} \;:=\; c \cdot \max\left\{\lambda_n, \tilde{\lambda}_{n,\zeta}\right\} .$$

Then,

$$\frac{|\Lambda_{n,\zeta} - \lambda_{n,\zeta}|}{\Lambda_{n,\zeta}\sqrt{\lambda_{n,\zeta}}} \;\xrightarrow[n\to\infty]{}\; 0 \qquad \textit{in probability} .$$

Proof For $a_1, a_2, b \in \mathbb{R}$, denote $a_1 \vee a_2 = \max\{a_1, a_2\}$ and note that $|a_1 \vee b - a_2 \vee b| \leq |a_1 - a_2|$. For every n, the definitions and (10) imply

$$\frac{|\Lambda_{n,\zeta} - \lambda_{n,\zeta}|}{\Lambda_{n,\zeta}\sqrt{\lambda_{n,\zeta}}} \leq \frac{\big|\Lambda_{n,\zeta} - c\cdot\big(\lambda_n \vee \tilde{\Lambda}_{n,\zeta}\big)\big| + c\cdot\big|\lambda_n \vee \tilde{\Lambda}_{n,\zeta} - \lambda_n \vee \tilde{\lambda}_{n,\zeta}\big|}{c^{\frac{3}{2}}\cdot\big(\lambda_n \vee \tilde{\Lambda}_{n,\zeta}\big)\sqrt{\lambda_n}} \leq \frac{c_n}{c^{\frac{3}{2}}\sqrt{\lambda_n}} + \frac{\big|\tilde{\Lambda}_{n,\zeta} - \tilde{\lambda}_{n,\zeta}\big|}{\sqrt{c}\lambda_n\sqrt{\lambda_n}} .$$

Hence, the statement follows from the assumption that $\lim_{n\to\infty} c_n/\sqrt{\lambda_n} = 0$ and from (32) in Lemma 5. ∎

According to (18), it suffices to show

$$\big\| f_{\mathbf{D}^\star_{n,\zeta},\Lambda_{n,\zeta}} - f_{P_{n,\zeta},\Lambda_{n,\zeta}} \big\|_H \;\xrightarrow[n\to\infty]{}\; 0 \qquad \text{in probability}$$

in order to prove convergence to 0 of the the second summand (25). To this end, note that

$$\begin{aligned}
&\big\| f_{\mathbf{D}^\star_{n,\zeta},\Lambda_{n,\zeta}} - f_{P_{n,\zeta},\Lambda_{n,\zeta}} \big\|_H \;\leq \\
&\qquad \leq \; \big\| f_{\mathbf{D}^\star_{n,\zeta},\Lambda_{n,\zeta}} - f_{\mathbf{D}^\star_{n,\zeta},\lambda_{n,\zeta}} \big\|_H + \big\| f_{\mathbf{D}^\star_{n,\zeta},\lambda_{n,\zeta}} - f_{P_{n,\zeta},\lambda_{n,\zeta}} \big\|_H + \big\| f_{P_{n,\zeta},\lambda_{n,\zeta}} - f_{P_{n,\zeta},\Lambda_{n,\zeta}} \big\|_H
\end{aligned}$$

and that $\big\| f_{\mathbf{D}^\star_{n,\zeta},\Lambda_{n,\zeta}} - f_{\mathbf{D}^\star_{n,\zeta},\lambda_{n,\zeta}} \big\|_H$ and $\big\| f_{P_{n,\zeta},\lambda_{n,\zeta}} - f_{P_{n,\zeta},\Lambda_{n,\zeta}} \big\|_H$ converge in probability to 0 according to part (i) of Lemma 9 (b), (19), and Lemma 7. Note that boundedness of the kernel K means that

$\sup_{x\in\mathcal{X}}\|\Phi(x)\|_H = \|K\|_\infty$. By defining $f(x,v)=f(x)$ for every $z=(x,v)\in\mathcal{X}\times(0,1)=\mathcal{Z}$ and $f\in H$, the RKHS H consisting of functions $f:\mathcal{X}\to\mathbb{R}$ can also be identified with an RKHS (again denoted by H) which consists of functions $f:\mathcal{Z}\to\mathbb{R}$; the kernel of this RKHS is given by $K(z,z')=K(x,x')$ for every $z=(x,v),z'=(x',u')\in\mathcal{X}\times(0,1)=\mathcal{Z}$; see, for example, the proof of (Christmann and Hable, 2012, Theorem 2). Fix $a=b_0+b_1\|K\|_\infty^q\big\|L(\cdot,0)\big\|_\infty^{q/2}c^{-q/2}$ and $n_0\in\mathbb{N}$ such that $\lambda_n\leq 1$ for every $n\geq n_0$. According to the definition of $\lambda_{n,\zeta}$, we have $\lambda_{n,\zeta}^{-q/2}\leq c^{-q/2}\lambda_n^{-q/2}\leq c^{-q/2}\lambda_n^{-1/2}$ for every $n\geq n_0$. According to part (ii) of Lemma 9 (b) and (19), for every $n\geq n_0$, there is a measurable function $h_{n,\zeta}:\mathcal{Z}\times\mathcal{Y}\to\mathbb{R}$ such that $\|h_{n,\zeta}\|_\infty\leq a\lambda_n^{-\frac{1}{2}}$ and

$$
\begin{aligned}
&\big\|f_{\mathbf{D}^\star_{n,\zeta},\lambda_{n,\zeta}}-f_{P_{n,\zeta},\lambda_{n,\zeta}}\big\|_H \;\leq\\
&\quad\leq\; \lambda_{n,\zeta}^{-1}\left\|\frac{1}{\sharp(I^\star_{n,\zeta})}\sum_{i\in I^\star_{n,\zeta}}h_{n,\zeta}(Z_i,Y_i)\Phi(X_i)-\int h_{n,\zeta}\Phi\,dP_{n,\zeta}\right\|_H \;\leq\\
&\quad\leq\; c^{-1}\lambda_n^{-1}\left|\frac{1}{\sharp(I^\star_{n,\zeta})}-\frac{1}{k_n}\right|\sum_{i\in I^\star_{n,\zeta}}\Big\|h_{n,\zeta}(Z_i,Y_i)\Phi(X_i)\Big\|_H+\\
&\qquad+c^{-1}\lambda_n^{-1}\left\|\frac{1}{k_n}\sum_{i\in I^\star_{n,\zeta}}h_{n,\zeta}(Z_i,Y_i)\Phi(X_i)-\int h_{n,\zeta}\Phi\,dP_{n,\zeta}\right\|_H \;\leq\\
&\quad\overset{(17)}{\leq}\; \lambda_n^{-\frac{3}{2}}\frac{\big|\sharp(I^\star_{n,\zeta})-k_n\big|}{k_n}\cdot ac^{-1}\|K\|_\infty+\\
&\qquad+\lambda_n^{-1}\frac{n}{k_n}\left\|\frac{1}{n}\sum_{i=1}^n h_{n,\zeta}(Z_i,Y_i)\Phi(X_i)I_{B_{n,\zeta}}(Z_i)-\int h_{n,\zeta}\Phi I_{B_{n,\zeta}}\,dQ_{Z,Y}\right\|_H\cdot c^{-1}.
\end{aligned}
$$

It follows from (28) that

$$
\lambda_n^{-\frac{3}{2}}\frac{\big|\sharp(I^\star_{n,\zeta})-k_n\big|}{k_n}\;\xrightarrow[n\to\infty]{}\;0\qquad\text{in probability}
$$

and it follows from Lemma 3 for $\Psi_n(z,y)=\lambda_n^{\frac{1}{2}}h_{n,\zeta}(z,y)\Phi(x)$, $z=(x,v)$, that

$$
\lambda_n^{-1}\frac{n}{k_n}\left\|\frac{1}{n}\sum_{i=1}^n h_{n,\zeta}(Z_i,Y_i)\Phi(X_i)I_{B_{n,\zeta}}(Z_i)-\int h_{n,\zeta}\Phi I_{B_{n,\zeta}}\,dQ_{Z,Y}\right\|_H\xrightarrow[n\to\infty]{}0\quad\text{in probability.}
$$

C.3 Convergence of the Third Summand (26)

For every $m\in\mathbb{N}$, define

$$
\alpha_m(y)\;=\;\sum_{j=-mM}^{mM}\frac{j}{m}I_{(\frac{j-1}{m},\frac{j}{m}]}(y)\,,\qquad y\in\mathbb{R}\,. \tag{38}
$$

That is, $\alpha_m(\mathcal{Y})\subset\big\{\frac{j}{m}\,\big|\,j\in\{-mM,\ldots,mM\}\big\}$ and

$$
\big|\alpha_m(y)-y\big|<\tfrac{1}{m}\quad\forall\,y\in\mathcal{Y}\,. \tag{39}
$$

According to Lemma 6, there is a set $B_1 \in \mathfrak{B}_X$ such that $P_X(B_1) = 1$ and such that, for all maps

$$h : \mathcal{X} \to \mathbb{R}, \quad x \mapsto P\Big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\Big|\, x\Big), \quad j \in \{-mM, \ldots, mM\}, \ m \in \mathbb{N},$$

(36) is fulfilled with $B_h = B_1$. Fix any $\zeta = (\xi, u) \in \mathcal{X} \times (0,1)$ such that $\xi \in B_0 \cap B_1$. It follows from (13) and (39) that, for every $m \in \mathbb{N}$,

$$\sup_{\substack{t \in [-M,M] \\ x \in \mathcal{X}}} \left| \int L(y,t)\, P(dy|x) - \int L(\alpha_m(y),t)\, P(dy|x) \right| \le \ell(\tfrac{1}{m}).$$

Since $\lim_{m\to\infty} \ell(\frac{1}{m}) = 0$, it is enough to show that, for every $m \in \mathbb{N}$,

$$\left| \int L\big(\alpha_m(y), \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) P(dy|\xi) - \iint L\big(\alpha_m(y), \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P(dy|x) P_{n,\zeta}\big(d(x,v)\big) \right|$$

converges to 0 in probability for $n \to \infty$. Next, it follows from

$$\int L\big(\alpha_m(y),t\big)\, P(dy|x) \overset{(38)}{=} \sum_{j=-mM}^{mM} L(\tfrac{j}{m},t) \cdot P\Big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\Big|\, x\Big) \qquad \forall t \in \mathbb{R} \quad \forall x \in \mathcal{X}$$

that it suffices to show that, for every $j \in \{-mM, \ldots, mM\}$ and $m \in \mathbb{N}$,

$$\left| L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, \xi\big) - \int L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, x\big) P_{n,\zeta}\big(d(x,v)\big) \right|$$

converges to 0 in probability for $n \to \infty$. The latter statement is shown in the following:

$$\begin{aligned}
&\left| L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, \xi\big) - \int L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, x\big) P_{n,\zeta}\big(d(x,v)\big) \right| \\
&\le \left| \int L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) \Big(P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, \xi\big) - P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, x\big) \Big) P_{n,\zeta}\big(d(x,v)\big) \right| \\
&\quad + \left| \int \Big(L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi)\big) - L\big(\tfrac{j}{m}, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) \Big) P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, x\big)\, P_{n,\zeta}\big(d(x,v)\big) \right| \\
&\le \sup_{t,y \in [-M,M]} L(y,t) \cdot \int \left| P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, \xi\big) - P\big(\big(\tfrac{j-1}{m}, \tfrac{j}{m}\big] \,\big|\, x\big) \right| P_{n,\zeta}\big(d(x,v)\big) \qquad (40) \\
&\quad + |L|_{M,1} \int \big| \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi) - \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x) \big|\, P_{n,\zeta}\big(d(x,v)\big) \qquad (41)
\end{aligned}$$

As $r_{n,\xi} \searrow 0$ (Lemma 2), it follows from the above definition of B_1 and $\xi \in B_1$ that the summand in (40) converges to 0 (in $\mathbb{R}$) for $n \to \infty$. In order to prove convergence (in probability) of the summand in (41), note that, according to the mean value theorem in several variables and Steinwart

and Christmann (2008, Corollary 4.36 and Equation (5.4)),

$$\begin{aligned}
&\int \left|\widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi) - \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\right| P_{n,\zeta}\big(d(x,v)\big) \;\leq \\
&\leq \int \left|f_{P_{n,\zeta},\Lambda_{n,\zeta}}(\xi) - f_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\right| P_{n,\zeta}\big(d(x,v)\big) \;\leq \\
&\leq \int \sup_{x' \in \overline{B}_{r_{n,\xi}}(\xi)} \left|\frac{\partial}{\partial x} f_{P_{n,\zeta},\Lambda_{n,\zeta}}(x')\right| \cdot |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big) \;\leq \\
&\leq \left\|f_{P_{n,\zeta},\Lambda_{n,\zeta}}\right\|_H \cdot \left(\sup_{x \in \overline{B}_{r_{n,\xi}}(\xi)} \sqrt{\partial^{1,1}K(x,x)}\right) \cdot \int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big) \;\leq \\
&\leq \left(\sup_{x \in \overline{B}_{r_{n,\xi}}(\xi)} \sqrt{\partial^{1,1}K(x,x)}\right) \cdot \sqrt{\mathcal{R}_{P_{n,\zeta}}(0)} \cdot \frac{\int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big)}{\sqrt{\Lambda_{n,\zeta}}}
\end{aligned}$$

where $\overline{B}_{r_{n,\xi}}(\xi)$ denotes the closed ball around ξ with radius $r_{n,\xi}$. As

$$\mathcal{R}_{P_{n,\zeta}}(0) \leq \sup_{y \in [-M,M]} L(y,0) < \infty \qquad \text{and} \qquad \lim_{n\to\infty} \sup_{x \in \overline{B}_{r_{n,\xi}}(\xi)} \sqrt{\partial^{1,1}K(x,x)} = \sqrt{\partial^{1,1}K(\xi,\xi)}\,,$$

it remains to show that

$$\frac{\int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big)}{\sqrt{\Lambda_{n,\zeta}}} \;\xrightarrow[n\to\infty]{}\; 0 \qquad \text{in probability .} \tag{42}$$

In order to prove (42), note that

$$\begin{aligned}
&\frac{\int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big)}{\sqrt{\Lambda_{n,\zeta}}} \;\leq \\
&\leq \frac{\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|}{\sqrt{\Lambda_{n,\zeta}}} + \frac{\left|\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi| - \int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big)\right|}{\sqrt{\Lambda_{n,\zeta}}} \;\leq \\
&\overset{(10)}{\leq} \frac{\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|}{\sqrt{c\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|^{\frac{3}{2}}}} + \qquad (43) \\
&\quad + c^{-\frac{1}{2}} \lambda_n^{-\frac{1}{2}} \left|\frac{1}{k_n} \sum_{i \in I_{n,\zeta}} |X_i - \xi| - \int |x-\xi| \, P_{n,\zeta}\big(d(x,v)\big)\right| . \qquad (44)
\end{aligned}$$

The summand in (44) converges to 0 in probability according to (32). The summand in (43) converges to 0 in probability because, by convexity of $z \mapsto z^{\frac{3}{2}}$ and $\sharp(I_{n,\zeta}) = k_n$, we get

$$\begin{aligned}
\frac{\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|}{\sqrt{c\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|^{\frac{3}{2}}}} \;&\leq\; \frac{\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|}{\sqrt{c\left(\frac{1}{k_n}\sum_{i \in I_{n,\zeta}} |X_i - \xi|\right)^{\frac{3}{2}}}} \;= \\
&= \; c^{-\frac{1}{2}} \Big(\frac{1}{k_n} \sum_{i \in I_{n,\zeta}} |X_i - \xi|\Big)^{\frac{1}{4}} \;\leq\; c^{-\frac{1}{2}} R_{n,\zeta}^{\frac{1}{4}} \;\xrightarrow[n\to\infty]{}\; 0 \quad \text{in probability}
\end{aligned}$$

according to (31).

C.4 Convergence of the Fourth Summand (27)

Let $\mathcal{M}_1(\mathcal{X}\times\mathcal{Y})$ be the set of all probability measures on $\mathcal{X}\times\mathcal{Y}$. For every $f\in H$, define the map $A_f:\ \mathcal{M}_1(\mathcal{X}\times\mathcal{Y})\times[0,\infty)\to\mathbb{R}$ by

$$A_f(P_0,\lambda)\ =\ \int L\big(y,f(x)\big)\,P_0\big(d(x,y)\big)\ +\ \lambda\|f\|_H^2 \tag{45}$$

for every $P_0\in\mathcal{M}_1(\mathcal{X}\times\mathcal{Y})$ and $\lambda\in[0,\infty)$. For every $f\in H$, the map $(x,y)\ \mapsto\ L\big(y,f(x)\big)$ is continuous and bounded on $\mathcal{X}\times\mathcal{Y}$ and, therefore, A_f is continuous with respect to weak convergence of probability measures (and the ordinary topology on $\mathbb{R}$ and $[0,\infty)$). Hence,

$$(P_0,\lambda)\ \mapsto\ \inf_{f\in H}A_f(P_0,\lambda)\qquad\text{is upper semi-continuous} \tag{46}$$

see, for example, (Denkowski et al., 2003, Prop. 1.1.36).

Let $\mathcal{C}_c(\mathcal{X}\times\mathcal{Y})$ be the set of all continuous functions $g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ with compact support. According to Denkowski et al. (2003, Theorem 2.6.24), there is a countable dense subset $\mathcal{S}\subset\mathcal{C}_c(\mathcal{X}\times\mathcal{Y})$ (with respect to uniform convergence).

According to Lemma 6, there is a set $B_2\in\mathfrak{B}_\mathcal{X}$ such that $P_\mathcal{X}(B_2)=1$ and such that, for all maps

$$h\ :\ \mathcal{X}\ \to\ \mathbb{R}\,,\qquad x\ \mapsto\ \int g(x,y)\,P(dy|x)\,,\qquad g\in\mathcal{S},$$

(36) is fulfilled with $B_h=B_2$. Fix any $\zeta=(\xi,u)\in\mathcal{X}\times(0,1)$ such that $\xi\in B_0\cap B_1\cap B_2$.

Lemma 8 *Let $(\Lambda_{n_j,\zeta})_{j\in\mathbb{N}}$ be a subsequence of $(\Lambda_{n,\zeta})_{n\in\mathbb{N}}$ which converges to zero Q-a.s. for $j\to\infty$. Then, Q - a.s.,*

$$\left(\int\!\!\int L\big(y,\widehat{f}_{P_{n_j,\zeta},\Lambda_{n_j,\zeta}}(x)\big)P(dy|x)P_{n_j,\zeta}\big(d(x,v)\big)-\inf_{t\in\mathbb{R}}\int L(y,t)\,P(dy|\xi)\right)\vee 0\ \ \xrightarrow[j\to\infty]{}\ 0\,.$$

Proof For every $n\in\mathbb{N}$, let $\tilde{P}_{n,\zeta}$ denote the conditional distribution of (X,Y) given $Z\in B_{n,\zeta}$. Then, for every integrable $g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$,

$$\int g(x,y)\tilde{P}_{n,\zeta}\big(d(x,y)\big)=\int g(x,y)P_{n,\zeta}\big(d(x,v,y)\big)=\int_\mathcal{Z}\!\int_\mathcal{Y} g(x,y)P(dy|x)P_{n,\zeta}\big(d(x,v)\big)$$

and, according to the definitions (45) and (6),

$$\inf_{f\in H}A_f(\tilde{P}_{n,\zeta},\lambda)\ =\ \int L\big(y,f_{P_{n,\zeta},\lambda}(x)\big)P_{n,\zeta}\big(d(x,v,y)\big)\ +\ \lambda\big\|f_{P_{n,\zeta},\lambda}\big\|_H^2 \tag{47}$$

for every $\lambda\in(0,\infty)$ and $n\in\mathbb{N}$. Analogously to the definition of $\tilde{P}_{n,\zeta}\in\mathcal{M}(\mathcal{X}\times\mathcal{Y})$, define $\tilde{P}_{0,\zeta}\in\mathcal{M}(\mathcal{X}\times\mathcal{Y})$ by

$$\int g(x,y)\,\tilde{P}_{0,\zeta}\big(d(x,y)\big)\ =\ \int_\mathcal{Y} g(\xi,y)\,P(dy|\xi)\qquad\text{for every integrable } g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}\,.$$

First, it is shown that

$$\tilde{P}_{n,\zeta}\ \xrightarrow[n\to\infty]{}\ \tilde{P}_{0,\zeta}\qquad\text{weakly in }\mathcal{M}(\mathcal{X}\times\mathcal{Y})\,. \tag{48}$$

According to Bauer (2001, Theorem 30.8), we have to show that

$$\int g\,d\tilde{P}_{n,\zeta} \xrightarrow[n\to\infty]{} \int g\,d\tilde{P}_{0,\zeta} \qquad \forall\, g \in \mathcal{C}_c(\mathcal{X}\times\mathcal{Y})\,. \tag{49}$$

Fix any $g \in \mathcal{C}_c(\mathcal{X}\times\mathcal{Y})$. Then, for every $\varepsilon > 0$, there is a $g_\varepsilon \in \mathcal{S}$ such that $\sup_{x,y} |g(x,y) - g_\varepsilon(x,y)| < \varepsilon$ and, therefore,

$$\begin{aligned}\left|\int g\,d\tilde{P}_{n,\zeta} - \int g\,d\tilde{P}_{0,\zeta}\right| \;&\le\; \int |g-g_\varepsilon|\,d\tilde{P}_{n,\zeta} + \left|\int g_\varepsilon\,d\tilde{P}_{n,\zeta} - \int g_\varepsilon\,d\tilde{P}_{0,\zeta}\right| + \int |g-g_\varepsilon|\,d\tilde{P}_{0,\zeta} \;\le \\ &\le\; 2\varepsilon + \int_{\mathcal{Z}} \left| \int_{\mathcal{Y}} g_\varepsilon(x,y)\,P(dy|x) - \int_{\mathcal{Y}} g_\varepsilon(\xi,y)\,P(dy|\xi)\right| P_{n,\zeta}\big(d(x,v)\big)\end{aligned}$$

The second summand converges to 0 for $n\to\infty$ because of $\xi \in B_2$, $g_\varepsilon \in \mathcal{S}$, and the definition of B_2. As $\varepsilon > 0$ can be arbitrarily close to 0, this shows (49) and, therefore, (48).
Next, fix any $\omega \in \Omega$ such that $\gamma_j := \Lambda_{n_j,\xi}(\omega) \longrightarrow 0$ for $j\to\infty$. Then,

$$\begin{aligned}&\limsup_{j\to\infty} \iint L\big(y, \widehat{f}_{P_{n_j,\zeta},\Lambda_{n_j,\zeta}(\omega)}(x)\big) P(dy|x) P_{n_j,\zeta}\big(d(x,v)\big) \;\le \\ &\quad\le\quad \limsup_{j\to\infty} \iint L\big(y, f_{P_{n_j,\zeta},\gamma_j}(x)\big) P(dy|x) P_{n_j,\zeta}\big(d(x,v)\big) + \gamma_j \big\| f_{P_{n_j,\zeta},\gamma_j}\big\|_H^2 \;= \\ &\overset{(47)}{=}\quad \limsup_{j\to\infty} \inf_{f\in H} A_f(\tilde{P}_{n_j,\zeta},\gamma_j) \overset{(46,48)}{\le} \inf_{f\in H} A_f(\tilde{P}_{0,\zeta},0) \;= \\ &\quad=\quad \inf_{f\in H} \int L\big(y, f(\xi)\big)\,P(dy|\xi) \;=\; \inf_{t\in\mathbb{R}} \int L(y,t)\,P(dy|\xi)\,.\end{aligned}$$

■

By use of the above lemma, we can complete the proof of part 4 now. The definition of $\Lambda_{n,\zeta}$ and (31) imply that $\Lambda_{n,\xi} \to 0$ in probability for $n\to\infty$. Then, via the characterization of convergence in probability by use of subsequences and almost sure convergence, it follows from Lemma 8 that

$$\left(\iint L\big(y, \widehat{f}_{P_{n,\zeta},\Lambda_{n,\zeta}}(x)\big) P(dy|x) P_{n,\zeta}\big(d(x,v)\big) - \inf_{t\in\mathbb{R}} \int L(y,t)\,P(dy|\xi)\right) \vee 0 \xrightarrow[n\to\infty]{} 0$$

in probability.

Appendix D. Stability Properties of Support Vector Machines

Part (a) of the following Lemma 9 shows: in order to ensure that empirical SVMs are continuous in the data, continuity of the loss function L is enough. This result strengthens (Steinwart and Christmann, 2008, Lemma 5.13) which assumes differentiability and also (Hable and Christmann, 2011, Corollary 3.5) which assumes Lipschitz-(equi-)continuity. Next, part (i) of Lemma 9 (b) considerably strengthens (Steinwart and Christmann, 2008, Corollary 5.19) in the sense that it quantifies the continuity of the map $\lambda \mapsto f_{P_0,\lambda}$. Finally, parts (ii) and (iii) of Lemma 9 (b) are just simple applications of the stability results in (Steinwart and Christmann, 2008, § 5.3).

Lemma 9 *Let $\mathcal{X}_0$ be a separable metric space and let $\mathcal{Y}_0 \subset \mathbb{R}$ be closed. Let $K : \mathcal{X}_0\times\mathcal{X}_0 \to \mathbb{R}$ be a continuous and bounded kernel with RKHS H and canonical feature map Φ. Let $L : \mathcal{Y}_0\times\mathbb{R} \to [0,\infty)$ be a convex loss function.*

(a) If $L : \mathcal{Y}_0 \times \mathbb{R} \to [0,\infty)$ is continuous, then the map

$$(\mathcal{X}_0 \times \mathcal{Y}_0)^n \;\to\; H\,, \qquad D \;\mapsto\; f_{D,\lambda}$$

is continuous for every $\lambda > 0$ and $n \in \mathbb{N}$.

(b) Assume that L has the local Lipschitz-property that, for every $a \in (0,\infty)$, there is an $|L|_{a,1} \in (0,\infty)$ such that

$$\sup_{y\in\mathcal{Y}_0} \bigl|L(y,t_1) - L(y,t_2)\bigr| \;\le\; |L|_{a,1} \cdot |t_1 - t_2| \qquad \forall\, t_1,t_2 \in [-a,a]\,.$$

(i) Then, for every probability measure P_0 on $(\mathcal{X}_0 \times \mathcal{Y}_0, \mathfrak{B}_{\mathcal{X}_0\times\mathcal{Y}_0})$ such that $\mathcal{R}_{P_0}(0) < \infty$ and for every $\lambda_0,\lambda_1 \in (0,\infty)$, it holds that

$$\bigl\|f_{P_0,\lambda_1} - f_{P_0,\lambda_0}\bigr\|_H \;\le\; \frac{|\lambda_1 - \lambda_0|}{\lambda_1\sqrt{\lambda_0}} 2\sqrt{\mathcal{R}_{P_0}(0)}\,.$$

(ii) If there are some $b_0,b_1 \in (0,\infty)$ and $q \in [0,\infty)$ such that, for every $a \in (0,\infty)$, $|L|_{a,1} = b_0 + b_1 a^q$, then: for every probability measures P_1 on $(\mathcal{X}_0 \times \mathcal{Y}_0, \mathfrak{B}_{\mathcal{X}_0\times\mathcal{Y}_0})$ such that $\mathcal{R}_{P_1}(0) < \infty$ and for every $\lambda \in (0,\infty)$, there is a measurable $h_{P_1,\lambda} : \mathcal{X}_0 \times \mathcal{Y}_0 \to \mathbb{R}$ such that

$$\bigl|h_{P_1,\lambda}(x,y)\bigr| \;\le\; b_0 + b_1\|K\|_\infty^q \left(\frac{\mathcal{R}_{P_1}(0)}{\lambda}\right)^{\frac{q}{2}} \tag{50}$$

and such that, for every P_2 on $(\mathcal{X}_0 \times \mathcal{Y}_0, \mathfrak{B}_{\mathcal{X}_0\times\mathcal{Y}_0})$ with $\mathcal{R}_{P_2}(0) < \infty$,

$$\begin{aligned}\bigl\|f_{P_1,\lambda} - f_{P_2,\lambda}\bigr\|_H \;&\le\; \lambda^{-1}\left\|\int h_{P_1,\lambda}\Phi\,dP_1 - \int h_{P_1,\lambda}\Phi\,dP_1\right\|_H = \\ &=\; \lambda^{-1}\sup_{\substack{f\in H\\ \|f\|_H\le 1}} \bigl|\mathbb{E}_{P_1}h_{P_1,\lambda}f - \mathbb{E}_{P_2}h_{P_1,\lambda}f\bigr|\,.\end{aligned}$$

(iii) If there are some $b_0,b_1 \in (0,\infty)$ and $q \in [0,\infty)$ such that, for every $a \in (0,\infty)$, $|L|_{a,1} = b_0 + b_1 a^q$, then: for every probability measures P_1 and P_2 on $(\mathcal{X}_0 \times \mathcal{Y}_0, \mathfrak{B}_{\mathcal{X}_0\times\mathcal{Y}_0})$ such that $\mathcal{R}_{P_1}(0) < \infty$ and $\mathcal{R}_{P_2}(0) < \infty$ and for every $\lambda \in (0,\infty)$,

$$\bigl\|f_{P_1,\lambda} - f_{P_2,\lambda}\bigr\|_H \;\le\; \|K\|_\infty\Bigl(b_0\lambda^{-1} + b_1\|K\|_\infty^q \mathcal{R}_{P_1}(0)^{\frac{q}{2}}\lambda^{-\frac{q}{2}-1}\Bigr)\bigl\|P_1 - P_2\bigr\|_{\mathrm{TV}}\,. \tag{51}$$

Proof In order to prove (a) by contradiction, assume that $D \mapsto f_{D,\lambda}$ is not continuous. Then, there is an $\varepsilon > 0$ and a sequence, such that

$$D^{(m)} \;\xrightarrow[m\to\infty]{}\; D^{(0)} \qquad \text{and} \qquad \bigl\|f_{D^{(m)},\lambda} - f_{D^{(0)},\lambda}\bigr\|_H \;\ge\; \varepsilon \quad \forall\, m \in \mathbb{N}\,. \tag{52}$$

Define $\mathcal{R}_D(f) = \frac{1}{n}\sum_{i=1}^n L(y_i, f(x_i))$ for every $D = \bigl((x_1,y_1),\ldots,(x_n,y_n)\bigr) \in (\mathcal{X}_0 \times \mathcal{Y}_0)^n$ and $f \in H$. According to (Steinwart and Christmann, 2008, (5.4)) and due to continuity of L,

$$\sup_{m\in\mathbb{N}} \bigl\|f_{D^{(m)},\lambda}\bigr\|_H \;\le\; \sup_{m\in\mathbb{N}} \sqrt{\lambda^{-1}\mathcal{R}_{D^{(m)}}(0)} \;<\; \infty\,. \tag{53}$$

Hence, there is a subsequence such that $f_{D^{(m_\ell)},\lambda}$ weakly converges to some $f_0 \in H$ in the Hilbert space H for $\ell \to \infty$; see, for example, (Dunford and Schwartz, 1958, Corollary IV.4.7). That is, there is also a sequence which fulfills (52) and such that, in addition, $f_{D^{(m)},\lambda}$ weakly converges to f_0 in H for some $f_0 \in H$. This implies

$$\lim_{m\to\infty} f_{D^{(m)},\lambda}(x) \;=\; \lim_{m\to\infty} \left\langle f_{D^{(m)},\lambda}, \Phi(x)\right\rangle_H \;=\; \left\langle f_0, \Phi(x)\right\rangle_H \;=\; f_0(x) \qquad \forall\, x \in \mathcal{X}_0$$

and, for $x^{(m)} \to x^{(0)}$ in $\mathcal{X}_0$,

$$\begin{aligned}
&\lim_{m\to\infty} \left| f_{D^{(m)},\lambda}(x^{(m)}) - f_0(x^{(0)})\right| \;\leq \\
&\quad\leq \lim_{m\to\infty} \left|\left\langle f_{D^{(m)},\lambda}, \Phi(x^{(m)}) - \Phi(x^{(0)})\right\rangle_H\right| + \lim_{m\to\infty} \left| f_{D^{(m)},\lambda}(x^{(0)}) - f_0(x^{(0)})\right| \\
&\quad\leq \lim_{m\to\infty} \left\| f_{D^{(m)},\lambda}\right\|_H \cdot \left\|\Phi(x^{(m)}) - \Phi(x^{(0)})\right\|_H \;=\; 0
\end{aligned}$$

where the last equality follows from (53) and continuity of the kernel K. Hence, it follows that

$$\lim_{m\to\infty} \mathcal{R}_{\boldsymbol{D}^{(m)}}\left(f_{D^{(m)},\lambda}\right) \;=\; \mathcal{R}_{\boldsymbol{D}^{(0)}}\left(f_0\right). \tag{54}$$

Therefore, lower semi-continuity of the H-norm with respect to weak convergence in H (e.g., Conway, 1985, Exercise V.1.9) implies

$$\liminf_{m\to\infty} \mathcal{R}_{\boldsymbol{D}^{(m)}}\left(f_{D^{(m)},\lambda}\right) + \lambda\left\|f_{D^{(m)},\lambda}\right\|_H^2 \;\geq\; \mathcal{R}_{\boldsymbol{D}^{(0)}}\left(f_0\right) + \lambda\left\|f_0\right\|_H^2. \tag{55}$$

Recall that the point-wise infimum of a familiy of continuous functions yields an upper semi-continuous function; see, for example, (Denkowski et al., 2003, Prop. 1.1.36). Then, the definition of $f_{D^{(m)},\lambda}$ and continuity of $D \mapsto \mathcal{R}_D(f) + \lambda\|f\|_H^2$ for every $f \in H$ imply

$$\begin{aligned}
&\mathcal{R}_{\boldsymbol{D}^{(0)}}\left(f_0\right) + \lambda\left\|f_0\right\|_H^2 \;\geq\; \inf_{f\in H}\left(\mathcal{R}_{\boldsymbol{D}^{(0)}}(f) + \lambda\|f\|_H^2\right) \;\geq \\
&\quad\geq\; \limsup_{m\to\infty} \inf_{f\in H}\left(\mathcal{R}_{\boldsymbol{D}^{(m)}}(f) + \lambda\|f\|_H^2\right) \;=\; \limsup_{m\to\infty} \mathcal{R}_{\boldsymbol{D}^{(m)}}\left(f_{D^{(m)},\lambda}\right) + \lambda\left\|f_{D^{(m)},\lambda}\right\|_H^2 \;\geq \\
&\quad\geq\; \liminf_{m\to\infty} \mathcal{R}_{\boldsymbol{D}^{(m)}}\left(f_{D^{(m)},\lambda}\right) + \lambda\left\|f_{D^{(m)},\lambda}\right\|_H^2 \;\overset{(55)}{\geq}\; \mathcal{R}_{\boldsymbol{D}^{(0)}}\left(f_0\right) + \lambda\left\|f_0\right\|_H^2.
\end{aligned}$$

Hence, it follows that $f_0 = f_{D^{(0)},\lambda}$ and

$$\lim_{m\to\infty} \mathcal{R}_{\boldsymbol{D}^{(m)}}\left(f_{D^{(m)},\lambda}\right) + \lambda\left\|f_{D^{(m)},\lambda}\right\|_H^2 \;=\; \mathcal{R}_{\boldsymbol{D}^{(0)}}\left(f_{D^{(0)},\lambda}\right) + \lambda\left\|f_{D^{(0)},\lambda}\right\|_H^2. \tag{56}$$

Then, $f_0 = f_{D^{(0)},\lambda}$, (54), and (56) imply that $\lim_{m\to\infty}\|f_{D^{(m)},\lambda}\|_H = \|f_{D^{(0)},\lambda}\|_H$. Since weak convergence in the Hilbert space H and this convergence of the H-norms imply norm convergence in H (see, e.g., Conway, 1985, Exercise V.1.8), we have shown that $\lim_{m\to\infty}\|f_{D^{(m)},\lambda} - f_{D^{(0)},\lambda}\|_H = 0$, which is a contradiction to (52).

The following proof of part (i) of Lemma 9 (b) is essentially a variant of the proof of (Steinwart and Christmann, 2008, Theorem 5.9) even though the statements are quite different. Let $\partial L(y,t_0)$ denote the subdifferential of the convex map $t \mapsto L(y,t)$ at the point t_0. According to (Steinwart and

Christmann, 2008, Corollary 5.10), there is a bounded measurable map $h :\in \mathcal{X}_0 \times \mathcal{Y}_0 \to \mathbb{R}$ such that $h(x,y) \in \partial L\big(y, f_{P_0,\lambda_0}(x)\big)$ for every $(x,y) \in \mathcal{X}_0 \times \mathcal{Y}_0$ and

$$f_{P_0,\lambda_0} \;=\; -\frac{1}{2\lambda_0} \int h \Phi \, dP_0 \,. \tag{57}$$

The definition of the subdifferential implies

$$h(x,y)\big(f_{P_0,\lambda_1}(x) - f_{P_0,\lambda_0}(x)\big) \;\le\; L\big(y, f_{P_0,\lambda_1}(x)\big) - L\big(y, f_{P_0,\lambda_0}(x)\big)$$

for every $(x,y) \in \mathcal{X}_0 \times \mathcal{Y}_0$ and integrating with respect to P_0 yields

$$\int h(x,y)\big(f_{P_0,\lambda_1}(x) - f_{P_0,\lambda_0}(x)\big) P_0\big(d(x,y)\big) \;\le\; \mathcal{R}(f_{P_0,\lambda_1}) - \mathcal{R}(f_{P_0,\lambda_0}) \,.$$

The reproducing property of the canonical feature map Φ and the property of the Bochner integral (Denkowski et al., 2003, Theorem 3.10.16) imply

$$\begin{aligned}
&\int h(x,y)\big(f_{P_0,\lambda_1}(x) - f_{P_0,\lambda_0}(x)\big) P_0\big(d(x,y)\big) \;= \\
&\quad= \int \big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, h(x,y)\Phi(x)\big\rangle_H \, P_0\big(d(x,y)\big) \;= \\
&\quad= \big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, \textstyle\int h\Phi \, dP_0\big\rangle_H \overset{(57)}{=} \big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, -2\lambda_0 f_{P_0,\lambda_0}\big\rangle_H \,.
\end{aligned}$$

That is,

$$\big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, -2\tfrac{\lambda_0}{\lambda_1} f_{P_0,\lambda_0}\big\rangle_H \;\le\; \tfrac{1}{\lambda_1}\big(\mathcal{R}_{P_0}(f_{P_0,\lambda_1}) - \mathcal{R}_{P_0}(f_{P_0,\lambda_0})\big) \,. \tag{58}$$

An elementary calculation with $\langle \, , \rangle_H$ shows that

$$2\big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0}, f_{P_0,\lambda_0}\big\rangle_H + \big\| f_{P_0,\lambda_1} - f_{P_0,\lambda_0}\big\|_H^2 = \big\| f_{P_0,\lambda_1}\big\|_H^2 - \big\| f_{P_0,\lambda_0}\big\|_H^2 . \tag{59}$$

Calculating (58)+(59) yields

$$\begin{aligned}
&\big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, 2(1-\tfrac{\lambda_0}{\lambda_1}) f_{P_0,\lambda_0}\big\rangle_H + \big\| f_{P_0,\lambda_1} - f_{P_0,\lambda_0}\big\|_H^2 \;\le \\
&\qquad\le\; \tfrac{1}{\lambda_1}\big(\mathcal{R}_{P_0}(f_{P_0,\lambda_1}) - \mathcal{R}_{P_0}(f_{P_0,\lambda_0})\big) + \big\| f_{P_0,\lambda_1}\big\|_H^2 - \big\| f_{P_0,\lambda_0}\big\|_H^2 \;= \\
&\qquad=\; \tfrac{1}{\lambda_1}\big(\mathcal{R}_{P_0,\lambda_1}(f_{P_0,\lambda_1}) - \mathcal{R}_{P_0,\lambda_1}(f_{P_0,\lambda_0})\big) \;\le\; 0 \,.
\end{aligned}$$

Hence,

$$\begin{aligned}
\big\| f_{P_0,\lambda_1} - f_{P_0,\lambda_0}\big\|_H^2 \;&\le\; \Big|\big\langle f_{P_0,\lambda_1} - f_{P_0,\lambda_0} \,, 2(1-\tfrac{\lambda_0}{\lambda_1}) f_{P_0,\lambda_0}\big\rangle_H\Big| \;\le \\
&\le\; \big\| f_{P_0,\lambda_1} - f_{P_0,\lambda_0}\big\|_H \cdot 2\big|1-\tfrac{\lambda_0}{\lambda_1}\big| \cdot \big\| f_{P_0,\lambda_0}\big\|_H \,.
\end{aligned}$$

Since $\big\| f_{P_0,\lambda_0}\big\|_H \le \sqrt{\lambda_0^{-1}\mathcal{R}_{P_0}(0)}$ (see, e.g., Steinwart and Christmann, 2008, (5.4)), this implies statement (i) of Lemma 9 (b).

In order to prove (ii) and (iii) of Lemma 9 (b), note that the properties of the Bochner-Integral (see, e.g., Denkowski et al., 2003, Theorem 3.10.16) imply $\langle \int h\Phi \, dP, f\rangle_H = \int hf \, dP$ for every integrable function $h : \mathcal{X}_0 \times \mathcal{Y}_0 \to \mathbb{R}$ because of the reproducing property $\langle \Phi(x), f\rangle_H = f(x)$. Due to the

assumptions on L, it follows from (Steinwart and Christmann, 2008, Corollary 5.10) that there is a measurable function $h_{P_1,\lambda} : \mathcal{X}_0 \times \mathcal{Y}_0 \to \mathbb{R}$ which fulfills (50) and

$$\left\|f_{P_1,\lambda} - f_{P_2,\lambda}\right\|_H \leq \frac{1}{\lambda}\left\|\int h_{P_1,\lambda}\Phi\, dP_1 - \int h_{P_1,\lambda}\Phi\, dP_2\right\|_H =$$
$$= \sup_{\substack{f\in H\\ \|f\|_H\leq 1}} \frac{1}{\lambda}\left\langle \int h_{P_1,\lambda}\Phi\, dP_1 - \int h_{P_1,\lambda}\Phi\, dP_2\ ,\ f\right\rangle_H = \sup_{\substack{f\in H\\ \|f\|_H\leq 1}} \frac{1}{\lambda}\left|\int h_{P_1,\lambda} f\, dP_1 - \int h_{P_1,\lambda} f\, dP_2\right|.$$

That is, we have shown (ii). Then, (iii) follows from (ii) and $\|f\|_\infty \leq \|K\|_\infty \|f\|_H$. ■

References

Heinz Bauer. *Measure and Integration Theory*. Walter de Gruyter & Co., Berlin, 2001.

Kristin P. Bennett and Jennifer A. Blue. A support vector machine approach to decision trees. In *Proceedings IEEE International Joint Conference on Neural Networks*, 1998.

Enrico Blanzieri and Anton Bryl. Instance-based spam filtering using SVM nearest neighbor classifier. In *The 20th International FLAIRS Conference*, pages 441–442, 2007a.

Enrico Blanzieri and Anton Bryl. Evaluation of the highest probability SVM nearest neighbor classifier with variable relative error cost. In *Fourth Conference on Email and Anti-Spam CEAS 2007*, 2007b. URL `http://www.ceas.cc/2007/papers/paper-42_upd.pdf`.

Enrico Blanzieri and Farid Melgani. Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Transactions on Geoscience and Remote Sensing*, 46(6): 1804–1811, 2008.

Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

Fu Chang, Chien-Yang Guo, Xiao-Rong Lin, and Chi-Jen Lu. Tree decomposition for large-scale SVM problems. *Journal of Machine Learning Research*, 11:2935–2972, 2010.

Haibin Cheng, Pang-Ning Tan, and Rong Jin. Localized support vector machine and its efficient algorithm. In *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007. URL `http://www.siam.org/proceedings/datamining/2007/dm07_045cheng.pdf`.

Haibin Cheng, Pang-Ning Tan, and Rong Jin. Efficient algorithm for localized support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, 22(4):537–549, 2010.

Andreas Christmann and Robert Hable. Consistency of support vector machines using additive kernels for additive models. *Computational Statistics and Data Analysis*, 56:854–873, 2012.

Andreas Christmann and Ingo Steinwart. Consistency and robustness of kernel-based regression in convex risk minimization. *Bernoulli*, 13(3):799–819, 2007.

John B. Conway. *A Course in Functional Analysis*. Springer-Verlag, New York, 1985.

Zdzislaw Denkowski, Stanislaw Migórski, and Nikolas S. Papageorgiou. *An Introduction to Nonlinear Analysis: Theory*. Kluwer Academic Publishers, Boston, 2003.

Luc Devroye, László Györfi, Adam Krzyżak, and Gábor Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *The Annals of Statistics*, 22:1371–1385, 1994.

Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.

Richard M. Dudley. *Real Analysis and Probability*. Cambridge University Press, Cambridge, 2002. Revised reprint of the 1989 original.

Nelson Dunford and Jacob T. Schwartz. *Linear Operators. I. General Theory*. Wiley-Interscience Publishers, New York, 1958.

Jinqing Fan and Irène Gijbels. *Local Polynomial Modelling and its Applications*. Chapman & Hall, London, 1996.

David H. Fremlin. *Measure Theory. Vol. 4*. Torres Fremlin, Colchester, 2006.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*. Springer, New York, 2002.

Robert Hable and Andreas Christmann. On qualitative robustness of support vector machines. *Journal of Multivariate Analysis*, 102:993–1007, 2011.

Steven G. Krantz and Harold R. Parks. *Geometric Integration Theory*. Birkhäuser, Basel, 2008.

Kalyanapuram R. Parthasarathy. *Probability Measures on Metric Spaces*. Academic Press Inc., New York, 1967.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

Nicola Segata and Enrico Blanzieri. Empirical assessment of classification accuracy of local SVM. In *The 18th Annual Belgian-Dutch Conference on Machine Learning (Benelearn 2009)*, pages 47–55, Tilburg, Belgium, 2009.

Nicola Segata and Enrico Blanzieri. Fast and scalable local kernel machines. *Journal of Machine Learning Research*, 11:1883–1926, 2010.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, New York, 2008.

Vladimir Vapnik and Léon Bottou. Local algorithms for pattern-recognition and dependencies estimation. *Neural Computation*, 5(6):893–909, 1993.

Donghui Wu, Kristin P. Bennett, Nello Cristianini, John Shawe-Taylor, and Royal Holloway. Large margin trees for induction and transduction. In *Proceedings of International Conference on Machine Learning*, pages 474–483, 1999.

Alon Zakai. *Towards a Theory of Learning in High-Dimensional Spaces*. PhD thesis, The Hebrew University of Jerusalem, 2008. URL `http://icnc.huji.ac.il/phd/theses/files/AlonZakai.pdf`.

Alon Zakai and Ya'acov Ritov. Consistency and localizability. *Journal of Machine Learning Research*, 10:827–856, 2009.

Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 2126–2136. IEEE Computer Society, 2006.

Journal of Machine Learning Research 14 (2013) 187-207 Submitted 12/11; Revised 9/12; Published 1/13

Lower Bounds and Selectivity of Weak-Consistent Policies in Stochastic Multi-Armed Bandit Problem

Antoine Salomon SALOMONA@IMAGINE.ENPC.FR
Jean-Yves Audibert* AUDIBERT@IMAGINE.ENPC.FR
Issam El Alaoui ISSAM.EL-ALAOUI.2007@POLYTECHNIQUE.ORG
Imagine, Université Paris-Est
6 Avenue Blaise Pascal
77455 Champs-sur-Marne, France

Editor: Nicolo Cesa-Bianchi

Abstract

This paper is devoted to regret lower bounds in the classical model of stochastic multi-armed bandit. A well-known result of Lai and Robbins, which has then been extended by Burnetas and Katehakis, has established the presence of a logarithmic bound for all consistent policies. We relax the notion of consistency, and exhibit a generalisation of the bound. We also study the existence of logarithmic bounds in general and in the case of Hannan consistency. Moreover, we prove that it is impossible to design an adaptive policy that would select the best of two algorithms by taking advantage of the properties of the environment. To get these results, we study variants of popular Upper Confidence Bounds (UCB) policies.

Keywords: stochastic bandits, regret lower bounds, consistency, selectivity, UCB policies

1. Introduction and Notations

Multi-armed bandits are a classical way to illustrate the difficulty of decision making in the case of a dilemma between exploration and exploitation. The denomination of these models comes from an analogy with playing a slot machine with more than one arm. Each arm has a given (and unknown) reward distribution and, for a given number of rounds, the agent has to choose one of them. As the goal is to maximize the sum of rewards, each round decision consists in a trade-off between exploitation (i.e., playing the arm that has been the more lucrative so far) and exploration (i.e., testing another arm, hoping to discover an alternative that beats the current best choice). One possible application is clinical trial: a doctor wants to heal as many patients as possible, the patients arrive sequentially, and the effectiveness of each treatment is initially unknown (Thompson, 1933). Bandit problems have initially been studied by Robbins (1952), and since then they have been applied to many fields such as economics (Lamberton et al., 2004; Bergemann and Valimaki, 2008), games (Gelly and Wang, 2006), and optimisation (Kleinberg, 2005; Coquelin and Munos, 2007; Kleinberg et al., 2008; Bubeck et al., 2009).

∗. Also at Willow, CNRS/ENS/INRIA - UMR 8548.

1.1 Setting

In this paper, we consider the following model. A stochastic multi-armed bandit problem is defined by:

- a number of rounds n,
- a number of arms $K \geq 2$,
- an environment $\theta = (\nu_1, \cdots, \nu_K)$, where each ν_k ($k \in \{1, \cdots, K\}$) is a real-valued measure that represents the distribution reward of arm k.

The number of rounds n may or may not be known by the agent, but this will not affect the present study.

We assume that rewards are bounded. Thus, for simplicity, each ν_k is a probability on $[0,1]$. Environment θ is initially unknown by the agent but lies in some known set Θ. For the problem to be interesting, the agent should not have great knowledges of its environment, so that Θ should not be too small and/or only contain too trivial distributions such as Dirac measures. To make it simple, we may assume that Θ contains all environments where each reward distribution is a Dirac distribution or a Bernoulli distribution. This will be acknowledged as Θ having the *Dirac/Bernoulli property*. For technical reason, we may also assume that Θ is of the form $\Theta_1 \times \ldots \times \Theta_K$, meaning that Θ_k is the set of possible reward distributions of arm k. This will be acknowledged as Θ having the *product property*.

The game is as follows. At each round (or time step) $t = 1, \cdots, n$, the agent has to choose an arm I_t in the set $\{1, \cdots, K\}$. This decision is based on past actions and observations, and the agent may also randomize his choice. Once the decision is made, the agent gets and observes a reward that is drawn from ν_{I_t} independently from the past. Thus a policy (or strategy) can be described by a sequence $(\sigma_t)_{t \geq 1}$ (or $(\sigma_t)_{1 \leq t \leq n}$ if the number of rounds n is known) such that each σ_t is a mapping from the set $\{1, \ldots, K\}^{t-1} \times [0,1]^{t-1}$ of past decisions and rewards into the set of arm $\{1, \ldots, K\}$ (or into the set of probabilities on $\{1, \ldots, K\}$, in case the agent randomizes his choices).

For each arm k and all time step t, let $T_k(t) = \sum_{s=1}^{t} \mathbb{1}_{I_s = k}$ denote the sampling time, that is, the number of times arm k was pulled from round 1 to round t, and $X_{k,1}, X_{k,2}, \ldots, X_{k,T_k(t)}$ the corresponding sequence of rewards. We denote by $\mathbb{P}_\theta$ the distribution on the probability space such that for any $k \in \{1, \ldots, K\}$, the random variables $X_{k,1}, X_{k,2}, \ldots, X_{k,n}$ are i.i.d. realizations of ν_k, and such that these K sequences of random variables are independent. Let $\mathbb{E}_\theta$ denote the associated expectation.

Let $\mu_k = \int x d\nu_k(x)$ be the mean reward of arm k. Introduce $\mu^* = \max_{k \in \{1,\ldots,K\}} \mu_k$ and fix an arm $k^* \in \operatorname{argmax}_{k \in \{1,\ldots,K\}} \mu_k$, that is, k^* has the best expected reward. The agent aims at minimizing its *regret*, defined as the difference between the cumulative reward he would have obtained by always drawing the best arm and the cumulative reward he actually received. Its regret is thus

$$R_n = \sum_{t=1}^{n} X_{k^*,t} - \sum_{t=1}^{n} X_{I_t, T_{I_t}(t)}.$$

As most of the publications on this topic, we focus on the expected regret, for which one can check that:

$$\mathbb{E}_\theta R_n = \sum_{k=1}^{K} \Delta_k \mathbb{E}_\theta[T_k(n)], \tag{1}$$

where Δ_k is the *optimality gap* of arm k, defined by $\Delta_k = \mu^* - \mu_k$. We also define Δ as the gap between the best arm and the second best arm, that is, $\Delta := \min_{k:\Delta_k>0} \Delta_k$.

Other notions of regret exist in the literature. One of them is the quantity

$$\max_k \sum_{t=1}^{n} X_{k,t} - X_{I_t,T_{I_t}(t)},$$

which is mostly used in adversarial settings. Results and ideas we want to convey here are more suited to expected regret, and considering other definitions of regret would only bring some more technical intricacies.

1.2 Consistency and Regret Lower Bounds

Former works have shown the existence of lower bounds on the expected regret of a large class of policies: intuitively, to perform well the agent has to explore all arms, and this requires a significant amount of suboptimal choices. In this way, Lai and Robbins (1985) proved a lower bound of order $\log n$ in a particular parametric framework, and they also exhibited optimal policies. This work has then been extended by Burnetas and Katehakis (1996). Both papers deal with *consistent* policies, meaning that they only consider policies such that:

$$\forall a > 0,\ \forall \theta \in \Theta,\ \mathbb{E}_\theta[R_n] = o(n^a). \tag{2}$$

Let us detail the bound of Burnetas and Katehakis, which is valid when Θ has the product property. Given an environment $\theta = (\nu_1, \cdots, \nu_K)$ and an arm $k \in \{1, \dots, K\}$, define:

$$D_k(\theta) := \inf_{\tilde{\nu}_k \in \Theta_k : \mathbb{E}[\tilde{\nu}_k] > \mu^*} KL(\nu_k, \tilde{\nu}_k),$$

where $KL(\nu, \mu)$ denotes the Kullback-Leibler divergence of measures ν and μ. Now fix a consistent policy and an environment $\theta \in \Theta$. If k is a suboptimal arm (i.e., $\mu_k \neq \mu^*$) such that $0 < D_k(\theta) < \infty$, then

$$\forall \varepsilon > 0,\ \lim_{n\to+\infty} \mathbb{P}\left[T_k(n) \geq \frac{(1-\varepsilon)\log n}{D_k(\theta)}\right] = 1.$$

This readily implies that:

$$\liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[T_k(n)]}{\log n} \geq \frac{1}{D_k(\theta)}.$$

Thanks to Formula (1), it is then easy to deduce a lower bound of the expected regret.

One contribution of this paper is to generalize the study of regret lower bounds, by considering weaker notions of consistency: α-consistency and Hannan consistency. We will define α-consistency ($\alpha \in [0,1)$) as a variant of Equation (2), where equality $\mathbb{E}_\theta[R_n] = o(n^a)$ only holds for all $a > \alpha$. We show that the logarithmic bound of Burnetas and Katehakis still holds, but coefficient $\frac{1}{D_k(\theta)}$ is turned into $\frac{1-\alpha}{D_k(\theta)}$. We also prove that the dependence of this new bound with respect to the term $1-\alpha$ is asymptotically optimal when $n \to +\infty$ (up to a constant).

We will also consider the case of Hannan consistency. Indeed, any policy achieves at most an expected regret of order n: because of the equality $\sum_{k=1}^{K} T_k(n) = n$ and thanks to Equation (1), one can show that $\mathbb{E}_\theta R_n \leq n \max_k \Delta_k$. More intuitively, this comes from the fact that the average cost of pulling an arm k is a constant Δ_k. As a consequence, it is natural to wonder what happens when

dealing with policies whose expected regret is only required to be $o(n)$, which is equivalent to Hannan consistency. This condition is less restrictive than any of the previous notions of consistency. In this larger class of policy, we show that the lower bounds on the expected regret are no longer logarithmic, but can be much smaller.

Finally, even if no logarithmic lower bound holds on the whole set Θ, we show that there necessarily exist some environments θ for which the expected regret is at least logarithmic. The latter result is actually valid without any assumptions on the considered policies, and only requires a simple property on Θ.

1.3 Selectivity

As we exhibit new lower bounds, we want to know if it is possible to provide optimal policies that achieve these lower bounds, as it is the case in the classical class of consistent policies. We answer negatively to this question, and for this we solve the more general problem of selectivity. Given a set of policies, we define selectivity as the ability to perform at least as good as the policy that is best suited to the current environment θ. Still, such an ability can not be implemented. As a by-product it is not possible to design a procedure that would specifically adapt to some kinds of environments, for example by selecting a particular policy. This contribution is linked with selectivity in on-line learning problem with perfect information, commonly addressed by prediction with expert advice (see, e.g., Cesa-Bianchi et al., 1997). In this spirit, a closely related problem is the one of regret against the best strategy from a pool studied by Auer et al. (2003). The latter designed an algorithm in the context of adversarial/nonstochastic bandit whose decisions are based on a given number of recommendations (experts), which are themselves possibly the rewards received by a set of given policies. To a larger extent, model selection has been intensively studied in statistics, and is commonly solved by penalization methods (Mallows, 1973; Akaike, 1973; Schwarz, 1978).

1.4 UCB Policies

Some of our results are obtained using particular Upper Confidence Bound algorithms. These algorithms were introduced by Lai and Robbins (1985): they basically consist in computing an index for each arm, and selecting the arm with the greatest index. A simple and efficient way to design such policies is as follows: choose each index as low as possible such that, conditional to past observations, it is an upper bound of the mean reward of the considered arm with high probability (or, say, with high confidence level). This idea can be traced back to Agrawal (1995), and has been popularized by Auer et al. (2002), who notably described a policy called UCB1. In this policy, each index $B_{k,s,t}$ is defined by an arm k, a time step t, an integer s that indicates the number of times arm k has been pulled before round t, and is given by:

$$B_{k,s,t} = \hat{X}_{k,s} + \sqrt{\frac{2\log t}{s}},$$

where $\hat{X}_{k,s}$ is the empirical mean of arm k after s pulls, that is, $\hat{X}_{k,s} = \frac{1}{s}\sum_{u=1}^{s} X_{k,u}$.

To summarize, UCB1 policy first pulls each arm once and then, at each round $t > K$, selects an arm k that maximizes $B_{k,T_k(t-1),t}$. Note that, by means of Hoeffding's inequality, the index $B_{k,T_k(t-1),t}$ is indeed an upper bound of μ_k with high probability (i.e., the probability is greater than $1 - 1/t^4$). Another way to understand this index is to interpret the empiric mean $\hat{X}_{k,T_k(t-1)}$ as an "exploitation"

term, and the square root $\sqrt{2\log t/s}$ as an "exploration" term (because the latter gradually increases when arm k is not selected).

Policy UCB1 achieves the logarithmic bound (up to a multiplicative constant), as it was shown that:

$$\forall \theta \in \Theta,\ \forall n \geq 3,\ \ \mathbb{E}_\theta[T_k(n)] \leq 12\frac{\log n}{\Delta_k^2} \ \text{ and } \ \mathbb{E}_\theta R_n \leq 12\sum_{k=1}^{K}\frac{\log n}{\Delta_k} \leq 12K\frac{\log n}{\Delta}.$$

Audibert et al. (2009) studied some variants of UCB1 policy. Among them, one consists in changing the $2\log t$ in the exploration term into $\rho\log t$, where $\rho > 0$. This can be interpreted as a way to tune exploration: the smaller ρ is, the better the policy will perform in simple environments where information is disclosed easily (for example when all reward distributions are Dirac measures). On the contrary, ρ has to be greater to face more challenging environments (typically when reward distributions are Bernoulli laws with close parameters).

This policy, that we denote UCB(ρ), was proven by Audibert et al. to achieve the logarithmic bound when $\rho > 1$, and the optimality was also obtained when $\rho > \frac{1}{2}$ for a variant of UCB(ρ). Bubeck (2010) showed in his PhD dissertation that their ideas actually enable to prove optimality of UCB(ρ) for $\rho > \frac{1}{2}$. Moreover, the case $\rho = \frac{1}{2}$ corresponds to a confidence level of $\frac{1}{t}$ (because of Hoeffding's inequality, as above), and several studies (Lai and Robbins, 1985; Agrawal, 1995; Burnetas and Katehakis, 1996; Audibert et al., 2009; Honda and Takemura, 2010) have shown that this level is critical.

We complete these works by a precise study of UCB(ρ) when $\rho \leq \frac{1}{2}$. We prove that UCB(ρ) is $(1-2\rho)$-consistent and that it is not α-consistent for any $\alpha < 1-2\rho$ (in view of the definition above, this means that the expected regret is roughly of order $n^{1-2\rho}$). Thus it does not achieve the logarithmic bound, but it performs well in simple environments, for example, environments where all reward distributions are Dirac measures.

Moreover, we exhibit expected regret bounds of general UCB policies, with the $2\log t$ in the exploration term of UCB1 replaced by an arbitrary function. We give sufficient conditions for such policies to be Hannan consistent and, as mentioned before, find that lower bounds need not be logarithmic any more.

1.5 Outline

The paper is organized as follows: in Section 2, we give bounds on the expected regret of general UCB policies and of UCB(ρ) ($\rho \leq \frac{1}{2}$), as preliminary results. In Section 3, we focus on α-consistent policies. Then, in Section 4, we study the problem of selectivity, and we conclude in Section 5 by general results on the existence of logarithmic lower bounds.

Throughout the paper $\lceil x \rceil$ denotes the smallest integer not less than x whereas $\lfloor x \rfloor$ denotes the largest integer not greater than x, $\mathbb{1}_A$ stands for the indicator function of event A, $Ber(p)$ is the Bernoulli law with parameter p, and δ_x is the Dirac measure centred on x.

2. Preliminary Results

In this section, we estimate the expected regret of UCB policies. This will be useful for the rest of the paper.

2.1 Bounds on the Expected Regret of General UCB Policies

We first study general UCB policies, defined by:

- Draw each arm once,
- then, at each round t, draw an arm

$$I_t \in \operatorname*{argmax}_{k\in\{1,\dots,K\}} B_{k,T_k(t-1),t},$$

where $B_{k,s,t}$ is defined by $B_{k,s,t} = \hat{X}_{k,s} + \sqrt{\frac{f_k(t)}{s}}$ and where functions f_k ($1 \le k \le K$) are increasing.

This definition is inspired by popular UCB1 algorithm, for which $f_k(t) = 2\log t$ for all k.

The following lemma estimates the performances of UCB policies in simple environments, for which reward distributions are Dirac measures.

Lemma 1 *Let $0 \le b < a \le 1$ and $n \ge 1$. For $\theta = (\delta_a, \delta_b)$, the random variable $T_2(n)$ is uniformly upper bounded by $\frac{1}{\Delta^2} f_2(n) + 1$. Consequently, the expected regret of* UCB *is upper bounded by $\frac{1}{\Delta} f_2(n) + 1$.*

Proof In environment θ, best arm is arm 1 and $\Delta = \Delta_2 = a - b$. Let us first prove the upper bound of the sampling time. The assertion is true for $n = 1$ and $n = 2$: the first two rounds consists in drawing each arm once, so that $T_2(n) \le 1 \le \frac{1}{\Delta^2} f_2(n) + 1$ for $n \in \{1,2\}$. If, by contradiction, the assertion is false, then there exists $t \ge 3$ such that $T_2(t) > \frac{1}{\Delta^2} f_2(t) + 1$ and $T_2(t-1) \le \frac{1}{\Delta^2} f_2(t-1) + 1$. Since $f_2(t) \ge f_2(t-1)$, this leads to $T_2(t) > T_2(t-1)$, meaning that arm 2 is drawn at round t. Therefore, we have $a + \sqrt{\frac{f_1(t)}{T_1(t-1)}} \le b + \sqrt{\frac{f_2(t)}{T_2(t-1)}}$, hence $a - b = \Delta \le \sqrt{\frac{f_2(t)}{T_2(t-1)}}$, which implies $T_2(t-1) \le \frac{1}{\Delta^2} f_2(t)$ and thus $T_2(t) \le \frac{1}{\Delta^2} f_2(t) + 1$. This contradicts the definition of t, and this ends the proof of the first statement. The second statement is a direct consequence of Formula (1). ■

Remark: throughout the paper, we will often use environments with $K = 2$ arms to provide bounds on expected regrets. However, we do not lose generality by doing so, because all corresponding proofs can be written almost identically to suit to any $K \ge 2$, by simply assuming that the distribution of each arm $k \ge 3$ is δ_0.

We now give an upper bound of the expected sampling time of any arm such that $\Delta_k > 0$. This bound is valid in any environment, and not only those of the form (δ_a, δ_b).

Lemma 2 *For any $\theta \in \Theta$ and any $\beta \in (0,1)$, if $\Delta_k > 0$ the following upper bound holds:*

$$\mathbb{E}_\theta[T_k(n)] \le u + \sum_{t=u+1}^{n} \left(1 + \frac{\log t}{\log(\frac{1}{\beta})}\right)\left(e^{-2\beta f_k(t)} + e^{-2\beta f_{k^*}(t)}\right),$$

where $u = \left\lceil \frac{4 f_k(n)}{\Delta_k^2} \right\rceil$.

An upper bound of the expected regret can be deduced from this lemma thanks to Formula 1.

Proof The core of the proof is a peeling argument and the use of Hoeffding's maximal inequality (see, e.g., Cesa-Bianchi and Lugosi, 2006, section A.1.3 for details). The idea is originally taken from Audibert et al. (2009), and the following is an adaptation of the proof of an upper bound of UCB(ρ) in the case $\rho > \frac{1}{2}$ which can be found in S. Bubeck's PhD dissertation.

First, let us notice that the policy selects an arm k such that $\Delta_k > 0$ at time step $t \leq n$ only if at least one of the three following equations holds:

$$B_{k^*,T_{k^*}(t-1),t} \leq \mu^*, \tag{3}$$

$$\hat{X}_{k,t} \geq \mu_k + \sqrt{\frac{f_k(t)}{T_k(t-1)}}, \tag{4}$$

$$T_k(t-1) < \frac{4f_k(n)}{\Delta_k^2}. \tag{5}$$

Indeed, if none of the equations is true, then:

$$B_{k^*,T_{k^*}(t-1),t} > \mu^* = \mu_k + \Delta_k \geq \mu_k + 2\sqrt{\frac{f_k(n)}{T_k(t-1)}} > \hat{X}_{k,t} + \sqrt{\frac{f_k(t)}{T_k(t-1)}} = B_{k,T_k(t-1),t},$$

which implies that arm k can not be chosen at time step t.

We denote respectively by $\xi_{1,t}, \xi_{2,t}$ and $\xi_{3,t}$ the events corresponding to Equations (3), (4) and (5).

We have:

$$\mathbb{E}_\theta[T_k(n)] \quad = \quad \mathbb{E}_\theta\left[\sum_{t=1}^n \mathbb{1}_{I_t=k}\right] = \mathbb{E}_\theta\left[\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}}\right] + \mathbb{E}_\theta\left[\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\setminus\xi_{3,t}}\right].$$

Let us show that the sum $\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}}$ is almost surely lower than $u := \lceil 4f_k(n)/\Delta_k^2 \rceil$. We assume by contradiction that $\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} > u$. Then there exists $m < n$ such that $\sum_{t=1}^{m-1} \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} < 4f_k(n)/\Delta_k^2$ and $\sum_{t=1}^m \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} = \lceil 4f_k(n)/\Delta_k^2 \rceil$. Therefore, for any $s > m$, we have:

$$T_k(s-1) \geq T_k(m) = \sum_{t=1}^m \mathbb{1}_{\{I_t=k\}} \geq \sum_{t=1}^m \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} = \left\lceil \frac{4f_k(n)}{\Delta_k^2} \right\rceil \geq \frac{4f_k(n)}{\Delta_k^2},$$

so that $\mathbb{1}_{\{I_s=k\}\cap\xi_{3,s}} = 0$. But then

$$\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} = \sum_{t=1}^m \mathbb{1}_{\{I_t=k\}\cap\xi_{3,t}} = \left\lceil \frac{4f_k(n)}{\Delta_k^2} \right\rceil \leq u,$$

which is the contradiction expected.

We also have $\sum_{t=1}^n \mathbb{1}_{\{I_t=k\}\setminus\xi_{3,t}} = \sum_{t=u+1}^n \mathbb{1}_{\{I_t=k\}\setminus\xi_{3,t}}$: since $T_k(t-1) \leq t-1$, event $\xi_{3,t}$ always happens at time step $t \in \{1,\ldots,u\}$.

And then, since event $\{I_t = k\}$ is included in $\xi_{1,t} \cup \xi_{2,t} \cup \xi_{3,t}$:

$$\mathbb{E}_\theta\left[\sum_{t=u+1}^n \mathbb{1}_{\{I_t=k\}\setminus\xi_{3,t}}\right] \leq \mathbb{E}_\theta\left[\sum_{t=u+1}^n \mathbb{1}_{\xi_{1,t}\cup\xi_{2,t}}\right] \leq \sum_{t=u+1}^n \mathbb{P}_\theta(\xi_{1,t}) + \mathbb{P}_\theta(\xi_{2,t}).$$

It remains to find upper bounds of $\mathbb{P}_\theta(\xi_{1,t})$ and $\mathbb{P}_\theta(\xi_{2,t})$. To this aim, we apply the peeling argument with a geometric grid over the time interval $[1,t]$:

$$\begin{aligned}
\mathbb{P}_\theta(\xi_{1,t}) &= \mathbb{P}_\theta\left(B_{k^*,T_{k^*}(t-1),t} \le \mu^*\right) = \mathbb{P}_\theta\left(\hat{X}_{k^*,T_{k^*}(t-1)} + \sqrt{\frac{f_{k^*}(t)}{T_{k^*}(t-1)}} \le \mu^*\right) \\
&\le \mathbb{P}_\theta\left(\exists s \in \{1,\cdots,t\},\, \hat{X}_{k^*,s} + \sqrt{\frac{f_{k^*}(t)}{s}} \le \mu^*\right) \\
&\le \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \mathbb{P}_\theta\left(\exists s : \{\beta^{j+1}t < s \le \beta^j t\},\, \hat{X}_{k^*,s} + \sqrt{\frac{f_{k^*}(t)}{s}} \le \mu^*\right) \\
&\le \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \mathbb{P}_\theta\left(\exists s : \{\beta^{j+1}t < s \le \beta^j t\},\, \sum_{l=1}^{s}(X_{k^*,l} - \mu^*) \le -\sqrt{s f_{k^*}(t)}\right) \\
&\le \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \mathbb{P}_\theta\left(\exists s : \{\beta^{j+1}t < s \le \beta^j t\},\, \sum_{l=1}^{s}(X_{k^*,l} - \mu^*) \le -\sqrt{\beta^{j+1} t f_{k^*}(t)}\right) \\
&= \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \mathbb{P}_\theta\left(\max_{\beta^{j+1}t < s \le \beta^j t} \sum_{l=1}^{s}(\mu^* - X_{k^*,l}) \ge \sqrt{\beta^{j+1} t f_{k^*}(t)}\right) \\
&\le \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \mathbb{P}_\theta\left(\max_{s \le \beta^j t} \sum_{l=1}^{s}(\mu^* - X_{k^*,l}) \ge \sqrt{\beta^{j+1} t f_{k^*}(t)}\right).
\end{aligned}$$

As the range of the random variables $(X_{k^*,l})_{1\le l\le s}$ is $[0,1]$, Hoeffding's maximal inequality gives:

$$\mathbb{P}_\theta(\xi_{1,t}) \le \sum_{j=0}^{\left\lfloor \frac{\log t}{\log(1/\beta)} \right\rfloor} \exp\left(-\frac{2\left(\sqrt{\beta^{j+1} t f_{k^*}(t)}\right)^2}{\beta^j t}\right) \le \left(\frac{\log t}{\log(1/\beta)} + 1\right) e^{-2\beta f_{k^*}(t)}.$$

Similarly, we have:

$$\mathbb{P}_\theta(\xi_{2,t}) \le \left(\frac{\log t}{\log(1/\beta)} + 1\right) e^{-2\beta f_k(t)},$$

and the result follows from the combination of previous inequalities. ■

2.2 Bounds on the Expected Regret of UCB(ρ), $\rho \le \frac{1}{2}$

We study the performances of UCB(ρ) policy, with $\rho \in (0, \frac{1}{2}]$. We recall that UCB($\rho$) is the UCB policy defined by $f_k(t) = \rho \log(t)$ for all k, that is, $B_{k,s,t} = \hat{X}_{k,s} + \sqrt{\frac{\rho \log t}{s}}$. Small values of ρ can be interpreted as a low level of experimentation in the balance between exploration and exploitation. Precise regret bound orders of UCB(ρ) when $\rho \in (0, \frac{1}{2}]$ are not documented in the literature.

We first give an upper bound of expected regret in simple environments, where it is supposed to perform well. As stated in the following proposition (which is a direct consequence of Lemma 1), the order of the bound is $\frac{\rho \log n}{\Delta}$.

Lemma 3 *Let $0 \leq b < a \leq 1$ and $n \geq 1$. For $\theta = (\delta_a, \delta_b)$, the random variable $T_2(n)$ is uniformly upper bounded by $\frac{\rho}{\Delta^2}\log(n) + 1$. Consequently, the expected regret of* UCB(ρ) *is upper bounded by $\frac{\rho}{\Delta}\log(n) + 1$.*

One can show that the expected regret of UCB(ρ) is actually equivalent to $\frac{\rho \log n}{\Delta}$ as n goes to infinity. These good performances are compensated by poor results in more complex environments, as showed in the following theorem. We exhibit an expected regret upper bound which is valid for any $\theta \in \Theta$, and which is roughly of order $n^{1-2\rho}$. We also show that this upper bound is asymptotically optimal. Thus, with $\rho \in (0, \frac{1}{2})$, UCB($\rho$) does not perform enough exploration to achieve the logarithmic bound, as opposed to UCB(ρ) with $\rho \in (\frac{1}{2}, +\infty)$.

Theorem 4 *For any $\rho \in (0, \frac{1}{2}]$, any $\theta \in \Theta$ and any $\beta \in (0,1)$, one has*

$$\mathbb{E}_\theta[R_n] \leq \sum_{k:\Delta_k > 0} \frac{4\rho \log n}{\Delta_k} + \Delta_k + 2\Delta_k \left(\frac{\log n}{\log(1/\beta)} + 1 \right) \frac{n^{1-2\rho\beta}}{1-2\rho\beta}.$$

Moreover, if Θ has the Dirac/Bernoulli property, then for any $\varepsilon > 0$ there exists $\theta \in \Theta$ such that

$$\lim_{n \to +\infty} \frac{\mathbb{E}_\theta[R_n]}{n^{1-2\rho-\varepsilon}} = +\infty.$$

The value $\rho = \frac{1}{2}$ is critical, but we can deduce from the upper bound of this theorem that UCB($\frac{1}{2}$) is consistent in the classical sense of Lai and Robbins (1985) and of Burnetas and Katehakis (1996).

Proof We set $u = \left\lceil \frac{4\rho \log n}{\Delta_k^2} \right\rceil$. By Lemma 2 we get:

$$\begin{aligned}
\mathbb{E}_\theta[T_k(n)] &\leq u + 2\sum_{t=u+1}^{n} \left(\frac{\log t}{\log(1/\beta)} + 1 \right) e^{-2\beta\rho\log(t)} \\
&= u + 2\sum_{t=u+1}^{n} \left(\frac{\log t}{\log(1/\beta)} + 1 \right) \frac{1}{t^{2\rho\beta}} \\
&\leq u + 2\left(\frac{\log n}{\log(1/\beta)} + 1 \right) \sum_{t=1}^{n} \frac{1}{t^{2\rho\beta}} \\
&\leq u + 2\left(\frac{\log n}{\log(1/\beta)} + 1 \right) \left(1 + \sum_{t=2}^{n} \frac{1}{t^{2\rho\beta}} \right) \\
&\leq u + 2\left(\frac{\log n}{\log(1/\beta)} + 1 \right) \left(1 + \int_1^{n-1} \frac{1}{t^{2\rho\beta}} dt \right) \\
&\leq u + 2\left(\frac{\log n}{\log(1/\beta)} + 1 \right) \frac{n^{1-2\rho\beta}}{1-2\rho\beta}.
\end{aligned}$$

As usual, the upper bound of the expected regret follows from Formula (1).

Now, let us show the lower bound. The result is obtained by considering an environment θ of the form $\left(Ber(\frac{1}{2}), \delta_{\frac{1}{2}-\Delta} \right)$, where Δ lies in $(0, \frac{1}{2})$ and is such that $2\rho(1+\sqrt{\Delta})^2 < 2\rho + \varepsilon$. This notation is obviously consistent with the definition of Δ as an optimality gap. We set $T_n := \lceil \frac{\rho \log n}{\Delta} \rceil$, and define the event ξ_n by:

$$\xi_n = \left\{ \hat{X}_{1,T_n} < \frac{1}{2} - (1 + \frac{1}{\sqrt{\Delta}})\Delta \right\}.$$

When event ξ_n occurs, one has for any $t \in \{T_n, \dots, n\}$

$$\begin{aligned} \hat{X}_{1,T_n} + \sqrt{\frac{\rho \log t}{T_n}} &\leq \hat{X}_{1,T_n} + \sqrt{\frac{\rho \log n}{T_n}} < \frac{1}{2} - (1 + \frac{1}{\sqrt{\Delta}})\Delta + \sqrt{\Delta} \\ &\leq \frac{1}{2} - \Delta, \end{aligned}$$

so that arm 1 is chosen no more than T_n times by UCB(ρ) policy. Consequently:

$$\mathbb{E}_\theta [T_2(n)] \geq \mathbb{P}_\theta(\xi_n)(n - T_n).$$

We will now find a lower bound of the probability of ξ_n thanks to Berry-Esseen inequality. We denote by C the corresponding constant, and by Φ the c.d.f. of the standard normal distribution. For convenience, we also define the following quantities:

$$\sigma := \sqrt{\mathbb{E}\left[\left(X_{1,1} - \frac{1}{2}\right)^2\right]} = \frac{1}{2},\ M_3 := \mathbb{E}\left[\left|X_{1,1} - \frac{1}{2}\right|^3\right] = \frac{1}{8}.$$

Using the fact that $\Phi(-x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}x}\beta(x)$ with $\beta(x) \xrightarrow[x \to +\infty]{} 1$, we have:

$$\begin{aligned} \mathbb{P}_\theta(\xi_n) &= \mathbb{P}_\theta\left(\frac{\hat{X}_{1,T_n} - \frac{1}{2}}{\sigma}\sqrt{T_n} \leq -2\left(1 + \frac{1}{\sqrt{\Delta}}\right)\Delta\sqrt{T_n}\right) \\ &\geq \Phi\left(-2(\Delta + \sqrt{\Delta})\sqrt{T_n}\right) - \frac{CM_3}{\sigma^3\sqrt{T_n}} \\ &\geq \frac{\exp\left(-2(\Delta + \sqrt{\Delta})^2 T_n\right)}{2\sqrt{2\pi}(\Delta + \sqrt{\Delta})\sqrt{T_n}}\beta\left(2(\Delta + \sqrt{\Delta})\sqrt{T_n}\right) - \frac{CM_3}{\sigma^3\sqrt{T_n}} \\ &\geq \frac{\exp\left(-2(\Delta + \sqrt{\Delta})^2(\frac{\rho \log n}{\Delta} + 1)\right)}{2\sqrt{2\pi}(\Delta + \sqrt{\Delta})\sqrt{T_n}}\beta\left(2(\Delta + \sqrt{\Delta})\sqrt{T_n}\right) - \frac{CM_3}{\sigma^3\sqrt{T_n}} \\ &\geq \frac{n^{-2\rho(1+\sqrt{\Delta})^2}}{\sqrt{T_n}} \frac{\exp\left(-2(\Delta + \sqrt{\Delta})^2\right)}{2\sqrt{2\pi}(\Delta + \sqrt{\Delta})}\beta\left(2(\Delta + \sqrt{\Delta})\sqrt{T_n}\right) - \frac{CM_3}{\sigma^3\sqrt{T_n}}. \end{aligned}$$

Previous calculations and Formula (1) give

$$\mathbb{E}_\theta[R_n] = \Delta\mathbb{E}_\theta[T_2(n)] \geq \Delta\mathbb{P}_\theta(\xi_n)(n - T_n),$$

so that we finally obtain a lower bound of $\mathbb{E}_\theta[R_n]$ of order $\frac{n^{1-2\rho(1+\sqrt{\Delta})^2}}{\sqrt{\log n}}$. Therefore, $\frac{\mathbb{E}_\theta[R_n]}{n^{1-2\rho-\varepsilon}}$ is at least of order $\frac{n^{2\rho+\varepsilon-2\rho(1+\sqrt{\Delta})^2}}{\sqrt{\log n}}$. Since $2\rho + \varepsilon - 2\rho(1 + \sqrt{\Delta})^2 > 0$, the numerator goes to infinity, faster than $\sqrt{\log n}$. This concludes the proof. ■

3. Bounds on the Class α-consistent Policies

In this section, our aim is to find how the classical results of Lai and Robbins (1985) and of Burnetas and Katehakis (1996) can be generalised if we do not restrict the study to consistent policies. As a by-product, we will adapt their results to the present setting, which is simpler than their parametric frameworks.

We recall that a policy is consistent if its expected regret is $o(n^a)$ for all $a > 0$ in all environments $\theta \in \Theta$. A natural way to relax this definition is the following.

Definition 5 *A policy is α-consistent if*

$$\forall a > \alpha,\ \forall \theta \in \Theta,\ \mathbb{E}_\theta[R_n] = o(n^a).$$

For example, we showed in the previous section that, for any $\rho \in (0, \frac{1}{2}]$, UCB($\rho$) is $(1-2\rho)$-consistent and not α-consistent if $\alpha < 1-2\rho$.

Note that the relevant range of α in this definition is $[0,1)$: the case $\alpha = 0$ corresponds to the standard definition of consistency (so that throughout the paper the term "consistent" also means "0-consistent"), and any value $\alpha \geq 1$ is pointless as any policy is then α-consistent. Indeed, the expected regret of any policy is at most of order n. This also lead us to wonder what happens if we only require the expected regret to be $o(n)$:

$$\forall \theta \in \Theta,\ \mathbb{E}_\theta[R_n] = o(n).$$

This requirement corresponds to the definition of Hannan consistency. The class of Hannan consistent policies includes consistent policies and α-consistent policies for any $\alpha \in [0,1)$. Some results about this class will be obtained in Section 5.

We focus on regret lower bounds on α-consistent policies. We first show that the main result of Burnetas and Katehakis can be extended in the following way.

Theorem 6 *Assume that Θ has the product property. Fix an α-consistent policy and $\theta \in \Theta$. If $\Delta_k > 0$ and if $0 < D_k(\theta) < \infty$, then*

$$\forall \varepsilon > 0,\ \lim_{n\to+\infty} \mathbb{P}_\theta \left[T_k(n) \geq (1-\varepsilon) \frac{(1-\alpha)\log n}{D_k(\theta)} \right] = 1.$$

Consequently

$$\liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[T_k(n)]}{\log n} \geq \frac{1-\alpha}{D_k(\theta)}.$$

Remind that the lower bound of the expected regret is then deduced from Formula (1), and that coefficient $D_k(\theta)$ is defined by:

$$D_k(\theta) := \inf_{\tilde{\nu}_k \in \Theta_k : \mathbb{E}[\tilde{\nu}_k] > \mu^*} KL(\nu_k, \tilde{\nu}_k),$$

where $KL(\nu,\mu)$ denotes the Kullback-Leibler divergence of measures ν and μ.

Note that, as opposed to Burnetas and Katehakis (1996), there is no optimal policy in general (i.e., a policy that would achieve the lower bound in all environment θ). This can be explained intuitively as follows. If by contradiction there existed such a policy, its expected regret would be of order $\log n$ and consequently it would be (0-)consistent. Then the lower bounds in the case of

0-consistency would necessarily hold. This can not happen if $\alpha > 0$ because $\frac{1-\alpha}{D_k(\theta)} < \frac{1}{D_k(\theta)}$.
Nevertheless, this argument is not rigorous because the fact that the regret would be of order $\log n$ is only valid for environments θ such that $0 < D_k(\theta) < \infty$. The non-existence of optimal policies is implied by a stronger result of the next section (yet, only if $\alpha > 0.2$).

Proof We adapt Proposition 1 in Burnetas and Katehakis (1996) and its proof. Let us denote $\theta = (\nu_1, \dots, \nu_K)$. We fix $\varepsilon > 0$, and we want to show that:

$$\lim_{n\to+\infty} \mathbb{P}_\theta \left(\frac{T_k(n)}{\log n} < \frac{(1-\varepsilon)(1-\alpha)}{D_k(\theta)} \right) = 0.$$

Set $\delta > 0$ and $\delta' > \alpha$ such that $\frac{1-\delta'}{1+\delta} > (1-\varepsilon)(1-\alpha)$. By definition of $D_k(\theta)$, there exists $\tilde{\nu}_k$ such that $\mathbb{E}[\tilde{\nu}_k] > \mu^*$ and

$$D_k(\theta) < KL(\nu_k, \tilde{\nu}_k) < (1+\delta) D_k(\theta).$$

Let us set $\tilde{\theta} = (\nu_1, \dots, \nu_{k-1}, \tilde{\nu}_k, \nu_{k+1}, \dots, \nu_K)$. Environment $\tilde{\theta}$ lies in Θ by the product property and arm k is its best arm. Define $I^\delta = KL(\nu_k, \tilde{\nu}_k)$ and

$$A_n^{\delta'} := \left\{ \frac{T_k(n)}{\log n} < \frac{1-\delta'}{I^\delta} \right\}, \; C_n^{\delta''} := \left\{ \log L_{T_k(n)} \le \left(1-\delta''\right) \log n \right\},$$

where δ'' is such that $\alpha < \delta'' < \delta'$ and L_t is defined by $\log L_t = \sum_{s=1}^t \log\left(\frac{d\nu_k}{d\tilde{\nu}_k}(X_{k,s}) \right)$.

Now, we show that $\mathbb{P}_\theta(A_n^{\delta'}) = \mathbb{P}_\theta(A_n^{\delta'} \cap C_n^{\delta''}) + \mathbb{P}_\theta(A_n^{\delta'} \setminus C_n^{\delta''}) \xrightarrow[n\to+\infty]{} 0$.

On the one hand, one has:

$$\begin{aligned}
\mathbb{P}_\theta(A_n^{\delta'} \cap C_n^{\delta''}) &\le n^{1-\delta''} \mathbb{P}_{\tilde{\theta}}(A_n^{\delta'} \cap C_n^{\delta''}) && (6)\\
&\le n^{1-\delta''} \mathbb{P}_{\tilde{\theta}}(A_n^{\delta'}) = n^{1-\delta''} \mathbb{P}_{\tilde{\theta}} \left(n - T_k(n) > n - \frac{1-\delta'}{I^\delta} \log n \right) \\
&\le \frac{n^{1-\delta''} \mathbb{E}_{\tilde{\theta}}[n - T_k(n)]}{n - \frac{1-\delta'}{I^\delta} \log n} && (7)\\
&= \frac{n^{-\delta''} \mathbb{E}_{\tilde{\theta}} \left[\sum_{l=1}^K T_\ell(n) - T_k(n) \right]}{n - \frac{1-\delta'}{I^\delta} \frac{\log n}{n}} \\
&\le \frac{\sum_{\ell \neq k} n^{-\delta''} \mathbb{E}_{\tilde{\theta}}[T_\ell(n)]}{1 - \frac{1-\delta'}{I^\delta} \frac{\log n}{n}} \xrightarrow[n\to+\infty]{} 0. && (8)
\end{aligned}$$

Equation (6) results from a partition of $A_n^{\delta'}$ into events $\{T_k(n) = a\}$, $0 \le a < \left\lceil \frac{1-\delta'}{I^\delta} \log n \right\rceil$. Each event $\{T_k(n) = a\} \cap C_n^{\delta''}$ equals $\{T_k(n) = a\} \cap \left\{ \prod_{s=1}^a \frac{d\nu_k}{d\tilde{\nu}_k}(X_{k,s}) \le n^{1-\delta''} \right\}$ and is measurable with respect to $X_{k,1}, \dots, X_{k,a}$ and to $X_{\ell,1}, \dots, X_{\ell,n}$ ($\ell \neq k$). Thus, $\mathbb{1}_{\{T_k(n)=a\} \cap C_n^{\delta''}}$ can be written as a function f of the latter r.v. and we have:

$$\begin{aligned}
\mathbb{P}_\theta \left(\{T_k(n) = a\} \cap C_n^{\delta''} \right) &= \int f\left((x_{k,s})_{1\le s\le a}, (x_{\ell,s})_{\ell\neq k, 1\le s\le n} \right) \prod_{\substack{\ell \neq k \\ 1 \le s \le n}} d\nu_\ell(x_{\ell,s}) \prod_{1\le s\le a} d\nu_k(x_{k,s}) \\
&\le \int f\left((x_{k,s})_{1\le s\le a}, (x_{\ell,s})_{\ell\neq k, 1\le s\le n} \right) \prod_{\substack{\ell \neq k \\ 1 \le s \le n}} d\nu_\ell(x_{\ell,s}) n^{1-\delta''} \prod_{1\le s\le a} d\tilde{\nu}_k(x_{k,s}) \\
&= n^{1-\delta''} \mathbb{P}_{\tilde{\theta}} \left(\{T_k(n) = a\} \cap C_n^{\delta''} \right).
\end{aligned}$$

Equation (7) is a consequence of Markov's inequality, and the limit in (8) is a consequence of α-consistency.

On the other hand, we set $b_n := \frac{1-\delta'}{I^\delta}\log n$, so that

$$\begin{aligned}\mathbb{P}_\theta(A_n^{\delta'} \setminus C_n^{\delta''}) &\leq \mathbb{P}\left(\max_{j\leq\lfloor b_n\rfloor} \log L_j > (1-\delta'')\log n\right)\\ &\leq \mathbb{P}\left(\frac{1}{b_n}\max_{j\leq\lfloor b_n\rfloor} \log L_j > I^\delta \frac{1-\delta''}{1-\delta'}\right).\end{aligned}$$

This term tends to zero, as a consequence of the law of large numbers.

Now that $\mathbb{P}_\theta(A_n^{\delta'})$ tends to zero, the conclusion results from

$$\frac{1-\delta'}{I^\delta} > \frac{1-\delta'}{(1+\delta)D_k(\theta)} \geq \frac{(1-\varepsilon)(1-\alpha)}{D_k(\theta)}.$$

■

The previous lower bound is asymptotically optimal with respect to its dependence in α, as claimed in the following proposition.

Proposition 7 *Assume that Θ has the Dirac/Bernoulli property. There exist $\theta \in \Theta$ and a constant $c > 0$ such that, for any $\alpha \in [0,1)$, there exists an α-consistent policy such that:*

$$\liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[T_k(n)]}{(1-\alpha)\log n} \leq c,$$

for any k satisfying $\Delta_k > 0$.

Proof In any environment of the form $\theta = (\delta_a, \delta_b)$ with $a \neq b$, Lemma 3 implies the following estimate for UCB(ρ):

$$\liminf_{n\to+\infty} \frac{\mathbb{E}_\theta T_k(n)}{\log n} \leq \frac{\rho}{\Delta^2},$$

where $k \neq k^*$.

Because $\frac{1-\alpha}{2} \in (0, \frac{1}{2})$ and since UCB(ρ) is $(1-2\rho)$-consistent for any $\rho \in (0, \frac{1}{2}]$ (Theorem 4), we obtain the result by choosing the α-consistent policy UCB($\frac{1-\alpha}{2}$) and by setting $c = \frac{1}{2\Delta^2}$.

■

4. Selectivity

In this section, we address the problem of selectivity. By selectivity, we mean the ability to adapt to the environment as and when rewards are observed. More precisely, a set of two (or more) policies is given. The one that performs the best depends on environment θ. We wonder if there exists an adaptive procedure that, given any environment θ, would be as good as any policy in the given set. Two major reasons motivate this study.

On the one hand this question was answered by Burnetas and Katehakis within the class of consistent policies. They exhibits an asymptotically optimal policy, that is, that achieves the regret

lower bounds they have proven. The fact that a policy performs as best as any other one obviously solves the problem of selectivity.

On the other hand, this problem has already been studied in the context of adversarial bandit by Auer et al. (2003). Their setting differs from our not only because their bandits are nonstochastic, but also because their adaptive procedure takes only into account a given number of recommendations, whereas in our setting the adaptation is supposed to come from observing rewards of the chosen arms (only one per time step). Nevertheless, one can wonder if an "exponentially weighted forecasters" procedure like EXP4 could be transposed to our context. The answer is negative, as stated in the following theorem.

To avoid confusion, we make the notations of the regret and of sampling time more precise by adding the considered policy: under policy $\mathcal{A}$, R_n and $T_k(n)$ will be respectively denoted $R_n(\mathcal{A})$ and $T_k(n,\mathcal{A})$.

Theorem 8 *Let $\tilde{\mathcal{A}}$ be a consistent policy and let ρ be a real number in $(0,0.4)$. If Θ has the Dirac/Bernoulli property and the product property, there is no policy which can both beat $\tilde{\mathcal{A}}$ and* UCB*(ρ), that is:*

$$\forall \mathcal{A},\ \exists \theta \in \Theta,\ \limsup_{n\to+\infty} \frac{\mathbb{E}_\theta[R_n(\mathcal{A})]}{\min(\mathbb{E}_\theta[R_n(\tilde{\mathcal{A}})], \mathbb{E}_\theta[R_n(\mathrm{UCB}(\rho))])} > 1.$$

Thus the existence of optimal policies does not hold when we extend the notion of consistency. Precisely, as UCB(ρ) is $(1-2\rho)$-consistent, we have shown that there is no optimal policy within the class of α-consistent policies, with $\alpha > 0.2$. Consequently, there do not exist optimal policies in the class of Hannan consistent policies either.

Moreover, Theorem 8 shows that methods that would be inspired by related literature in adversarial bandit can not apply to our framework. As we said, this impossibility may come from the fact that we can not observe at each time step the decisions and rewards of more than one algorithm. If we were able to observe a given set of policies from step to step, then it would be easy to beat them all: it would be sufficient to aggregate all the observations and simply pull the arm with the greater empiric mean. The case where we only observe decisions (and not rewards) of a set of policies may be interesting, but is left outside of the scope of this paper.

Proof Assume by contradiction that

$$\exists \mathcal{A},\ \forall \theta \in \Theta,\ \limsup_{n\to+\infty} u_{n,\theta} \leq 1,$$

where $u_{n,\theta} = \frac{\mathbb{E}_\theta[R_n(\mathcal{A})]}{\min(\mathbb{E}_\theta[R_n(\tilde{\mathcal{A}})], \mathbb{E}_\theta[R_n(UCB(\rho))])}$.

For any θ, we have

$$\mathbb{E}_\theta[R_n(\mathcal{A})] = \frac{\mathbb{E}_\theta[R_n(\mathcal{A})]}{\mathbb{E}_\theta[R_n(\tilde{\mathcal{A}})]} \mathbb{E}_\theta[R_n(\tilde{\mathcal{A}})] \leq u_{n,\theta} \mathbb{E}_\theta[R_n(\tilde{\mathcal{A}})], \tag{9}$$

so that the fact that $\tilde{\mathcal{A}}$ is a consistent policy implies that $\mathcal{A}$ is also consistent. Consequently the lower bound of Theorem 6 also holds for policy $\mathcal{A}$.

For the rest of the proof, we focus on environments of the form $\theta = (\delta_0, \delta_\Delta)$ with $\Delta > 0$. In this case, arm 2 is the best arm, so that we have to compute $D_1(\theta)$. On the one hand, we have:

$$D_1(\theta) = \inf_{\tilde{\nu}_1 \in \Theta_1 : \mathbb{E}[\tilde{\nu}_1] > \mu^*} KL(\nu_1, \tilde{\nu}_1) = \inf_{\tilde{\nu}_1 \in \Theta_1 : \mathbb{E}[\tilde{\nu}_1] > \Delta} KL(\delta_0, \tilde{\nu}_1) = \inf_{\tilde{\nu}_1 \in \Theta_1 : \mathbb{E}[\tilde{\nu}_1] > \Delta} \log\left(\frac{1}{\tilde{\nu}_1(0)}\right).$$

As $\mathbb{E}[\tilde{\nu}_1] \leq 1 - \tilde{\nu}_1(0)$, we get:

$$D_1(\theta) \geq \inf_{\tilde{\nu}_1 \in \Theta_1 : 1-\tilde{\nu}_1(0) \geq \Delta} \log\left(\frac{1}{\tilde{\nu}_1(0)}\right) \geq \log\left(\frac{1}{1-\Delta}\right).$$

One the other hand, we have for any $\varepsilon > 0$:

$$D_1(\theta) \leq KL(\delta_0, Ber(\Delta+\varepsilon)) = \log\left(\frac{1}{1-\Delta-\varepsilon}\right)$$

Consequently $D_1(\theta) = \log\left(\frac{1}{1-\Delta}\right)$, and the lower bound of Theorem 6 reads:

$$\liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[T_1(n,\mathcal{A})]}{\log n} \geq \frac{1}{\log\left(\frac{1}{1-\Delta}\right)}.$$

Just like Equation (9), we have:

$$\mathbb{E}_\theta[R_n(\mathcal{A})] \leq u_{n,\theta}\mathbb{E}_\theta[R_n(UCB(\rho))].$$

Moreover, Lemma 3 provides:

$$\mathbb{E}_\theta[R_n(UCB(\rho))] \leq 1 + \frac{\rho \log n}{\Delta}.$$

Now, by gathering the three previous inequalities and Formula (1), we get:

$$\begin{aligned}
\frac{1}{\log\left(\frac{1}{1-\Delta}\right)} &\leq \liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[T_1(n,\mathcal{A})]}{\log n} = \liminf_{n\to+\infty} \frac{\mathbb{E}_\theta[R_n(\mathcal{A})]}{\Delta \log n} \\
&\leq \liminf_{n\to+\infty} \frac{u_{n,\theta}\mathbb{E}_\theta[R_n(UCB(\rho))]}{\Delta \log n} \leq \liminf_{n\to+\infty} \frac{u_{n,\theta}}{\Delta \log n}\left(1 + \frac{\rho \log n}{\Delta}\right) \\
&\leq \liminf_{n\to+\infty} \frac{u_{n,\theta}}{\Delta \log n} + \liminf_{n\to+\infty} \frac{\rho u_{n,\theta}}{\Delta^2} = \frac{\rho}{\Delta^2} \liminf_{n\to+\infty} u_{n,\theta} \leq \frac{\rho}{\Delta^2} \limsup_{n\to+\infty} u_{n,\theta} \\
&\leq \frac{\rho}{\Delta^2}.
\end{aligned}$$

This means that ρ has to be lower bounded by $\frac{\Delta^2}{\log\left(\frac{1}{1-\Delta}\right)}$, but this is greater than 0.4 if $\Delta = 0.75$, hence the contradiction.

■

Note that this proof gives a simple alternative to Theorem 4 to show that UCB(ρ) is not consistent (if $\rho \leq 0.4$). Indeed if it were consistent, then in environment $\theta = (\delta_0, \delta_\Delta)$ the same contradiction between the lower bound of Theorem 6 and the upper bound of Lemma 3 would hold.

5. General Bounds

In this section, we study lower bounds on the expected regret with few requirements on Θ and on the class of policies. With a simple property on Θ but without any assumption on the policy, we show that there always exist logarithmic lower bounds for some environments θ. Then, still with a

simple property on Θ, we show that there exists a Hannan consistent policy for which the expected regret is sub-logarithmic for some environment θ.

Note that the policy that always pulls arm 1 has a 0 expected regret in environments where arm 1 has the best mean reward, and an expected regret of order n in other environments. So, for this policy, expected regret is sub-logarithmic in some environments. Nevertheless, this policy is not Hannan consistent because its expected regret is not always $o(n)$.

5.1 The Necessity of a Logarithmic Regret in Some Environments

The necessity of a logarithmic regret in some environments can be explained by a simple sketch proof. Assume that the agent knows the number of rounds n, and that he balances exploration and exploitation in the following way: he first pulls each arm $s(n)$ times, and then selects the arm that has obtained the best empiric mean for the rest of the game. Denote by $p_{s(n)}$ the probability that the best arm does not have the best empiric mean after the exploration phase (i.e., after the first $Ks(n)$ rounds). The expected regret is then of the form

$$c_1(1-p_{s(n)})s(n)+c_2 p_{s(n)}n. \tag{10}$$

Indeed, if the agent manages to match the best arm then he only suffers the pulls of suboptimal arms during the exploration phase. That represents an expected regret of order $s(n)$. If not, the number of pulls of suboptimal arms is of order n, and so is the expected regret.

Now, let us approximate $p_{s(n)}$. It has the same order as the probability that the best arm gets an empiric mean lower than the second best mean reward. Moreover, $\frac{X_{k^*,s(n)}-\mu^*}{\sigma}\sqrt{s(n)}$ (where σ is the variance of $X_{k^*,1}$) has approximately a standard normal distribution by the central limit theorem. Therefore, we have:

$$\begin{aligned}
p_{s(n)} &\approx \mathbb{P}_\theta(X_{k^*,s(n)} \le \mu^* - \Delta) = \mathbb{P}_\theta\left(\frac{X_{k^*,s(n)}-\mu^*}{\sigma}\sqrt{s(n)} \le -\frac{\Delta\sqrt{s(n)}}{\sigma}\right) \\
&\approx \frac{1}{\sqrt{2\pi}}\frac{\sigma}{\Delta\sqrt{s(n)}}\exp\left(-\frac{1}{2}\left(\frac{\Delta\sqrt{s(n)}}{\sigma}\right)^2\right) \\
&\approx \frac{1}{\sqrt{2\pi}}\frac{\sigma}{\Delta\sqrt{s(n)}}\exp\left(-\frac{\Delta^2 s(n)}{2\sigma^2}\right).
\end{aligned}$$

It follows that the expected regret has to be at least logarithmic. Indeed, to ensure that the second term $c_2 p_{s(n)}n$ of Equation (10) is sub-logarithmic, $s(n)$ has to be greater than $\log n$. But then first term $c_1(1-p_{s(n)})s(n)$ is greater than $\log n$.

Actually, the necessity of a logarithmic regret can be written as a consequence of Theorem 6. Indeed, if we assume by contradiction that $\limsup_{n\to+\infty}\frac{\mathbb{E}_\theta R_n}{\log n}=0$ for all θ (i.e., $\mathbb{E}_\theta R_n = o(\log n)$), the considered policy is consistent. Consequently, Theorem 6 implies that

$$\limsup_{n\to+\infty}\frac{\mathbb{E}_\theta R_n}{\log n} \ge \liminf_{n\to+\infty}\frac{\mathbb{E}_\theta R_n}{\log n} > 0.$$

Yet, this reasoning needs Θ having the product property, and conditions of the form $0 < D_k(\theta) < \infty$ also have to hold.

The following proposition is a rigorous version of our sketch, and it shows that the necessity of a logarithmic lower bound can be based on a simple property on Θ.

Proposition 9 *Assume that there exist two environments* $\theta = (\nu_1, \ldots, \nu_K) \in \Theta$, $\tilde{\theta} = (\tilde{\nu}_1, \ldots, \tilde{\nu}_K) \in \Theta$, *and an arm* $k \in \{1, \ldots, K\}$ *such that*

1. *k has the best mean reward in environment θ,*
2. *k is not the winning arm in environment $\tilde{\theta}$,*
3. $\nu_k = \tilde{\nu}_k$ *and there exists* $\eta \in (0,1)$ *such that*

$$\prod_{\ell \neq k} \frac{d\nu_\ell}{d\tilde{\nu}_\ell}(X_{\ell,1}) \geq \eta \;\; \mathbb{P}_{\tilde{\theta}} - a.s. \tag{11}$$

Then, for any policy, there exists $\hat{\theta} \in \Theta$ *such that*

$$\limsup_{n \to +\infty} \frac{\mathbb{E}_{\hat{\theta}} R_n}{\log n} > 0.$$

Let us explain the logic of the three conditions of the proposition. If $\nu_k = \tilde{\nu}_k$, and in case ν_k seems to be the reward distribution of arm k, then arm k has to be pulled often enough for the regret to be small if the environment is θ. Nevertheless, one has to explore other arms to know whether the environment is actually $\tilde{\theta}$. Moreover, Inequality (11) makes sure that the distinction between θ and $\tilde{\theta}$ is tough to make: it ensures that pulling any arm $\ell \neq k$ gives a reward which is likely in both environments. Without such an assumption the problem may be very simple, and providing a logarithmic lower bound is hopeless. Indeed, the distinction between any pair of tricky environments $(\theta, \tilde{\theta})$ may be solved in only one pull of a given arm $\ell \neq k$, that would almost surely give a reward that is possible in only one of the two environments.

The third condition can be seen as an alternate version of condition $0 < D_k(\theta) < \infty$ in Theorem 6, though there is no logical connection with it. Finally, let us remark that one can check that any set Θ that has the Dirac/Bernoulli property satisfies the conditions of Proposition 9.

Proof The proof consists in writing a proper version of Expression (10). To this aim we compute a lower bound of $\mathbb{E}_{\tilde{\theta}} R_n$, expressed as a function of $\mathbb{E}_\theta R_n$ and of an arbitrary function $g(n)$.

In the following, $\tilde{\Delta}_k$ denotes the optimality gap of arm k in environment $\tilde{\theta}$. As event $\left\{\sum_{\ell \neq k} T_\ell(n) \leq g(n)\right\}$ is measurable with respect to $X_{\ell,1}, \ldots, X_{\ell,\lfloor g(n) \rfloor}$ ($\ell \neq k$) and to $X_{k,1}, \ldots, X_{k,n}$, we also introduce the function q such that

$$\mathbb{1}_{\left\{\sum_{\ell \neq k} T_\ell(n) \leq g(n)\right\}} = q\big((X_{\ell,s})_{\ell \neq k,\, s=1..\lfloor g(n) \rfloor}, (X_{k,s})_{s=1..n}\big).$$

We have:

$$
\begin{aligned}
\mathbb{E}_{\tilde{\theta}} R_n &\geq \tilde{\Delta}_k \mathbb{E}_{\tilde{\theta}}[T_k(n)] \geq \tilde{\Delta}_k (n - g(n)) \mathbb{P}_{\tilde{\theta}} \left(T_k(n) \geq n - g(n) \right) && (12) \\
&= \tilde{\Delta}_k (n - g(n)) \mathbb{P}_{\tilde{\theta}} \left(n - \sum_{\ell \neq k} T_\ell(n) \geq n - g(n) \right) \\
&= \tilde{\Delta}_k (n - g(n)) \mathbb{P}_{\tilde{\theta}} \left(\sum_{\ell \neq k} T_\ell(n) \leq g(n) \right) \\
&= \tilde{\Delta}_k (n - g(n)) \int q\big((x_{\ell,s})_{\ell \neq k,\ s=1..\lfloor g(n) \rfloor}, (x_{k,s})_{s=1..n} \big) \prod_{\substack{\ell \neq k \\ s=1..\lfloor g(n) \rfloor}} d\tilde{\nu}_\ell(x_{\ell,s}) \prod_{s=1..n} d\tilde{\nu}_k(x_{k,s}) \\
&\geq \tilde{\Delta}_k (n - g(n)) \int q\big((x_{\ell,s})_{\ell \neq k,\ s=1..\lfloor g(n) \rfloor}, (x_{k,s})_{s=1..n} \big) \eta^{\lfloor g(n) \rfloor} \prod_{\substack{\ell \neq k \\ s=1..\lfloor g(n) \rfloor}} d\nu_\ell(x_{\ell,s}) \prod_{s=1..n} d\nu_k(x_{k,s}) && (13) \\
&\geq \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \int q\big((x_{\ell,s})_{\ell \neq k,\ s=1..\lfloor g(n) \rfloor}, (x_{k,s})_{s=1..n} \big) \prod_{\substack{\ell \neq k \\ s=1..\lfloor g(n) \rfloor}} d\nu_\ell(x_{\ell,s}) \prod_{s=1..n} d\nu_k(x_{k,s}) \\
&= \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \mathbb{P}_\theta \left(\sum_{\ell \neq k} T_\ell(n) \leq g(n) \right) \\
&= \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \left(1 - \mathbb{P}_\theta \left(\sum_{\ell \neq k} T_\ell(n) > g(n) \right) \right) \\
&\geq \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \left(1 - \frac{\mathbb{E}_\theta \left(\sum_{\ell \neq k} T_\ell(n) \right)}{g(n)} \right) && (14) \\
&\geq \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \left(1 - \frac{\mathbb{E}_\theta \left(\sum_{\ell \neq k} \Delta_\ell T_\ell(n) \right)}{\Delta g(n)} \right) && (15) \\
&\geq \tilde{\Delta}_k (n - g(n)) \eta^{g(n)} \left(1 - \frac{\mathbb{E}_\theta R_n}{\Delta g(n)} \right),
\end{aligned}
$$

where the first inequality of (12) is a consequence of Formula (1), the second inequality of (12) and inequality (14) come from Markov's inequality, Inequality (13) is a consequence of (11), and Inequality (15) results from the fact that $\Delta_\ell \geq \Delta$ for all ℓ.

Now, let us conclude. If $\frac{\mathbb{E}_\theta R_n}{\log n} \xrightarrow[n \to +\infty]{} 0$, we set $g(n) = \frac{2\mathbb{E}_\theta R_n}{\Delta}$, so that $g(n) \leq \min\left(\frac{n}{2}, \frac{-\log n}{2 \log \eta} \right)$ for n large enough. Then, we have:

$$
\mathbb{E}_{\tilde{\theta}} R_n \geq \tilde{\Delta}_k \frac{n - g(n)}{2} \eta^{g(n)} \geq \tilde{\Delta}_k \frac{n}{4} \eta^{\frac{-\log n}{2 \log \eta}} = \tilde{\Delta}_k \frac{\sqrt{n}}{4}.
$$

In particular, $\frac{\mathbb{E}_{\tilde{\theta}} R_n}{\log n} \xrightarrow[n \to +\infty]{} +\infty$, and the result follows.

■

5.2 Hannan Consistency

We will prove that there exists a Hannan consistent policy such that there can not be a logarithmic lower bound for every environment θ of Θ. To this aim, we make use of general UCB policies again (cf. Section 2.1). Let us first give sufficient conditions on the f_k for UCB policy to be Hannan consistent.

Proposition 10 *Assume that $f_k(n) = o(n)$ for all $k \in \{1,\dots,K\}$. Assume also that there exist $\gamma > \frac{1}{2}$ and $N \geq 3$ such that $f_k(n) \geq \gamma \log\log n$ for all $k \in \{1,\dots,K\}$ and for all $n \geq N$. Then* UCB *is Hannan consistent.*

Proof Fix an arm k such that $\Delta_k > 0$ and choose $\beta \in (0,1)$ such that $2\beta\gamma > 1$. By means of Lemma 2, we have for n large enough:

$$\mathbb{E}_\theta[T_k(n)] \leq u + 2\sum_{t=u+1}^{n} \left(1 + \frac{\log t}{\log(\frac{1}{\beta})}\right) e^{-2\beta\gamma\log\log t},$$

where $u = \left\lceil \frac{4f_k(n)}{\Delta_k^2} \right\rceil$.

Consequently, we have:

$$\mathbb{E}_\theta[T_k(n)] \leq u + 2\sum_{t=2}^{n} \left(\frac{1}{(\log t)^{2\beta\gamma}} + \frac{1}{\log(\frac{1}{\beta})} \frac{1}{(\log t)^{2\beta\gamma-1}} \right). \tag{16}$$

Sums of the form $\sum_{t=2}^{n} \frac{1}{(\log t)^c}$ with $c > 0$ are equivalent to $\frac{n}{(\log n)^c}$ as n goes to infinity. Indeed, on the one hand we have

$$\sum_{t=3}^{n} \frac{1}{(\log t)^c} \leq \int_2^n \frac{dx}{(\log x)^c} \leq \sum_{t=2}^{n} \frac{1}{(\log t)^c},$$

so that $\sum_{t=2}^{n} \frac{1}{(\log t)^c} \sim \int_2^n \frac{dx}{(\log x)^c}$. On the other hand, we have

$$\int_2^n \frac{dx}{(\log x)^c} = \left[\frac{x}{(\log x)^c}\right]_2^n + c\int_2^n \frac{dx}{(\log x)^{c+1}}.$$

As both integrals are divergent we have $\int_2^n \frac{dx}{(\log x)^{c+1}} = o\left(\int_2^n \frac{dx}{(\log x)^c}\right)$, so that $\int_2^n \frac{dx}{(\log x)^c} \sim \frac{n}{(\log n)^c}$.

Combining the fact that $\sum_{t=2}^{n} \frac{1}{(\log t)^c} \sim \frac{n}{(\log n)^c}$ with Equation (16), we get the existence of a constant $C > 0$ such that

$$\mathbb{E}_\theta[T_k(n)] \leq \left\lceil \frac{4f_k(n)}{\Delta^2} \right\rceil + \frac{Cn}{(\log n)^{2\beta\gamma-1}}.$$

Since $f_k(n) = o(n)$ and $2\beta\gamma - 1 > 0$, the latter inequality shows that $\mathbb{E}_\theta[T_k(n)] = o(n)$. The result follows. ■

We are now in the position to prove the main result of this section.

Theorem 11 *If Θ has the Dirac/Bernoulli property, there exist Hannan consistent policies for which the expected regret can not be lower bounded by a logarithmic function in all environments θ.*

Proof If $f_1(n) = f_2(n) = \log\log n$ for $n \geq 3$, UCB is Hannan consistent by Proposition 10. According to Lemma 1, the expected regret is then of order $\log\log n$ in environments of the form (δ_a, δ_b), $a \neq b$. Hence the conclusion on the non-existence of logarithmic lower bounds. ■

Thus we have obtained a lower bound of order $\log\log n$. This order is critical regarding the methods we used. Yet, we do not know if this order is optimal.

Acknowledgments

This work has been supported by the French National Research Agency (ANR) through the COSINUS program (ANR-08-COSI-004: EXPLO-RA project).

References

R. Agrawal. Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Mathematics*, 27:1054–1078, 1995.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, volume 1, pages 267–281. Springer Verlag, 1973.

J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

D. Bergemann and J. Valimaki. Bandit problems. In *The New Palgrave Dictionary of Economics, 2nd ed.* Macmillan Press, 2008.

S. Bubeck. *Bandits Games and Clustering Foundations*. PhD thesis, Université Lille 1, France, 2010.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. Online optimization in X-armed bandits. In *Advances in Neural Information Processing Systems 21*, pages 201–208. 2009.

A.N. Burnetas and M.N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, 2006.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

P.A. Coquelin and R. Munos. Bandit algorithms for tree search. In *Uncertainty in Artificial Intelligence*, 2007.

S. Gelly and Y. Wang. Exploration exploitation in go: UCT for Monte-Carlo go. In *Online Trading between Exploration and Exploitation Workshop, Twentieth Annual Conference on Neural Information Processing Systems (NIPS 2006)*, 2006.

J. Honda and A. Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Proceedings of the Twenty-Third Annual Conference on Learning Theory (COLT)*, 2010.

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 681–690, 2008.

R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17*, pages 697–704. 2005.

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

D. Lamberton, G. Pagès, and P. Tarrès. When can the two-armed bandit algorithm be trusted? *Annals of Applied Probability*, 14(3):1424–1454, 2004.

C.L. Mallows. Some comments on cp. *Technometrics*, pages 661–675, 1973.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.

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

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Journal of Machine Learning Research 14 (2013) 209-242 Submitted 10/11; Revised 6/12; Published 1/13

MAGIC Summoning: Towards Automatic Suggesting and Testing of Gestures With Low Probability of False Positives During Use

Daniel Kyu Hwa Kohlsdorf DKOHL@TZI.DE
Thad E. Starner THAD@CC.GATECH.EDU
GVU & School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332

Editors: Isabelle Guyon and Vassilis Athitsos

Abstract

Gestures for interfaces should be short, pleasing, intuitive, and easily recognized by a computer. However, it is a challenge for interface designers to create gestures easily distinguishable from users' normal movements. Our tool MAGIC Summoning addresses this problem. Given a specific platform and task, we gather a large database of unlabeled sensor data captured in the environments in which the system will be used (an "Everyday Gesture Library" or EGL). The EGL is quantized and indexed via multi-dimensional Symbolic Aggregate approXimation (SAX) to enable quick searching. MAGIC exploits the SAX representation of the EGL to suggest gestures with a low likelihood of false triggering. Suggested gestures are ordered according to brevity and simplicity, freeing the interface designer to focus on the user experience. Once a gesture is selected, MAGIC can output synthetic examples of the gesture to train a chosen classifier (for example, with a hidden Markov model). If the interface designer suggests his own gesture and provides several examples, MAGIC estimates how accurately that gesture can be recognized and estimates its false positive rate by comparing it against the natural movements in the EGL. We demonstrate MAGIC's effectiveness in gesture selection and helpfulness in creating accurate gesture recognizers.

Keywords: gesture recognition, gesture spotting, false positives, continuous recognition

1. Introduction

The success of the Nintendo Wii, Microsoft Kinect, and Google's and Apple's mobile devices demonstrates the popularity of gesture-based interfaces. Gestural interfaces can be expressive, quick to access, and intuitive (Guimbretière and Winograd, 2000; Pirhonen et al., 2002; Starner et al., 1998; Witt, 2007). Yet gesture-based interfaces may trigger functionality incorrectly, confusing normal movement with a command. For example, the Apple iPod's "shake-to-shuffle" gesture, which is intended to signal when the user wants to skip a song and randomly select another, tends to trigger falsely while the user is walking (see Figure 1a). Part of the difficulty is that the recognizer must constantly monitor an accelerometer to determine if the gesture is being performed. Some accelerometer or gyro-based interfaces constrain the problem by requiring the user to segment the gesture by pressing a button. For example, in Nintendo's Wii Bowling the player presses the "B" trigger when beginning to swing his arm and releases the trigger at the end of the swing to release the virtual bowling ball. Such a push-to-gesture approach is similar to the push-to-talk method that speech recognition researchers use to improve performance. Yet such mechanisms can slow interactions, confuse users, and limit the utility of gesture interaction. For example, the fast,

easy-to-access nature of the shake-to-shuffle gesture would be impeded if the user needed to hold a button to perform the gesture. Ideally, such free-space "motion gestures" (Ashbrook, 2009) should be short, pleasing to perform, intuitive, and easily recognized by a computer against a background of the user's normal movements.

Touchpad gesture shortcuts, which upon execution can start an affiliated application on a laptop or mobile phone (Ouyang and Li, 2012), are another example of command gestures that must be differentiated from everyday motions. Fortunately, these gestures are naturally isolated in time from each other since most touchpad hardware does not even provide data to the operating system when no touches are being sensed. However, an interface designer must still create gesture commands that are not easily confused with normal click or drag and drop actions (see Figure 1b).

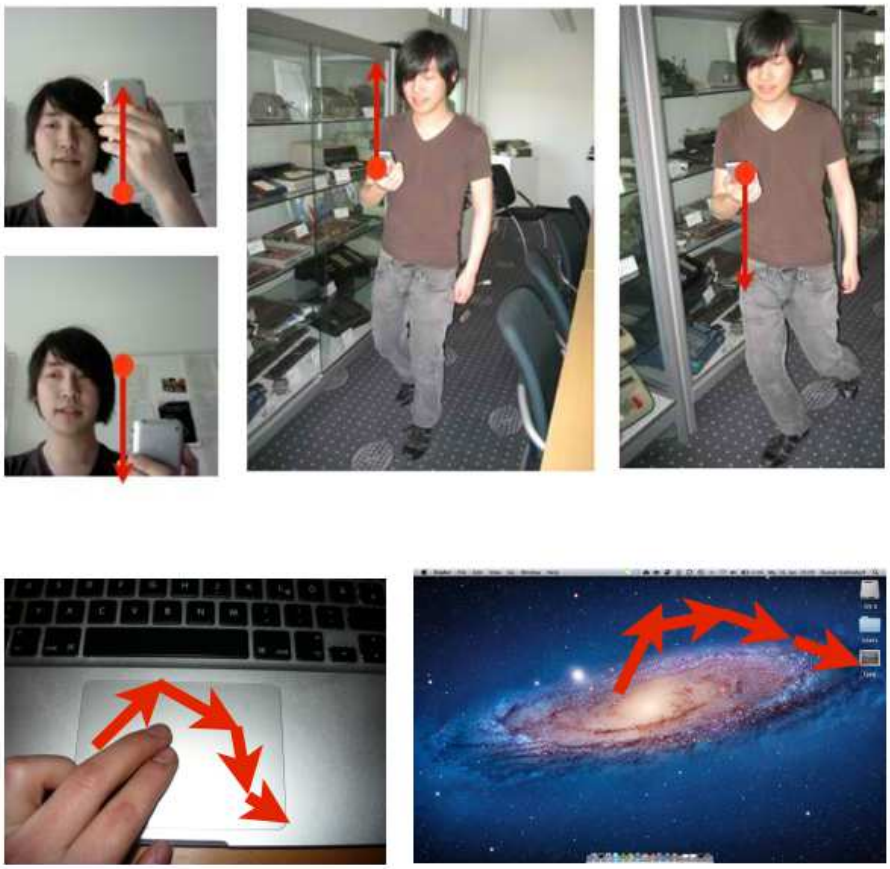


Figure 1: *Top:* A "shake-to-shuffle" gesture (*left*) can be confused with normal up-and-down movement while walking (*right*). *Bottom:* A touchpad shortcut gesture (*left*) can be confused with normal cursor movement (*right*).

Many "direct manipulation" (Hutchins et al., 1985) gestures such as pointing gestures and pinch-to-zoom gestures are used in modern interfaces. These gestures provide the user continuous feedback while the gesture is occurring, which allows the user to adjust to sensing errors or cancel the interaction quickly. However, representational gestures that are intended to trigger a discrete action

are less common. We posit that their relative scarcity relates to the difficulty of discovering appropriate gestures for the task. Our previous studies have shown that designing command gestures that do not trigger accidentally during normal, everyday use is difficult for both human computer interaction (HCI) and pattern recognition experts (Ashbrook and Starner, 2010). In addition, the current process to determine the viability of a gesture is challenging and expensive. Gestures are often found to be inappropriate only after the system has entered user testing. If a gesture is found to trigger accidentally during testing, the gesture set has to be changed appropriately, and the testing has to be repeated. Such an iterative design cycle can waste a month or more with each test. Thus, we posit the need for a tool to help designers quickly judge the suitability of a gesture from a pattern recognition perspective while they focus on the user experience aspects of the gestural interface.

Several gesture design tools have been described in the HCI literature (Dannenberg and Amon, 1989; Long, 2001; Fails and Olsen, 2003; Maynes-Aminzade et al., 2007; Dey et al., 2004), yet none address the issue of false positives. Similarly, most gesture recognition toolkits in the pattern recognition and related literature focus on isolated gestures (Wobbrock et al., 2007; Lyons et al., 2007) or the recognition of strings of gestures, such as for sign language (Westeyn et al., 2003). Rarely do such tools focus on gesture spotting (Yang et al., 2009) for which the critical metric is false positives per hour.

Ashbrook and Starner (2010) introduced the Multiple Action Gesture Interface Creation (MAGIC) Toolkit. A MAGIC user could specify gesture classes by providing examples of each gesture. MAGIC provided feedback on each example and each gesture class by visualizing intra- and inter-class distances and estimating the prototype recognizer's accuracy by classifying all provided gesture examples in isolation. Unlike the above tools, MAGIC could predict whether a query gesture would tend to trigger falsely by comparing the gesture to a database of movements recorded in the everyday lives of users. Primarily designed as an HCI Tool, the system used a nearest neighbor method with a dynamic time warping (DTW) distance measure (Fu et al., 2008).

One shortcoming of this work was that the relative false positive rates predicted in user studies were not compared to the actual false positive rates of a gesture recognizer running in the field. Another shortcoming was the long time (up to 20 minutes) needed to search for potential hits in a database of everyday user movements (an "Everyday Gesture Library" or EGL) even while using approximations like scaling with matching (Fu et al., 2008). MAGIC was designed as an interactive tool, yet due to the delay in feedback, gesture interaction designers waited until all gestures were designed before testing them against the EGL. Often, when doing an EGL test in batch, the interface designers discovered that many of their gestures were poor choices. Designers "learned to fear the EGL." Faster feedback would allow designers to compare candidate gestures to the EGL as they perform each example, speeding the process and allowing more exploration of the space of acceptable gestures. Another result from previous studies is that users were frustrated by encountering too many false positives in the Everyday Gesture Library (Ashbrook and Starner, 2010). In other words, many designed gestures are rejected since the number of predicted false positives is too high.

Here, we focus on the pattern recognition tasks needed to create MAGIC Summoning, a completely new, web-based MAGIC implementation designed to address the needs discovered from using the original. Section 2 introduces the basic operation of the tool. Section 3 describes an indexing method for the EGL using a multi-dimensional implementation of indexable Symbolic Aggregate approXimation (iSAX) that speeds EGL comparisons by an order of magnitude over the DTW implementation. While not as accurate as DTW or other methods such as HMMs, our system's speed allows interface designers to receive feedback after every gesture example input instead

of waiting to test the gesture set in batch. We compare the iSAX approach to linear searches of the EGL with HMMs and DTW to show that our approach, while returning fewer matches, does predict the relative suitability of different gestures. Section 4.4 continues this comparison to show that the predictions made by MAGIC match observations made when the resulting gesture recognizers are tested in a real continuous gesture recognition setting. Sections 5 and 6 provide additional details. The first describes a method of using the EGL to create a null (garbage) class that improves the performance of a HMM classifier and a DTW classifier when compared to a typical thresholding method. The second demonstrates the stability of our method by examining its sensitivity to its parameters and provides a method capable of learning reasonable defaults for those parameters in an unsupervised manner. These sections expand significantly upon previous work published in Face and Gesture (Kohlsdorf et al., 2011), while the remaining sections represent unpublished concepts.

Section 7 may be of the most interest to many readers. This section describes how MAGIC Summoning suggests novel gestures that are predicted to have a low probability of false positives. While the capability may be surprising at first, the technique follows directly from the iSAX indexing scheme. In Section 7.2 we show that the suggested gestures have low false positive rates during a user study in a real life setting. In our tests, the space of gestures that are not represented in EGLs tends to be large. Thus, there are many potential gestures from which to choose. Section 7.3 describes our attempts at finding metrics that enable ordering of the suggested gestures with regard to brevity, simplicity, and "quality."

2. MAGIC Summoning Web-based Toolkit

MAGIC Summoning is a web-based toolkit that helps users design motion-based gestural commands (as opposed to static poses) that are expected not to trigger falsely in everyday usage (Kohlsdorf, 2011; Kohlsdorf et al., 2011). All MAGIC experiments described in this paper focus on creating **user-independent** recognizers. This choice reflects our interest in creating useful gesture interfaces and is also due to practicality; collecting large data sets for the EGL from a single user is time consuming and onerous. To ground the discussion with a practical problem, we focus on the challenge of designing gestures performed by moving an Android phone in one's hand. We assume a three-axis accelerometer, which is always included in modern Android phones. The goal is to create gestures (and an appropriate classifier) that, when recognized, trigger functions like "open mailbox" or "next song." Without a push-to-gesture trigger, such gestures are highly susceptible to false positives (Ashbrook, 2009), which emphasizes the need for the MAGIC tool.

2.1 Creating Gesture Classes And Testing For Confusion Between Classes

MAGIC Summoning has two software components: a gesture recorder running on the Android device and the MAGIC web application. The first step in gesture creation is to start a new project in the web service. The interface designer specifies the set of gestures through collecting training data for each of the gestures using the recorder. In order to record a training example, the interaction designer opens the recorder on his smart phone and performs the gesture. The recorder automatically estimates when the gesture starts and when it ends using the method described by Ashbrook (2009). Specifically, the recorder tracks the variance of the accelerometer data in a sliding window. If the variance is above a user-defined threshold, recording starts. If it falls below the threshold, then recording ends.

After the example is recorded, the designer is asked to associate the example with an appropriate gesture label, and the recorder uploads the example to the web. The designer evaluates the gesture in the web application to determine how well it can be distinguished from other gestures. All gestures and their examples are listed in MAGIC Summoning's sidebar (see Figure 2). Examples marked with a red cross are misclassified given the current model, and instances marked with a green circle indicate correct classification. By default, MAGIC Summoning uses a one nearest neighbor classifier with dynamic time warping (NN-DTW) to classify gestures, although other classifiers such as a hidden Markov model (HMM) could be substituted. By clicking on an instance, the designer can see the raw sensor data plotted for that example as well as the predicted number of false positives in the EGL (the method used to calculated this number is explained in Section 3).

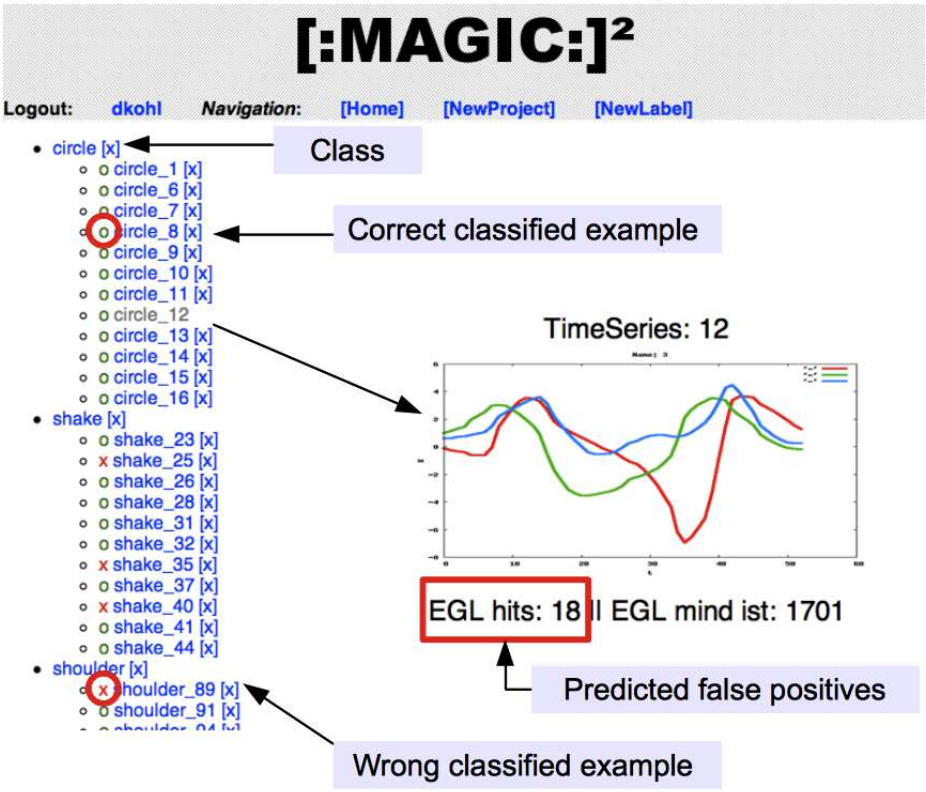


Figure 2: Magic Summoning showing the gesture classes, their examples, and the number of EGL hits (lower numbers are better).

Clicking on a gesture in the sidebar opens a view with statistics about it. One statistic is the goodness of the gesture. The goodness is defined as the harmonic mean of precision and recall (Ashbrook, 2009):

$$goodness = 2 * \frac{precision * recall}{precision + recall}.$$

Similar to the original work by Ashbrook (2009), MAGIC Summoning provides users with information about the inter-class distance and the intra-class distance of the gesture. Both are visualized using a mean and standard deviation plot. In an intra-class distance plot we calculate the means and standard deviations of the distances from all examples in a class to all other examples in that class and visualize the result as a box plot (see Figure 3). In an inter-class distance plot we calculate the means and standard deviations from one class to all the others in the training set. The distance between two classes is the mean distance of all examples of one class to all examples of another. These statistics and visualizations help designers find inconsistencies in the examples of a given gesture class as well as unintentional similarities between classes.

2.2 Android Phone Accelerometer Everyday Gesture Library

We collected a large EGL ($>$ 1.5 million seconds or 19 days total) using six participants' Android phones in Bremen, Germany. The age of the participants ranged from 20 to 30 years. We implemented a background process that wrote the three-axis accelerometer data to the phone's flash memory. Unfortunately, the sampling frequency varied as the models of Android phones we used return samples only when the change in the accelerometer reading exceeds a factory-defined threshold. The phones used are the Motorola Droid, the Samsung Galaxy, HTC Nexus One, the HTC Legend, and the HTC Desire. Other EGLs loadable in MAGIC include movements sensed with a Microsoft Kinect and gestures made on trackpads. We focus mostly on our EGL created with Android phones, but readers interested in experiments with other sensors can refer to Kohlsdorf (2011) for more information.

2.3 Testing For False Positives With The EGL

The original Macintosh-based MAGIC tool displayed a timeline that showed which candidate gesture matched the EGL and at which time. However, gesture designers did not care when or why a given gesture showed a given false positive in the EGL; they just wished to know how many "hits" occurred in the EGL so that they could accept or reject the gesture (Ashbrook, 2009). Thus, we omitted the timeline for simplicity in the web-based application. In the following section we will describe our accelerated method for testing a gesture for potential false positives against the EGL. This method enables rapid iteration on different gesture sets by the interaction designer.

If a user is displeased by the results after testing, he can delete gestures suspected of high false positive rates or misclassification errors and design new gestures. When the user is satisfied with the gesture set, MAGIC Summoning can train a classifier based on hidden Markov models (HMMs) or the default NN-DTW method. The user can then download the trained recognizer. Note that we do not suggest using the iSAX method used to search the EGL as a gesture recognizer as we have tuned the method for speed, not accuracy.

3. False Positive Prediction

When testing a gesture set against the EGL, the original MAGIC calculates the DTW distance for every example of each candidate gesture, sliding a window through time across the EGL and allowing the window to grow or shrink to better match the example when a potential close match is discovered. If the resulting distance is above a certain user-defined threshold it counts as a false

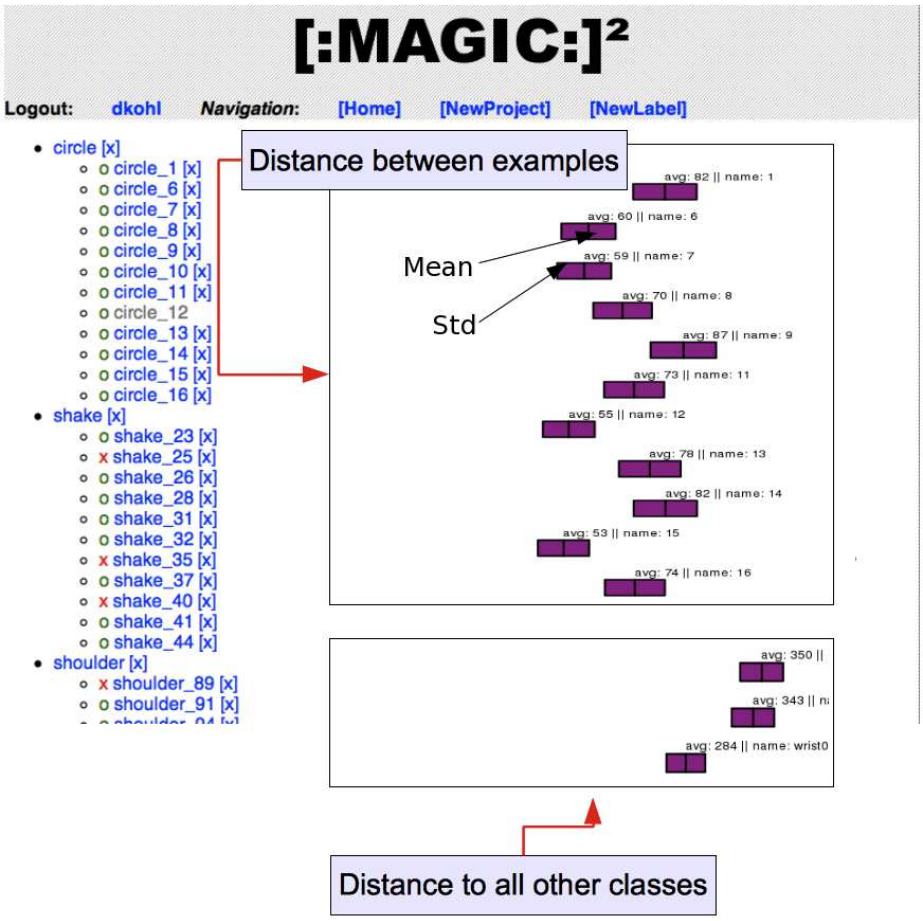


Figure 3: Mean and standard deviation of the distance between each example in a class and of the class as a whole in relation to other classes.

positive "hit." Ashbrook and Starner (2010) assert that the sum of the hits predicts how well the gesture will perform in everyday life (an assertion supported by our experiments described later).

In optimizing the speed of the EGL comparison, Ashbrook (2009) observed that not all regions of the EGL need checking. Since we are interested in motion-based gestures instead of static poses, parts of the EGL with low variance in their signal need not be examined. Thus, we pre-process the EGL to find "interesting" regions where the average variance over all dimensions in the sensor data in a region defined by a sliding window over 10 samples exceeds a given threshold (see Figure 4).[1] Eliminating regions from the EGL that can not possibly match candidate gestures significantly speeds EGL search. Note that a similar technique was described earlier to segment gestures when the interface designer is creating examples of candidate gestures. All experiments in this paper will use these techniques.

1. Word spotting algorithms in speech recognition perform similar checks, rejecting regions of "silence" before employing more computationally intensive comparisons.

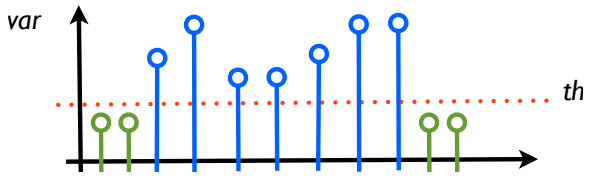


Figure 4: When finding start and stop points of a gesture or finding interesting regions in the EGL, we run a sliding window over the raw recorded time series and calculate the sample variance in that window when a new sample is inserted. If the variance is above a certain threshold, the gesture or interesting region starts. It stops when the variance falls below that threshold.

Searching the EGL parallelizes well, as each processor can be devoted to different regions of the EGL. However, even on a high-end, eight-core Macintosh workstation, searches were too slow for an interactive system. For a small, five-hour EGL with three-axis accelerometer data sampled at 40Hz, each example required between 5-25 seconds to check. Thus, one gesture with 10 examples could require minutes to search in the EGL. This slowness causes interface designers to create gestures in batch and then check them against the EGL. Testing a set of eight gestures with all their examples could take up to 20 minutes, leading to a relatively long and frustrating development cycle for the designer (Ashbrook and Starner, 2010). In the following sections, we describe a method to speed the EGL search using iSAX. We start with an overview of our method and our assumptions. We then provide the specific methods we used to adapt iSAX to our problem.

3.1 Overview Of EGL Search Method And Assumptions

In MAGIC Summoning, we first segment the EGL into interesting regions as defined previously. Each region is divided into four even subregions to form a "word" of length four. The region is then encoded into a string of symbols using the standard SAX quantization method. The string is entered into an iSAX tree representing the EGL. The iSAX tree is initialized with cardinality two but quickly grows as many regions hash to the same leaf on the suffix tree and the leaf needs to be split (Shieh and Keogh, 2008). As each region is encoded into the iSAX tree, its location in the EGL is recorded in the leaf. Once the EGL is completely encoded into an iSAX tree, we can perform "approximate search" using a gesture example as a query (Shieh and Keogh, 2008). The query is split into four regions and SAX-encoded in much the same way as the interesting regions of the EGL. An approximate search to determine the number of matches between the query and the EGL becomes a simple matter of matching the query string to the appropriate branch of the iSAX suffix tree and returning the number of strings contained in that branch.

One failing of this approach is that the interesting regions may be significantly larger or smaller than the candidate gestures. Regions significantly smaller than the command gestures are not of concern as they will never falsely match a command gesture in practice. We can eliminate such regions out-of-hand from the comparison. However, regions of movement that might match the query gesture may be hidden within longer regions in the EGL.

A key insight, which will be used repeatedly, is that we need not recover every region of the EGL that might cause a false match with the query. We are not intending iSAX to be used as a

gesture recognizer. Instead, our goal is to allow the designer to compare the suitability of a gesture relative to other candidates quickly. As long as the movement occurs repeatedly in the EGL at isolated times as well as in longer regions, the iSAX method will report a number of "hits," which will be sufficient to warn the interaction designer of a problem.

A second insight is that users of gesture interfaces often pause before and after they perform a command gesture. Gesture recognizers exploit this behavior and use these pauses to help identify the command gesture. Movements that look like command gestures embedded in long regions of user motion are unlikely to be matched in practice by these recognizers. However, short everyday user motions that are similar to a command gesture are a particular worry for false positives. Thus, the iSAX encoding scheme of the EGL above seems suitable for our needs. However, if the goal of the interaction designer is to create gestures that can be chained together to issue a series of commands quickly, these longer regions in the EGL will need to be encoded more formally using constraints on how long a section can be encoded in each symbol. Such constraints can be derived from the length of expected command gestures (usually between 1-4 seconds in our experience), and the length of SAX word defined by the system.

A final insight is that a more precise comparison against the EGL can be made at the end of the gesture design process with the gesture recognizer that is output by MAGIC. During gesture design, all we require of the EGL search method is that it is fast enough to be interactive and that it provides an early warning when a given gesture may be susceptible to false triggering. Given the above operational scenario, we tune our iSAX implementation to provide fast feedback to the user. Details on the implementation follow below.

3.2 SAX Encoding

SAX quantizes time series in both time and value and encodes them into a string of symbols (Lin et al., 2007). For example, the time series in Figure 5 is divided into four equal portions (for a "word" length of four) and converted into a string using a four symbol vocabulary (a "cardinality" of four).

To be more precise, we first normalize the time series to have a zero mean and standard deviation of one. Assuming the original time series $T = t_1, ..., t_j, ...t_n$ has n samples, we want to first quantize the time series into a shorter time series $\bar{T} = \bar{t}_1, ..., \bar{t}_i, ...\bar{t}_w$ of word length w. The i^{th} element of $\bar{T}$ can be calculated by

$$\bar{t}_i = \frac{w}{n} \sum_{k=(\frac{n}{w}(i-1)+1)}^{\frac{n}{w}i} t_k.$$

Given the values in the compressed time series, we next convert them into symbols using a small alphabet of size (cardinality) a. Imagine the y-axis divided into an arbitrary number of regions bounded by $a-1$ breakpoints. Each of these regions is assigned to a symbol from the alphabet. Since we wish each symbol in the vocabulary to be used approximately the same amount, we place a normal Gaussian curve centered at 0 on the y-axis and place the breakpoints such that the area under the Gaussian for each section is equal. By performing the SAX process on the EGL and each gesture example separately, we are able to compare the changes in the signals through time without concern regarding their offsets from zero or relative amplitudes.

One convenience of the SAX representation is that there exists a distance calculation between two strings, defined as MINDIST by Lin et al. (2007), that is a lower bound on the Euclidean

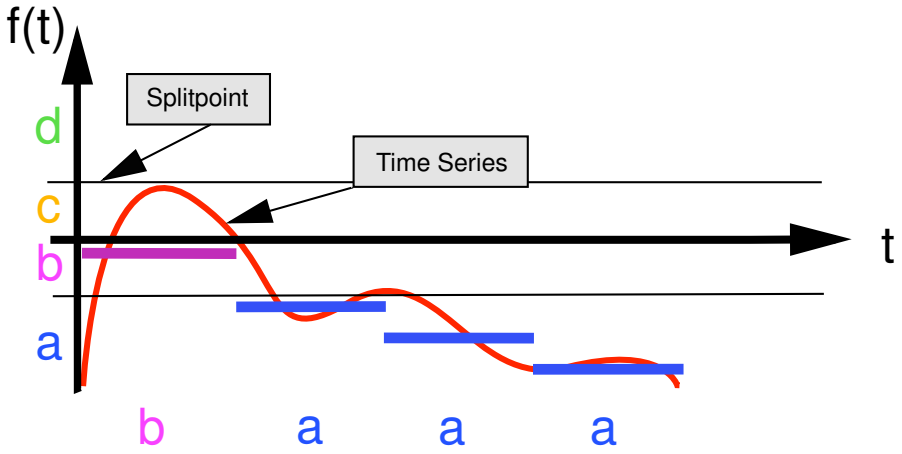


Figure 5: SAX process used to convert a time series to a string. The raw data is segmented into a user-specified word length, in this case four. Then each segment is replaced by a symbol associated with that region on the y-axis, based on the average value. The resulting string is represented by the string of symbols with superscripts indicating the number of symbols used to quantize each region: $b^4a^4a^4a^4$.

distance between the original two time series. Thus, we can search the EGL for possible false positives with some measure of confidence.

Another convenience of the representation is that the cardinality of each separate region can be increased whenever more precision is needed. For example, suppose we increase the cardinality of the first region in Figure 5 to eight (thus, the vocabulary would include letters a-h). The string might then be $d^8a^4a^4a^4$, as the region of the y-axis formerly covered by symbols a and b would now be covered by symbols a, b, c, and d. We can compare strings with regions of different cardinality by observing that we know that each time series is normalized before SAX encoding and that the regions are defined by a normal Gaussian centered at zero with all regions having an equal area under the Gaussian's curve. Thus, we still know the minimal distance possible between each region, and we can still use MINDIST to determine a lower bound on the Euclidean distance between the original two time series. This capability will be useful in our upcoming discussion on iSAX and its application to the EGL.

3.3 Multi-Dimensional iSAX Indexing And EGL Search

iSAX is a tree-based method for time series indexing introduced in Shieh and Keogh (2008). For encoding the EGL, our goal is create an iSAX tree that can be traversed quickly when searching for a match to a SAX-encoded example of a gesture. Each leaf of the tree contains the number of occurrences of that string in the EGL as well as the position of each occurrence. To begin, assume we are searching an EGL represented by the simple iSAX tree in Figure 6 with a query represented by $a^2a^2b^2b^2$ (for the sake of argument, assume we decided to represent the example gesture crudely, with regions of cardinality two). Immediately, we see that there is no branch of the tree with an a^2 in

the first position, and we return no matches in the EGL. Now assume that we are searching the EGL for a query of $b^2b^2b^2b^2$. We find that there is a node of the EGL that contains that string, and that node has children (that is, the node is an "internal node"). Looking at the children in that branch, we see that we need to re-code the query gesture to have cardinality three in the first region. Re-coding reveals that the query gesture is better represented by the sequence $c^3b^2b^2b^2$, which matches one of the terminal leaves in the tree. The number of sequences from the EGL stored in that leaf is returned as the number of "hits" in the EGL.

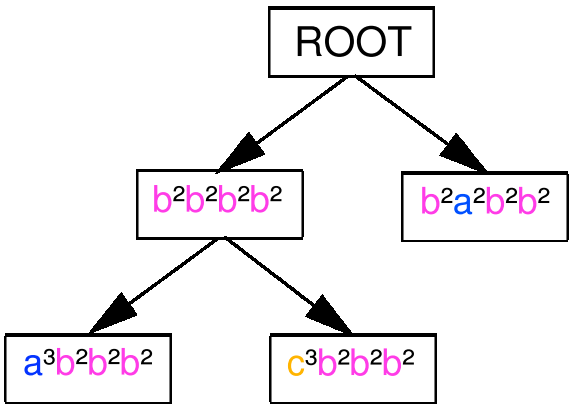


Figure 6: iSAX tree with three leaves. On the first level all symbols' cardinalities are equal. The node $b^2b^2b^2b^2$ is an internal node. For the children under this node, the cardinality of the first region is increased by one.

Next we describe how to encode a one-dimensional EGL into an iSAX tree. First, we find all the "interesting" regions in the EGL using the variance method discussed earlier. We divide the regions evenly into four sections and encode them using SAX with cardinality two, allowing for sixteen possible strings. Note that each node in an iSAX tree holds a hash table mapping child nodes to an iSAX word. Thus, when inserting a region into the iSAX tree, we compare the region's SAX string to the hash table in the root node. If there is no match, we create a child node and enter it into the hash table using its SAX string. If the SAX string is found, we examine the node to see if it is a terminal leaf. Each leaf points to a file (called a "bucket") stored on disk holding all of the regions that have mapped to it. The leaf also contains the position of each of the regions in the EGL and a count of the number of regions contained in the leaf. If the number of regions in the bucket exceeds a user specified size (called the "bucket size"), it is deleted, and the cardinality of the iSAX word is increased at one position (picked by round robin). At the deleted node's position we insert a new internal node. All the time series of the deleted node are inserted into the new node but with a higher cardinality. Children of the internal node are created as needed, effectively splitting the previous leaf into several new leaves. When we encounter a internal node during the insertion of a region, we search the node's hash table for children that match and proceed normally, creating a new leaf node if no matching child exists.

Note that this method of creating the iSAX tree dynamically adjusts the size of the vocabulary to better distinguish similar regions in the EGL. Given a bigger vocabulary, the SAX word will fit more exactly to the region. In other words, this method of encoding devotes more bits to describing

similar movements that are repeated often in the EGL. Thus, when a query gesture is compared to the EGL iSAX tree, MAGIC will quickly return with no or few hits (depending on the specified bucket size) if the query is very distinct from the EGL. If the query is similar to motions in the EGL, the search process will traverse deeper in the tree, examining finer and finer distinctions between the query and the regions contained in the EGL.

The above discussion assumed that the data was one-dimensional. For multi-dimensional data, such as is used in the experiments described below, we create n iSAX trees, one for each dimension of the recorded data. We index all dimensions separately and join those n trees under one new root node (see Figure 7).

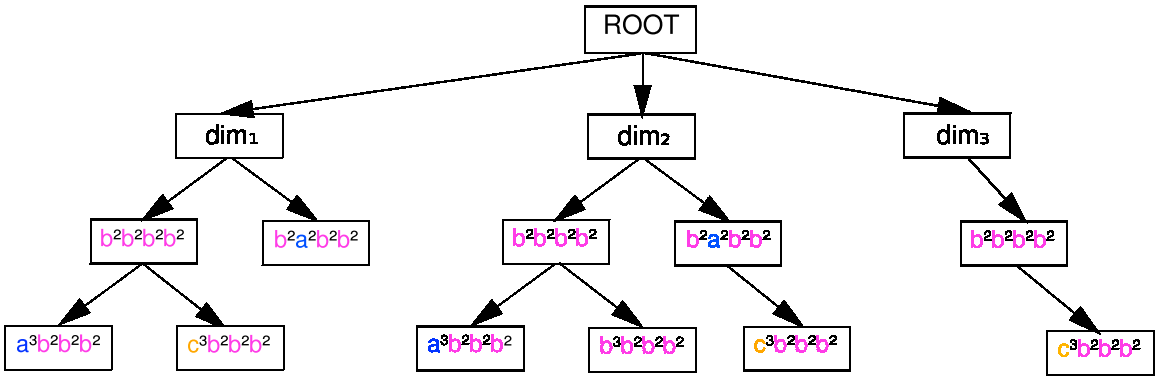


Figure 7: A multi-dimensional iSAX tree. Under the root node there is a dimension layer. Each node in this layer is the root node for a one-dimensional iSAX tree. During search, we search all iSAX trees, one for each dimension.

We query the EGL iSAX tree (constructed from the EGL) in all n dimensions. The result of that search is n files, one for each dimension. The number of hits can then be calculated by counting the number of places where each hit from each dimension overlap for all dimensions. Comparing the timestamps can be costly, so we introduced an approximation based on the observation that there can never be more overlapping time series than the number in the dimension with the lowest number of matches. For example, consider the result of a search in three dimensions (x, y, z) where the number of hits in the EGL are $x = 4$, $y = 20$ and $z = 6$. There can never be more then four hits total if we require that hits must overlap in all dimensions. The overall EGL testing method is summarized in Table 1.

Upon reflection, the EGL search procedure described above raises several questions and possibilities. What are reasonable values for the bucket size, word size, and cardinalities used in encoding the EGL, and how sensitive is MAGIC to these parameters? This question will be examined in detail in Section 6. A nice side effect of EGL search is that we can use the matches found to train a class of gestures that a recognizer should ignore (a "garbage" or NULL class). Section 5 will explore this option. Searching for which SAX strings are not contained in the EGL tree can suggest which gestures are not made during everyday movement. In Section 7, we exploit this attribute to recommend gestures to the interaction designer. However, first we will provide evidence that searching the EGL does indeed predict the number of false positives during the usage of a gesture interface.

```
Test preparation:
0) Collect a large data base of user movements in advance.
1) Find interesting regions by applying variance thresholding.
2) Build an n dimensional iSAX tree.

Gesture testing:
0) Find start and end point of gesture.
1) Search the iSAX tree in all n dimensions.
2) Return the number of time series in the minimum file.
```

Table 1: Testing gestures for potential false positives against a database of pre-recorded device usage.

4. Experimental Verification

In the following section we describe two experiments that suggest that an iSAX search of the EGL is a viable means to predict false positives. Our first goal is to show that false positive prediction using iSAX is correlated with the previous method of searching the EGL linearly using dynamic time warping (Ashbrook, 2009). We will also conduct an experiment in which we will show that the EGL is able to predict the relative number of false positives when using a gesture interface in everyday life. We describe the data used for the experiments and our experimental method before presenting our findings.

4.1 EGLs And Gestures Used In Evaluations

We use three different data sets to serve as EGL databases. The first is our Android accelerometer data set as described earlier. Before indexing the recorded data, we extracted the interesting regions, applying a threshold of $th = 0.001$ (triggering at almost any movement) and a window size of $N = 10$ (0.25 sec at 40Hz). The average duration of the interesting regions is $11,696ms$. The second EGL is based on the Alkan database[2] of everyday movements collected with an iPhone (Hattori et al., 2011). The third data set is another collection of everyday movements collected on Android phones for a different project at Georgia Tech. These two latter EGLs were processed in the same manner as the first.

We collected a reference data set of gestures for evaluation purposes. We acted as interaction designers and designed four gestures by performing them while holding a smart phone. For each gesture we collected 10 examples, resulting in 40 examples total. The four gestures are: drawing a circle in the air, touching your shoulder, shaking the phone up and down, and hacking (a motion similar to swinging an ax). The average duration of the gestures is between one and two seconds.

4.2 Comparison Conditions: NN-DTW And HMMs

When comparing the dynamic time warping EGL search method to a search in iSAX index space we will use the following procedure. The DTW method compares each interesting region from the EGL to each gesture example (Ashbrook, 2009). We calculate the dynamic time warping distance

2. Alkan web site can be found at: `http://alkan.jp/`.

of a new gesture to all examples in the EGL and apply a threshold chosen empirically. All regions for which the distance is below this threshold for any example count as a false positive (in keeping with MAGIC's ability to output a one nearest neighbor classifier for live gesture recognition).

For yet another comparison, we use hidden Markov models to search the EGL for false positives. For the experiments in this paper, we use a six-state HMM (ignoring initial and end states) with one skip transition and one Gaussian output probability per state per dimension (see Figure 8). We collect all the examples for our gesture set first and then train a HMM for each of the gestures. We classify each region in the EGL and apply a threshold based on maximum likelihood to determine if a region in the EGL is close enough to the gesture to count as a false positive. We chose both the maximum likelihood threshold as well as the distance threshold so that classifier accuracy stayed high (93% for NN-DTW and 100% for HMM).

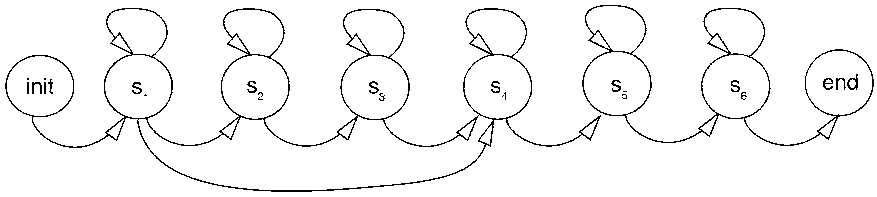


Figure 8: The topology of the left-right, six-state HMM used in our experiments. The first state is the start state, and the eighth state is the end state. Each internal state transitions to itself and its successor. We include a skip transition to help recognize shorter gestures.

4.3 Comparison Of iSAX To NN-DTW And HMM In Searching EGLs

We wish to compare our iSAX EGL search method to the more conventional NN-DTW and HMM techniques described above. When selecting between two candidate gestures, the interaction designer wishes to choose the one with a lower number of predicted false positives. Thus, if a first gesture has few hits when NN-DTW or HMMs are used and a second gesture has many hits, that same trend should be shown with iSAX. The absolute number of EGL hits does not matter, but there should be a strong correlation between the relative number of hits returned by iSAX and the other two techniques when run on the same set of gestures. We use the Pearson correlation coefficient as a metric to compare the techniques.

Regardless of the search method used, we store the number of hits in a vector. Each entry of that vector corresponds to the overall number of false positives for a given gesture. For iSAX and NN-DTW, the overall number of false positives for a gesture is calculated by searching the EGL for each example of that gesture and summing the resulting numbers of hits. For HMM models, thresholding on the log likelihood probability is used. For our set of four test gestures, testing returns three vectors (one for each method) of four elements (one for each gesture). We calculate the Pearson correlation coefficient between the iSAX vector and the NN-DTW vector and between the iSAX vector and the HMM vector.

To reassure ourselves that this technique produces a meaningful metric, we performed Monte Carlo simulation experiments. Indeed, the correlation of random vectors with four elements show low r values.

First, we compare the search methods on the EGL from Bremen. We chose the iSAX parameters empirically:

word length: 4

base cardinality: 2

bucket: 6000.

Figure 9 compares the number of hits per hour returned by each method. The hits per hour metric reflects the number of matches found in the EGL divided by the original time required to record the EGL. One can see that our iSAX search approximation returns many fewer hits than NN-DTW or HMMs. However, the magnitude of the iSAX values correlate strongly with the NN-DTW ($r = 0.96$) and HMM ($r = 0.97$) results. Thus, a high number of hits returned by iSAX on the EGL (high compared to other gestures tested with iSAX) is a good indicator for when a gesture should be discarded. The remaining gestures are suitable candidates for user testing.

We also measured the time needed to complete the search for each method on a 2.0GHz Intel Core Duo T2500 Macbook with 2GB of RAM. The NN-DTW and HMM methods require more then 10 minutes to complete the search on all 40 gesture examples whereas iSAX search required 22 seconds, a 27X increase in speed. With such speed, each of the gesture examples could have been checked as it was entered by the interaction designer. In fact, the EGL search would require less than a second for each gesture example, which is less than the amount of time required to check a new example for confusion against all the other gesture examples with NN-DTW when creating a eight gesture interface (Ashbrook, 2009). Thus, we have obtained our goal of maintaining interactivity during gesture design.

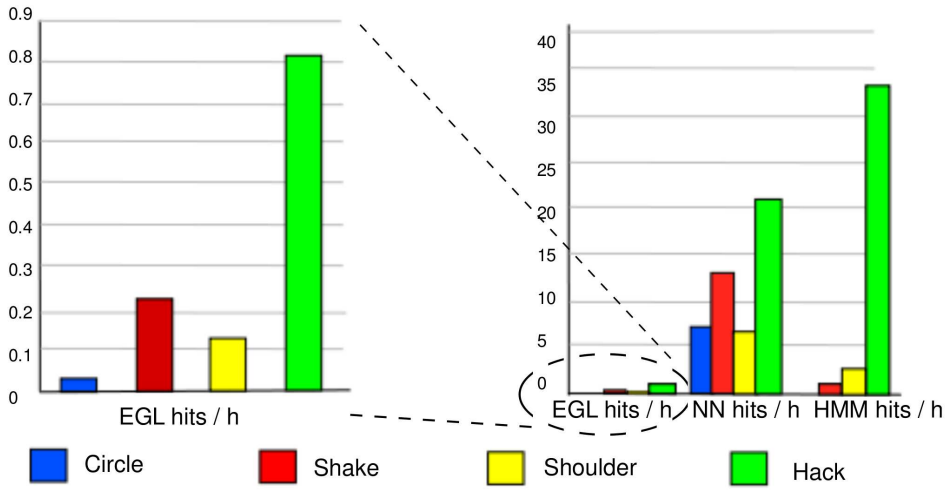


Figure 9: *Left:* The hits per hour in the EGL based on iSAX search. *Right:* A comparison of the number of hits per hour returned by iSAX, NN-DTW, and HMMs from the EGL.

We were curious as to how much EGL data is needed to predict poor command gestures. We generated three random subsets of the EGL by picking 100, 200 and 500 interesting regions at random from the data set and comparing the correlation coefficient between iSAX and NN-DTW. The correlation between the results remained surprisingly high, even with an EGL containing only 100 regions:

- **n = 100:** $r = 0.89$
- **n = 200:** $r = 0.93$
- **n = 500:** $r = 0.93$.

As later experiments show, more data is better, but even a relatively small EGL can help the interaction designer avoid choosing troublesome gestures. We also compared iSAX versus NN-DTW in the Alkan and Georgia Tech EGLs, with similar results to the original Bremen EGL:

- **Alkan:** $r = 0.94$
- **Georgia Tech:** $r = 0.99$.

Our results suggest that the results of an iSAX search on the EGL correlate highly with those of the slower EGL search methods. Even though the absolute number of hits found by the iSAX method are significantly fewer than the other methods, the relative number of hits can be used to compare the desirability of one candidate gesture versus another.

4.4 Comparison Of iSAX Predictions To HMM And NN-DTW Gesture Recognizer Use In Practice

Next, we examine whether our iSAX EGL search method is able to predict false positives in everyday life. In fact, this experiment is the first to verify that any EGL search is able to predict false positive rates of a gesture recognizer in practice.

We exported NN-DTW and HMM recognizers from MAGIC Summoning for the four gestures trained during the process described in the previous experiment. We integrated the HMM classifier into an interactive system. Next, we recruited four Android phone users who had not contributed to the EGLs nor the training of the gestures.

In order to understand how difficult it was to perform the gestures correctly, we asked the users to perform each gesture 10 times without feedback. The HMM classifier performed at 60% accuracy, which is not surprising given the gestures and testing procedure. Next we allowed the users to train with the HMM recognizer to become more familiar with how to perform the gestures so that they could be more easily recognized. This way of learning can be found in commercial systems like the Nintendo Wii, which uses avatars to help users learn control gestures. Not surprisingly, the four users' average accuracy with the HMM recognizer improved to 95% after training.

After the users completed their training, we installed a software application on their phones that notified the users when to perform one randomly selected gesture, once every hour. Otherwise, the users performed their normal activities, and the application records all the users' movements. We searched the recorded data for the intended gestures. The HMM classifier found $50\% - 70\%$ of the intentional gestures whereas NN-DTW search found all of them. However, the NN-DTW classifier

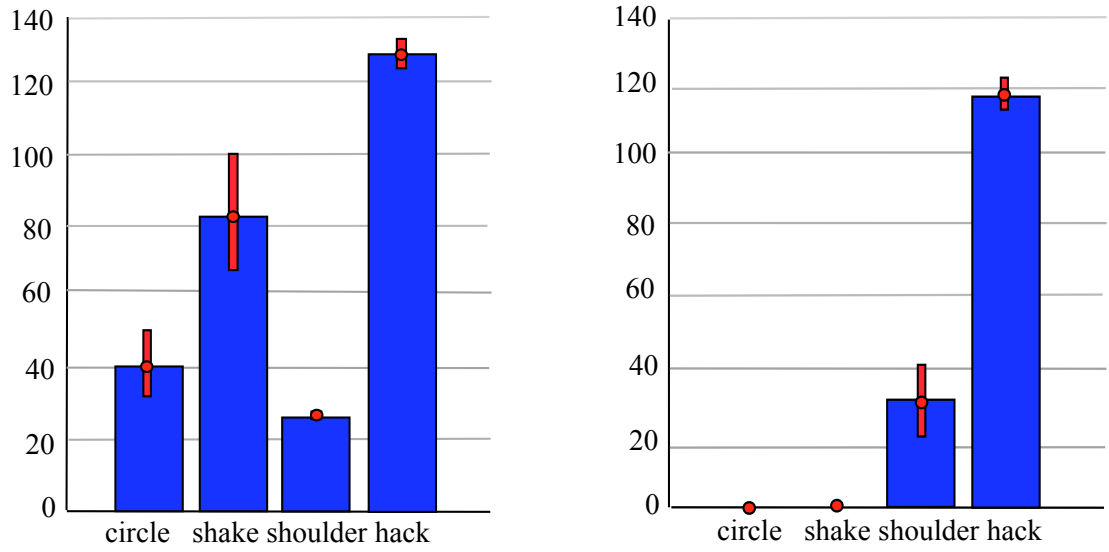


Figure 10: The EGL hits per hour found during deployment. *Left:* The EGL hits for NN-DTW search per gesture. *Right:* The EGL hits for HMM search per gesture. The EGL hits for a gesture are the average hits over all four users. The bars correspond to one standard deviation.

had lower precision than the HMMs. Given that we specifically allowed gestures that were known to be poor (from EGL testing) and that the system did not provide feedback to the users, such poor performance is to be expected (and desired from the point of the experiment).

Figure 10 shows the false positive rates for each gesture and recognizer. We observed a high correlation ($r = 0.84$) between the relative false positive rates predicted by the iSAX search on the original EGL and the actual, tested NN-DTW performance on the users' data. The correlation was even higher ($r = 0.97$) for the HMM classifier. These results support our hypothesis that MAGIC Summoning can be used to predict gestures at risk of having many false positives when deployed in gesture recognizers in practice.

5. Improving Recognition Through A NULL Class Created From EGL Search

In the experiments in the previous section, we needed to specify a threshold to avoid false positives when distinguishing the four gestures from our four users' everyday motions. For NN-DTW, the threshold was a distance, while with HMMs it was a probability. Setting this threshold requires more pattern recognition experience than an interaction designer may possess, and often gestures are not separable from everyday movements with a simple threshold. Another option is to create a NULL (garbage) class, which attempts to capture all the motion not matching the gestures of interest. With this technique, the recognizer runs continually but does not return a result when the sensor data matches the NULL class.

Here, we use EGL data to train a NULL class automatically so that a user-defined threshold is not needed. Multi-dimensional iSAX search of the EGL returns time series similar to a query

gesture. Thus, it is a simple matter to collect the EGL hits from all examples of all gestures in the gesture interface to train a NULL gesture (using either technique).

The following experiment is based on the data collected while our four users performed the four requested gestures during their daily activities. We adjusted the thresholds upward for the HMM and NN-DTW recognizers to avoid misclassifications in the EGL while still detecting the gestures from the training set. We also trained NULL classes for both recognizers. Figure 11 shows the results of all four recognizers running on the user study data. Using the EGL NULL class method resulted in a statistically significant improvement of both the NN-DTW ($p << 0.0001$) and HMM ($p < 0.05$) recognizers. Both avoided more false positives using the NULL class instead of a threshold. Gesture recognition accuracy and correlation to the iSAX EGL hits remained consistent with the experiment in the previous section. The results suggest that training a NULL class based on EGL hits can be a successful way to improve performance and reduce complexity for the interaction designer. Note that many variations of this technique are possible and might further improve results. For example, a different NULL class could be trained for each gesture.

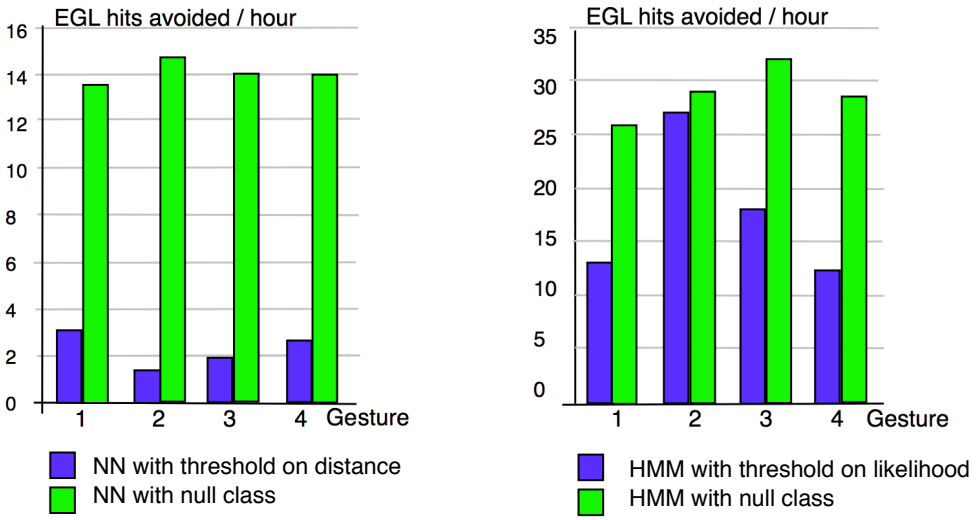


Figure 11: *Left:* The false positives per hour avoided using a NULL class for each gesture based on EGL hits versus the simple threshold. *Right:* The false positives per hour avoided for HMMs using the NULL class versus the simple threshold.

6. iSAX Parameter Sensitivity Experiments

In Section 4.4, iSAX was able to predict the relative performance of a gesture during continuous recognition. However, the process required setting several parameters: word length, bucket size, and initial cardinality. In addition, we compared the false positive predictions to that of the NN-DTW method, which itself required a distance threshold (when a NULL class is not used). How sensitive

is our method to these parameters? We use the same four test gestures (circle, shake, shoulder, hack) and EGL as in our user study to explore this issue.

Observe that the cardinality of the sequences is automatically adjusted during the creation of the EGL iSAX tree, quickly changing from its initial minimal setting of two. Effectively, this parameter is not set by the user, and we can remove it from the following experiments on parameter sensitivity by holding it at a reasonable value. We choose a base cardinality of four ($card = 4$), given that this level of complexity was judged sufficient from observations in the original iSAX experiments (Shieh, 2010; Shieh and Keogh, 2008) and in our own work (Kohlsdorf et al., 2011).

In the experiments below, we compare the iSAX results, while varying the bucket size and word length, to the NN-DTW method using the correlation method described above. We also tried comparing the iSAX method to NN-DTW with different reasonable distance thresholds (3.3, 5, 10), but we found little change in the results. For example, the bottom of Figure 12 shows graphs comparing iSAX word length to correlation with NN-DTW at each of the distance thresholds. The graphs are very similar, indicating that the comparison with NN-DTW is relatively stable with respect to the distance threshold used. Thus, we turn our attention to word length and bucket size.

In the first part of the experiment we test the correlation of the EGL searches using NN-DTW and iSAX trees constructed with different iSAX word lengths (4,5,6, ... ,13) and bucket sizes (1000, 2000, ... , 10000). Figure 12 plots the results. Changes in bucket size cause minor variations in correlation; however, word length has significant effects.

Since the performance of our method seems mostly dependent on one parameter, we propose an automatic parameter tuning method that does not require any data except a pre-recorded EGL. The central concept is to choose random regions from the EGL to serve as a gesture training set and to tune the iSAX parameters to that set using hill climbing.

We require the user to specify the number of gestures in the data set (N), how many examples we want to collect for each gesture (M), and a threshold on the dynamic time warping distance over which two time series are distinct. We pick N regions of motion ("interesting" regions) at random from the EGL to serve as "reference gestures." For those N reference gestures we extract M examples from the EGL where the DTW distance to the reference gesture is smaller than a threshold. Then we compute the false positives for this gesture set using the NN-DTW method. In order to find the appropriate word length we use hill climbing in the iSAX parameter space. At each step, we perform false positive prediction using iSAX and compare the results to the NN-DTW results using the Pearson correlation coefficient as an objective function.

We ran an experiment to test this hill-climbing technique, allowing the procedure to set the word length automatically and comparing the results to NN-DTW. We started the word length at 4 and increased it to 13. If the observed correlation at a given word length is followed by a smaller one when the next word length is tried, the algorithm stops and returns the last word length. As one can see in Figure 12, after 3 iterations iSAX finds a local maximum. However, this sequential method is not optimal. For example, if the word length which maximizes the correlation is 9 and the local maximum at the word length 6 is smaller, we would stop too early. However, this problem can be solved by including simulated annealing or stochastic gradient descent in the future.

In this chapter, we showed that the iSAX EGL search relies on several parameters but that the parameters can be tuned automatically. Word length seems the primary parameter that needs to be tuned.

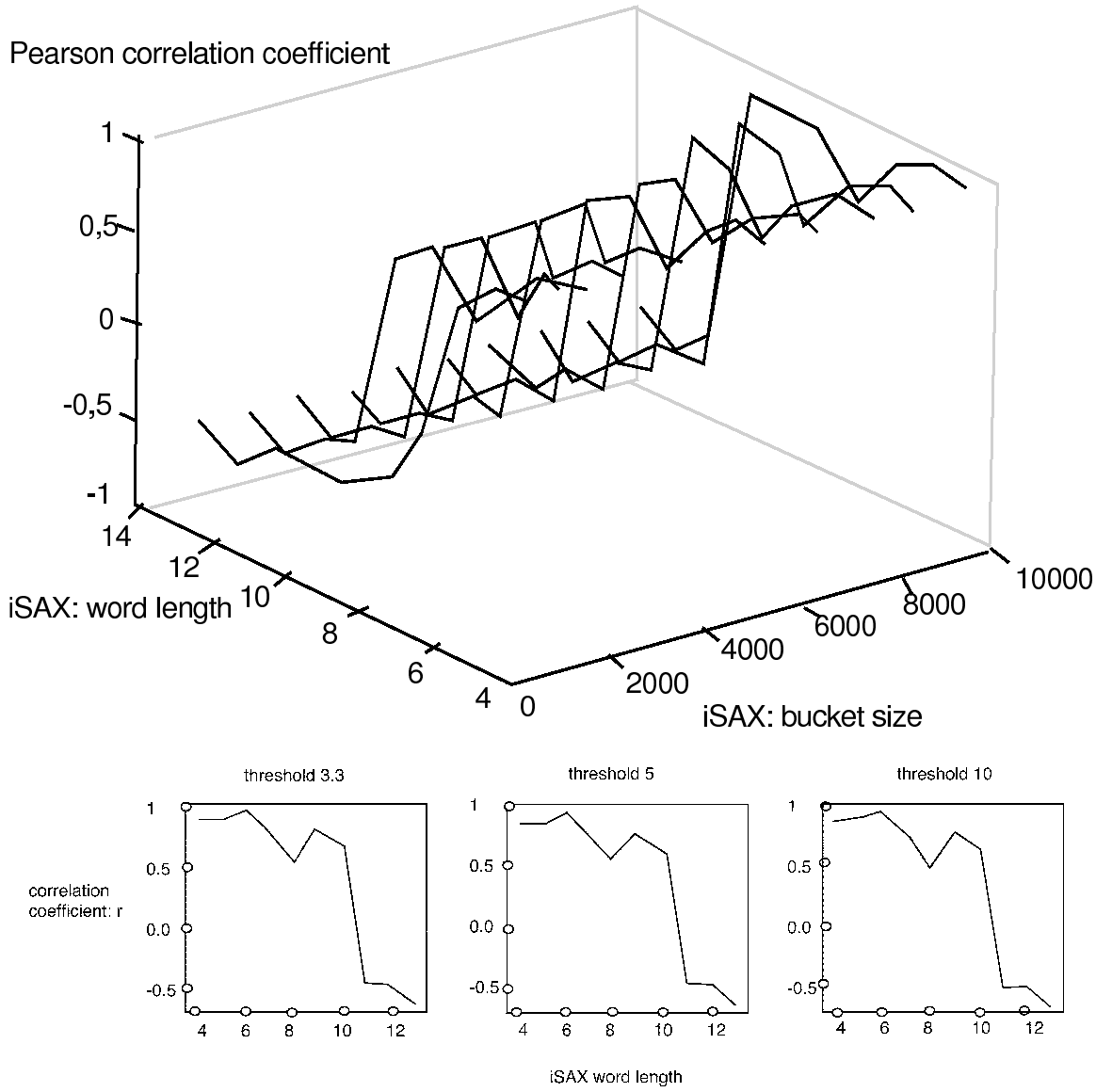


Figure 12: Top: correlation to NN-DTW vs. iSAX word length vs iSAX bucket size. Bottom: iSAX word length vs. correlation to NN-DTW for distance thresholds of 3.3, 5, and 10, respectively.

7. MAGIC Summoning: Suggesting Gestures With Low Probability Of False Positives During Use

To this point, we have focused on efficient gesture testing. However, when using MAGIC to design gestures in previous studies, our participants wished to have MAGIC suggest potential gestures instead of creating their own. Often the gestures designed by the participants showed high false positive rates when tested against the EGL, leading to frustration. MAGIC users said they would

rather select from a set of gestures that were "known good" than experiment blindly with constraints they did not understand (Ashbrook, 2009).

In the next section, we describe a method for suggesting gestures based on a pre-recorded EGL. We then perform an experiment where we test suggested gestures for false positives during normal device usage by naive subjects. Finally, we examine different possible metrics to order the suggestions for easier selection by the designer. While we have mostly used accelerometers in our experiments to date, here we concentrate on capacitive trackpads, specifically those used on Apple's laptops. Data from inertial sensors are hard to visualize for an interaction designer without a inverse kinematic system to map the sensor readings into limb movement. While such systems are now feasible with adequate accuracy, we wished to avoid the additional complexity for these first experiments. Trackpads provide two dimensional data that are easy to visualize for an interaction designer, and trackpads are commonly used in everyday office work. In addition, industry has begun to include more complex command gestures in their trackpad-based products (Li, 2010).

7.1 Synthesizing And Visualizing Gestures

We introduce a method for proposing gestures that do not collide with every day movements using four steps, briefly outlined here. First, we collect an EGL that is representative of the usage of the device or sensor. Next, we build an iSAX tree based on the EGL. We systematically enumerate the possible SAX strings and check for those which are NOT contained in the tree. Finally, we visualize these gestures and present them to the interaction designer. Once the designer selects a set of gestures for his interface, MAGIC Summoning can train a recognizer for the gestures using synthesized data.

7.1.1 COLLECTING AN EGL

Collecting a representative EGL is often time-consuming and is best done by someone familiar both with the specific sensor involved and pattern recognition in general. Fortunately, the process is only necessary once for the device of interest and then can be used for different interface designers and tasks. Mostly, the EGL will be collected across multiple people to ensure that the resulting gestures can be user independent. Ideally, the EGL should be collected across every situation and physical context where the device might be used (for example, sitting at a desk or driving) to make sure that incidental motions are well represented. If the resulting gesture recognizer is intended to work across different devices (for example, across multiple version of Android phones), the EGL should be collected from a representative sample of those devices.

7.1.2 REPRESENTING THE EGL AND GENERATING GESTURES

Next, we convert the EGL into a simplified iSAX tree structure. Unlike the work above, here we only care that a given string occurred in the EGL instead of how many times it occurred. Thus, we can use a simpler indexing method that will allow easier gesture building later. We convert interesting regions from the EGL to SAX words and build the set of all strings observed in the EGL. Since the sensor input is multivariate, we build the SAX word in each dimension and concatenate the words. Thus, for n dimensions and a word length of w, the indexing key grows to $n * w$. Given the cardinalities in the word, discovering gestures that are not represented in the EGL is a simple matter of combinatorics. We generate all possible gestures and store the gesture as a viable candidate if it is not contained in the EGL.

7.1.3 VISUALIZING CANDIDATE GESTURES AND TRAINING GESTURE RECOGNIZERS

In order for the interface designer to select between the different candidate gestures, we must visualize them. Specifically, we need to convert the candidate gesture from a SAX string into a real valued time series. For each SAX symbol, we know that valid values are somewhere between the upper and lower breakpoint of the area assigned to the symbol. We choose a random point between these breakpoints for each symbol. We then use spline interpolation or re-sampling to fit a curve through the resulting values from each SAX symbol. We used an exponential moving average to smooth the resulting curve. The overall process is shown in Figure 13. Note that by repeating this process we can generate a synthetic set of time series that could have generated the SAX word. This synthetic data is used to visualize acceptable versions of the trackpad gesture to the interaction designer. We will also use this synthetic data to train a recognizer for the gesture if it is selected (see below).

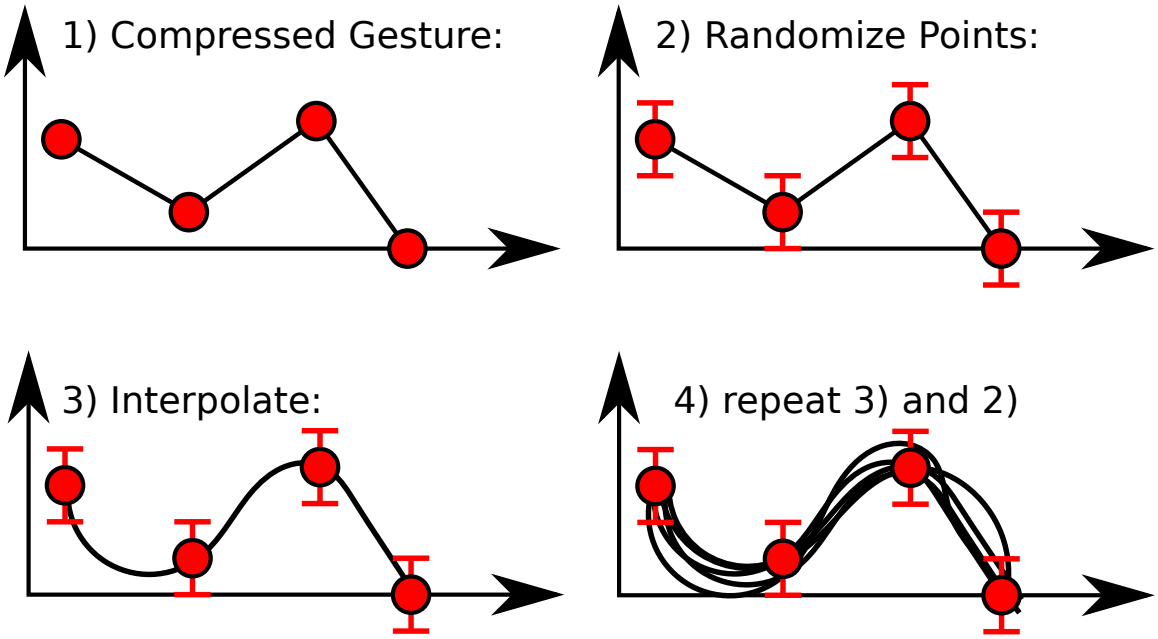


Figure 13: Converting a SAX word to example gestures.

Figure 14 shows MAGIC Summoning's user interface for gesture suggestion. In the center of the window we display a synthesized gesture. The color of the lines indicates the time course of the gesture as it is performed on the trackpad (from dark to light). Many synthetic examples of a given SAX word are drawn to give the interaction designer a sense of the possible shapes of the gesture. New suggestions are displayed periodically, effectively creating a movie of potential gestures. In our first implementation, gesture suggestions were selected randomly, keeping a list of previously viewed gestures so as to avoid repetition. If the interaction designer sees a desirable gesture, he stops the presentation with a key press.

If other gestures have already been selected by the user, the similarity of the currently displayed gesture to the already selected gestures is shown in a bar plot in a window at the bottom left. Based on these similarity scores, the user can retain the gesture or continue searching other suggestions. In this case, we decided to use the $1 Recognizer (Wobbrock et al., 2007) both for generating

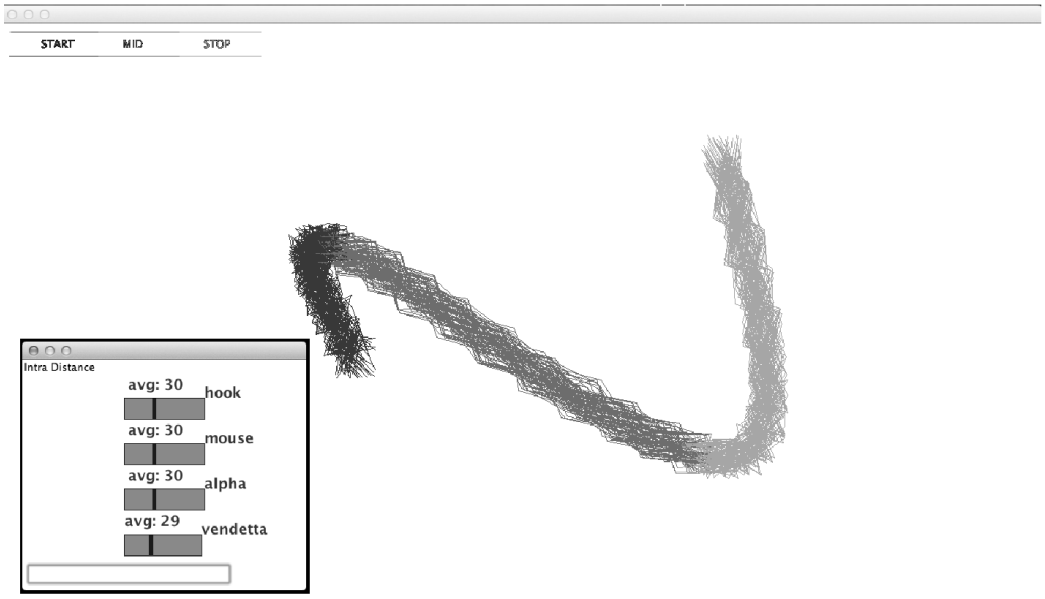


Figure 14: The MAGIC Summoning gesture suggestion interface.

similarity scores and for gesture recognition. To train the gesture recognizer, we simply used the synthetic examples generated during the visualization process.

7.1.4 $1 RECOGNIZER

Since the $1 Recognizer is widely used in HCI research (Belatar and Coldefy, 2010; Dang and André, 2010) but is not necessarily known to machine learning researchers, we give a quick overview here. The recognizer is optimized for single stroke gestures and can be considered instance-based learning. Each instance or template is re-sampled to be of equal length with all others and then rotated, scaled, and translated to a canonical form before being used. During recognition the query gesture is compared to all the stored templates using an angular distance metric. In continuous recognition we can apply a threshold on that distance, and the rest of the recognition process is similar to the dynamic time warping approach described earlier. The authors report recognition accuracies of 99%, which is comparable to DTW implementations on the same data sets. The method is simple, fast to compute, and understandable by pattern recognition novices. Thus, the algorithm is well-suited for experimentation by interface designers. With MAGIC Summoning, interaction designers do not need to collect any training data for the recognizer. The training data is produced synthetically from the EGL as described above. Note that we can use the $1 Recognizer as a distance measure for EGL search (albeit slowly compared to iSAX), which will be useful for comparison experiments below.

7.2 Testing Suggested Gestures And Recognizers In Practice

We collected an EGL consisting of ten participants using their personal Mac laptops for one week. Figure 15 visualizes the EGL. While indexing the EGL, we set the SAX word length to four. For

a two dimensional touchpad, the length doubles to eight. Setting the cardinality to four leads to a total number of 65536 (4^8) possible strings.

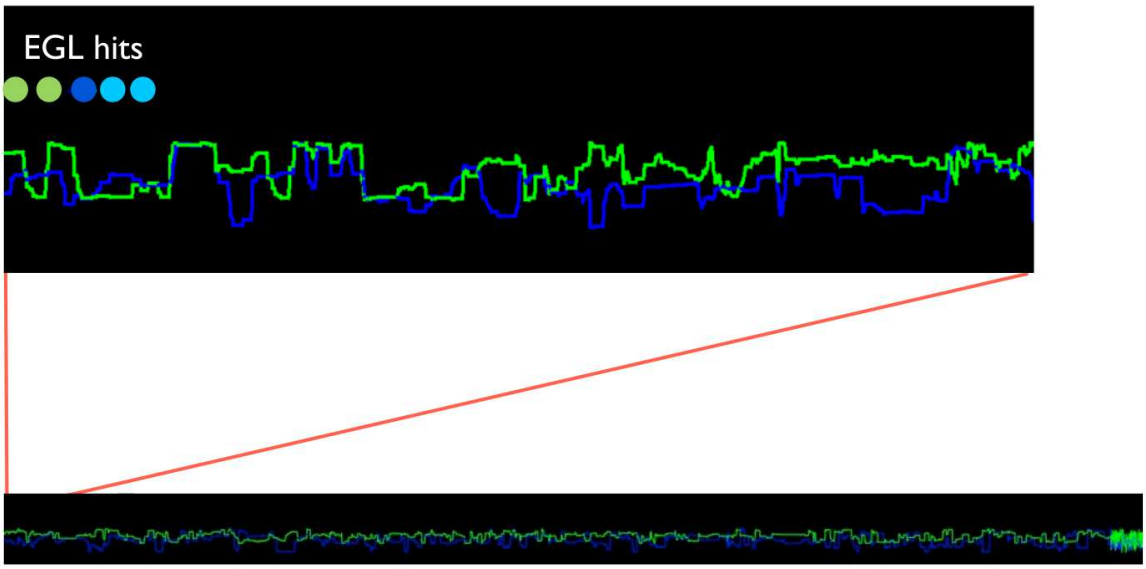


Figure 15: Bottom: The touch pad EGL. Top: An excerpt from the EGL showing five false positives during testing of a gesture, indicated as colored bubbles.

We observed 1222 unique strings in the collected EGL. The space is surprisingly sparse; there are 64314 strings not found in the EGL, suggesting that there are a large number of gestures that could be made with a low probability of false positives.

We performed an experiment to evaluate if the proposed suggestion and selection process described in the previous section can produce gestures that show a low false positive rate in everyday life. In addition, we were concerned as to whether synthetic data would be sufficient to train a high accuracy recognizer for this domain. We acted as an interaction designer and selected six gestures using the visualization tool above (see Figure 16). We preferred gestures that were simple and memorable. Figure 17 demonstrates 70 other gestures suggested by the system that were not used. We trained a $1 Recognizer for each of the six gestures selected using synthetic data generated by MAGIC.

We designed a six user study with users who did not contribute to the EGL. As in the false positive prediction experiments from the previous section, we asked users to practice with the recognition system so that they could perform the gestures with confidence. Users were able to improve their performance from $\approx 46\%$ to $\approx 90\%$ quickly. Afterward, the users worked on their computers for four hours while all touchpad movements were recorded. Every 10 minutes we sent a notification to the users asking them to perform one of the six gestures, resulting in four examples of each gesture for each participant. Thus, we collected 24 hours of data and 144 gesture examples.

The gesture recognizer was able to recognize 98% of the performed gestures. Even though synthetic data was use to train the recognizer, these findings are similar to those of Wobbrock et al. (2007), who reported a 99% accuracy in their experiments. The false positive rates of the gestures

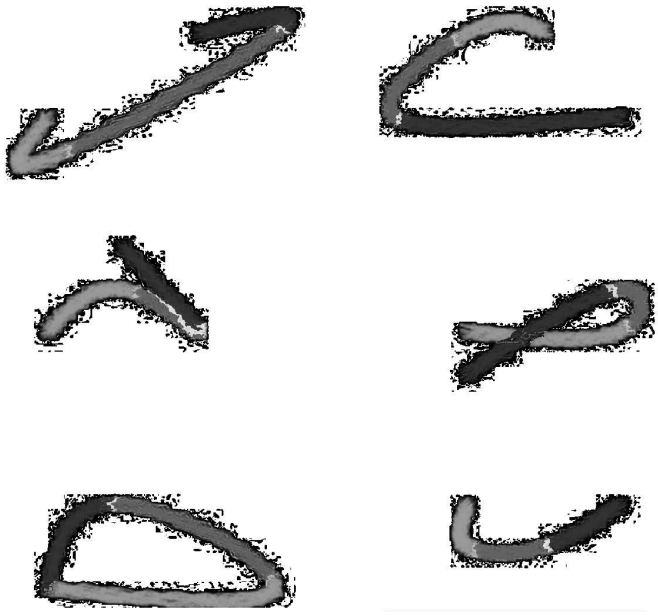


Figure 16: The six gestures used in the study. Gestures are drawn from dark to light.

are low except for one gesture (see Figure 18). Thus, the experiment supports the hypothesis that MAGIC Summoning can suggest gestures and aid the interaction designer in creating a gesture system that results in low false positives. However, several questions remain. Can we order the suggestions so as to present the "best" gestures first? Also, the experiment as described has no control condition. What would have been the result if we had tried suggesting random gestures from the $64,314$ available?

7.3 Ordering Gesture Suggestions

In this section we will explore possible ways of ordering gestures such that users can quickly find desirable gestures from the large number of possibilities. Our intuition is that users prefer simple gestures since they can be accessed quickly and are easy to memorize.

Our first approach is defining the complexity of a gesture as the entropy of its SAX word (Mitchell, 1997):

$$H(word) = -\sum_{i=0}^{card} p(symbol_i) * log(symbol_i).$$

However, if we want to prefer simpler gestures, we should check to determine if false positive rates in real usage are correlated with simplicity. Otherwise, proposing simpler gestures first could be counterproductive. Intuitively, one would think that simpler gestures would trigger more often in everyday life. To investigate this question we trained the $1 Recognizer with 100 randomly chosen gestures and searched the EGL with it. For each gesture we calculated the entropy and compared the false positive rate to the entropy and found no correlation ($r^2 \approx 0.04$). Thus, there seems to be little additional risk to suggesting lower entropy gestures first.

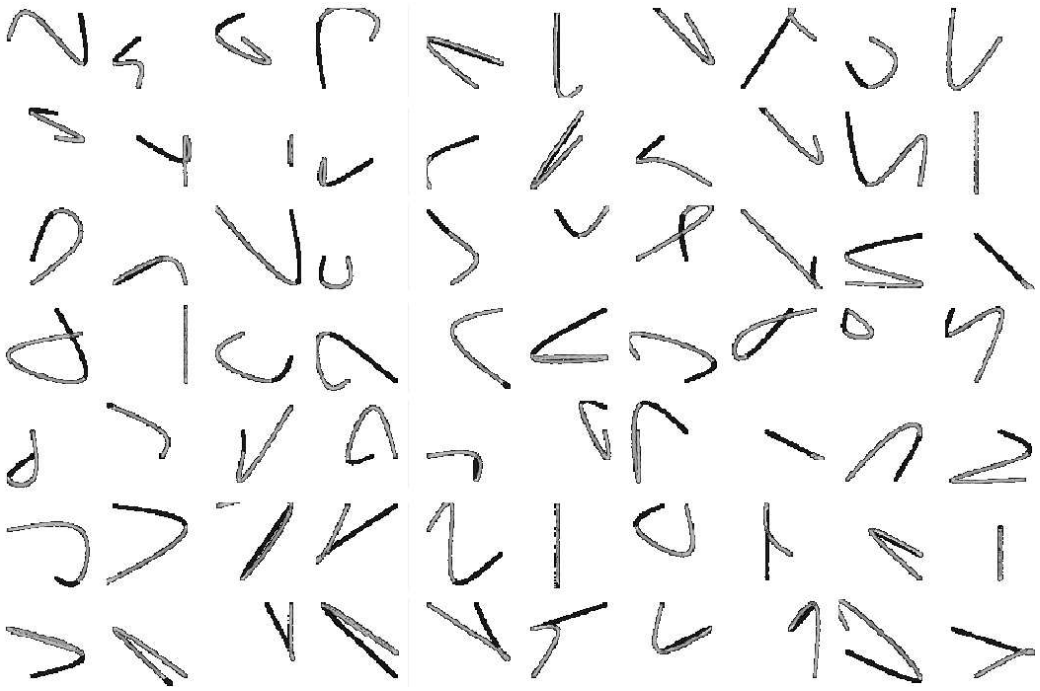


Figure 17: 70 generated gestures with potential low false positive rates. Gestures ordered from left-to-right and from top-to-bottom with increasing entropy.

The above heuristic seems logical for ordering suggestions. Low entropy gestures would seem to be simpler and easier to perform. To confirm this intuition we ran a small user study. We generated 100 gestures and sorted them using the above score. We examined the 20 best-ranked gestures and rejected ones that required significant overlap of the strokes (see Figure 19) as the static visualization of the strokes could confuse subjects. For each of the 10 remaining gestures we asked six users to perform the gesture in the air, on the table or on their touchpad and asked them to assign a score of performability between 1 and 10. All participants received the same gestures. Interestingly, we were not able to find a correlation between the entropy of a gesture's SAX word and the users' ratings ($r^2 = 0.09$).

Given the above result, we desire gestures not in the EGL but that are known to be performable. With a trackpad, all suggested gestures should be physically possible, but in future work with inertial sensors the suggestions could become impossible without constraining the system in some manner.

We decided to prefer gesture suggestions where the substrings of the SAX word representing the candidate gesture are represented in the EGL, but the gesture string itself was not present. We will assume one dimension for ease of illustration. If a gesture ACBD is not in the EGL, but the subcomponents AC, CB, and BD or ACB and CBD were well represented in the EGL, we might conclude that ACBD is possible for the user to perform. In other words, we will prefer gestures where the most n-grams from the EGL are included in the suggested gesture's string. Intuitively, though, such a heuristic causes concern that such gestures might have a higher chance of false triggering.

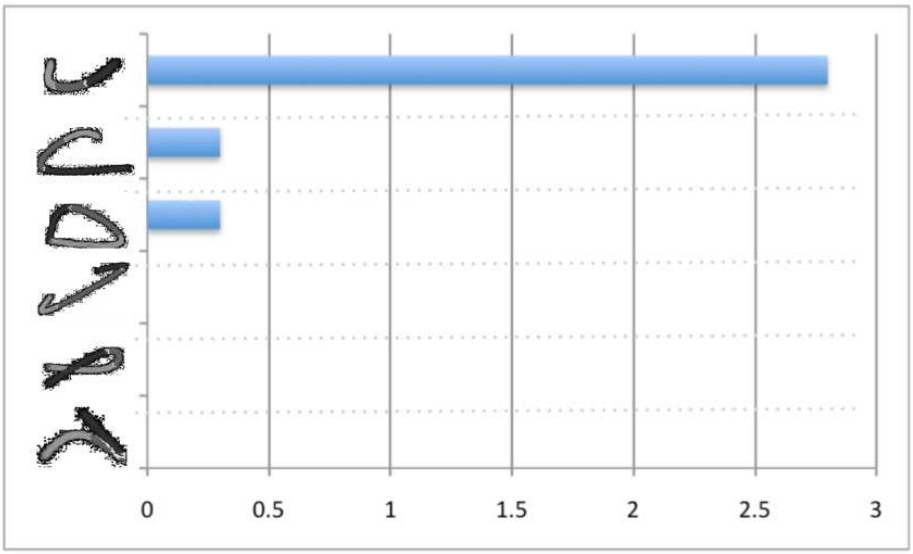


Figure 18: Results of the trackpad gesture user study in false positives per hour. All but one of the gestures suggested by MAGIC Summoning show a low false positive rate.

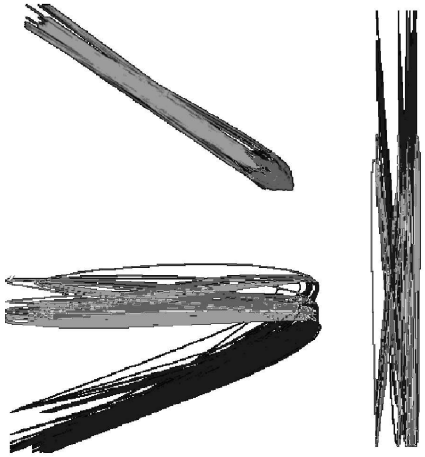


Figure 19: MAGIC Summoning gestures with significant overlap of the strokes were rejected to avoid user confusion.

To investigate this possibility, we extracted bi-grams and tri-grams from the EGL, created candidate gestures from them, and tried to find a correlation between the false positives in the EGL and the number of n-grams in the gesture's string. Note that this method of composition creates gestures with a variety of properties: ones common in the EGL, rare in the EGL, and not present in the EGL.

A correlation would indicate an increased risk with this method of ordering the suggestions, but we did not find one, giving a modicum of assurance in the method:

Bi-grams $r^2 = 0.000676$

Tri-grams $r^2 = 0.000256$.

Beside low false positives, another criteria for a good gesture system is that there should be a low chance of confusion between gestures. If the user is creating a control system with six gestures and has already selected five of them, we should prefer suggestions that are distinct from the five gestures already chosen. We measure the distinguishably of a gesture using the Hamming distance (Hamming, 1950) of the gesture's SAX word. Thus, when ordering gestures, we sort using a score defined as

$$score(word) = \frac{dist(word)}{(1 + entropy(word))}$$

where the distance of the word is the average Hamming distance to all other gestures in the gesture set. This metric provides a high distance to the other gestures and a low entropy. Note that we use $(1 + entropy(word))$ to avoid unreasonably high or infinite scores when the entropy value is near 0.

Given the results of the above experiments, we are now tuning MAGIC Summoning to generate gestures composed from parts of the EGL and to suggest gestures that are most dissimilar to each other. We intend to test this ordering system in our future work with suggesting gestures for use with inertial sensors.

7.4 How Selective Are MAGIC Summoning's Suggestions?

In the above user study, we selected six gestures by hand from MAGIC Summoning's suggestions and tested the $1 Recognizer that MAGIC output for both accuracy and false triggering. However, there were many possible gestures that the system could have output instead. In this last section we will investigate if suggesting gestures based on our method is better generated ones by chance.

As we have seen previously, using iSAX results in fewer hits being identified in an EGL than those found by typical gesture recognizers (HMM, NN-DTW, $1 Recognizer, etc.). The sole reason to use iSAX is that it quickly returns whether or not a candidate gesture is worthwhile to investigate further. However, we do not need to generate gesture suggestions in real time. In fact, as soon as an EGL is collected, the same "overnight" process that generates the EGL's iSAX tree representation for prediction could track the gestures not represented in the EGL. Once these gestures are known, the recognizer of choice could be trained with synthetic data of the gesture, and the recognizer could be run on the EGL for a more precise estimate of the expected hits. The number of false positives returned should allow a finer discrimination between candidate gestures. In the following experiment, we use this new procedure to generate suggested gestures and test ones with the lowest number of false positives on the test data collected from subjects not represented in the EGL.

In this experiment, we generated 2000 random gestures from SAX strings not in the EGL. For each of the gestures we synthesized 40 examples and trained a $1 recognizer with them. We used this recognizer to test search the EGL in the classic way, that is testing each interesting region using the trained recognizer. We used a typical threshold ($th = .85$) for the $1 score. All results above that threshold count as a hit with the EGL. Figure 20 orders the gestures by least to most number of hits per hour in the EGL. Clearly the $1 Recognizer identifies many potential false positives, yet most of the gestures still have low rates.

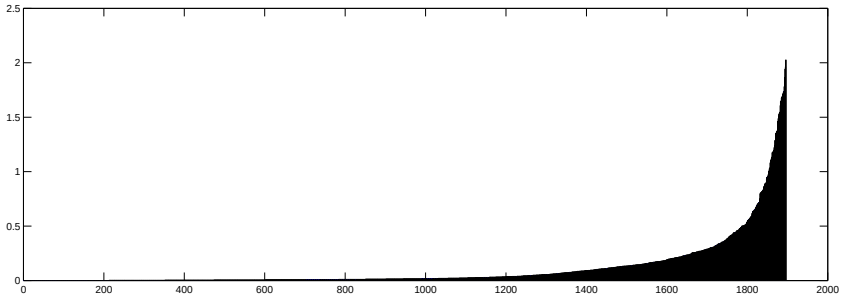


Figure 20: Number of false positives identified in the EGL using the $1 Recognizer for each of 2000 gestures synthesized from SAX strings not represented in the EGL. Gestures with more than 2 hits per hour are not graphed to preserve scale.

Figure 21, top, shows another view of this data. Note that over 35% of the 2000 gestures have 0 - 0.0001 false positives/hour. Compare this rate to that of Figure 21, bottom. This graph was generated using all the SAX strings represented in the EGL. Less than 12% of these gestures have such low false positive rates. Clearly, the SAX representation does have considerable predictive power on which suggested gestures are least likely to trigger falsely using the $1 Recognizer in the EGL. In fact, better than one in three of the gestures suggested by choosing SAX strings not in the EGL will be candidates for very low false positive rates with the synthetically trained $1 Recognizer.

The above observation suggests a relatively efficient method for creating gesture candidates for the interaction designer. First, randomly choose a unrepresented SAX string in the EGL. Train the desired recognizer using synthetic data. Run the recognizer on the EGL. If the rate of false positives per hour is less than 0.0001, keep the gesture. Otherwise, discard it. Generate as many gesture suggestions as is possible given time constraints. (Approximately 25 minutes is required to generate 100 gesture suggestions using a modern laptop, but such a process is highly parallelizable and can be run in batch before the interaction designer approaches the system.) Order the suggestions as described above and present them to the interaction designer for selection.

We conducted an experiment evaluating this algorithm. We split the collected EGL for touchpad gestures into two subsets. Each subset contains randomly chosen, distinct time series from the original EGL. The intersection between the subsets is empty. We used the first subset to generate 100 randomly chosen, distinct gestures candidates that show less then 0.0001 false positives per hour using the $1 Recognizer. We used these recognizers to then search the data in the second subset. On average we found the gestures to trigger 0.0022 times per hour, with a standard deviation of 0.003. These rates correspond to an average time between false triggerings of 455 hours, or approximately one month assuming usage 16 hours/day. Thus, this method of choosing gestures to suggest to an interaction designer seems desirable as well as practical.

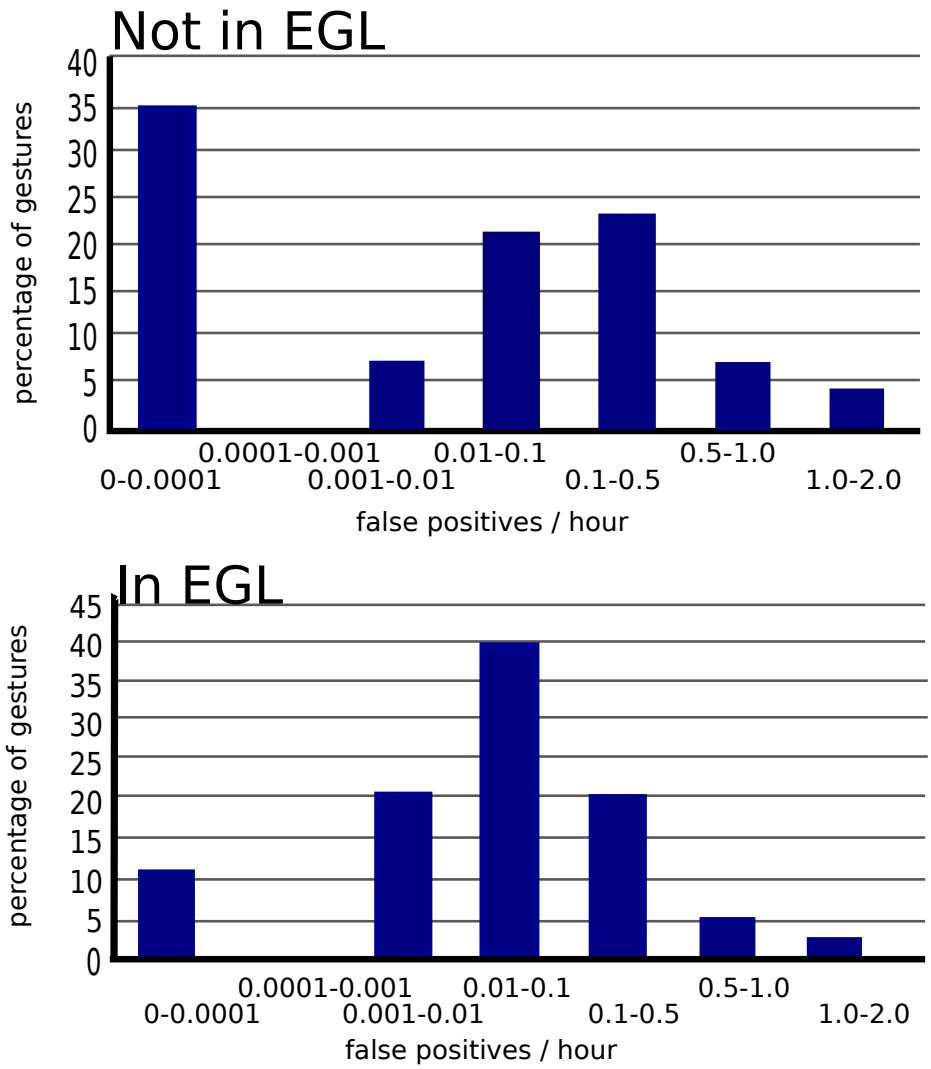


Figure 21: Histogram demonstrating the percentages of the number of false positives per hour for gestures with SAX representations not in the EGL (top) and all gestures with SAX representations in the EGL (bottom).

8. Future Work

To date, the task for most gesture recognition systems has been to optimize accuracy given a set of gestures to be recognized. In this paper, we have reversed the problem, seeking to discover which gestures might be most suitable for recognition.

However, improved suggestion ordering is an area for improvement. Performability might be improved by modeling how gestures are produced (Cao and Zhai, 2007) and prioritizing those gestures with least perceived effort. For domains where the coupling between sensor data and limb movement are not as apparent, such as accelerometer-based motion gestures, inverse kinematic models and 3D avatars seem appropriate both for prioritizing suggestions and for visualizing the

gesture for the interaction designer. For situations with many degrees of freedom, such as whole body movement as tracked by the Microsoft Kinect©, the space of potential gestures may be extremely large. Physical and behavioral constraints might be applied to reduce the search space for the interaction designer. While MAGIC and MAGIC Summoning have been applied to multiple domains, we have only applied the gesture suggestion functions to trackpads. We are eager to investigate MAGIC Summoning's usefulness and usability in other domains.

9. Conclusion

We have described two pattern recognition tasks that can be used to help interaction designers create gesture interfaces: testing a user-defined gesture (and its classifier) against a previously captured database of typical usage sensor data to determine its tendency to trigger falsely and suggesting gestures automatically to the designer. We have shown that iSAX can be used to provide near immediate feedback to the user as to whether a gesture is inappropriate. While this method is approximate and recovers only a fraction of the total false positives in the EGL, MAGIC Summoning's results correlate strongly with those of HMMs, DTW, and the $1 Recognizer and can thus be used to provide guidance during training. We showed that MAGIC Summoning and the EGL could be used to create a null class of close false matches that increase the performance of the chosen classifier.

To suggest gestures to the interaction designer that may have low chance of triggering falsely, we exploited the SAX representation used to index the EGL. MAGIC Summoning generates all the strings not in the EGL, converts the SAX strings back into a gesture visualization, and suggests appropriate gestures to the designer. MAGIC Summoning also outputs classifiers for the gesture, trained on synthetic data generated from the SAX string. Using the task of finding command gestures for Mac trackpads, we showed that the gestures generated by MAGIC Summoning have generally low false positive rates when deployed and that the classifiers output by the system were adequate to the task of spotting the gesture.

Even if iSAX search of an EGL is not a perfect predictor for the false positives of a gesture in every day usage, we find that the approximations are sufficient to speed interface design significantly. MAGIC's methods are not intended to replace user testing with the final device. However, we believe that the tool will decrease the number of iterations needed to build a fast and stable gesture recognition interface.

Acknowledgments

This material is based upon work supported, in part, by the National Science Foundation under Grant No. 0812281. We would also like to thank Google for their support of the most recent advances in this project. Thanks also to David Quigley for sharing his Android EGL data set and Daniel Ashbrook for his original MAGICal efforts and collaborations.

References

Dan Ashbrook. *Enabling Mobile Microinteractions*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2009.

Daniel Ashbrook and Thad Starner. MAGIC: a motion gesture design tool. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 2159–2168, New York, New York, 2010.

Mohammed Belatar and François Coldefy. Sketched menus and iconic gestures, techniques designed in the context of shareable interfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 143–146, New York, New York, 2010.

Xiang Cao and Shumin Zhai. Modeling human performance of pen stroke gestures. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 1495–1504, New York, New York, 2007.

Chi Tai Dang and Elisabeth André. Surface-poker: multimodality in tabletop games. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 251–252, New York, New York, 2010.

Roger Dannenberg and Dale Amon. A gesture based user interface prototyping system. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 127–132, New York, New York, 1989.

Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 33–40, New York, New York, 2004.

Jerry Fails and Dan Olsen. A design tool for camera-based interaction. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 449–456, New York, New York, 2003.

Ada Wai-Chee Fu, Eamonn Keogh, Leo Yung Lau, Chotirat Ann Ratanamahatana, and Raymond Chi-Wing Wong. Scaling and time warping in time series querying. *The International Journal on Very Large Data Bases*, 17(4):899–921, 2008.

François Guimbretière and Terry Winograd. Flowmenu: combining command, text, and data entry. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 213–216, 2000.

Richard Hamming. Error detecting and error correcting codes. *Bell Systems Technical Journal*, 29: 147–160, 1950.

Yuichi Hattori, Sozo Inoue, and Go Hirakawa. A large scale gathering system for activity data with mobile sensors. In *Proceedings of the IEEE International Symposium on Wearable Computers*, pages 97–100, Washington, District of Columbia, 2011.

Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Human-Computer Interaction*, 1(4):311–338, December 1985. ISSN 0737-0024.

Daniel Kohlsdorf, Thad Starner, and Daniel Ashbrook. MAGIC 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 1–6, Washington, District of Columbia, 2011.

Danile Kohlsdorf. Motion gesture: False positive prediction and prevention. Master's thesis, University of Bremen, Bremen, Germany, 2011.

Yang Li. Gesture search: a tool for fast mobile data access. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 87–96, New York, New York, 2010.

Jessica Lin, Li Wei, and Eamonn Keogh. Experiencing sax: A novel symbolic representation of time series. *Journal of Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

Chris Long. *Quill: A Gesture Design Tool for Pen-based User Interfaces*. PhD thesis, University of California, Berkeley, California, 2001.

Kent Lyons, Helene Brashear, Tracy Westeyn, Jung Soo Kim, and Thad Starner. GART: the gesture and activity recognition toolkit. In *Proceedings of the International Conference on Human-Computer Interaction: Intelligent Multimodal Interaction Environments*, pages 718–727, Berlin, Germany, 2007.

Dan Maynes-Aminzade, Terry Winograd, and Takeo Igarashi. Eyepatch: prototyping camera-based interaction through examples. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 33–42, New York, New York, 2007.

Tom M. Mitchell. *Machine Learning*. McGraw Hill, New York, New York, 1997.

Tom Ouyang and Yang Li. Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 2895–2904, 2012.

Antti Pirhonen, Stephen Brewster, and Christopher Holguin. Gestural and audio metaphors as a means of control for mobile devices. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 291–298, New York, New York, 2002.

Jin Shieh and Eamonn Keogh. iSAX: indexing and mining terabyte sized time series. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–631, New York, New York, 2008.

Jin-Wien Shieh. *Time Series Retrievel: Indexing and Mapping Large Datasets*. PhD thesis, University California, Riverside, California, 2010.

Thad Starner, Joshua Weaver, and Alex Pentland. Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371 – 1375, December 1998.

Tracy Westeyn, Helene Brashear, Amin Atrash, and Thad Starner. Georgia Tech Gesture Toolkit: supporting experiments in gesture recognition. In *Proceedings of the International Conference on Multimodal Interfaces*, pages 85–92, New York, New York, 2003.

Hendrik Witt. *Human-Computer Interfaces for Wearable Computers*. PhD thesis, University Bremen, Bremen, Germany, 2007.

Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 159–168, New York, New York, 2007.

Hee-Deok Yang, Stan Sclaroff, and Seong-Whan Lee. Sign language spotting with a threshold model based on conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1264–1277, 2009.

Journal of Machine Learning Research 14 (2013) 243-280 Submitted 1/11; Revised 1/12; Published 1/13

Sparse Single-Index Model

Pierre Alquier PIERRE.ALQUIER@UCD.IE
School of Mathematical Sciences
University College Dublin
James Joyce Library, Belfield
Dublin 4, Ireland

Gérard Biau* GERARD.BIAU@UPMC.FR
LSTA, LPMA and Institut universitaire de France
Université Pierre et Marie Curie – Paris VI
Boîte 158, Tour 15-25, 2ème étage
4 place Jussieu, 75252 Paris Cedex 05, France

Editor: John Shawe-Taylor

Abstract

Let $(\mathbf{X},Y)$ be a random pair taking values in $\mathbb{R}^p \times \mathbb{R}$. In the so-called single-index model, one has $Y = f^\star(\theta^{\star T}\mathbf{X}) + W$, where $f^\star$ is an unknown univariate measurable function, $\theta^\star$ is an unknown vector in $\mathbb{R}^d$, and W denotes a random noise satisfying $\mathbb{E}[W|\mathbf{X}] = 0$. The single-index model is known to offer a flexible way to model a variety of high-dimensional real-world phenomena. However, despite its relative simplicity, this dimension reduction scheme is faced with severe complications as soon as the underlying dimension becomes larger than the number of observations ("p larger than n" paradigm). To circumvent this difficulty, we consider the single-index model estimation problem from a sparsity perspective using a PAC-Bayesian approach. On the theoretical side, we offer a sharp oracle inequality, which is more powerful than the best known oracle inequalities for other common procedures of single-index recovery. The proposed method is implemented by means of the reversible jump Markov chain Monte Carlo technique and its performance is compared with that of standard procedures.

Keywords: single-index model, sparsity, regression estimation, PAC-Bayesian, oracle inequality, reversible jump Markov chain Monte Carlo method

1. Introduction

Let $\mathcal{D}_n = \{(\mathbf{X}_1,Y_1),\ldots,(\mathbf{X}_n,Y_n)\}$ be a collection of independent observations, distributed as a generic independent pair $(\mathbf{X},Y)$ taking values in $\mathbb{R}^p \times \mathbb{R}$ and satisfying $\mathbb{E}Y^2 < \infty$. Throughout, we let $\mathbf{P}$ be the distribution of $(\mathbf{X},Y)$, so that the sample $\mathcal{D}_n$ is distributed according to $\mathbf{P}^{\otimes n}$. In the regression function estimation problem, the goal is to use the data $\mathcal{D}_n$ in order to construct an estimate $r_n : \mathbb{R}^p \to \mathbb{R}$ of the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. In the classical parametric linear model, one assumes

$$Y = \theta^{\star T}\mathbf{X} + W,$$

∗. Also at DMA, Ecole Normale Supérieure, 45 rue d'Ulm, 75230 Paris Cedex 05, France.

where $\theta^\star = (\theta_1^\star, \ldots, \theta_p^\star)^T \in \mathbb{R}^p$ and $\mathbb{E}[W|\mathbf{X}] = 0$. Here

$$r(\mathbf{x}) = \theta^{\star T}\mathbf{x} = \sum_{j=1}^{p} \theta_j^\star x_j$$

is a linear function of the components of $\mathbf{x} = (x_1, \ldots, x_p)^T$. More generally, we may define

$$Y = f^\star(\theta^{\star T}\mathbf{X}) + W, \tag{1}$$

where $f^\star$ is an unknown univariate measurable function. This is the celebrated single-index model, which is recognized as a particularly useful variation of the linear formulation and can easily be interpreted: The model changes only in the direction $\theta^\star$, and the way it changes in this direction is described by the function $f^\star$. This model has applications to a variety of fields, such as discrete choice analysis in econometrics and dose-response models in biometrics, where high-dimensional regression models are often employed. There are too many references to be included here, but the monographs of McCullagh and Nelder (1983) and Horowitz (1998) together with the references Härdle et al. (1993), Ichimura (1993), Delecroix et al. (2006), Dalalyan et al. (2008) and Lopez (2009) will provide the reader with good introductions to the general subject area.

One of the main advantages of the single-index model is its supposed ability to deal with the problem of high dimension (Bellman, 1961). It is known that estimating the regression function is especially difficult whenever the dimension p of $\mathbf{X}$ becomes large. As a matter of fact, the optimal mean square convergence rate $n^{-2k/(2k+p)}$ for the estimation of a k-times differentiable regression function converges to zero dramatically slowly if the dimension p is large compared to k. This leads to an unsatisfactory accuracy of estimation for moderate sample sizes, and one possibility to circumvent this problem is to impose additional assumptions on the regression function. Thus, in particular, if $r(\mathbf{x}) = f^\star(\theta^{\star T}\mathbf{x})$ holds for every $\mathbf{x} \in \mathbb{R}^p$, then the underlying structural dimension of the model is 1 (instead of p) and the estimation of r can hopefully be performed easier. In this regard, it is shown in Gaïffas and Lecué (2007) that the optimal rate of convergence over the single-index model class is $n^{-2k/(2k+1)}$ (instead of $n^{-2k/(2k+p)}$), thereby answering a conjecture of Stone (1982).

Nevertheless, practical estimation of the link function $f^\star$ and the index $\theta^\star$ still requires a degree of statistical smoothing. Perhaps the most common approach to reach this goal is to use a nonparametric smoother (for instance, a kernel or a local polynomial method) to construct an approximation $\hat{f}_n$ of $f^\star$, then substitute $\hat{f}_n$ into an empirical version $R_n(\theta)$ of the mean square error $R(\theta) = \mathbb{E}[Y - f(\theta^T\mathbf{X})]^2$, and finally choose $\hat{\theta}_n$ to minimize $R_n(\theta)$ (see, e.g., Härdle et al., 1993; Delecroix et al., 2006, where the procedure is discussed in detail). The rationale behind this type of two-stage approach, which is asymptotic in spirit, is that it produces a $\sqrt{n}$-consistent estimate of θ, thereby devolving the difficulty to the simpler problem of computing a good estimate for the one-dimensional function $f^\star$. However, the relative simplicity of this strategy is accompanied by severe difficulties (overfitting) when the dimension p becomes larger than the number of observations n. Estimation in this setting (called "p larger than n" paradigm) is generally acknowledged as an important challenge in contemporary statistics, see, for example, the recent monograph of Bühlmann and van de Geer (2011). In fact, this drawback considerably reduces the ability of the single-index model to behave as an effective dimension reduction technique.

On the other hand, 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 allow for a good approximation of

the original image. Such signals have few nonzero 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).

In the present document, we consider the single-index model (1) from a sparsity perspective, that is, we assume that $\theta^\star$ has only a few coordinates different from 0. In the dimension reduction scenario we have in mind, the ambient dimension p 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 $\theta^\star$ are nonzero. This assumption is helpful at least for two reasons: If p is large and the number of nonzero coordinates is small enough, then the model is easier to interpret and its efficient estimation becomes possible. Our setting is close in spirit of the approach of Cohen et al. (2012), who study approximation from queries of functions of the form $f(\theta^T \mathbf{x})$, where θ is approximately sparse (in the sense that it belongs to a weak-ℓ_p space). However, these authors do not provide any statistical study of their model. Our modus operandi will rather rely on the so-called PAC-Bayesian approach, originally developed in the classification context by Shawe-Taylor and Williamson (1997), McAllester (1998) and Catoni (2004, 2007). This strategy was further investigated for regression by Audibert (2004) and Alquier (2008) and, more recently, worked out in the sparsity framework by Dalalyan and Tsybakov (2008, 2012) and Alquier and Lounici (2011). The main message of Dalalyan and Tsybakov (2008, 2012) and Alquier and Lounici (2011) is that aggregation with a properly chosen prior is able to deal nicely with the sparsity issue. Contrary to procedures such as the Lasso, the Dantzig selector and other penalized least square methods, which achieve fast rates under rather restrictive assumptions on the Gram matrix associated to the predictors, PAC-Bayesian aggregation requires only minimal assumptions on the model. Besides, it is computationally feasible even for a large p and exhibits good statistical performance.

The paper is organized as follows. In Section 2, we first set out some notation and introduce the single-index estimation procedure. Then we state our main result (Theorem 2), which offers a sparsity oracle inequality more powerful than the best known oracle inequalities for other common procedures of single-index recovery. Section 3 is devoted to the practical implementation of the estimate via a reversible jump Markov chain Monte Carlo (MCMC) algorithm, and to numerical experiments on both simulated and real-life data sets. In order to preserve clarity, proofs have been postponed to Section 4 and the description of the MCMC method in its full length is given in the Appendix Section 5.

Note finally that our techniques extend to the case of multiple-index models, of the form

$$Y = f^\star(\theta_1^{\star T}\mathbf{X}, \ldots, \theta_m^{\star T}\mathbf{X}) + W,$$

where the underlying structural dimension m is supposed to be larger than 1 but substantially smaller than p. However, to keep things simple, we let $m = 1$ and leave the reader the opportunity to adapt the results to the more general situation $m \geq 1$.

2. Sparse Single-index Estimation

We start this section with some notation and basic requirements.

2.1 Notation

Throughout the document, we suppose that the recorded data $\mathcal{D}_n$ is generated according to the single-index model (1). More precisely, for each $i = 1, \ldots, n$,

$$Y_i = f^\star(\theta^{\star T}\mathbf{X}_i) + W_i,$$

where $f^\star$ is a univariate measurable function, $\theta^\star$ is a p-variate vector, and $W_1, \ldots, W_n$ are independent copies of W. We emphasize that it is implicitly assumed that the observations are drawn according to the true model under study.

Recall that, in model (1), $\mathbb{E}[W|\mathbf{X}] = 0$ and, consequently, that $\mathbb{E}W = 0$. However, the distribution of W (in particular, the variance) may depend on $\mathbf{X}$. We shall not precisely specify this dependence, and will rather require the following condition on the distribution of W.

Assumption N. There exist two positive constants σ and L such that, for all integers $k \geq 2$,

$$\mathbb{E}\left[|W|^k \,|\, \mathbf{X}\right] \leq \frac{k!}{2}\sigma^2 L^{k-2}.$$

Observe that Assumption **N** holds in particular if $W = \Phi(\mathbf{X})\varepsilon$, where ε is a standard Gaussian random variable independent of $\mathbf{X}$ and $\Phi(\mathbf{X})$ is almost surely bounded.

Let $\|\theta\|_1$ denote the ℓ_1-norm of the vector $\theta = (\theta_1, \ldots, \theta_p)^T$, that is, $\|\theta\|_1 = \sum_{j=1}^p |\theta_j|$. Without loss of generality, it will be assumed throughout the document that the index $\theta^\star$ belongs to $\mathcal{S}^p_{1,+}$, where $\mathcal{S}^p_{1,+}$ is the set of all $\theta \in \mathbb{R}^p$ such that $\|\theta\|_1 = 1$ and the first nonzero coordinate of θ is positive.

Denoting by $\|\mathbf{X}\|_\infty$ the supremum norm of $\mathbf{X}$, we will also require that the random variable $\|\mathbf{X}\|_\infty$ is almost surely bounded by a constant which, without loss of generality, can be taken equal to 1. Moreover, it will also be assumed that the link function $f^\star$ is bounded by some known positive constant C. Thus, letting $\|f^\star\|_\infty$ be the functional supremum norm of $f^\star$ over $[-1,1]$, we set:

Assumption B. The condition $\|\mathbf{X}\|_\infty \leq 1$ holds almost surely and there exists a positive constant C larger than 1 such that $\|f^\star\|_\infty \leq C$.

Remark 1 *To keep a sufficient degree of clarity, no attempt was made to optimize the constants. In particular, the requirement $C \geq 1$ is purely technical. It is always satisfied by taking $C = \max(\|f^\star\|_\infty, 1)$.*

In order to approximate the link function $f^\star$, we shall use the vector space $\mathcal{F}$ spanned by a given countable dictionary of measurable functions $\{\varphi_j\}_{j=1}^\infty$. Put differently, the approximation space $\mathcal{F}$ is the set of (finite) linear combinations of functions of the dictionary. Each φ_j of the collection is assumed to be defined on $[-1,1]$ and to take values in $[-1,1]$. To avoid getting into too much technicalities, we will also assume that each φ_j is differentiable and such that, for some positive constant ℓ, $\|\varphi_j'\|_\infty \leq \ell \times j$. This assumption is satisfied by the (non-normalized) trigonometric system

$$\varphi_1(t) = 1, \varphi_{2j}(t) = \cos(\pi j t), \varphi_{2j+1}(t) = \sin(\pi j t), \quad j = 1, 2, \ldots$$

Finally, for any measurable $f : \mathbb{R}^p \to \mathbb{R}$ and $\theta \in \mathcal{S}^p_{1,+}$, we let

$$R(\theta, f) = \mathbb{E}\left[\left(Y - f(\theta^T \mathbf{X})\right)^2\right]$$

and denote by

$$R_n(\theta, f) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - f(\theta^T \mathbf{X}_i)\right)^2$$

the empirical counterpart of $R(\theta, f)$ based on the sample $\mathcal{D}_n$.

2.2 Estimation Procedure

We are now in a position to describe our estimation procedure. The method which is presented here is inspired by the approach developed by Catoni (2004, 2007). It strongly relies on the choice of a probability measure π on $\mathcal{S}^p_{1,+} \times \mathcal{F}$, called the prior, which in our framework should enforce the sparsity properties of the target regression function. With this objective in mind, we first let

$$d\pi(\theta, f) = d\mu(\theta) d\nu(f),$$

that is, we assume that the distribution over the indexes is independent of the distribution over the link functions. With respect to the parameter θ, we put

$$d\mu(\theta) = \frac{\displaystyle\sum_{i=1}^{p} 10^{-i} \sum_{I \subset \{1,\dots,p\}, |I|=i} \binom{p}{i}^{-1} d\mu_I(\theta)}{1 - (\frac{1}{10})^p}, \tag{2}$$

where $|I|$ denotes the cardinality of I and $d\mu_I(\theta)$ is the uniform probability measure on the set

$$\mathcal{S}^p_{1,+}(I) = \{\theta = (\theta_1, \dots, \theta_p) \in \mathcal{S}^p_{1,+} : \theta_j = 0 \text{ if and only if } j \notin I\}.$$

We see that $\mathcal{S}^p_{1,+}(I)$ may be interpreted as the set of "active" coordinates in the single-index regression of Y on $\mathbf{X}$, and note that the prior on $\mathcal{S}^p_{1,+}$ is a convex combination of uniform probability measures on the subsets $\mathcal{S}^p_{1,+}(I)$. The weights of this combination depend only on the size of the active coordinate subset I. As such, the value $|I|$ characterizes the sparsity of the model: The smaller $|I|$, the smaller the number of variables involved in the model. The factor 10^{-i} penalizes models of high dimension, in accordance with the sparsity idea.

The choice of the prior ν on $\mathcal{F}$ is more involved. To begin with, we define, for any positive integer $M \leq n$ and all $\Lambda > 0$,

$$\mathcal{B}_M(\Lambda) = \left\{(\beta_1, \dots, \beta_M) \in \mathbb{R}^M : \sum_{j=1}^{M} j|\beta_j| \leq \Lambda \text{ and } \beta_M \neq 0\right\}.$$

Next, we let $\mathcal{F}_M(\Lambda) \subset \mathcal{F}$ be the image of $\mathcal{B}_M(\Lambda)$ by the map

$$\begin{array}{rccc} \Phi_M & : \mathbb{R}^M & \to & \mathcal{F} \\ & (\beta_1, \dots, \beta_M) & \mapsto & \sum_{j=1}^{M} \beta_j \varphi_j. \end{array}$$

It is worth pointing out that, roughly, Sobolev spaces are well approximated by $\mathcal{F}_M(\Lambda)$ as M grows (more on this in Section 2.3). Finally, we define $\nu_M(\mathrm{d}f)$ on the set $\mathcal{F}_M(C+1)$ as the image of the uniform measure on $\mathcal{B}_M(C+1)$ induced by the map Φ_M, and take

$$\mathrm{d}\nu(f) = \frac{\sum_{M=1}^{n} 10^{-M} \mathrm{d}\nu_M(f)}{1-(\frac{1}{10})^n}. \tag{3}$$

Some comments are in order here. First, we note that the prior π is defined on $\mathcal{S}_{1,+}^p \times \mathcal{F}_n(C+1)$ endowed with its canonical Borel σ-field. The choice of $C+1$ instead of C in the definition of the prior support is essentially technical. This bound ensures that when the target $f^\star$ belongs to $\mathcal{F}_n(C)$, then a small ball around it is contained in $\mathcal{F}_n(C+1)$. It could be safely replaced by $C+u_n$, where $\{u_n\}_{n=1}^\infty$ is any positive sequence vanishing sufficiently slowly as $n \to \infty$. Next, the integer M should be interpreted as a measure of the "dimension" of the function f—the larger M, the more complex the function—and the prior ν adapts again to the sparsity idea by penalizing large-dimensional functions f. The coefficients 10^{-i} and 10^{-M} which appear in (2) and (3) show that more complex models have a geometrically decreasing influence. Note however that the value 10, which has been chosen because of its good practical results, is somehow arbitrary. It could be, in all generality, replaced by a more general coefficient α at the price of a more technical analysis (and with no consequences on the rates of convergence). Finally, we observe that, for each $f = \sum_{j=1}^M \beta_j \varphi_j \in \mathcal{F}_M(C+1)$,

$$\|f\|_\infty \leq \sum_{j=1}^{M} |\beta_j| \leq C+1.$$

Now, let λ be a positive real number, called the inverse temperature parameter hereafter. The estimates $\hat{\theta}_\lambda$ and $\hat{f}_\lambda$ of $\theta^\star$ and $f^\star$, respectively, are simply obtained by randomly drawing

$$(\hat{\theta}_\lambda, \hat{f}_\lambda) \sim \hat{\rho}_\lambda,$$

where $\hat{\rho}_\lambda$ is the so-called Gibbs posterior distribution over $\mathcal{S}_{1,+}^p \times \mathcal{F}_n(C+1)$, defined by the probability density

$$\frac{\mathrm{d}\hat{\rho}_\lambda}{\mathrm{d}\pi}(\theta, f) = \frac{\exp\left[-\lambda R_n(\theta, f)\right]}{\int \exp\left[-\lambda R_n(\theta, f)\right] \mathrm{d}\pi(\theta, f)}.$$

[The notation $\mathrm{d}\hat{\rho}_\lambda/\mathrm{d}\pi$ means the density of $\hat{\rho}_\lambda$ with respect to π.] The estimate $(\hat{\theta}_\lambda, \hat{f}_\lambda)$ has a simple interpretation. Firstly, the level of significance of each pair (θ, f) is assessed via its least square error performance on the data $\mathcal{D}_n$. Secondly, a Gibbs distribution with respect to the prior π enforcing those pairs (θ, f) with the most empirical significance is assigned on the space $\mathcal{S}_{1,+}^p \times \mathcal{F}_n(C+1)$. Finally, the resulting estimate is just a random realization (conditional to the data) of this Gibbs posterior distribution.

2.3 Sparsity Oracle Inequality

For any $I \subset \{1, \ldots, p\}$ and any positive integer $M \leq n$, we set

$$\left(\theta_{I,M}^\star, f_{I,M}^\star\right) \in \arg \min_{(\theta,f) \in \mathcal{S}_{1,+}^p(I) \times \mathcal{F}_M(C)} R(\theta, f).$$

At this stage, it is very important to note that, for each M, the infimum $f^\star_{I,M}$ is defined on $\mathcal{F}_M(C)$, whereas the prior charges a slightly bigger set, namely $\mathcal{F}_M(C+1)$.

The main result of the paper is the following theorem. Here and everywhere, the wording "with probability $1-\delta$" means the probability evaluated with respect to the distribution $\mathbf{P}^{\otimes n}$ of the data $\mathcal{D}_n$ *and* the conditional probability measure $\hat{\rho}_\lambda$. Recall that ℓ is a positive constant such that $\|\varphi_j'\|_\infty \leq \ell \times j$.

Theorem 2 *Assume that Assumption* **N** *and Assumption* **B** *hold. Set*

$$w = 8(2C+1)\max[L, 2C+1]$$

and take

$$\lambda = \frac{n}{w + 2\left[(2C+1)^2 + 4\sigma^2\right]}. \tag{4}$$

Then, for all $\delta \in]0,1[$*, with probability at least* $1-\delta$ *we have*

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi \inf_{\substack{I \subset \{1,\dots,p\} \\ 1 \leq M \leq n}} \left\{ R(\theta^\star_{I,M}, f^\star_{I,M}) - R(\theta^\star, f^\star) + \frac{M\log(Cn) + |I|\log(pn) + \log\left(\frac{2}{\delta}\right)}{n} \right\},$$

where Ξ *is a positive constant, depending on* L*,* C*,* σ *and* ℓ *only.*

Remark 3 *Interestingly enough, analysis of the estimate* $(\hat{\theta}_\lambda, \hat{f}_\lambda)$ *is still possible when Assumption* **N** *is not satisfied. Indeed, even if Bernstein's inequality (see Lemma 5) is not valid, a recent paper by Seldin et al. (2011) provides us with a nice alternative inequality assuming less restrictive assumptions. However, we would then suffer a loss in the upper bound of Theorem 2. It is also interesting to note that recent results by Audibert and Catoni (2011) allow the study of PAC-Bayesian estimates without Assumption* **N***. However, the results of these authors are valid for linear models only, and it is therefore not clear to what extent their technique can be transposed to our setting.*

Theorem 2 can be given a simple interpretation. Indeed, we see that if there is a "small" I and a "small" M such that $R(\theta^\star_{I,M}, f^\star_{I,M})$ is close to $R(\theta^\star, f^\star)$, then $R(\hat{\theta}_\lambda, \hat{f}_\lambda)$ is also close to $R(\theta^\star, f^\star)$ up to terms of order $1/n$. However, if no such I or M exists, then one of the terms $M\log(Cn)/n$ and $|I|\log(pn)/n$ starts to dominate, thereby deteriorating the general quality of the bound. A good approximation with a "small" I is typically possible when $\theta^\star$ is sparse or, at least, when it can be approximated by a sparse parameter. On the other hand, a good approximation with a "small" M is possible if $f^\star$ has a sufficient degree of regularity.

To illustrate the latter remark, assume for instance that $\{\varphi_j\}_{j=1}^\infty$ is the (non-normalized) trigonometric system and suppose that the target $f^\star$ belongs to the Sobolev ellipsoid, defined by

$$\mathcal{W}\left(k, \frac{6C^2}{\pi^2}\right) = \left\{ f \in L_2([-1,1]) : f = \sum_{j=1}^\infty \beta_j \varphi_j \text{ and } \sum_{j=1}^\infty j^{2k}\beta_j^2 \leq \frac{6C^2}{\pi^2} \right\}$$

for some unknown regularity parameter $k \geq 2$ (see, e.g., Tsybakov, 2009). Observe that, in this context, the approximation sets $\mathcal{F}_M(C+1)$ take the form

$$\mathcal{F}_M(C+1) = \left\{ f \in L_2([-1,1]) : f = \sum_{j=1}^M \beta_j \varphi_j, \sum_{j=1}^M j|\beta_j| \leq C+1 \text{ and } \beta_M \neq 0 \right\}.$$

It is important to note that the regularity parameter k is assumed to be unknown, and this casts our results in the so-called adaptive setting. The following additional assumption will be needed:

Assumption D. The random variable $\theta^{\star T}\mathbf{X}$ has a probability density on $[-1,1]$, bounded from above by a positive constant B.

Last, we let $I^\star$ be the set I such that $\theta^\star \in \mathcal{S}^p_{1,+}(I)$ and set $\|\theta^\star\|_0 = |I^\star|$.

Corollary 4 *Assume that Assumption* **N**, *Assumption* **B** *and Assumption* **D** *hold. Suppose also that $f^\star$ belongs to the Sobolev ellipsoid $\mathcal{W}(k, 6C^2/\pi^2)$, where the real number $k \geq 2$ is an (unknown) regularity parameter. Set $w = 8(2C+1)\max[L, 2C+1]$ and take λ as in (4). Then, for all $\delta \in]0,1[$, with probability at least $1-\delta$ we have*

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi' \left\{ \left(\frac{\log(Cn)}{n} \right)^{\frac{2k}{2k+1}} + \frac{\|\theta^\star\|_0 \log(pn)}{n} + \frac{\log\left(\frac{2}{\delta}\right)}{n} \right\}, \tag{5}$$

where Ξ' is a positive constant, depending on L, C, σ, ℓ and B only.

As far as we are aware, all existing methods achieving rates of convergence similar to the ones provided by Corollary 4 are valid in an asymptotic setting only (p fixed and $n \to \infty$). The strength of Corollary 4 is to provide a finite sample bound and to show that our estimate still behaves well in a nonasymptotic situation if the intrinsic dimension (i.e., the sparsity) is small with respect to n. To understand this remark, just assume that p is a function of n such that $p \to \infty$ as $n \to \infty$. Whereas a classical asymptotic approach cannot say anything useful about this situation, our bounds still provide some information, provided the model is sparse enough (i.e., $\|\theta^\star\|_0$ is sufficiently small with respect to n).

We see that, asymptotically (p fixed and $n \to \infty$), the leading term on the right-hand side of inequality (5) is $(\log(n)/n)^{\frac{2k}{2k+1}}$. This is the minimax rate of convergence over a Sobolev class, up to a $\log(n)$ factor. However, when n is "small" and $\theta^\star$ is not sparse (i.e., $\|\theta^\star\|_0$ is not "small"), the term $\|\theta^\star\|_0 \log(pn)/n$ starts to emerge and cannot be neglected. Put differently, in large dimension, the estimation of $\theta^\star$ itself is a problem—this phenomenon is not taken into account by asymptotic studies.

It is worth mentioning that the approach developed in the present article does not offer any guarantee on the point of view of variable (feature) selection. To reach this objective, an interesting route to follow is the sufficient dimension reduction (SDR) method proposed by Chen et al. (2010), which can be applied to the single-index model to estimate consistently the parameter $\theta^\star$ and perform variable selection in a sparsity framework. Note however that such results require strong assumptions on the distribution of the data.

Finally, it should be stressed that the choice of λ in Theorem 2 and Corollary 4 is not the best possible and may eventually be improved, at the price of a more technical analysis however.

3. Implementation and Numerical Results

A series of experiments was conducted, both on simulated and real-life data sets, in order to assess the practical capabilities of the proposed method and compare its performance with that of standard procedures. Prior to analysis, we first need to discuss its concrete implementation, which has been carried out via a Markov Chain Monte Carlo (MCMC) method.

3.1 Implementation via Reversible Jump MCMC

The use of MCMC methods has become a popular way to compute Bayesian estimates. For an introduction to the domain, one should refer to the comprehensive monograph of Marin and Robert (2007) and the references therein. Importantly, in this computational framework, an adaptation of the well-known Hastings-Metropolis algorithm to the case where the posterior distribution gives mass to several models of different dimensions was proposed by Green (1995) under the name Reversible Jump MCMC (RJMCMC) method. In the PAC-Bayesian setting, MCMC procedures were first considered by Catoni (2004), whereas Dalalyan and Tsybakov (2008, 2012) and Alquier and Lounici (2011) explore their practical implementation in the sparse context using Langevin Monte Carlo and RJMCMC, respectively. Regarding the single-index model, MCMC algorithms were used to compute Bayesian estimates by Antoniadis et al. (2004) and, more recently, by Wang (2009), who develop a fully Bayesian method to analyze the single-index model. Our implementation technique is close in spirit to the one of Wang (2009).

As a starting point for the approximate computation of our estimate, we used the RJMCMC method of Green (1995), which is in fact an adaptation of the Hastings-Metropolis algorithm to the case where the objective posterior probability distribution (here, $\hat{\rho}_\lambda$) assigns mass to several different models. The idea is to start from an initial given pair $(\theta^{(0)}, f^{(0)}) \in \mathcal{S}^p_{1,+} \times \mathcal{F}_n(C+1)$ and then, at each step, to iteratively compute $(\theta^{(t+1)}, f^{(t+1)})$ from $(\theta^{(t)}, f^{(t)})$ via the following chain of rules:

- Sample a random pair $(\tau^{(t)}, h^{(t)})$ according to some proposal conditional density $k_t(\,.\,|(\theta^{(t)}, f^{(t)}))$ with respect to the prior π;
- Take
$$(\theta^{(t+1)}, f^{(t+1)}) = \begin{cases} (\tau^{(t)}, h^{(t)}) & \text{with probability } \alpha_t \\ (\theta^{(t)}, f^{(t)}) & \text{with probability } 1-\alpha_t, \end{cases}$$
where
$$\alpha_t = \min\left(1, \frac{\frac{\mathrm{d}\hat{\rho}_\lambda}{\mathrm{d}\pi}(\tau^{(t)}, h^{(t)}) \times k_t\left((\theta^{(t)}, f^{(t)})|(\tau^{(t)}, h^{(t)})\right)}{\frac{\mathrm{d}\hat{\rho}_\lambda}{\mathrm{d}\pi}(\theta^{(t)}, f^{(t)}) \times k_t\left((\tau^{(t)}, h^{(t)})|(\theta^{(t)}, f^{(t)})\right)}\right).$$

This protocol ensures that the sequence $\{(\theta^{(t)}, f^{(t)})\}_{t=0}^{\infty}$ is a Markov chain with invariant probability distribution $\hat{\rho}_\lambda$ (see, e.g., Marin and Robert, 2007). A usual choice is to take $k_t \equiv k$, so that the Markov chain is homogeneous. However, in our context, it is more convenient to let $k_t = k_1$ if t is odd and $k_t = k_2$ if t is even. Roughly, the effect of k_1 is to modify the index $\theta^{(t)}$ while k_2 will essentially act on the link function $f^{(t)}$. While the ideas underlying the proposal densities k_1 and k_2 are quite simple, a precise description in its full length turns out to be more technical. Thus, in order to preserve the readability of the paper, the explicit construction of k_1 and k_2 has been postponed to the Appendix Section 5.

From a theoretical point of view, it is clear that the implementation of our method requires knowledge of the constant C (the upper bound on $\|f^\star\|_\infty$). A too small C will result in a smaller model, which is unable to perform a good approximation. On the other hand, a larger C induces a poor bound in Theorem 2.1. In practice, however, the influence of C turns out to be secondary compared to the impact of the parameter λ. Indeed, it was found empirically that a very large choice of C (e.g., $C = 10^{100}$) does not deteriorate the overall quality of the results, as soon as λ is appropriately chosen. This is the approach that was followed in the experimental testing process.

Besides, the time for the Markov chains to converge depends strongly on the ambient dimension p and the starting point of the simulations. When the dimension is small (typically, $p \leq 10$), the chains converge fast and any value may be chosen as a starting point. In this case, we let the MCMC run 1000 steps and obtained satisfying results. On the other hand, when the dimension is larger (typically, $p > 10$), the convergence is very slow, in the sense that $R_n(\theta^{(t)}, f^{(t)})$ takes a very long time to stabilize. However, using as a starting point for the chains the preliminary estimate $\hat{\theta}_{\text{HHI}}$ (see below) significantly reduces the number of steps needed to reach convergence—we let the chains run 5000 steps in this context. Nevertheless, as a general rule, we encourage the users to inspect the convergence of the chains by checking if $R_n(\theta^{(t)}, f^{(t)})$ is stabilized, and to run several chains starting from different points to avoid their attraction into local minima.

3.2 Simulation Study

In this subsection, we illustrate the finite sample performance of the presented estimation method on three synthetic data sets and compare its predictive capabilities with those of three standard statistical procedures. In all our experiments, we took as dictionary the (non-normalized) trigonometric system $\{\varphi_j\}_{j=1}^{\infty}$ and denote accordingly the resulting regression function estimate defined in Section 2 by $\hat{F}_{\text{Fourier}}$. In accordance with the order of magnitude indicated by the theoretical results, we set $\lambda = 4n$. This choice can undoubtedly be improved a bit but, as the numerical results show, it seems sufficient for our procedure to be fairly competitive.

The tested competing methods are the Lasso (Tibshirani, 1996), the standard regression kernel estimate (Nadaraya, 1964, 1970; Watson, 1964; Tsybakov, 2009), and the estimation strategy discussed in Härdle et al. (1993). While the procedure of Härdle et al. (1993) is specifically tailored for single-index models, the Lasso is designed to deal with the estimation of sparse linear models. On the other hand, the nonparametric kernel method is one of the best options when no obvious assumption (such as the single-index one) can be made on the shape of the targeted regression function.

We briefly recall that, for a linear model of the form $Y = \theta^{\star T}\mathbf{X} + W$, the Lasso estimate takes the form $\hat{F}_{\text{Lasso}}(\mathbf{x}) = \hat{\theta}_{\text{Lasso}}^T \mathbf{x}$, where

$$\hat{\theta}_{\text{Lasso}} \in \arg\min_{\theta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \theta^T \mathbf{X}_i\right)^2 + \xi \sum_{j=1}^{p} |\theta_j| \right\}$$

and $\xi > 0$ is a regularization parameter. Theoretical results (see, e.g., Bunea et al., 2007) indicate that ξ should be of the order $\xi^\star = \sigma\sqrt{\log(p)/n}$. Throughout, σ is assumed to be known, and we let $\xi = \xi^\star/3$, since this choice is known to give good practical results. The Nadaraya-Watson kernel estimate will be denoted by $\hat{F}_{\text{NW}}$. It is defined by

$$\hat{F}_{\text{NW}}(\mathbf{x}) = \frac{\sum_{i=1}^n Y_i K_h(\mathbf{x} - \mathbf{X}_i)}{\sum_{i=1}^n K_h(\mathbf{x} - \mathbf{X}_i)}$$

for some nonnegative kernel K on $\mathbb{R}^p$ and $K_h(\mathbf{z}) = K(\mathbf{z}/h)/h$. In the experiments, we let K be the Gaussian kernel $K(\mathbf{z}) = \exp(-\mathbf{z}^T\mathbf{z})$ and chose the smoothing parameter h via a classical leave-one-out procedure on the grid $\mathcal{G} = \{0.75^k, k = 0, \ldots, \lfloor \log(n) \rfloor\}$, see, for example, Györfi et al. (2002) (notation $\lfloor . \rfloor$ stands for the floor function). Finally, the estimation procedure advocated in Härdle

et al. (1993) takes the form

$$\hat{F}_{\mathrm{HHI}}(\mathbf{x}) = \frac{\sum_{i=1}^{n} Y_i G_{\hat{h}}\left(\hat{\theta}_{\mathrm{HHI}}^T(\mathbf{x}-\mathbf{X}_i)\right)}{\sum_{i=1}^{n} G_{\hat{h}}\left(\hat{\theta}_{\mathrm{HHI}}^T(\mathbf{x}-\mathbf{X}_i)\right)}$$

for some kernel G on $\mathbb{R}$, with $G_h(\mathbf{z}) = G(\mathbf{z}/h)/h$ and

$$\left(\hat{h}, \hat{\theta}_{\mathrm{HHI}}\right) \in \arg\min_{h>0, \theta\in\mathbb{R}^p} \sum_{i=1}^{n} \left[Y_i - \frac{\sum_{j\neq i} Y_j G_h\left(\theta^T(\mathbf{X}_j - \mathbf{X}_i)\right)}{\sum_{j\neq i} G_h\left(\theta^T(\mathbf{X}_j - \mathbf{X}_i)\right)} \right]^2.$$

All calculations were performed with the Gaussian kernel. We used the grid $\mathcal{G}$ for the optimization with respect to h, whereas the best search for θ was implemented via a pathwise coordinate optimization.

The various methods were tested for the general regression model

$$Y_i = F(\mathbf{X}_i) + W_i, \quad i = 1, \ldots, n,$$

for three different choices of F (single-index or not) and two values of n, namely $n = 50$ and $n = 100$. In each of these models, the observations $\mathbf{X}_i$ take values in $\mathbb{R}^p$, with $p = 10$ and $p = 50$, and have independent components uniformly distributed on $[-1, 1]$. The noise variables $W_1, \ldots, W_n$ are independently distributed according to a Gaussian $\mathcal{N}(0, \sigma^2)$, with $\sigma = 0.2$. It is worth pointing out that for $n = 50$ and $p = 50$, p and n are of the same order, which means that the setting is nonasymptotic. It is essentially in this case that the use of estimates tailored to sparsity, which reduce the variance, is expected to improve the performance over generalist methods. On the other hand, the situation $n = 100$ and $p = 10$ is less difficult and mimics the asymptotic setting.

The three examined functions $F(\mathbf{x})$, for $\mathbf{x} = (x_1, \ldots, x_p)$, were the following ones:

[**Model 1**] A linear model $F_{\mathrm{Linear}}(\mathbf{x}) = 2\theta^{\star T}\mathbf{x}$.

[**Model 2**] A single-index function $F_{\mathrm{SI}}(\mathbf{x}) = 2(\theta^{\star T}\mathbf{x})^2 + \theta^{\star T}\mathbf{x}$.

[**Model 3**] A purely nonparametric model $F_{\mathrm{NP}}(\mathbf{x}) = 2|x_2|\sqrt{|x_1|} - x_3^3$,

where, in the first and second model, $\theta^\star = (0.5, 0.5, 0, \ldots, 0)^T$. Thus, in [**Model 1**] and [**Model 2**], even if the ambient dimension is large, the intrinsic dimension of the model is in fact equal to 2.

For each experiment, a learning set of size n was generated to compute the estimates and their performance, in terms of mean square prevision error, was evaluated on a separate test set of the same size. The results are shown in Table 1 ($p = 10$) and Table 2 ($p = 50$). As each experiment was repeated 20 times, these tables report the median, the mean and the standard deviation (s.d.) of the prevision error of each procedure.

Some comments are in order. First, we note without surprise that:

1. The Lasso performs well in the linear setting [**Model 1**].
2. The single-index methods $\hat{F}_{\mathrm{Fourier}}$ and $\hat{F}_{\mathrm{HHI}}$ are the best ones when the targeted regression function really involves a single-index model [**Model 2**].
3. The kernel method gives good results in the purely nonparametric setting [**Model 3**].

$n = 50$	$p = 10$	$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
F_{Linear}	median	0.061	0.063	**0.046**	0.293
	mean	0.061	0.063	**0.047**	0.290
	s.d.	0.016	0.014	0.011	0.063
F_{SI}	median	**0.050**	0.067	0.307	0.198
	mean	**0.069**	0.080	0.338	0.208
	s.d.	0.081	0.057	0.082	0.072
F_{NP}	median	0.375	0.405	0.830	**0.354**
	mean	0.402	0.407	0.890	**0.336**
	s.d.	0.166	0.110	0.176	0.006
$n = 100$	$p = 10$	$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
F_{Linear}	median	0.053	0.051	**0.042**	0.227
	mean	0.056	0.050	**0.043**	0.237
	s.d.	0.011	0.006	0.004	0.044
F_{SI}	median	**0.047**	0.052	0.332	0.209
	mean	**0.049**	0.053	0.337	0.218
	s.d.	0.009	0.012	0.063	0.045
F_{NP}	median	**0.305**	0.343	0.793	0.333
	mean	**0.321**	0.338	0.833	0.324
	s.d.	0.092	0.042	0.145	0.041

Table 1: Numerical results for the simulated data, with $n = 50$ and $n = 100$, $p = 10$. The characters in bold indicate the best performance.

Interestingly, $\hat{F}_{\text{Fourier}}$ provides slightly better results than the single-index-tailored estimate $\hat{F}_{\text{HHI}}$, especially for $p = 50$. This observation can be easily explained by the fact that $\hat{F}_{\text{HHI}}$ does not integrate any sparsity information regarding the parameter $\theta^\star$, whereas $\hat{F}_{\text{Fourier}}$ tries to focus on the dimension of the active coordinates, which is equal to 2 in this simulation. As a general finding, we retain that $\hat{F}_{\text{Fourier}}$ is the most robust of all the tested procedures.

3.3 Real Data

The real-life data sets used in this second series of experiments are from two different sources. The first one, called **AIR-QUALITY** data ($n = 111$, $p = 3$), has been first used by Chambers et al. (1983) and has been later considered as a benchmark in the study and comparison of single-index models (see, for example, Antoniadis et al., 2004; Wang, 2009, , among others). This data set originated from an environmental study relating $n = 111$ ozone concentration measures at $p = 3$ meteorological variables, namely wind speed, temperature and radiation. The data is available as a package in the software **R** (R Development Core Team, 2008), which we employed in all the numerical experiments. The programs are available upon request from the authors.

The second category of data arises from the UC Irvine Machine Learning Repository `http://archive.ics.uci.edu/ml`, where the following packages have been downloaded from:

- **AUTO-MPG** (Quinlan, 1993, $n = 392$, $p = 7$).

$n = 50$	$p = 50$	$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
F_{Linear}	median	**0.057**	1.156	0.060	0.507
	mean	0.095	1.124	**0.066**	0.533
	s.d.	0.143	0.241	0.026	0.081
F_{SI}	median	**0.050**	0.502	0.795	0.308
	mean	**0.051**	0.539	0.776	0.326
	s.d.	0.011	0.200	0.208	0.109
F_{NP}	median	**0.358**	0.788	1.910	0.374
	mean	0.504	0.771	1.931	**0.391**
	s.d.	0.320	0.168	0.468	0.101
$n = 100$	$p = 50$	$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
F_{Linear}	median	0.053	0.092	**0.050**	0.519
	mean	0.054	0.100	**0.050**	0.508
	s.d.	0.007	0.026	0.006	0.026
F_{SI}	median	**0.047**	0.242	0.503	0.329
	mean	**0.070**	0.267	0.502	0.339
	s.d.	0.099	0.111	0.106	0.073
F_{NP}	median	**0.361**	0.736	1.968	0.418
	mean	0.557	0.765	2.045	**0.406**
	s.d.	0.519	0.226	0.546	0.076

Table 2: Numerical results for the simulated data, with $n = 50$ and $n = 100$, $p = 50$. The characters in bold indicate the best performance.

- **CONCRETE** (Yeh, 1998, $n = 1030$, $p = 8$).
- **HOUSING** (Harrison and Rubinfeld, 1978, $n = 508$, $p = 13$).
- **SLUMP-1**, **SLUMP-2** and **SLUMP-3**, which correspond to the concrete slump test data introduced by Yeh (2007) ($n = 51$, $p = 7$). Since there are 3 different output variables Y in the original data set, we created a single experiment for each of these variables (1 refers to the output "slump", 2 to the output "flow" and 3 to the output "28-day Compressive Strength").
- **WINE-RED** and **WINE-WHITE** (Cortez et al., 2009, $n = 1599$, $n = 4898$, $p = 11$).

We refer to the above-mentioned references for a precise description of the meaning of the variables involved in these data sets. For homogeneity reasons, all data were normalized to force the input variables to lie in $[-1,1]$—in accordance with the setting of our method—and to ensure that all output variables have standard deviation 0.5. In two data sets (**AIR-QUALITY** and **AUTO-MPG**) there were some missing values and the corresponding observations were simply removed.

For each method and each of the nine data sets, we randomly split the observations in a learning and a test set of equal sizes, computed the estimate on the learning set, evaluated the prediction error on the test set, and repeated this protocol 20 times. The results are summarized in Table 3.

We see that all the tested methods provide reasonable results on most data sets. The Lasso is very competitive, especially in the nonasymptotic framework. The estimation procedure $\hat{F}_{\text{Fourier}}$

Data set		$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
AIR QUALITY	median	0.117	**0.099**	0.107	0.129
$n = 111$	mean	0.128	**0.096**	0.113	0.130
$p = 3$	s.d.	0.044	0.029	0.029	0.035
AUTO-MPG	median	**0.044**	0.049	0.070	0.068
$n = 392$	mean	0.051	**0.050**	0.072	0.069
$p = 7$	s.d.	0.017	0.006	0.011	0.009
CONCRETE	median	0.089	**0.087**	0.106	0.094
$n = 1030$	mean	0.091	**0.087**	0.107	0.094
$p = 8$	s.d.	0.008	0.003	0.005	0.004
HOUSING	median	0.074	**0.059**	0.086	0.086
$n = 508$	mean	0.076	**0.061**	0.085	0.088
$p = 11$	s.d.	0.015	0.013	0.012	0.016
SLUMP-1	median	0.289	**0.171**	0.201	0.208
$n = 51$	mean	0.244	**0.187**	0.213	0.226
$p = 7$	s.d.	0.062	0.050	0.049	0.047
SLUMP-2	median	0.219	0.196	**0.172**	0.215
$n = 51$	mean	0.216	0.194	**0.171**	0.213
$p = 7$	s.d.	0.053	0.025	0.019	0.022
SLUMP-3	median	0.065	0.070	**0.053**	0.116
$n = 51$	mean	0.073	0.079	**0.052**	0.126
$p = 7$	s.d.	0.033	0.027	0.010	0.026
WINE-RED	median	0.173	**0.171**	0.183	0.171
$n = 1599$	mean	0.174	**0.170**	0.174	0.183
$p = 11$	s.d.	0.009	0.008	0.007	0.010
WINE-WHITE	median	0.191	0.187	0.185	**0.184**
$n = 4898$	mean	0.202	0.188	0.186	**0.185**
$p = 11$	s.d.	0.045	0.003	0.004	0.004

Table 3: Numerical results for the real-life data sets. The characters in bold indicate the best performance.

offers outcomes which are similar to the ones of $\hat{F}_{\text{HHI}}$, with a slight advantage for the latter method however. Altogether, $\hat{F}_{\text{Fourier}}$ and $\hat{F}_{\text{HHI}}$ provide the best performance in terms of prediction error in 6 out of 9 experiments. Besides, when it is not the best, the method $\hat{F}_{\text{Fourier}}$ is close to the best one, as for example in **SLUMP-3** and **WINE-RED**. As an illustrative example, the plot of the resulting fit of our procedure to the data set **AUTO-MPG** is shown in Figure 1.

Clearly, all data sets under study have a dimension p which is small compared to n. To correct this situation, we ran the same series of experiments by adding some additional irrelevant dimensions to the data. Specifically, the observations were embedded into a space of dimension $p \times 4$ by letting the new fake coordinates follow independent uniform $[0,1]$ random variables. The results are shown in Table 4. In this nonasymptotic framework, the method $\hat{F}_{\text{HHI}}$—which is not designed

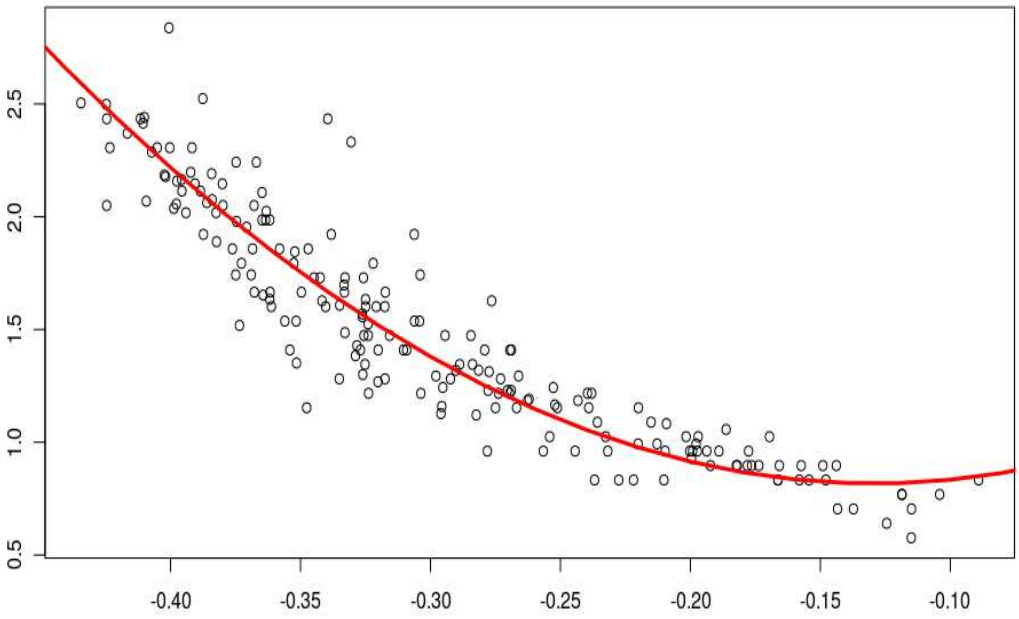


Figure 1: **AUTO-MPG** example: Estimated link function by the method $\hat{F}_{\text{Fourier}}$.

for sparsity—collapses, whereas $\hat{F}_{\text{Fourier}}$ takes a clear advantage over its competitors. In fact, it provides the best results in 3 out of 9 experiments (**AUTO-MPG**, **CONCRETE** and **HOUSING**). Besides, when it is not the best, the method $\hat{F}_{\text{Fourier}}$ is very close to the best one, as for example in **SLUMP-3** and **WINE-RED**.

Thus, as a general conclusion to this experimental section, we may say that our PAC-Bayesian oriented procedure has an excellent predictive ability, even in nonasymptotic/high-dimensional situations. It is fast, robust, and exhibits performance at the level of the gold standard Lasso. Moreover, as seen in the artificial data analysis, it is expected to perform better than the Lasso if the data cannot be explained approximately by a linear model.

4. Proofs

We start with some preliminary results that will play an important role throughout this section.

4.1 Preliminary Results

Throughout this section, we let π be the prior probability measure on $\mathbb{R}^p \times \mathcal{F}_n(C+1)$ equipped with its canonical Borel σ-field. Recall that $\mathcal{F}_n(C+1) \subset \mathcal{F}$ and that, for each $f \in \mathcal{F}_n(C+1)$, we have $\|f\|_\infty \leq C+1$.

Augmented data set		$\hat{F}_{\text{Fourier}}$	$\hat{F}_{\text{HHI}}$	$\hat{F}_{\text{Lasso}}$	$\hat{F}_{\text{NW}}$
AIR QUALITY	median	0.172	0.272	**0.164**	0.281
$n = 111$	mean	0.244	0.291	**0.163**	0.291
$p = 12$	s.d.	0.163	0.116	0.038	0.046
AUTO-MPG	median	**0.043**	0.062	0.085	0.202
$n = 392$	mean	**0.044**	0.072	0.086	0.203
$p = 28$	s.d.	0.009	0.018	0.008	0.014
CONCRETE	median	**0.087**	0.093	0.113	0.245
$n = 1030$	mean	**0.087**	0.094	0.112	0.094
$p = 32$	s.d.	0.007	0.008	0.005	0.009
HOUSING	median	**0.071**	0.199	0.092	0.226
$n = 508$	mean	**0.075**	0.181	0.095	0.227
$p = 44$	s.d.	0.023	0.084	0.013	0.018
SLUMP-1	median	**0.270**	0.426	0.276	0.271
$n = 51$	mean	0.290	0.409	0.274	**0.262**
$p = 28$	s.d.	0.101	0.079	0.055	0.042
SLUMP-2	median	0.276	0.332	**0.195**	0.253
$n = 51$	mean	0.285	0.349	**0.198**	0.254
$p = 28$	s.d.	0.075	0.063	0.043	0.034
SLUMP-3	median	0.079	0.371	**0.061**	0.372
$n = 51$	mean	0.082	0.361	**0.058**	0.279
$p = 28$	s.d.	0.025	0.079	0.013	0.031
WINE-RED	median	0.178	0.222	**0.172**	0.245
$n = 1599$	mean	0.176	0.226	**0.174**	0.246
$p = 44$	s.d.	0.085	0.033	0.006	0.029
WINE-WHITE	median	0.199	0.239	**0.187**	0.252
$n = 4898$	mean	0.204	0.256	**0.188**	0.260
$p = 44$	s.d.	0.091	0.041	0.005	0.019

Table 4: Numerical results for the real-life data sets augmented with noise variables. The characters in bold indicate the best performance.

Besides, since $\mathbb{E}[Y|\mathbf{X}] = f^\star(\theta^{\star T}\mathbf{X})$ almost surely, we note once and for all that for all $(\theta, f) \in \mathcal{S}^p_{1,+} \times \mathcal{F}_n(C+1)$,

$$\begin{aligned} R(\theta, f) - R(\theta^\star, f^\star) &= \mathbb{E}\left[Y - f(\theta^T\mathbf{X})\right]^2 - \mathbb{E}\left[Y - f^\star(\theta^{\star T}\mathbf{X})\right]^2 \\ &= \mathbb{E}\left[f(\theta^T\mathbf{X}) - f^\star(\theta^{\star T}\mathbf{X})\right]^2 \end{aligned}$$

(Pythagora's theorem). We start with four technical lemmas. Lemma 5 is a version of Bernstein's inequality, whose proof can be found in Massart (2007, Chapter 2, inequality (2.21)). Lemma 6 is a classical result, whose proof can be found, for example, in Catoni (2007, page 4). For a random variable Z, the notation $(Z)_+$ means the positive part of Z.

Lemma 5 *Let $T_1, \ldots, T_n$ be independent real-valued random variables. Assume that there exist two positive constants v and w such that, for all integers $k \geq 2$,*

$$\sum_{i=1}^{n} \mathbb{E}\left[(T_i)_+^k\right] \leq \frac{k!}{2} v w^{k-2}.$$

Then, for any $\zeta \in]0, 1/w[$,

$$\mathbb{E}\left[\exp\left(\zeta \sum_{i=1}^{n} [T_i - \mathbb{E}T_i]\right)\right] \leq \exp\left(\frac{v\zeta^2}{2(1-w\zeta)}\right).$$

Given a measurable space $(E, \mathcal{E})$ and two probability measures μ_1 and μ_2 on $(E, \mathcal{E})$, we denote by $\mathcal{K}(\mu_1, \mu_2)$ the Kullback-Leibler divergence of μ_1 with respect to μ_2, defined by

$$\mathcal{K}(\mu_1, \mu_2) = \begin{cases} \displaystyle\int \log\left(\frac{\mathrm{d}\mu_1}{\mathrm{d}\mu_2}\right) \mathrm{d}\mu_1 & \text{if } \mu_1 \ll \mu_2, \\ \infty & \text{otherwise.} \end{cases}$$

(Notation $\mu_1 \ll \mu_2$ means "μ_1 is absolutely continuous with respect to μ_2".) In the next lemma, notation $\circ$ stands for the function composition operator.

Lemma 6 *Let $(E, \mathcal{E})$ be a measurable space. For any probability measure μ on $(E, \mathcal{E})$ and any measurable function $h : E \to \mathbb{R}$ such that $\int (\exp \circ h) \mathrm{d}\mu < \infty$, we have*

$$\log \int (\exp \circ h) \mathrm{d}\mu = \sup_m \left(\int h \mathrm{d}m - \mathcal{K}(m, \mu) \right), \tag{6}$$

where the supremum is taken over all probability measures on $(E, \mathcal{E})$ and, by convention, $\infty - \infty = -\infty$. Moreover, as soon as h is bounded from above on the support of μ, the supremum with respect to m on the right-hand side of (6) is reached for the Gibbs distribution g given by

$$\frac{\mathrm{d}g}{\mathrm{d}\mu}(e) = \frac{\exp[h(e)]}{\displaystyle\int (\exp \circ h) \mathrm{d}\mu}, \quad e \in E.$$

Lemma 7 *Assume that Assumption* **N** *holds. Set $w = 8(2C+1)\max[L, 2C+1]$ and take*

$$\lambda \in \left]0, \frac{n}{w + [(2C+1)^2 + 4\sigma^2]}\right[.$$

Then, for all $\delta \in]0, 1[$ and any data-dependent probability measure $\hat{\rho}$ absolutely continuous with respect to π we have, with probability at least $1 - \delta$,

$$\begin{aligned} &R(\hat{\theta}, \hat{f}) - R(\theta^\star, f^\star) \\ &\quad \leq \frac{1}{1 - \frac{\lambda[(2C+1)^2 + 4\sigma^2]}{n - w\lambda}} \left(R_n(\hat{\theta}, \hat{f}) - R_n(\theta^\star, f^\star) + \frac{\log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\hat{\theta}, \hat{f})\right) + \log\left(\frac{1}{\delta}\right)}{\lambda} \right), \end{aligned}$$

where the pair $(\hat{\theta}, \hat{f})$ is distributed according to $\hat{\rho}$.

Proof Fix $\theta \in \mathcal{S}_{1,+}^p$ and $f \in \mathcal{F}_n(C+1)$. The proof starts with an application of Lemma 5 to the random variables

$$T_i = -\left(Y_i - f(\theta^T \mathbf{X}_i)\right)^2 + \left(Y_i - f^\star(\theta^{\star T} \mathbf{X}_i)\right)^2, \quad i = 1, \ldots, n.$$

Note that these random variables are independent, identically distributed, and that

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E} T_i^2 &= \sum_{i=1}^n \mathbb{E}\left\{ \left[2Y_i - f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right]^2 \left[f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right]^2 \right\} \\
&= \sum_{i=1}^n \mathbb{E}\left\{ \left[2W_i + f^\star(\theta^{\star T} \mathbf{X}_i) - f(\theta^T \mathbf{X}_i)\right]^2 \left[f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right]^2 \right\} \\
&\leq \sum_{i=1}^n \mathbb{E}\left\{ \left[4W_i^2 + (2C+1)^2\right] \left[f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right]^2 \right\} \\
&\quad (\text{since } \mathbb{E}[W_i | \mathbf{X}_i] = 0).
\end{aligned}$$

Thus, by Assumption **N**,

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E} T_i^2 &\leq \left[(2C+1)^2 + 4\sigma^2\right] \sum_{i=1}^n \mathbb{E}\left[f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right]^2 \\
&\leq v,
\end{aligned}$$

where we set

$$v = 2n\left[(2C+1)^2 + 4\sigma^2\right]\left[R(\theta, f) - R(\theta^\star, f^\star)\right]. \tag{7}$$

More generally, for all integers $k \geq 3$,

$$\begin{aligned}
&\sum_{i=1}^n \mathbb{E}\left[(T_i)_+^k\right] \\
&\quad \leq \sum_{i=1}^n \mathbb{E}\left\{ \left|2Y_i - f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^k \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^k \right\} \\
&\quad = \sum_{i=1}^n \mathbb{E}\left\{ \left|2W_i + f^\star(\theta^{\star T} \mathbf{X}_i) - f(\theta^T \mathbf{X}_i)\right|^k \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^k \right\} \\
&\quad \leq 2^{k-1} \sum_{i=1}^n \mathbb{E}\left\{ \left[2^k |W_i|^k + (2C+1)^k\right] (2C+1)^{k-2} \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^2 \right\}.
\end{aligned}$$

In the last inequality, we used the fact that $|a+b|^k \leq 2^{k-1}(|a|^k + |b|^k)$ together with

$$\begin{aligned}
\left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^k &= \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^{k-2} \times \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^2 \\
&\leq (2C+1)^{k-2} \left|f(\theta^T \mathbf{X}_i) - f^\star(\theta^{\star T} \mathbf{X}_i)\right|^2.
\end{aligned}$$

Therefore, by Assumption **N**,

$$\begin{aligned}
&\sum_{i=1}^{n} \mathbb{E}\left[(T_i)_+^k\right] \\
&\quad \leq \sum_{i=1}^{n} \left[2^{2k-2}k!\sigma^2 L^{k-2} + 2^{k-1}(2C+1)^k\right](2C+1)^{k-2}\left[R(\theta,f) - R(\theta^\star,f^\star)\right] \\
&\quad = v \times \frac{\left[2^{2k-2}k!\sigma^2 L^{k-2} + 2^{k-1}(2C+1)^k\right](2C+1)^{k-2}}{\left[(2C+1)^2 + 4\sigma^2\right]} \\
&\quad \leq v \times \frac{8^{k-2}k!\max\left[L^{k-2},(2C+1)^{k-2}\right](2C+1)^{k-2}}{2} \\
&\quad = \frac{k!}{2} v w^{k-2},
\end{aligned}$$

with $w = 8(2C+1)\max[L, 2C+1]$.

Thus, for any inverse temperature parameter $\lambda \in]0, n/w[$, taking $\zeta = \lambda/n$, we may write by Lemma 5

$$\mathbb{E}\Big\{ \exp\left[\lambda\left(R(\theta,f) - R(\theta^\star,f^\star) - R_n(\theta,f) + R_n(\theta^\star,f^\star)\right)\right] \Big\} \leq \exp\left(\frac{v\lambda^2}{2n^2(1-\frac{w\lambda}{n})}\right).$$

Therefore, using the definition of v, we obtain

$$\begin{aligned}
\mathbb{E}\Bigg\{ \exp\Bigg[&\left(\lambda - \frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta,f) - R(\theta^\star,f^\star)\right) \\
&+ \lambda\left(-R_n(\theta,f) + R_n(\theta^\star,f^\star)\right) - \log\left(\frac{1}{\delta}\right)\Bigg]\Bigg\} \leq \delta.
\end{aligned}$$

Next, we use a standard PAC-Bayesian approach (Catoni, 2004, 2007; Audibert, 2004; Alquier, 2008). Let us remind the reader that π is a prior probability measure on the set $\mathcal{S}_{1,+}^p \times \mathcal{F}_n(C+1)$. We have

$$\begin{aligned}
\int \mathbb{E}\Bigg\{ \exp\Bigg[&\left(\lambda - \frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta,f) - R(\theta^\star,f^\star)\right) \\
&+ \lambda\left(-R_n(\theta,f) + R_n(\theta^\star,f^\star)\right) - \log\left(\frac{1}{\delta}\right)\Bigg]\Bigg\} d\pi(\theta,f) \leq \delta
\end{aligned}$$

and consequently, using Fubini's theorem,

$$\begin{aligned}
\mathbb{E}\Bigg\{ \int \exp\Bigg[&\left(\lambda - \frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta,f) - R(\theta^\star,f^\star)\right) \\
&+ \lambda\left(-R_n(\theta,f) + R_n(\theta^\star,f^\star)\right) - \log\left(\frac{1}{\delta}\right)\Bigg] d\pi(\theta,f)\Bigg\} \leq \delta.
\end{aligned}$$

Therefore, for any data-dependent posterior probability measure $\hat{\rho}$ absolutely continuous with respect to π, adopting the convention $\infty \times 0 = 0$,

$$\mathbb{E}\Bigg\{ \int \exp\Bigg[\Bigg(\lambda - \frac{\lambda^2 \left[(2C+1)^2 + 4\sigma^2 \right]}{n(1-\frac{w\lambda}{n})} \Bigg) \left(R(\theta, f) - R(\theta^\star, f^\star) \right) + \lambda \left(-R_n(\theta, f) + R_n(\theta^\star, f^\star) \right) - \log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\theta, f) \right) - \log\left(\frac{1}{\delta} \right) \Bigg] \mathrm{d}\hat{\rho}(\theta, f) \Bigg\} \leq \delta.$$

Recalling that $\mathbf{P}^{\otimes n}$ stands for the distribution of the sample $\mathcal{D}_n$, the latter inequality can be more conveniently written as

$$\mathbb{E}_{\mathcal{D}_n \sim \mathbf{P}^{\otimes n}} \mathbb{E}_{(\hat{\theta}, \hat{f}) \sim \hat{\rho}} \Bigg\{ \exp\Bigg[\Bigg(\lambda - \frac{\lambda^2 \left[(2C+1)^2 + 4\sigma^2 \right]}{n(1-\frac{w\lambda}{n})} \Bigg) \left(R(\hat{\theta}, \hat{f}) - R(\theta^\star, f^\star) \right) + \lambda \left(-R_n(\hat{\theta}, \hat{f}) + R_n(\theta^\star, f^\star) \right) - \log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\hat{\theta}, \hat{f}) \right) - \log\left(\frac{1}{\delta} \right) \Bigg] \Bigg\} \leq \delta.$$

Thus, using the elementary inequality $\exp(\lambda x) \geq \mathbf{1}_{\mathbb{R}_+}(x)$ we obtain, with probability at most δ,

$$\Bigg(1 - \frac{\lambda \left[(2C+1)^2 + 4\sigma^2 \right]}{n(1-\frac{w\lambda}{n})} \Bigg) \left(R(\hat{\theta}, \hat{f}) - R(\theta^\star, f^\star) \right) \geq R_n(\hat{\theta}, \hat{f}) - R_n(\theta^\star, f^\star) + \frac{\log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\hat{\theta}, \hat{f}) \right) + \log\left(\frac{1}{\delta} \right)}{\lambda},$$

where the probability is evaluated with respect to the distribution $\mathbf{P}^{\otimes n}$ of the data $\mathcal{D}_n$ *and* the conditional probability measure $\hat{\rho}$. Put differently, letting

$$\lambda \in \left] 0, \frac{n}{w + \left[(2C+1)^2 + 4\sigma^2 \right]} \right[,$$

we have, with probability at least $1 - \delta$,

$$R(\hat{\theta}, \hat{f}) - R(\theta^\star, f^\star) \leq \frac{1}{1 - \frac{\lambda[(2C+1)^2 + 4\sigma^2]}{n - w\lambda}} \left(R_n(\hat{\theta}, \hat{f}) - R_n(\theta^\star, f^\star) + \frac{\log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\hat{\theta}, \hat{f}) \right) + \log\left(\frac{1}{\delta} \right)}{\lambda} \right).$$

This concludes the proof of Lemma 7. ■

Lemma 8 *Under the conditions of Lemma 7 we have, with probability at least* $1-\delta$*,*

$$\int R_n(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R_n(\theta^\star,f^\star)$$
$$\leq \left(1+\frac{\lambda\left[(2C+1)^2+4\sigma^2\right]}{n-w\lambda}\right)\left(\int R(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R(\theta^\star,f^\star)\right) + \frac{\mathcal{K}(\hat{\rho},\pi)+\log\left(\frac{1}{\delta}\right)}{\lambda}.$$

Proof The beginning of the proof is similar to the one of Lemma 7. More precisely, we apply Lemma 5 with $T_i = (Y_i - f(\theta^T\mathbf{X}_i))^2 - (Y_i - f^\star(\theta^{\star T}\mathbf{X}_i))^2$ and obtain, for any inverse temperature parameter $\lambda \in]0, n/w[$,

$$\mathbb{E}\left\{\exp\left[\lambda\left(R(\theta^\star,f^\star) - R(\theta,f) - R_n(\theta^\star,f^\star) + R_n(\theta,f)\right)\right]\right\} \leq \exp\left(\frac{v\lambda^2}{2n^2(1-\frac{w\lambda}{n})}\right)$$

(see (7) for the definition of v). Thus, using the definition of v,

$$\mathbb{E}\Bigg\{\exp\Bigg[\left(\lambda+\frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta^\star,f^\star)-R(\theta,f)\right)$$
$$+\lambda\left(R_n(\theta,f)-R_n(\theta^\star,f^\star)\right)-\log\left(\frac{1}{\delta}\right)\Bigg]\Bigg\} \leq \delta.$$

Integrating with respect to π leads to

$$\int \mathbb{E}\Bigg\{\exp\Bigg[\left(\lambda+\frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta^\star,f^\star)-R(\theta,f)\right)$$
$$+\lambda\left(R_n(\theta,f)-R_n(\theta^\star,f^\star)\right)-\log\left(\frac{1}{\delta}\right)\Bigg]\Bigg\}\mathrm{d}\pi(\theta,f) \leq \delta$$

whence, by Fubini's theorem,

$$\mathbb{E}\Bigg\{\int\exp\Bigg[\left(\lambda+\frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta^\star,f^\star)-R(\theta,f)\right)$$
$$+\lambda\left(R_n(\theta,f)-R_n(\theta^\star,f^\star)\right)-\log\left(\frac{1}{\delta}\right)\Bigg]\mathrm{d}\pi(\theta,f)\Bigg\} \leq \delta.$$

Thus, for any data-dependent posterior probability measure $\hat{\rho}$ absolutely continuous with respect to π,

$$\mathbb{E}\Bigg\{\int\exp\Bigg[\left(\lambda+\frac{\lambda^2\left[(2C+1)^2+4\sigma^2\right]}{n(1-\frac{w\lambda}{n})}\right)\left(R(\theta^\star,f^\star)-R(\theta,f)\right)$$
$$+\lambda\left(R_n(\theta,f)-R_n(\theta^\star,f^\star)\right)$$
$$-\log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\theta,f)\right)-\log\left(\frac{1}{\delta}\right)\Bigg]\mathrm{d}\hat{\rho}(\theta,f)\Bigg\}$$
$$\leq \delta.$$

Therefore, by Jensen's inequality,

$$\begin{aligned}
&\mathbb{E}\Bigg\{ \exp \int \Bigg[\left(\lambda + \frac{\lambda^2 \left[(2C+1)^2 + 4\sigma^2 \right]}{n(1-\frac{w\lambda}{n})} \right) (R(\theta^\star, f^\star) - R(\theta, f)) \\
&\qquad\qquad + \lambda (R_n(\theta, f) - R_n(\theta^\star, f^\star)) \\
&\qquad\qquad - \log\left(\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\pi}(\theta, f) \right) - \log\left(\frac{1}{\delta} \right) \Bigg] \mathrm{d}\hat{\rho}(\theta, f) \Bigg\} \\
&= \mathbb{E}\Bigg\{ \exp\Bigg[\left(\lambda + \frac{\lambda^2 \left[(2C+1)^2 + 4\sigma^2 \right]}{n(1-\frac{w\lambda}{n})} \right) \left(R(\theta^\star, f^\star) - \int R(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) \right) \\
&\qquad\qquad + \lambda \left(\int R_n(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R_n(\theta^\star, f^\star) \right) \\
&\qquad\qquad - \mathcal{K}(\hat{\rho}, \pi) - \log\left(\frac{1}{\delta} \right) \Bigg] \Bigg\} \\
&\leq \delta.
\end{aligned}$$

Consequently, by the elementary inequality $\exp(\lambda x) \geq \mathbf{1}_{\mathbb{R}_+}(x)$, we obtain, with probability at most δ,

$$\begin{aligned}
&\int R_n(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R_n(\theta^\star, f^\star) \\
&\quad \geq \left(1 + \frac{\lambda \left[(2C+1)^2 + 4\sigma^2 \right]}{n - w\lambda} \right) \left(\int R(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R(\theta^\star, f^\star) \right) \\
&\qquad\qquad + \frac{\mathcal{K}(\hat{\rho}, \pi) + \log\left(\frac{1}{\delta} \right)}{\lambda}.
\end{aligned}$$

Equivalently, with probability at least $1 - \delta$,

$$\begin{aligned}
&\int R_n(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R_n(\theta^\star, f^\star) \\
&\quad \leq \left(1 + \frac{\lambda \left[(2C+1)^2 + 4\sigma^2 \right]}{n - w\lambda} \right) \left(\int R(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R(\theta^\star, f^\star) \right) \\
&\qquad\qquad + \frac{\mathcal{K}(\hat{\rho}, \pi) + \log\left(\frac{1}{\delta} \right)}{\lambda}.
\end{aligned}$$

■

4.2 Proof of Theorem 2

The proof starts with an application of Lemma 7 with $\hat{\rho} = \hat{\rho}_\lambda$ (the Gibbs distribution) as posterior distribution. More precisely, we know that, with probability larger than $1-\delta$,

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \le \frac{1}{1 - \frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \left(R_n(\hat{\theta}_\lambda, \hat{f}_\lambda) - R_n(\theta^\star, f^\star) + \frac{\log\left(\frac{\mathrm{d}\hat{\rho}_\lambda}{\mathrm{d}\pi}(\hat{\theta}_\lambda, \hat{f}_\lambda)\right) + \log\left(\frac{1}{\delta}\right)}{\lambda} \right),$$

where the probability is evaluated with respect to the distribution $\mathbf{P}^{\otimes n}$ of the data $\mathcal{D}_n$ *and* the conditional probability measure $\hat{\rho}_\lambda$. Observe that

$$\begin{aligned}\log\left(\frac{\mathrm{d}\hat{\rho}_\lambda}{\mathrm{d}\pi}(\hat{\theta}_\lambda, \hat{f}_\lambda)\right) &= \log\left(\frac{\exp\left[-\lambda R_n(\hat{\theta}_\lambda, \hat{f}_\lambda)\right]}{\int \exp\left[-\lambda R_n(\theta, f)\right] \mathrm{d}\pi(\theta, f)}\right) \\ &= -\lambda R_n(\hat{\theta}_\lambda, \hat{f}_\lambda) - \log \int \exp\left[-\lambda R_n(\theta, f)\right] \mathrm{d}\pi(\theta, f).\end{aligned}$$

Consequently, with probability at least $1-\delta$,

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \le \frac{1}{\lambda\left(1 - \frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}\right)} \left(-\log \int \exp\left[-\lambda R_n(\theta, f)\right] \mathrm{d}\pi(\theta, f) - \lambda R_n(\theta^\star, f^\star) + \log\left(\frac{1}{\delta}\right) \right).$$

Next, using Lemma 6 we deduce that, with probability at least $1-\delta$,

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \le \frac{1}{1 - \frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\hat{\rho}} \left\{ \int R_n(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R_n(\theta^\star, f^\star) + \frac{\mathcal{K}(\hat{\rho}, \pi) + \log\left(\frac{1}{\delta}\right)}{\lambda} \right\},$$

where the infimum is taken over all probability measures on $\mathcal{S}^p_{1,+} \times \mathcal{F}_n(C+1)$. In particular, letting $\mathcal{M}(I, M)$ be the set of all probability measures on $\mathcal{S}^p_{1,+}(I) \times \mathcal{F}_M(C+1)$, we have, with probability at least $1-\delta$,

$$\begin{aligned}&R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \\ &\le \frac{1}{1 - \frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\substack{I \subset \{1,\dots,p\} \\ 1 \le M \le n}} \inf_{\hat{\rho} \in \mathcal{M}(I,M)} \left\{ \int R_n(\theta, f) \mathrm{d}\hat{\rho}(\theta, f) - R_n(\theta^\star, f^\star) + \frac{\mathcal{K}(\hat{\rho}, \pi) + \log\left(\frac{1}{\delta}\right)}{\lambda} \right\}.\end{aligned}$$

Next, observe that, for $\hat{\rho} \in \mathcal{M}(I,M)$,

$$\mathcal{K}(\hat{\rho},\pi) = \mathcal{K}(\hat{\rho},\mu\otimes\nu) = \mathcal{K}(\hat{\rho},\mu_I\otimes\nu_M) + \log\left[\frac{\left(1-\left(\frac{1}{10}\right)^p\right)\left(1-\left(\frac{1}{10}\right)^n\right)\binom{p}{|I|}}{10^{-|I|-M}}\right]$$
$$\leq \mathcal{K}(\hat{\rho},\mu_I\otimes\nu_M) + \log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right]. \tag{8}$$

Therefore, with probability at least $1-\delta$,

$$R(\hat{\theta}_\lambda,\hat{f}_\lambda) - R(\theta^\star,f^\star)$$
$$\leq \frac{1}{1-\frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\substack{I\subset\{1,\dots,p\}\\ 1\leq M\leq n}} \inf_{\hat{\rho}\in\mathcal{M}(I,M)} \left\{ \int R_n(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R_n(\theta^\star,f^\star) \right.$$
$$\left. + \frac{\mathcal{K}(\hat{\rho},\mu_I\otimes\nu_M) + \log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right] + \log\left(\frac{1}{\delta}\right)}{\lambda} \right\}. \tag{9}$$

By Lemma 8 and inequality (8), for any data-dependent distribution $\hat{\rho}\in\mathcal{M}(I,M)$, with probability at least $1-\delta$,

$$\int R_n(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R_n(\theta^\star,f^\star)$$
$$\leq \left(1+\frac{\lambda\left[(2C+1)^2+4\sigma^2\right]}{n-w\lambda}\right)\left(\int R(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R(\theta^\star,f^\star)\right)$$
$$+ \frac{\mathcal{K}(\hat{\rho},\mu_I\otimes\nu_M) + \log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right] + \log\left(\frac{1}{\delta}\right)}{\lambda}. \tag{10}$$

Thus, combining inequalities (9) and (10), we may write, with probability at least $1-2\delta$,

$$R(\hat{\theta}_\lambda,\hat{f}_\lambda) - R(\theta^\star,f^\star)$$
$$\leq \frac{1}{1-\frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\substack{I\subset\{1,\dots,p\}\\ 1\leq M\leq n}} \inf_{\hat{\rho}\in\mathcal{M}(I,M)} \left\{ \right.$$
$$\left(1+\frac{\lambda\left[(2C+1)^2+4\sigma^2\right]}{n-w\lambda}\right)\left(\int R(\theta,f)\mathrm{d}\hat{\rho}(\theta,f) - R(\theta^\star,f^\star)\right)$$
$$\left. + 2\,\frac{\mathcal{K}(\hat{\rho},\mu_I\otimes\nu_M) + \log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right] + \log\left(\frac{1}{\delta}\right)}{\lambda} \right\}. \tag{11}$$

For any subset I of $\{1,\dots,p\}$, any positive integer $M\leq n$ and any $\eta,\gamma\in]0,1/n]$, let the probability measure $\rho_{I,M,\eta,\gamma}$ be defined by

$$\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f) = \mathrm{d}\rho^1_{I,M,\eta}(\theta)\mathrm{d}\rho^2_{I,M,\gamma}(f),$$

with

$$\frac{\mathrm{d}\rho^1_{I,M,\eta}}{\mathrm{d}\mu_I}(\theta) \propto \mathbf{1}_{[\|\theta-\theta^\star_{I,M}\|_1 \le \eta]}$$

and

$$\frac{\mathrm{d}\rho^2_{I,M,\gamma}}{\mathrm{d}\nu_M}(f) \propto \mathbf{1}_{[\|f-f^\star_{I,M}\|_M \le \gamma]}$$

where, for $f = \sum_{j=1}^M \beta_j \varphi_j \in \mathcal{F}_M(C+1)$, we put

$$\|f\|_M = \sum_{j=1}^M j|\beta_j|.$$

With this notation, inequality (11) leads to

$$\begin{aligned} &R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \\ &\le \frac{1}{1-\frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\substack{I\subset\{1,\dots,p\}\\ 1\le M\le n}} \inf_{\eta,\gamma>0} \Bigg\{ \\ &\qquad \left(1+\frac{\lambda\left[(2C+1)^2+4\sigma^2\right]}{n-w\lambda}\right)\left(\int R(\theta,f)\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f) - R(\theta^\star,f^\star)\right) \\ &\qquad + 2\,\frac{\mathcal{K}(\rho_{I,M,\eta,\gamma}, \mu_I\otimes\nu_M) + \log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right] + \log\left(\frac{1}{\delta}\right)}{\lambda} \Bigg\}. \end{aligned} \tag{12}$$

To finish the proof, we have to control the different terms in (12). Note first that

$$\log\binom{p}{|I|} \le |I|\log\left(\frac{pe}{|I|}\right)$$

and, consequently,

$$\log\left[\frac{\binom{p}{|I|}}{10^{-|I|-M}}\right] \le |I|\log\left(\frac{pe}{|I|}\right) + (|I|+M)\log 10. \tag{13}$$

Next,

$$\begin{aligned} \mathcal{K}(\rho_{I,M,\eta,\gamma}, \mu_I\otimes\nu_M) &= \mathcal{K}(\rho^1_{I,M,\eta}\otimes\rho^2_{I,M,\gamma}, \mu_I\otimes\nu_M) \\ &= \mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) + \mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M). \end{aligned}$$

By technical Lemma 9, we know that

$$\mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) \le (|I|-1)\log\left(\max\left[|I|, \frac{4}{\eta}\right]\right).$$

Similarly, by technical Lemma 10,

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = M\log\left(\frac{C+1}{\gamma}\right).$$

Putting all the pieces together, we are led to

$$\mathcal{K}(\rho_{I,M,\eta,\gamma}, \mu_I \otimes \nu_M) \leq (|I|-1)\log\left(\max\left[|I|, \frac{4}{\eta}\right]\right) + M\log\left(\frac{C+1}{\gamma}\right). \tag{14}$$

Finally, it remains to control the term

$$\int R(\theta, f)\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f).$$

To this aim, we write

$$\begin{aligned}
&\int R(\theta, f)\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f) \\
&\quad = \int \mathbb{E}\left[\left(Y - f(\theta^T\mathbf{X})\right)^2\right]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f) \\
&\quad = \int \mathbb{E}\big[\big(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) + f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X}) \\
&\qquad\quad + f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\big)^2\big]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f) \\
&\quad = R(\theta^\star_{I,M}, f^\star_{I,M}) \\
&\qquad + \int \mathbb{E}\Big[\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right)^2 \\
&\qquad + \left(f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\right)^2 \\
&\qquad + 2\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right) \\
&\qquad + 2\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\right) \\
&\qquad + 2\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\right)\Big]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f) \\
&\quad := R(\theta^\star_{I,M}, f^\star_{I,M}) + \mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D} + \mathbf{E}.
\end{aligned}$$

4.2.1 COMPUTATION OF C

By Fubini's theorem,

$$\begin{aligned}
\mathbf{C} &= \mathbb{E}\left[\int 2\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right)\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f)\right] \\
&= \mathbb{E}\Bigg\{\int\Bigg[2\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right) \\
&\qquad\qquad \times \int\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right)\mathrm{d}\rho^2_{I,M,\gamma}(f)\Bigg]\mathrm{d}\rho^1_{I,M,\eta}(\theta)\Bigg\}.
\end{aligned}$$

By the triangle inequality, for $f = \sum_{j=1}^M \beta_j\varphi_j$ and $f^\star_{I,M} = \sum_{j=1}^M (\beta^\star_{I,M})_j\varphi_j$, it holds

$$\sum_{j=1}^M j|\beta_j| \leq \sum_{j=1}^M j\left|\beta_j - (\beta^\star_{I,M})_j\right| + \sum_{j=1}^M j\left|(\beta^\star_{I,M})_j\right|.$$

Since $f^\star_{I,M} \in \mathcal{F}_M(C)$, we have $\sum_{j=1}^M j|(\beta^\star_{I,M})_j| \le C$, so that $\sum_{j=1}^M j|\beta_j| \le C+1$ as soon as $\|f - f^\star_{I,M}\|_M \le 1$. This shows that the set

$$\left\{ f = \sum_{j=1}^M \beta_j \varphi_j : \|f - f^\star_{I,M}\|_M \le \gamma \right\}$$

is contained in the support of ν_M. In particular, this implies that $\rho^2_{I,M,\gamma}$ is centered at $f^\star_{I,M}$ and, consequently,

$$\int \left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X}) \right) \mathrm{d}\rho^2_{I,M,\gamma}(f) = 0.$$

This proves that $\mathbf{C} = 0$.

4.2.2 CONTROL OF A

Clearly,

$$\mathbf{A} \le \int \sup_{y \in \mathbb{R}} \left((f^\star_{I,M}(y) - f(y)) \right)^2 \mathrm{d}\rho^2_{I,M,\gamma}(f) \le \gamma^2.$$

4.2.3 CONTROL OF B

We have

$$\begin{aligned}
\mathbf{B} &= \int \mathbb{E}\left[\left(f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T \mathbf{X}) \right)^2 \right] \mathrm{d}\rho_{I,M,\eta,\gamma}(\theta, f) \\
&\le \int \mathbb{E}\left[\left(\ell(C+1)(\theta^{\star T}_{I,M} - \theta^T)\mathbf{X} \right)^2 \right] \mathrm{d}\rho^1_{I,M,\eta}(\theta) \\
&\quad \text{(by the mean value theorem)} \\
&\le \ell^2 (C+1)^2 \mathbb{E}\left[\|\mathbf{X}\|^2_\infty \right] \int \|\theta^\star_{I,M} - \theta\|_1^2 \mathrm{d}\rho^1_{I,M,\eta}(\theta) \\
&\le \ell^2 (C+1)^2 \eta^2 \\
&\quad \text{(by Assumption } \mathbf{D}\text{)}.
\end{aligned}$$

4.2.4 CONTROL OF E

Write

$$
\begin{aligned}
|\mathbf{E}| &\leq 2\int \mathbb{E}\Big[\big|f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\big| \\
&\qquad\qquad \times \big|f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\big|\Big]\,\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f) \\
&\leq 2\int \mathbb{E}\Big[\big|f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\big| \\
&\qquad\qquad \times \ell(C+1)\big|(\theta^{\star T}_{I,M} - \theta^T)\mathbf{X}\big|\Big]\,\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f) \\
&\leq 2\left(\int \mathbb{E}\left[\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^{\star T}_{I,M}\mathbf{X})\right)^2\right]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f)\right)^{\frac{1}{2}} \\
&\qquad\qquad \left(\int \mathbb{E}\left[\left(\ell(C+1)(\theta^{\star T}_{I,M} - \theta^T)\mathbf{X}\right)^2\right]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f)\right)^{\frac{1}{2}} \\
&\quad \text{(by the Cauchy-Schwarz inequality)} \\
&\leq 2\left(\gamma^2\right)^{\frac{1}{2}}\left(\ell^2(C+1)^2\eta^2\right)^{\frac{1}{2}} \\
&= 2\ell(C+1)\gamma\eta.
\end{aligned}
$$

4.2.5 CONTROL OF D

Finally,

$$
\begin{aligned}
\mathbf{D} &= 2\int \mathbb{E}\left[\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f(\theta^{\star T}_{I,M}\mathbf{X}) - f(\theta^T\mathbf{X})\right)\right]\mathrm{d}\rho_{I,M,\eta,\gamma}(\theta,f) \\
&= 2\int \mathbb{E}\left[\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right)\right]\mathrm{d}\rho^1_{I,M,\eta}(\theta) \\
&\quad \text{(since } \int f\mathrm{d}\rho^2_{I,M,\gamma}(f) = f^\star_{I,M}\text{)} \\
&= 2\mathbb{E}\left[\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)\int\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right)\mathrm{d}\rho^1_{I,M,\eta}(\theta)\right] \\
&\leq 2\sqrt{\mathbb{E}\left[\left(Y - f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X})\right)^2\right]} \\
&\quad \times \sqrt{\mathbb{E}\left[\int\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right)\mathrm{d}\rho^1_{I,M,\eta}(\theta)\right]^2} \\
&\quad \text{(by the Cauchy-Schwarz inequality)} \\
&= 2\sqrt{R(\theta^\star_{I,M}, f^\star_{I,M})}\sqrt{\mathbb{E}\left[\int\left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right)\mathrm{d}\rho^1_{I,M,\eta}(\theta)\right]^2}.
\end{aligned}
$$

The inequality

$$
\begin{aligned}
\big|f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\big| &\leq \ell(C+1)\big|(\theta^{\star T}_{I,M} - \theta^T)\mathbf{X}\big| \\
&\leq \ell(C+1)\|\theta^\star_{I,M} - \theta\|_1
\end{aligned}
$$

leads to

$$\left[\int \left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right) \mathrm{d}\rho^1_{I,M,\eta}(\theta)\right]^2 \leq \ell^2(C+1)^2 \left[\int \|\theta^\star_{I,M} - \theta\|_1 \mathrm{d}\rho^1_{I,M,\eta}(\theta)\right]^2.$$

Consequently,

$$\left[\int \left(f^\star_{I,M}(\theta^{\star T}_{I,M}\mathbf{X}) - f^\star_{I,M}(\theta^T\mathbf{X})\right) \mathrm{d}\rho^1_{I,M,\eta}(\theta)\right]^2 \leq \ell^2(C+1)^2\eta^2,$$

and therefore

$$\begin{aligned}\mathbf{D} &\leq 2\ell(C+1)\eta\sqrt{R(0,0)/2} \\ &\leq \sqrt{2}\ell(C+1)\eta\sqrt{C^2+\sigma^2}.\end{aligned}$$

Thus, taking $\eta = \gamma = 1/n$ and putting all the pieces together, we obtain

$$\mathbf{A}+\mathbf{B}+\mathbf{C}+\mathbf{D}+\mathbf{E} \leq \frac{\Xi_1}{n},$$

where Ξ_1 is a positive constant, depending on C, σ and ℓ. Combining this inequality with (12)-(14) yields, with probability larger than $1-2\delta$,

$$\begin{aligned}&R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \\ &\quad\leq \frac{1}{1 - \frac{\lambda[(2C+1)^2+4\sigma^2]}{n-w\lambda}} \inf_{\substack{I\subset\{1,\dots,p\} \\ 1\leq M\leq n}} \left\{\left(1 + \frac{\lambda\left[(2C+1)^2+4\sigma^2\right]}{n-w\lambda}\right)\left(R(\theta^\star_{I,M}, f^\star_{I,M})\right.\right. \\ &\qquad\left.\left. - R(\theta^\star, f^\star) + \frac{\Xi_1}{n}\right) + 2\,\frac{M\log(10(C+1)n) + |I|\log(40epn) + \log\left(\frac{1}{\delta}\right)}{\lambda}\right\}.\end{aligned}$$

Choosing finally

$$\lambda = \frac{n}{w + 2\left[(2C+1)^2+4\sigma^2\right]},$$

we obtain that there exists a positive constant Ξ_2, function of L, C, σ and ℓ such that, with probability at least $1-2\delta$,

$$\begin{aligned}R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi_2 \inf_{\substack{I\subset\{1,\dots,p\} \\ 1\leq M\leq n}} \left\{R(\theta^\star_{I,M}, f^\star_{I,M}) - R(\theta^\star, f^\star)\right. \\ \left. + \frac{M\log(10Cn) + |I|\log(40epn) + \log\left(\frac{1}{\delta}\right)}{n}\right\}.\end{aligned}$$

This concludes the proof of Theorem 2.

4.3 Proof of Corollary 4

We already know, by Theorem 2, that with probability at least $1-\delta$,

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi \inf_{\substack{I \subset \{1,\dots,p\} \\ 1 \leq M \leq n}} \left\{ R(\theta^\star_{I,M}, f^\star_{I,M}) - R(\theta^\star, f^\star) + \frac{M \log(Cn) + |I| \log(pn) + \log\left(\frac{2}{\delta}\right)}{n} \right\}.$$

By definition, for all $(\theta, f) \in \mathcal{S}^p_{1,+}(I) \times \mathcal{F}_M(C)$,

$$R(\theta^\star_{I,M}, f^\star_{I,M}) \leq R(\theta, f).$$

In particular, if $I^\star$ is such that $\theta^\star \in \mathcal{S}^p_{1,+}(I^\star)$, then

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi \inf_{\substack{1 \leq M \leq n \\ f \in \mathcal{F}_M(C)}} \left\{ R(\theta^\star, f) - R(\theta^\star, f^\star) + \frac{M \log(Cn) + |I^\star| \log(pn) + \log\left(\frac{2}{\delta}\right)}{n} \right\}. \tag{15}$$

Observe that, for any $f \in \mathcal{F}_M(C)$,

$$\begin{aligned} R(\theta^\star, f) - R(\theta^\star, f^\star) &= \int_{\mathbb{R}^p} \left[f\left(\theta^{\star T} \mathbf{x}\right) - f^\star\left(\theta^{\star T} \mathbf{x}\right) \right]^2 \mathrm{d}\mathbf{P}(\mathbf{x}, y) \\ &\leq B^2 \int_{-1}^{1} \left[f(t) - f^\star(t) \right]^2 \mathrm{d}t. \end{aligned}$$

Since $f^\star \in L_2([-1,1])$, we may write

$$f^\star = \sum_{j=1}^{\infty} \beta^\star_j \varphi_j$$

and apply (15) with

$$f = \sum_{j=1}^{M} \beta^\star_j \varphi_j.$$

In order to do so, we just need to check that $f \in \mathcal{F}_M(C)$, that is

$$\sum_{j=1}^{M} j |\beta^\star_j| \leq C.$$

But, by the Cauchy-Schwarz inequality,

$$\begin{aligned} \sum_{j=1}^{M} j |\beta^\star_j| &= \sum_{j=1}^{M} j^k |\beta^\star_j| j^{1-k} \\ &\leq \sqrt{\sum_{j=1}^{M} j^{2k} (\beta^\star_j)^2} \sqrt{\sum_{j=1}^{M} j^{2-2k}}. \end{aligned}$$

Thus,

$$\begin{aligned}\sum_{j=1}^{M} j|\beta_j^\star| &\leq \frac{\pi}{\sqrt{6}}\sqrt{\sum_{j=1}^{M} j^{2k}(\beta_j^\star)^2} \\ &\quad \text{(since, by assumption, } k \geq 2) \\ &\leq C \\ &\quad \text{(since } f^\star \in \mathcal{W}(k, 6C^2/\pi^2)).\end{aligned}$$

Next, with this choice of f,

$$\int_{-1}^{1} [f(t) - f^\star(t)]^2 \, \mathrm{d}t \leq \Lambda M^{-2k}$$

for some positive constant Λ depending only on k and C (see, for instance, Tsybakov, 2009). Therefore, inequality (15) leads to

$$R(\hat{\theta}_\lambda, \hat{f}_\lambda) - R(\theta^\star, f^\star) \leq \Xi \inf_{1 \leq M \leq n} \left\{ \Lambda M^{-2k} + \frac{M \log(Cn) + |I^\star| \log(pn) + \log\left(\frac{2}{\delta}\right)}{n} \right\}. \qquad (16)$$

Letting $\lceil . \rceil$ be the ceiling function and choosing $M = \lceil (n/\log(Cn))^{\frac{1}{2\beta+1}} \rceil$ in (16) concludes the proof.

4.4 Some Technical Lemmas

Lemma 9 *For any subset I of $\{1, \ldots, p\}$, any positive integer $M \leq n$ and any $\eta \in]0, 1/n]$, let the probability measure $\rho^1_{I,M,\eta}$ be defined by*

$$\frac{\mathrm{d}\rho^1_{I,M,\eta}}{\mathrm{d}\mu_I}(\theta) \propto \mathbf{1}_{[\|\theta - \theta^\star_{I,M}\|_1 \leq \eta]}.$$

Then

$$\mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) \leq (|I| - 1) \log\left(\max\left[|I|, \frac{4}{\eta} \right] \right).$$

Proof For simplicity, we assume that $I = \{1, \ldots, |I|\}$. Up to a permutation of the coordinates, the proof remains valid for any subset I of $\{1, \ldots, p\}$. Still for simplicity, we let $\tilde{\theta}$ denote $\theta^\star_{I,M}$. By a symmetry argument, it can be assumed that $\tilde{\theta}$ has nonnegative coordinates—this just means that $\tilde{\theta}$ is arbitrarily fixed in one of the $2^{|I|-1}$ faces of $\mathcal{S}^p_{1,+}(I)$. We denote by $\mathcal{FA}$ this face and note that

$$\mathcal{FA} = \left\{ \theta \in (\mathbb{R}_+)^{|I|} \times \{0\}^{p-|I|} : \sum_{j=1}^{|I|} \theta_j = 1 \right\}.$$

Finally, without loss of generality, we suppose that the largest coordinate in $\tilde{\theta}$ is $\tilde{\theta}_1$, and let χ be the uniform probability measure on $\mathcal{FA}$, defined by

$$\frac{\mathrm{d}\chi}{\mathrm{d}\mu_I}(\theta) = 2^{|I|-1} \mathbf{1}_{[\theta \in \mathcal{FA}]}.$$

Set $u = \min(1/|I|, \eta/2)$, and let

$$\begin{aligned} T_2 &= (\tilde{\theta}_1 - u, \tilde{\theta}_2 + u, \tilde{\theta}_3, \ldots, \tilde{\theta}_{|I|}, 0, \ldots, 0), \\ T_3 &= (\tilde{\theta}_1 - u, \tilde{\theta}_2, \tilde{\theta}_3 + u, \ldots, \tilde{\theta}_{|I|}, 0, \ldots, 0), \\ &\vdots \qquad\qquad\qquad\qquad \vdots \\ T_{|I|} &= (\tilde{\theta}_1 - u, \tilde{\theta}_2, \tilde{\theta}_3, \ldots, \tilde{\theta}_{|I|} + u, 0, \ldots, 0). \end{aligned}$$

Note that $u \leq 1/|I| \leq \tilde{\theta}_1$. Therefore, for each j, all the coordinates of T_j are nonnegative. Obviously $\|T_j\|_1 = 1$, so that, for all j, $T_j \in \mathcal{FA}$. Denoting by K the convex hull of the set $\{\tilde{\theta}, T_2, \ldots, T_{|I|}\}$, we also have $K \subset \mathcal{FA}$. Next, observe that $\|T_j - \tilde{\theta}\|_1 = 2u \leq \eta$, which implies $K \subset \{\theta \in \mathbb{R}^p : \|\theta - \tilde{\theta}\|_1 \leq \eta\}$.

Clearly,

$$\begin{aligned} \mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) &= \log \left(\frac{1}{\int \mathbf{1}_{[\|\theta - \theta^\star_{I,M}\|_1 \leq \eta]} \mathrm{d}\mu_I(\theta)} \right) \\ &\leq \log \left(\frac{1}{\int \mathbf{1}_{[\theta \in \mathcal{FA}]} \mathbf{1}_{[\|\theta - \theta^\star_{I,M}\|_1 \leq \eta]} \mathrm{d}\mu_I(\theta)} \right). \end{aligned}$$

Thus,

$$\begin{aligned} \mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) &\leq \log \left(\frac{2^{|I|-1}}{\int \mathbf{1}_{[\|\theta - \theta^\star_{I,M}\|_1 \leq \eta]} \mathrm{d}\chi(\theta)} \right) \\ &\leq \log \left(\frac{2^{|I|-1}}{\int \mathbf{1}_{[\theta \in K]} \mathrm{d}\chi(\theta)} \right). \end{aligned}$$

Observe that K is homothetic to $\mathcal{FA}$, by a factor of u. This means that

$$\int \mathbf{1}_{[\theta \in K]} \mathrm{d}\chi(\theta) = u^{|I|-1}.$$

Consequently, we obtain

$$\mathcal{K}(\rho^1_{I,M,\eta}, \mu_I) \leq \log \left(\left(\frac{2}{u} \right)^{|I|-1} \right) \leq (|I| - 1) \log \left(\max \left[|I|, \frac{4}{\eta} \right] \right).$$

■

Lemma 10 *For any subset I of $\{1, \ldots, p\}$, any positive integer $M \leq n$ and any $\gamma \in]0, 1/n]$, let the probability measure $\rho^2_{I,M,\gamma}$ be defined by*

$$\frac{\mathrm{d}\rho^2_{I,M,\gamma}}{\mathrm{d}\nu_M}(f) \propto \mathbf{1}_{[\|f - f^\star_{I,M}\|_M \leq \gamma]}$$

where, for $f = \sum_{j=1}^{M} \beta_j \varphi_j \in \mathcal{F}_M(C+1)$, *we put*

$$\|f\|_M = \sum_{j=1}^{M} j|\beta_j|.$$

Then

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = M \log\left(\frac{C+1}{\gamma}\right).$$

Proof Observe that

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = \int \log\left(\frac{\mathrm{d}\rho^2_{I,M,\gamma}}{\mathrm{d}\nu_M}(f)\right) \mathrm{d}\rho^2_{I,M,\gamma}(f).$$

Now,

$$\frac{\mathrm{d}\rho^2_{I,M,\gamma}}{\mathrm{d}\nu_M}(f) = \frac{\mathbf{1}_{[\|f-f^\star_{I,M}\|_M \le \gamma]}(f)}{\zeta},$$

where $\zeta = \int \mathbf{1}_{[\|f-f^\star_{I,M}\|_M \le \gamma]}(f)\mathrm{d}\nu_M(f)$. It easily follows, using the fact that the support of $\rho^2_{I,M,\gamma}$ is included in the set $\{f \in \mathcal{F}_M(C+1) : \|f - f^\star_{I,M}\| \le \gamma\}$, that

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = \log(1/\zeta).$$

Note that

$$\begin{aligned}\zeta &= \int \mathbf{1}_{[\|f-f^\star_{I,M}\|_M \le \gamma]}(f)\mathrm{d}\nu_M(f) \\ &= \frac{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j - (\beta^\star_{I,M})_j| \le \gamma]}(\beta)\mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]}(\beta)\mathrm{d}\beta}{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]}(\beta)\mathrm{d}\beta},\end{aligned}$$

where the second equality is true since ν_M is (the image of) the uniform probability measure on $\{\beta \in \mathbb{R}^M : \sum_{j=1}^M j|\beta_j| \le C+1\}$. This implies

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = \log\left(\frac{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]}(\beta)\mathrm{d}\beta}{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j - (\beta^\star_{I,M})_j| \le \gamma]}(\beta)\mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]}(\beta)\mathrm{d}\beta}\right).$$

By the triangle inequality,

$$\sum_{j=1}^{M} j|\beta_j| \le \sum_{j=1}^{M} j\left|\beta_j - (\beta^\star_{I,M})_j\right| + \sum_{j=1}^{M} j\left|(\beta^\star_{I,M})_j\right|.$$

Since $f^\star_{I,M} \in \mathcal{F}_M(C)$, we have $\sum_{j=1}^M j|(\beta^\star_{I,M})_j| \le C$, so that

$$\mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]} \ge \mathbf{1}_{[\sum_{j=1}^M j|\beta_j - (\beta^\star_{I,M})_j| \le \gamma]}$$

as soon as $\gamma \le 1$. We conclude that

$$\mathcal{K}(\rho^2_{I,M,\gamma}, \nu_M) = \log\left(\frac{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j| \le C+1]}\mathrm{d}\beta}{\int \mathbf{1}_{[\sum_{j=1}^M j|\beta_j - (\beta^\star_{I,M})_j| \le \gamma]}\mathrm{d}\beta}\right) = M \log\left(\frac{C+1}{\gamma}\right).$$

■

5. Annex: Description of the MCMC Algorithm

This annex is intended to make thoroughly clear the specification of the proposal conditional densities k_1 and k_2 introduced in Section 3.

5.1 Notation

In order to provide explicit formulas for the conditional densities $k_1((\tau,h)|(\theta,f))$ and $k_2((\tau,h)|(\theta,f))$, we first set

$$f = \sum_{j=1}^{m_f} \beta_{f,j}\varphi_j \quad \text{and} \quad h = \sum_{j=1}^{m_h} \beta_{h,j}\varphi_j,$$

where it is recalled that $\{\varphi_j\}_{j=1}^{\infty}$ denotes the (non-normalized) trigonometric system. We let I (respectively, J) be the set of nonzero coordinates of the vector θ (respectively, τ), and denote finally by θ_I (respectively, τ_J) the vector of dimension $|I|$ (respectively, $|J|$) which contains the nonzero coordinates of θ (respectively, $|\tau|$). Recall that all densities are defined with respect to the prior π, which is made explicit in Section 2.2.

For a generic $h \in \mathcal{F}_{m_h}(C+1)$, given $\tau \in \mathcal{S}_{1,+}^p$ and $s>0$, we let the density $\mathrm{dens}_s(h|\tau,m_h)$ with respect to π be defined by

$$\mathrm{dens}_s(h|\tau,m_h) \propto \exp\left[-\frac{1}{2s^2}\sum_{j=1}^{m_h}\left(\beta_{h,j}-\tilde{\beta}_j(\tau,m_h)\right)^2\right]\mathbf{1}\left[\sum_{j=1}^{m_h} j|\beta_{h,j}| \le C+1\right],$$

where the $\tilde{\beta}_j(\tau,m_h)$ are the empirical least square coefficients given by

$$\left\{\tilde{\beta}_j(\tau,m_h)\right\}_{j=1,\ldots,m_h} \in \arg\min_{b\in\mathbb{R}^{m_h}} \sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{m_h} b_j\varphi_j(\tau^T\mathbf{X}_i)\right)^2.$$

In the experiments, we fixed $s=0.1$. Note that simulating with respect to $\mathrm{dens}_s(h|\tau,m_h)$ is an easy task, since one just needs to compute a least square estimate and then draw from a truncated Gaussian distribution.

5.2 Description of k_1

We take

$$\begin{aligned} k_1(\cdot|(\theta,f)) = {} & \frac{2k_{1,=}(\cdot|(\theta,f)) + k_{1,+}(\cdot|(\theta,f))}{3}\mathbf{1}_{[|I|=1]} \\ & + \frac{k_{1,-}(\cdot|(\theta,f)) + 2k_{1,=}(\cdot|(\theta,f)) + k_{1,+}(\cdot|(\theta,f))}{4}\mathbf{1}_{[1<|I|<p]} \\ & + \frac{k_{1,-}(\cdot|(\theta,f)) + 2k_{1,=}(\cdot|(\theta,f))}{3}\mathbf{1}_{[|I|=p]}. \end{aligned}$$

Roughly, the idea is that $k_{1,-}$ tries to remove one component in θ, $k_{1,=}$ keeps the same number of components, whereas $k_{1,+}$ adds one component. The density $k_{1,=}$ takes the form

$$k_{1,=}((\tau,h)|(\theta,f)) = k_{1,=}(\tau|\theta)\mathrm{dens}_s(h|\tau,m_f).$$

The density $k_{1,=}(.|\theta)$ is the density of τ when $J = I$ and

$$\tau_I = \frac{\theta_I + E}{\|\theta_I + E\|_1} \mathrm{sgn}\left((\theta_I + E)_{j(\theta_I+E)}\right),$$

where $E = (E_1, \ldots, E_{|I|})$ and the E_i are independent random variables uniformly distributed in $[-\delta, \delta]$. Throughout, the value of δ was fixed at 0.5. It is noteworthy that when we change the parameter from θ to τ, then we also change the function from f to h. Thus, with this procedure, the link function h is more "adapted" to τ and the subsequent move is more likely to be accepted in the Hastings-Metropolis algorithm.

In the case where we are to remove one component, $k_{1,-}$ is given by

$$k_{1,-}((\tau,h)|(\theta,f)) = \sum_{j\in I} c_j \mathbf{1}_{[\tau=\theta_{-j}]} \mathrm{dens}_s(h|\tau, m_f),$$

where θ_{-j} is just obtained from θ by setting the j-th component to 0 and by renormalizing the parameter in order to have $\|\theta_{-j}\|_1 = 1$. We set

$$c_j = \frac{\exp(-|\theta_j|)\,\mathbf{1}_{[|\theta_j|<\delta]}}{\sum_{\ell\in I}\exp(-|\theta_\ell|)\,\mathbf{1}_{[|\theta_\ell|<\delta]}}.$$

The idea is that smaller components are more likely to be removed than larger ones. Finally, the density $k_{1,+}$ takes the form

$$k_{1,+}((\tau,h)|(\theta,f)) = \sum_{j\notin I} c'_j \mathbf{1}_{[\tau_{-j}=\theta]} \frac{\mathbf{1}_{[|\tau_j|<\delta]}}{2\delta} \mathrm{dens}_s(h|\tau, m_f).$$

We set

$$c'_j = \frac{\exp\left(\left|\sum_{i=1}^n \left(Y_i - f(\theta^T \mathbf{X}_i)\right)(\mathbf{X}_i)_j\right|\right)}{\sum_{\ell\notin I}\exp\left(\left|\sum_{i=1}^n (Y_i - f(\theta^T\mathbf{X}_i))(\mathbf{X}_i)_\ell\right|\right)}$$

where $(\mathbf{X}_i)_j$ denotes the j-th component of $\mathbf{X}_i$. In words, the idea is that a new nonzero coordinate in θ is more likely to be interesting in the model if the corresponding feature is correlated with the current residual.

5.3 Description of k_2

In the same spirit, we let the conditional density k_2 be defined by

$$\begin{aligned} k_2(\cdot|(\theta,f)) = {} & \frac{2k_{2,=}(\cdot|(\theta,f)) + k_{2,+}(\cdot|(\theta,f))}{3}\mathbf{1}_{[m_f=1]} \\ & + \frac{k_{2,-}(\cdot|(\theta,f)) + 2k_{2,=}(\cdot|(\theta,f)) + k_{2,+}(\cdot|(\theta,f))}{4}\mathbf{1}_{[1<m_f<n]} \\ & + \frac{k_{2,-}(\cdot|(\theta,f)) + 2k_{2,=}(\cdot|(\theta,f))}{3}\mathbf{1}_{[m_f=n]}. \end{aligned}$$

We choose

$$k_{2,=}((\tau,h)|(\theta,f)) = \mathbf{1}_{[\tau=\theta]}\mathrm{dens}_s(h|\tau, m_f)$$

and

$$k_{2,+}((\tau,h)|(\theta,f)) = \mathbf{1}_{[\tau=\theta]}\mathrm{dens}_s(h|\tau, m_f+1).$$

With this choice, $m_h = m_f + 1$, which means that the proposal density tries to add one coefficient in the expansion of h, while leaving θ unchanged. Finally

$$k_{2,-}\left((\tau,h)|(\theta,f)\right) = \mathbf{1}_{[\tau=\theta]}\mathrm{dens}_s(h|\tau,m_f-1),$$

and the proposal tries to remove one coefficient in h.

Acknowledgments

The authors thank four referees for valuable comments and insightful suggestions, which lead to a substantial improvement of the paper. They also thank John O'Quigley for his careful reading of the article. They would like to acknowledge support for this project from the French National Research Agency under grants ANR-09-BLAN-0128 "PARCIMONIE" and ANR-09-BLAN-0051-02 "CLARA", and from the INRIA project "CLASSIC" hosted by Ecole Normale Supérieure and CNRS.

References

P. Alquier. PAC-Bayesian bounds for randomized empirical risk minimizers. *Mathematical Methods of Statistics*, 17:279–304, 2008.

P. Alquier and K. Lounici. PAC-Bayesian bounds for sparse regression estimation with exponential weights. *Electronic Journal of Statistics*, 5:127–145, 2011.

A. Antoniadis, G. Grégoire, and I.W. McKeague. Bayesian estimation in single-index models. *Statistica Sinica*, 14:1147–1164, 2004.

J.-Y. Audibert. Aggregated estimators and empirical complexity for least square regression. *Annales de l'Institut Henri Poincaré: Probability and Statistics*, 40:685–736, 2004.

J.-Y. Audibert and O. Catoni. Robust linear least squares regression. *The Annals of Statistics*, 39: 2766–2794, 2011.

R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37:1705–1732, 2009.

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 S. van de Geer. *Statistics for High-Dimensional Data*. Springer, New York, 2011.

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.

O. Catoni. *Statistical Learning Theory and Stochastic Optimization*. Springer, 2004.

O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Lecture Notes-Monograph Series*. IMS, 2007.

J.M. Chambers, W.S. Cleveland, B. Kleiner, and P.A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth & Brooks, Belmont, 1983.

X. Chen, C. Zou, and R.D. Cook. Coordinate-independent sparse sufficient dimension reduction and variable selection. *The Annals of Statistics*, 38:3696–3723, 2010.

A. Cohen, I. Daubechies, R. DeVore, G. Kerkyacharian, and D. Picard. Capturing ridge functions in high dimension from point queries. *Constructive Approximation*, 35:225–243, 2012.

P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47:547–553, 2009.

A.S. Dalalyan and A.B. Tsybakov. Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity. *Machine Learning*, 72:39–61, 2008.

A.S. Dalalyan and A.B. Tsybakov. Sparse regression learning by aggregation and Langevin Monte-Carlo. *Journal of Computer and System Sciences*, 78:1423–1443, 2012.

A.S. Dalalyan, A. Juditsky, and V. Spokoiny. A new algorithm for estimating the effective dimension-reduction subspace. *Journal of Machine Learning Research*, 9:1647–1678, 2008.

M. Delecroix, M. Hristache, and V. Patilea. On semiparametric M-estimation in single-index regression. *Journal of Statistical Planning and Inference*, 136:730–769, 2006.

S. Gaïffas and G. Lecué. Optimal rates and adaptation in the single-index model using aggregation. *Electronic Journal of Statistics*, 1:538–573, 2007.

P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, New York, 2002.

W. Härdle, P. Hall, and H. Ichimura. Optimal smoothing in single-index models. *The Annals of Statistics*, 21:157–178, 1993.

D. Jr. Harrison and D.L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102, 1978.

J.L. Horowitz. *Semiparametric Methods in Econometrics*. Springer, 1998.

H. Ichimura. Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. *Journal of Econometrics*, 58:71–120, 1993.

O. Lopez. Single-index regression models with right-censored responses. *Journal of Statistical Planning and Inference*, 139:1082–1097, 2009.

J.-M. Marin and C. Robert. *Bayesian Core: A Practical Approach to Computational Bayesian Analysis*. Springer, New York, 2007.

P. Massart. *Concentration Inequalities and Model Selection*. Springer, Berlin, 2007.

D.A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 230–234, New York, 1998. ACM.

P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1983.

E.A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

E.A. Nadaraya. Remarks on nonparametric estimates for density functions and regression curves. *Theory of Probability and its Applications*, 15:134–137, 1970.

J.R. Quinlan. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 236–243, Amherst, 1993. Morgan Kaufmann.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, 2008.

Y. Seldin, N. Cesa-Bianchi, F. Laviolette, P. Auer, J. Shawe-Taylor, and J. Peters. *PAC-Bayesian analysis of the exploration-exploitation trade-off*. arXiv:1105.4585, 2011.

J. Shawe-Taylor and R. Williamson. A PAC analysis of a Bayes estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 2–9, New York, 1997. ACM.

C.J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10:1040–1053, 1982.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

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.

H.B. Wang. Bayesian estimation and variable selection for single index models. *Computational Statistics and Data Analysis*, 53:2617–2627, 2009.

G.S. Watson. Smooth regression analysis. *Sankhyā Series A*, 26:359–372, 1964.

I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28:1797–1808, 1998.

I.-C. Yeh. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29:474–480, 2007.

Journal of Machine Learning Research 14 (2013) 281-301 Submitted 1/12; Revised 8/12; Published 1/13

Derivative Estimation with Local Polynomial Fitting

Kris De Brabanter KRIS.DEBRABANTER@ESAT.KULEUVEN.BE
Jos De Brabanter* JOS.DEBRABANTER@ESAT.KULEUVEN.BE
Bart De Moor* BART.DEMOOR@ESAT.KULEUVEN.BE
Department of Electrical Engineering SCD-SISTA
KU Leuven
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium

Irène Gijbels IRENE.GIJBELS@WIS.KULEUVEN.BE
Department of Mathematics & Leuven Statistics Research Centre (LStat)
KU Leuven
Celestijnenlaan 200B
B-3001 Leuven, Belgium

Editor: Xiaotong Shen

Abstract

We present a fully automated framework to estimate derivatives nonparametrically without estimating the regression function. Derivative estimation plays an important role in the exploration of structures in curves (jump detection and discontinuities), comparison of regression curves, analysis of human growth data, etc. Hence, the study of estimating derivatives is equally important as regression estimation itself. Via empirical derivatives we approximate the qth order derivative and create a new data set which can be smoothed by any nonparametric regression estimator. We derive L_1 and L_2 rates and establish consistency of the estimator. The new data sets created by this technique are no longer independent and identically distributed (i.i.d.) random variables anymore. As a consequence, automated model selection criteria (data-driven procedures) break down. Therefore, we propose a simple factor method, based on bimodal kernels, to effectively deal with correlated data in the local polynomial regression framework.

Keywords: nonparametric derivative estimation, model selection, empirical derivative, factor rule

1. Introduction

The next section describes previous methods and objectives for nonparametric derivative estimation. Also, a brief summary of local polynomial regression is given.

1.1 Previous Methods And Objectives

Ever since the introduction of nonparametric estimators for density estimation, regression, etc. in the mid 1950s and early 1960s, their popularity has increased over the years. Mainly, this is due to the fact that statisticians realized that pure parametric thinking in curve estimations often does not

*. Bart De Moor and Jos De Brabanter are with IBBT-KU Leuven Future Health Department, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. Jos De Brabanter is also with the Departement Industrieel Ingenieur, KaHo Sint Lieven (Associatie KU Leuven), G. Desmetstraat 1, B-9000 Gent, Belgium.

meet the need for flexibility in data analysis. Many of their properties have been rigorously investigated and are well understood, see, for example, Fan and Gijbels (1996), Györfi et al. (2002) and Tsybakov (2009). Although the importance of regression estimation is indisputable, sometimes the first or higher order derivatives of the regression function can be equally important. This is the case in the exploration of structures in curves (Chaudhuri and Marron, 1999; Gijbels and Goderniaux, 2004) (jump detection and discontinuities), inference of significant features in data, trend analysis in time series (Rondonotti et al., 2007), comparison of regression curves (Park and Kang, 2008), analysis of human growth data (Müller, 1988; Ramsay and Silverman, 2002), the characterization of submicroscopic nanoparticles from scattering data (Charnigo et al., 2007) and inferring chemical compositions. Also, estimation of derivatives of the regression function is required for plug-in bandwidth selection strategies (Wand and Jones, 1995) and in the construction of confidence intervals (Eubank and Speckman, 1993).

It would be tempting to differentiate the estimated nonparametric estimate $\hat{m}(x)$ w.r.t. the independent variable to obtain the first order derivative of the regression function. However, such a procedure can only work well if the original regression function is extremely well estimated. Otherwise, it can lead to wrong derivative estimates when the data is noisy. Therefore, it can be expected that straightforward differentiation of the regression estimate $\hat{m}(x)$ will result in an accumulation of errors which increase with the order of the derivative.

In the literature there are two main approaches to nonparametric derivative estimation: Regression/smoothing splines and local polynomial regression. In the context of derivative estimation, Stone (1985) has shown that spline derivative estimators can achieve the optimal L_2 rate of convergence. Asymptotic bias and variance properties and asymptotic normality have been established by Zhou and Wolfe (2000). In case of smoothing splines, Ramsay (1998) noted that choosing the smoothing parameter is tricky. He stated that data-driven methods are generally poor guides and some user intervention is nearly always required. In fact, Wahba and Wang (1990) demonstrated that the smoothing parameter for a smoothing spline depends on the integer q while minimizing $\sum_{i=1}^{n}(\hat{m}^{(q)}(x_i) - m^{(q)}(x_i))^2$. Jarrow et al. (2004) suggested an empirical bias bandwidth criterion to estimate the first derivative via semiparametric penalized splines.

Early works discussing kernel based derivative estimation include Gasser and Müller (1984) and Härdle and Gasser (1985). Müller et al. (1987) and Härdle (1990) proposed a generalized version of the cross-validation technique to estimate the first derivative via kernel smoothing using difference quotients. Their cross-validation technique is related to modified cross-validation for correlated errors proposed by Chu and Marron (1991). Although the use of difference quotients may be natural, their variances are proportional to n^2 in case of equispaced design. Therefore, this type of cross-validation will be spoiled due to the large variability. In order to improve on the previous methods, Müller et al. (1987) also proposed a factor method to estimate a derivative via kernel smoothing. A variant of the factor method was also used by Fan and Gijbels (1995).

In case of local polynomial regression (Fan and Gijbels, 1996), the estimation of the qth derivative is straightforward. One can estimate $m^{(q)}(x)$ via the intercept coefficient of the qth derivative (local slope) of the local polynomial being fitted at x, assuming that the degree p is larger or equal to q. Note that this estimate of the derivative is, in general, not equal to the qth derivative of the estimated regression function $\hat{m}(x)$. Asymptotic properties as well as asymptotic normality were established by Fan and Gijbels (1996). Strong uniform consistency properties were shown by Delecroix and Rosa (2007).

As mentioned before, two problems inherently present in nonparametric derivative estimation are the unavailability of the data for derivative estimation (only regression data is given) and bandwidth or smoothing selection. In what follows we investigate a new way to compute derivatives of the regression function given the data $(x_1,Y_1),\ldots,(x_n,Y_n)$. This procedure is based on the creation of a new data set via empirical derivatives. A minor drawback of this approach is the fact the data are correlated and hence poses a threat to classical bandwidth selection methods. In order to deal with correlated data we extend our previous work (De Brabanter et al., 2011) and derive a factor method based on bimodal kernels to estimate the derivatives of the unknown regression function.

This paper is organized as follows. Next, we give a short introduction to local polynomial fitting. Section 2 illustrates the principle of empirical first order derivatives and their use within the local polynomial regression framework. We derive bias and variance of empirical first order derivatives and establish pointwise consistency. Further, the behavior at the boundaries of empirical first order derivatives is described. Section 3 generalizes the idea of empirical first order derivatives to higher order derivatives. Section 4 discusses the problem of bandwidth selection in the presence of correlated data. In Section 5 we conduct a Monte Carlo experiment to compare the proposed method with two often used methods for derivative estimation. Finally, Section 6 states the conclusions.

1.2 Local Polynomial Regression

Consider the bivariate data $(x_1,Y_1),\ldots,(x_n,Y_n)$ which form an independent and identically distributed (i.i.d) sample from a population (x,Y) where x belongs to $\mathcal{X}\subseteq\mathbb{R}$ and $Y\in\mathbb{R}$. If $\mathcal{X}$ denotes the closed real interval $[a,b]$ then $x_i = a+(i-1)(b-a)/(n-1)$. Denote by $m(x)=\mathbf{E}[Y]$ the regression function. The data is regarded to be generated from the model

$$Y = m(x)+e, \tag{1}$$

where $\mathbf{E}[e]=0$, $\mathbf{Var}[e]=\sigma^2<\infty$, x and e are independent and m is twice continuously differentiable on $\mathcal{X}$. Suppose that $(p+1)^{\text{th}}$ derivative of m at the point x_0 exists. Then, the unknown regression function m can be locally approximated by a polynomial of order p. A Taylor expansion yields, for x in a neighborhood of x_0,

$$m(x)\approx\sum_{j=0}^{p}\frac{m^{(j)}(x_0)}{j!}(x-x_0)^j\equiv\sum_{j=0}^{p}\beta_j(x-x_0)^j. \tag{2}$$

This polynomial is fitted locally by the following weighted least squares regression problem:

$$\min_{\beta_j\in\mathbb{R}}\sum_{i=1}^{n}\Big\{Y_i-\sum_{j=0}^{p}\beta_j\,(x_i-x_0)^j\Big\}^2K_h(x_i-x_0), \tag{3}$$

where β_j are the solutions to the weighted least squares problem, h is the bandwidth controlling the size of the local neighborhood and $K_h(\cdot)=K(\cdot/h)/h$ with K a kernel function assigning weights to each point. From the Taylor expansion (2) it is clear that $\hat{m}^{(q)}(x_0)=q!\hat{\beta}_q$ is an estimator for $m^{(q)}(x_0)$, $q=0,1,\ldots,p$. For local polynomial fitting $p-q$ should be taken to be odd as shown in Ruppert and Wand (1994) and Fan and Gijbels (1996). In matrix notation (3) can be written as:

$$\min_{\beta}\{(\mathbf{y}-X\beta)^T\mathbf{W}(\mathbf{y}-X\beta)\},$$

where $\mathbf{y} = (Y_1, \ldots, Y_n)^T$, $\beta = (\beta_0, \ldots, \beta_p)^T$ and

$$\mathbf{X} = \begin{pmatrix} 1 & (x_1 - x_0) & \cdots & (x_1 - x_0)^p \\ \vdots & \vdots & & \vdots \\ 1 & (x_n - x_0) & \cdots & (x_n - x_0)^p \end{pmatrix},$$

and $\mathbf{W}$ the $n \times n$ diagonal matrix of weights

$$\mathbf{W} = \text{diag}\{K_h(x_i - x_0)\}.$$

The solution vector is given by least squares theory and yields

$$\hat{\beta} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{y}.$$

2. Derivative Estimation

In this section we first illustrate the principle of empirical first order derivatives and how they can be used within the local polynomial regression framework to estimate first order derivatives of the unknown regression function.

2.1 Empirical Derivatives And Its Properties

Given a local polynomial regression estimate (3), it would be tempting to differentiate it w.r.t. the independent variable. Such a procedure can lead to wrong derivative estimates when the data is noisy and will deteriorate quickly when calculating higher order derivatives. A possible solution to avoid this problem is by using the first order difference quotient

$$Y_i^{(1)} = \frac{Y_i - Y_{i-1}}{x_i - x_{i-1}}$$

as a noise corrupted version of $m'(x_i)$ where the superscript (1) signifies that $\hat{Y}_i^{(1)}$ is a noise corrupted version of the first (true) derivative. Such an approach has been used by Müller et al. (1987) and Härdle (1990) to estimate first order derivatives via kernel smoothing. Such an approach produces a very noisy estimate of the derivative which is of the order $O(n^2)$ and as a result it will be difficult to estimate the derivative function. For equispaced design yields

$$\mathbf{Var}(Y_i^{(1)}) = \frac{1}{(x_i - x_{i-1})^2}(\mathbf{Var}(Y_i) + \mathbf{Var}(Y_{i-1})) = \frac{2\sigma^2}{(x_i - x_{i-1})^2} = \frac{2\sigma^2 (n-1)^2}{d(\mathcal{X})^2},$$

where $d(\mathcal{X}) := \sup \mathcal{X} - \inf \mathcal{X}$. In order to reduce the variance we use a variance-reducing linear combination of symmetric (about i) difference quotients

$$Y_i^{(1)} = Y^{(1)}(x_i) = \sum_{j=1}^{k} w_j \cdot \left(\frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}} \right), \tag{4}$$

where the weights $w_1, \ldots, w_k$ sum up to one. The linear combination (4) is valid for $k+1 \leq i \leq n-k$ and hence $k \leq (n-1)/2$. For $2 \leq i \leq k$ or $n-k+1 \leq i \leq n-1$ we define $Y_i^{(1)}$ by replacing $\sum_{j=1}^{k}$ in (4) by $\sum_{j=1}^{k(i)}$ where $k(i) = \min\{i-1, n-i\}$ and replacing $w_1, \ldots, w_{k(i)}$ by $w_1 / \sum_{j=1}^{k(i)} w_j, \ldots, w_{k(i)} / \sum_{j=1}^{k(i)} w_j$.

Finally, for $i = 1$ and $i = n$ we define $Y_1^{(1)}$ and $Y_n^{(1)}$ to coincide with $Y_2^{(1)}$ and $Y_{n-1}^{(1)}$. The proportion of indices i falling between $k+1$ and $n-k$ approaches 1 as n increases, so this boundary issue becomes smaller as n becomes larger. Alternatively, one may just leave $Y_i^{(1)}$ undefined for indices i not falling between $k+1$ and $n-k$. This latter approach will be used in the remaining of the paper, except in Figure 1 where we want to illustrate the boundary issues.

Linear combinations such as (4) are frequently used in finite element theory and are useful in the numerical solution of differential equations (Iserles, 1996). However, the weights used for solving differential equations are not appropriate here because of the random errors in model (1). Therefore, we need to optimize the weights so that minimum variance is attained. This result is stated in Proposition 1.

Proposition 1 *Assume model* (1) *holds with equispaced design and let* $\sum_{j=1}^{k} w_j = 1$. *Then, for* $k+1 \le i \le n-k$, *the weights*

$$w_j = \frac{6j^2}{k(k+1)(2k+1)}, \quad j = 1, \ldots, k$$

minimize the variance of $Y_i^{(1)}$ *in* (4).
Proof: see Appendix A. ■

Figure 1a displays the empirical first derivative for $k \in \{2,5,7,12\}$ generated from model (1) with $m(x) = \sqrt{x(1-x)}\sin((2.1\pi)/(x+0.05))$, $x \in [0.25,1]$ for 300 equispaced points and $e \sim \mathcal{N}(0, 0.1^2)$. For completeness the first order difference quotient is also shown. Even for a small k, it can be seen that the empirical first order derivatives are noise corrupted versions of the true derivative m'. In contrast, difference quotients produce an extreme noisy version of the true derivative (Figure 1b). Also, note the large amplitude of the signal constructed by difference quotients. When k is large, empirical first derivatives are biased near local extrema of the true derivative (see Figure 1f). Further, the boundary issues are clearly visible in Figure 1d through Figure 1f for $i \in [1, k+1] \cup [n-k, n]$.

The next two theorems give asymptotic results on the bias and variance and establish pointwise consistency of the empirical first order derivatives.

Theorem 2 *Assume model* (1) *holds with equispaced design and m is twice continuously differentiable on* $\mathcal{X} \subseteq \mathbb{R}$. *Further, assume that the second order derivative* $m^{(2)}$ *is finite on* $\mathcal{X}$. *Then the bias and variance of the empirical first order derivative, with weights assigned by Proposition 1, satisfy*

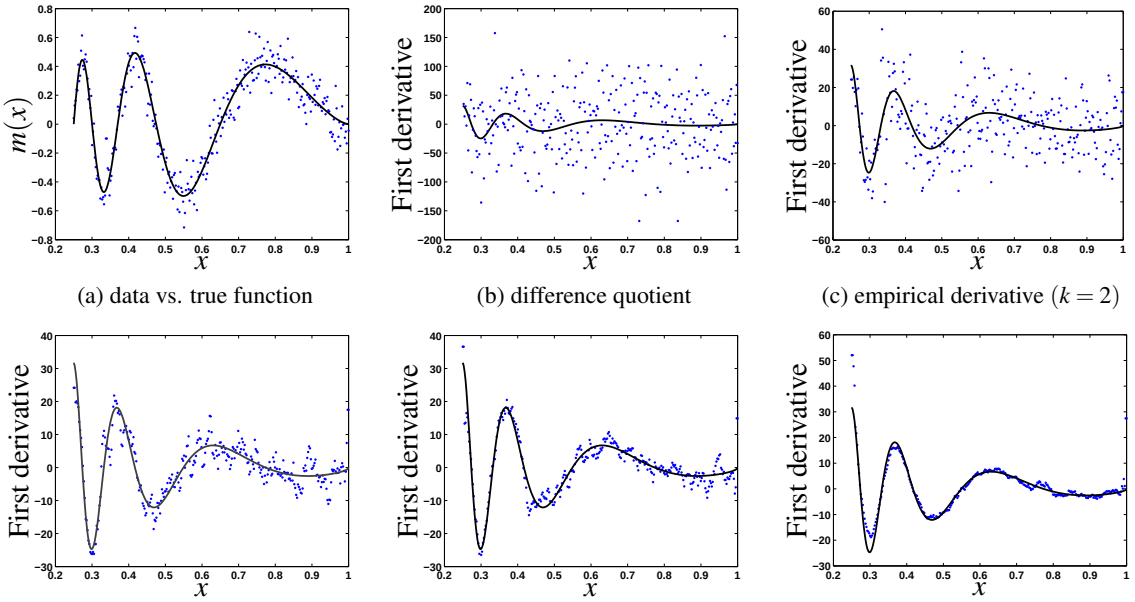
$$\text{bias}(Y_i^{(1)}) = O(n^{-1}k) \quad \textit{and} \quad \mathbf{Var}(Y_i^{(1)}) = O(n^2 k^{-3})$$

uniformly for $k+1 \le i \le n-k$.
Proof: see Appendix B. ■

Theorem 3 (Pointwise consistency) *Assume* $k \to \infty$ *as* $n \to \infty$ *such that* $nk^{-3/2} \to 0$ *and* $n^{-1}k \to 0$. *Further assume that m is twice continuously differentiable on* $\mathcal{X} \subseteq \mathbb{R}$. *Then, for the minimum variance weights given in Proposition 1, we have for any* $\varepsilon > 0$

$$\mathbf{P}\big(|Y_i^{(1)} - m'(x_i)| \ge \varepsilon\big) \to 0.$$

Proof: see Appendix C. ■



(a) data vs. true function (b) difference quotient (c) empirical derivative ($k = 2$)

(d) empirical derivative ($k = 5$) (e) empirical derivative ($k = 7$) (f) empirical derivative ($k = 12$)

Figure 1: (a) Simulated data set of size $n = 300$ equispaced points from model (1) with $m(x) = \sqrt{x(1-x)}\sin((2.1\pi)/(x+0.05))$ and $e \sim \mathcal{N}(0, 0.1^2)$; (b) first order difference quotients which are barely distinguishable from noise. As a reference, the true derivative is also displayed (full line); (c)-(f) empirical first derivatives for $k \in \{2,5,7,12\}$.

According to Theorem 2 and Theorem 3, the bias and variance of the empirical first order derivative tends to zero and $k \to \infty$ faster than $O(n^{2/3})$ but slower than $O(n)$. The optimal rate at which $k \to \infty$ such that the mean squared error (MSE) of the empirical first order derivatives will tend to zero at the fastest possible rate is a direct consequence of Theorem 2. This optimal L_2 rate is achieved for $k = O(n^{4/5})$ and consequently, the $\text{MSE}(Y_i^{(1)}) = \mathbf{E}(Y_i^{(1)} - m'(x_i))^2 = O(n^{-2/5} + n^{-1/5})$. Similar, one can also establish the rate of the mean absolute deviation (MAD) or L_1 rate of the estimator, that is, $\mathbf{E}\,|Y_i^{(1)} - m'(x_i)|$. By Jensen's inequality

$$\begin{aligned} \mathbf{E}\,|Y_i^{(1)} - m'(x_i)| &\leq \mathbf{E}\,|Y_i^{(1)} - \mathbf{E}(Y_i^{(1)})| + |\,\mathbf{E}(Y_i^{(1)}) - m'(x_i)| \\ &\leq \sqrt{\mathbf{Var}(Y_i^{(1)})} + \text{bias}(Y_i^{(1)}) = O(n^{-1/5}), \end{aligned}$$

for the optimal L_1 rate of $k = O(n^{4/5})$ (equal to the optimal L_2 rate). Under the same conditions as Theorem 3, it is easy to show that $\mathbf{E}\,|Y_i^{(1)} - m'(x_i)| \to 0$. Even though we know the optimal asymptotic order of k, the question still remains how to choose k in practice. In many data analyses, one would like to get a quick idea what the value of k should be. In such a case a rule of thumb can be very suitable. Such a rule can be somewhat crude but it possesses simplicity and is easily computable. In order to derive a suitable expression for the MSE, we start from the bias and variance expressions for the empirical derivatives. An upperbound for the MSE is given by (see also the proof

of Theorem 2)

$$\begin{aligned} \mathrm{MSE}(Y_i^{(1)}) &= \mathrm{bias}^2(Y_i^{(1)}) + \mathbf{Var}(Y_i^{(1)}) \\ &\leq \frac{9k^2(k+1)^2\mathcal{B}^2 d(\mathcal{X})^2}{16(n-1)^2(2k+1)^2} + \frac{3\sigma^2(n-1)^2}{k(k+1)(2k+1)d(\mathcal{X})^2}, \end{aligned} \tag{5}$$

where $\mathcal{B} = \sup_{x\in\mathcal{X}} |m^{(2)}(x)|$. Setting the derivative of (5) w.r.t. k to zero yields

$$3\mathcal{B}^2 d(\mathcal{X})^4 k^3 (1+k)^3 (1+2k+2k^2) = 8(1+8k+18k^2+12k^3)(n-1)^4\sigma^2. \tag{6}$$

Solving (6), with the constraint that $k > 0$, can be done by means of any root finding algorithm and will result in the value k for which the MSE is lowest. However, a much simpler rule of thumb and without much loss of accuracy is obtained by only considering the highest order terms yielding

$$k = \left(\frac{16\sigma^2}{\mathcal{B}^2 d(\mathcal{X})^4}\right)^{1/5} n^{4/5}.$$

The above quantity contains some unknown quantities and need to be estimated. The error variance σ^2 can be estimated by means of Hall's $\sqrt{n}$-consistent estimator (Hall et al., 1990)

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n-2}(0.809Y_i - 0.5Y_{i+1} - 0.309Y_{i+2})^2.$$

For the second unknown quantity $\mathcal{B}$ one can use the local polynomial regression estimate of order $p = 3$ leading to the following (rough) estimate of the second derivative $\hat{m}^{(2)}(x_0) = 2\hat{\beta}_2$ (see also Section 1). Consequently, a rule of thumb selector for k is given by

$$\hat{k} = \left(\frac{16\hat{\sigma}^2}{(\sup_{x_0\in\mathcal{X}} |\hat{m}^{(2)}(x_0)|)^2 d(\mathcal{X})^4}\right)^{1/5} n^{4/5}. \tag{7}$$

The result of the rule of thumb (7) is a value for k which is real. In practice we round the obtained k value closest to the next integer value. As an alternative, one could also consider cross-validation or complexity criteria in order to find an optimal value for k.

2.2 Behavior At The Boundaries

Recall that for the boundary region ($2 \leq i \leq k$ and $n-k+1 \leq i \leq n-1$) the weights in the derivative (4) and the range of the sum are slightly modified. Such a modification allows for an automatic bias correction at the boundaries. This can be seen as follows. Let the first $(q+1)$ derivatives of m be continuous on $\mathcal{X}$. Then a Taylor series of m in a neighborhood of x_i yields

$$m(x_{i+j}) = m(x_i) + \sum_{l=1}^{q}\frac{1}{l!}\left(\frac{jd(\mathcal{X})}{n-1}\right)^l m^{(l)}(x_i) + O\big((j/n)^{q+1}\big)$$

and

$$m(x_{i-j}) = m(x_i) + \sum_{l=1}^{q}\frac{1}{l!}\left(\frac{-jd(\mathcal{X})}{n-1}\right)^l m^{(l)}(x_i) + O\big((j/n)^{q+1}\big).$$

From the above series it follows that

$$\begin{aligned}\mathbf{E}(Y_i^{(1)}) &= \sum_{j=1}^{k} w_j \frac{m(x_{i+j}) - m(x_{i-j})}{x_{i+j} - x_{i-j}} \\ &= \frac{n-1}{2d(\mathcal{X})} \sum_{j=1}^{k} w_j \frac{\sum_{l=1}^{q} \frac{1}{l!} \left(\frac{jd(\mathcal{X})}{n-1}\right)^l m^{(l)}(x_i) - \sum_{l=1}^{q} \frac{1}{l!} \left(\frac{-jd(\mathcal{X})}{n-1}\right)^l m^{(l)}(x_i) + O\big((j/n)^{q+1}\big)}{j}.\end{aligned}$$

By noticing that all even orders of the derivative cancel out, the previous result can be written as

$$\begin{aligned}\mathbf{E}(Y_i^{(1)}) &= \frac{n-1}{2d(\mathcal{X})} \sum_{j=1}^{k} \frac{w_j}{j} \left[\frac{2jd(\mathcal{X})}{n-1} m'(x_i) + \sum_{l=3,5,\ldots}^{q} \frac{2}{l!} \left(\frac{jd(\mathcal{X})}{n-1}\right)^l m^{(l)}(x_i) + O\big((j/n)^{q+1}\big) \right] \\ &= m'(x_i) \sum_{j=1}^{k} w_j + \sum_{l=3,5,\ldots}^{q} m^{(l)}(x_i) \sum_{j=1}^{k} \frac{w_j}{l!} \frac{j^{l-1} d(\mathcal{X})^{l-1}}{(n-1)^{l-1}} + O\big((j/n)^q\big).\end{aligned}$$

For $2 \leq i \leq k$, the sum in the first term is not equal to 1. This immediately follows from the definition of the derivative in (4). Therefore, the length of the sum k has to be replaced with $k(i) = i-1$. Let $0 \leq \kappa = \sum_{j=1}^{k(i)} w_j < 1$ for $2 \leq i \leq k$. Then, the bias of the derivative (4) is given by

$$\text{bias}(Y_i^{(1)}) = (\kappa - 1) m'(x_i) + \sum_{l=3,5,\ldots}^{q} m^{(l)}(x_i) \sum_{j=1}^{k} \frac{w_j}{l!} \frac{j^{l-1} d(\mathcal{X})^{l-1}}{(n-1)^{l-1}} + O\big(n^{-q/5}\big),$$

where $\sum_{j=1}^{k} \frac{w_j}{l!} \frac{j^{l-1} d(\mathcal{X})^{l-1}}{(n-1)^{l-1}} = O(n^{-(l-1)/5})$ since $k = O(n^{4/5})$. However, in order to obtain an automatic bias correction at the boundaries, we can make $\kappa = 1$ by normalizing the sum leading to the following estimator

$$Y_i^{(1)} = \sum_{j=1}^{k(i)} \frac{w_j}{\sum_{j=1}^{k(i)} w_j} \left(\frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}} \right) \tag{8}$$

at the boundaries. Also notice that the bias at the boundaries is of the same order as in the interior.

Unfortunately, this bias correction comes at a prize, that is, increased variance at the boundaries. The variance of (8), for $k(i) = i-1$, is given by

$$\mathbf{Var}(Y_i^{(1)}) = \frac{\sigma^2 (n-1)^2}{2d(\mathcal{X})^2} \sum_{j=1}^{k(i)} \frac{w_j^2}{\left(\sum_{j=1}^{k(i)} w_j\right)^2} \frac{1}{j^2} = \frac{3\sigma^2 (n-1)^2}{d(\mathcal{X})^2} \frac{1}{i(i-1)(2i-1)}.$$

Then, at the boundary (for $2 \leq i \leq k$), it follows that an upper bound for the variance is given by

$$\mathbf{Var}(Y_i^{(1)}) \leq \frac{\sigma^2 (n-1)^2}{2d(\mathcal{X})^2}$$

and a lower bound by

$$\begin{aligned}\mathbf{Var}(Y_i^{(1)}) &\geq \frac{3\sigma^2 (n-1)^2}{d(\mathcal{X})^2} \frac{1}{k(k-1)(2k-1)} \\ &\geq \frac{3\sigma^2 (n-1)^2}{d(\mathcal{X})^2} \frac{1}{k(k+1)(2k+1)}.\end{aligned}$$

Hence, the variance will be largest (but limited) for $i = 2$ and will decrease for growing i till $i = k$. Also, from the last inequality it follows that variance at the boundaries will always be larger or equal than the variance of the interior. An analogue calculation shows the same result for $n-k+1 \leq i \leq n-1$ by setting $k(i) = n-i$.

3. Higher Order Empirical Derivatives

In this section, we generalize the idea of first order empirical derivatives to higher order derivatives. Let q denote the order of the derivative and assume further that $q \geq 2$, then higher order empirical derivatives can be defined inductively as

$$Y_i^{(l)} = \sum_{j=1}^{k_l} w_{j,l} \cdot \left(\frac{Y_{i+j}^{(l-1)} - Y_{i-j}^{(l-1)}}{x_{i+j} - x_{i-j}} \right) \qquad \text{with} \quad l \in \{2, \ldots, q\}, \tag{9}$$

where $k_1, k_2, \ldots, k_q$ are positive integers (not necessary equal), the weights at each level l sum up to one and $Y_i^{(0)} = Y_i$ by definition. As with the first order empirical derivative, a boundary issue arises with expression (9) when $i < \sum_{l=1}^{q} k_l + 1$ or $i > n - \sum_{l=1}^{q} k_l$. Similar to (4), a boundary correction can be used. Although, the qth order derivatives are linear in the weights at level q, they are not linear in the weights at all levels. As such, no simple formulas for variance minimizing weights exist. Fortunately, simple weight sequences exist which control the asymptotic bias and variance quite well assuming that $k_1, \ldots, k_q$ increase appropriately with n (see Theorem 4).

Theorem 4 *Assume model* (1) *holds with equispaced design and let* $\sum_{j=1}^{k_l} w_{j,l} = 1$. *Further assume that the first* $(q+1)$ *derivatives of* m *are continuous on the interval* $\mathcal{X}$. *Assume that there exist* $\lambda \in (0,1)$ *and* $c_l \in (0, \infty)$ *such that* $k_l n^{-\lambda} \to c_l$ *for* $n \to \infty$ *and* $l \in \{1, 2, \ldots, q\}$. *Further, assume that*

$$w_{j,1} = \frac{6j^2}{k_1(k_1+1)(2k_1+1)} \quad \textit{for} \quad j = 1, \ldots, k_1,$$

and

$$w_{j,l} = \frac{2j}{k_l(k_l+1)} \quad \textit{for} \quad j = 1, \ldots, k_l \quad \textit{and} \quad l \in \{2, \ldots, q\}.$$

Then the asymptotic bias and variance of the empirical q*th order derivative are given by*

$$\text{bias}(Y_i^{(q)}) = O(n^{\lambda - 1}) \quad \textit{and} \quad \mathbf{Var}(Y_i^{(q)}) = O(n^{2q - 2\lambda(q+1/2)})$$

uniformly for $\sum_{l=1}^{q} k_q + 1 < i < n - \sum_{l=1}^{q} k_q$.
Proof: see Appendix C. ■

An interesting consequence of Theorem 4 is that the order of the bias of the empirical derivative estimator does not depend on the order of the derivative q. The following two corollaries are a direct consequence of Theorem 4. Corollary 5 states that the L_2 rate of convergence (and L_1 rate) will be slower for increasing orders of derivatives q, that is, higher order derivatives are progressively more difficult to estimate. Corollary 5 suggests that the MSE of the qth order empirical derivative will tend to zero for $\lambda \in (\frac{2q}{2q+1}, 1)$ prescribing, for example, $k_q = O(n^{2(q+1)/(2q+3)})$. Similar results can be obtained for the MAD. Corollary 6 proves L_2 and L_1 consistency.

Corollary 5 *Under the assumptions of Theorem 4, for the weight sequences defined in Theorem 4, the asymptotic mean squared error and asymptotic mean absolute deviation are given by*

$$\mathbf{E}(Y_i^{(q)} - m^{(q)}(x_i))^2 = O(n^{2(\lambda-1)} + n^{2q-2\lambda(q+1/2)}) \quad \textit{and} \quad \mathbf{E}\,|Y_i^{(q)} - m^{(q)}(x_i)| = O(n^{\lambda-1} + n^{q-\lambda(q+1/2)}).$$

Corollary 6 *Under the assumptions of Theorem 4, for the weight sequences defined in Theorem 4 and* $\lambda \in (\frac{2q}{2q+1}, 1)$*, it follows that*

$$\mathbf{E}(Y_i^{(q)} - m^{(q)}(x_i))^2 \to 0 \quad \textit{and} \quad \mathbf{E}\,|Y_i^{(q)} - m^{(q)}(x_i)| \to 0, \quad n \to \infty.$$

4. Bandwidth Selection For Correlated Data

From (4), it is clear that for the newly generated data set the i.i.d. assumption is no longer valid since it is a weighted sum of differences of the original data set. In such cases, it is known that data-driven bandwidth selectors and plug-ins break down (Opsomer et al., 2001; De Brabanter et al., 2011). In this paper we extend the idea of De Brabanter et al. (2011) and develop a factor rule based on bimodal kernels to determine the bandwidth. They showed, under mild conditions on the kernel function and for equispaced design, that by using a kernel satisfying $K(0) = 0$ the correlation structure is removed without any prior knowledge about its structure. Further, they showed that bimodal kernels introduce extra bias and variance yielding in a slightly wiggly estimate. In what follows we develop a relation between the bandwidth of a unimodal kernel and the bandwidth of a bimodal kernel. Consequently, the estimate based on this bandwidth will be smoother than the one based on a bimodal kernel.

Assume the following model for the qth order derivative

$$Y^{(q)}(x) = m^{(q)}(x) + \varepsilon$$

and assume that m has two continuous derivatives. Further, let $\mathbf{Cov}(\varepsilon_i, \varepsilon_{i+l}) = \gamma_l < \infty$ for all l and assume that $\sum_{l=1}^{\infty} l|\gamma_l| < \infty$. Then, if $h \to \infty$ and $nh \to \infty$ as $n \to \infty$, the bandwidth h that minimizes the mean integrated squared error (MISE) of the local polynomial regression estimator (3) with p odd under correlation is given by (Simonoff, 1996; Fan and Gijbels, 1996)

$$\hat{h} = C_p(K) \left[\frac{(\sigma^2 + 2\sum_{l=1}^{\infty} \gamma_l)\, d(\mathcal{X})}{\int \{m^{(p+1)}(u)\}^2 \, du} \right]^{1/(2p+3)} n^{-1/(2p+3)}, \tag{10}$$

where

$$C_p(K) = \left[\frac{\{(p+1)!\}^2 \int K_p^{\star 2}(u)\, du}{2(p+1)\{\int u^{p+1} K_p^{\star}(u)\, du\}^2} \right]^{1/(2p+3)}$$

and $K_p^{\star}$ denotes the equivalent kernel defined as

$$K_p^{\star}(u) = (1\ 0\ \cdots\ 0) \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_p \\ \mu_1 & \mu_2 & \cdots & \mu_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_p & \mu_{p+1} & \cdots & \mu_{2p} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ u \\ \vdots \\ u^p \end{pmatrix} K(u),$$

with $\mu_j = \int u^j K(u)\,du$. Since the bandwidth h_b based on a symmetric bimodal kernel $\overline{K}$ has a similar expression as (10) for a unimodal kernel, one can express h as a function of h_b resulting into a factor method. It is easily verified that

$$\hat{h} = C_p(K,\overline{K})\hat{h}_b,$$

where

$$C_p(K,\overline{K}) = \left[\frac{\int K_p^{\star 2}(u)\,du\,\{\int u^{p+1}\overline{K}_p^{\star}(u)\,du\}^2}{\int \overline{K}_p^{\star 2}(u)\,du\,\{\int u^{p+1}K_p^{\star}(u)\,du\}^2}\right]^{1/(2p+3)}.$$

The factor $C_p(K,\overline{K})$ is easy to calculate and Table 1 lists some of these factors for different unimodal kernels and for various odd orders of polynomials p. We take $\overline{K}(u) = (2/\sqrt{\pi})u^2\exp(-u^2)$ as bimodal kernel.

p	Gaussian	Uniform	Epanechnikov	Triangular	Biweight	Triweight
1	1.16231	2.02248	2.57312	2.82673	3.04829	3.46148
3	1.01431	2.45923	2.83537	2.98821	3.17653	3.48541
5	0.94386	2.79605	3.09301	3.20760	3.36912	3.62470

Table 1: The factor $C_p(K,\overline{K})$ for different unimodal kernels and for various odd orders of polynomials p with $\overline{K}(u) = (2/\sqrt{\pi})u^2\exp(-u^2)$ as bimodal kernel.

5. Simulations

In what follows, we evaluate the proposed method for derivative estimation with several other methods used in the literature.

5.1 First Order Derivative Estimation

We evaluate the proposed method for derivative estimation with several other methods used in the literature, that is, via the local slope in local polynomial regression with $p = 3$ (*R* package `locpol` (Cabrera, 2009)) and penalized smoothing splines (*R* package `pspline` (Ramsey and Ripley, 2010)). For the latter we have used quintic splines (Newell and Einbeck, 2007) to estimate the first order derivative. All smoothing parameters were determined by weighted generalized cross-validation (WGCV$^{(q)}$) defined as

$$\text{WGCV}^{(q)} = \frac{1}{n}\sum_{i=1}^{n} s_i\left(\frac{Y_i^{(q)} - \hat{m}_n^{(q)}(x_i)}{1-\text{trace}(L)/n}\right)^2,$$

with $s_i = \mathbf{1}\{\sum_{l=1}^{q} k_l + 1 \le i \le n - \sum_{l=1}^{q} k_l\}$ and let L be the smoother matrix of the local polynomial regression estimate. The Gaussian kernel has been used for all kernel methods. The proposed method uses $\overline{K}(u) = (2/\sqrt{\pi})u^2\exp(-u^2)$ as bimodal kernel. The corresponding sets of bandwidths of the bimodal kernel h_b were $\{0.04, 0.045, \ldots, 0.095\}$ and k_1 was determined in each run by (7). Consider the following two functions

$$m(x) = \sin^2(2\pi x) + \log(4/3 + x) \quad \text{for} \quad x \in [-1,1] \tag{11}$$

and

$$m(x) = 32e^{-8(1-2x)^2}(1-2x) \quad \text{for} \quad x \in [0,1], \tag{12}$$

In a first simulation we show a typical result for the first order derivative ($q = 1$) of (11) and (12), its first order empirical derivative (see Figure 2). The data sets are of size $n = 1000$ and are generated from model (1) with $e \sim N(0,\sigma^2)$ for $\sigma = 0.03$ (regression function (11)) and $\sigma = 0.1$ (regression function (12)). To smooth the noisy derivative data we have chosen a local polynomial regression estimate of order $p = 3$. For the Monte Carlo study, we constructed data sets size with $n = 500$ and

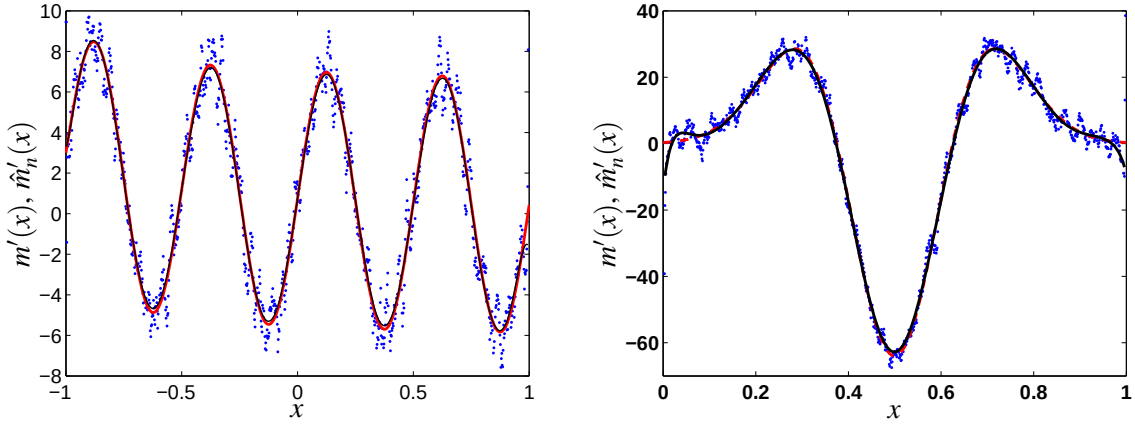


Figure 2: Illustration of the noisy empirical first order derivative (data points), smoothed empirical first order derivative based on a local polynomial regression estimate of order $p = 3$ (bold line) and true derivative (bold dashed line). (a) First order derivative of regression function (11) with $k_1 = 7$; (b) First order derivative of regression function (12) with $k_1 = 12$.

generated the function

$$m(x) = \sqrt{x(1-x)}\sin\left(\frac{2.1\pi}{x+0.05}\right) \quad \text{for} \quad x \in [0.25,1]$$

100 times according to model (1) with $e \sim N(0,\sigma^2)$ and $\sigma = 0.1$. As measure of comparison we chose the adjusted mean absolute error defined as

$$\text{MAEadjusted} = \frac{1}{481}\sum_{i=10}^{490} |\hat{m}'_n(x_i) - m'(x_i)|.$$

This criterion was chosen to ignore boundary effects in the estimation for the three methods. The result of the Monte Carlo study for (12) is given in Figure 3. From the Monte Carlo experiment, it is clear that all three methods yield similar results and no method supersedes the other.

5.2 Second Order Derivative Estimation

As before, all smoothing parameters were determined by weighted generalized cross-validation (WGCV$^{(q)}$) for $q = 2$. A typical result for the second order derivative ($q = 2$) of (11) and (12) and

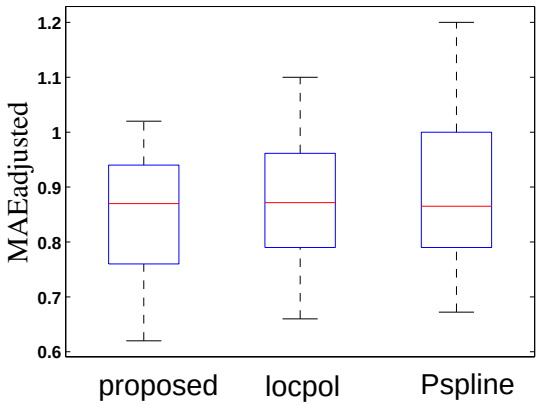


Figure 3: Result of the Monte Carlo study for the proposed method and two other well-known methods for first order derivative estimation.

its second order empirical derivative is shown in Figure 4. To smooth the noisy derivative data we have chosen a local polynomial regression estimate of order $p = 3$. The question that arises is the following: How to tune k_1 and k_2 for second order derivative estimation? Consider a set of candidate values of k_1 and k_2, for example, $\{5,\ldots,40\}$. Note that, according to Corollary 5, the order of k_q should increase with q. The size of the set is determined both by the computational time that one is willing to invest and by the maximum fraction of the observation weights $s_1,\ldots,s_n$ that one is willing to set to 0 in order to circumvent the aforementioned boundary issues. In order to have a fair comparison among the values of k_1 and k_2, one should use the same observation weights for all candidate values. Therefore, the largest value determines the weights. To choose the value k_1 and k_2 from the candidate set, we can take k_1 and k_2 that minimize $\mathrm{WGCV}^{(2)}$. A similar strategy can be used to determine k_q. We have chosen to tune k_1 according to the way described above and not via (7) because the optimal k_1 for first derivatives is not necessarily the optimal one to be used for estimating second derivatives. From the simulations, it is clear that the variance is larger for increasing q for $\lambda \in (\frac{2q}{2q+1}, 1)$ (the order of the bias remains the same). This was already confirmed by Theorem 4.

For the Monte Carlo study, we constructed data sets are of size $n = 1500$ and generated the function

$$m(x) = 8e^{-(1-5x)^3(1-7x)} \quad \text{for} \quad x \in [0, 0.5]$$

100 times according to model (1) with $e \sim N(0, \sigma^2)$ and $\sigma = 0.1$. As measure of comparison we chose the adjusted mean absolute error defined as

$$\mathrm{MAEadjusted} = \frac{1}{1401} \sum_{i=50}^{1450} |\hat{m}_n^{(2)}(x_i) - m^{(2)}(x_i)|.$$

This criterion was chosen to ignore boundary effects in the estimation. We evaluate the proposed method for derivative estimation with the local slope in local polynomial regression with $p = 5$ and penalized smoothing splines. For the latter we have used septic splines (Newell and Einbeck, 2007) to estimate the second order derivative. The result of the Monte Carlo study is shown in Figure 5. As before, all three methods perform equally well and show similar variances.

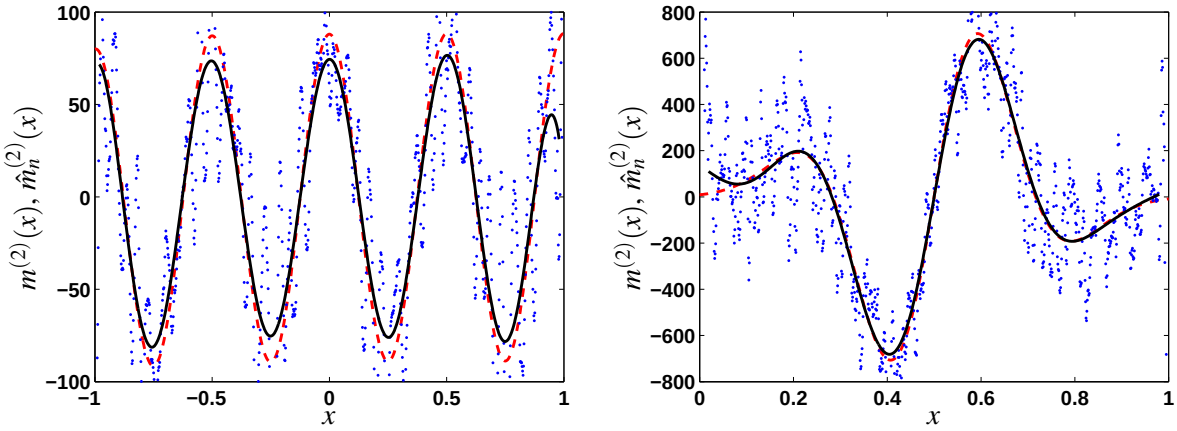


Figure 4: Illustration of the noisy empirical second order derivative (data points), smoothed empirical second order derivative based on a local polynomial regression estimate of order $p = 3$ (bold line) and true derivative (bold dashed line). (a) Second order derivative of regression function (11) with $k_1 = 6$ and $k_2 = 10$; (b) Second order derivative of regression function (12) with $k_1 = 3$ and $k_2 = 25$.

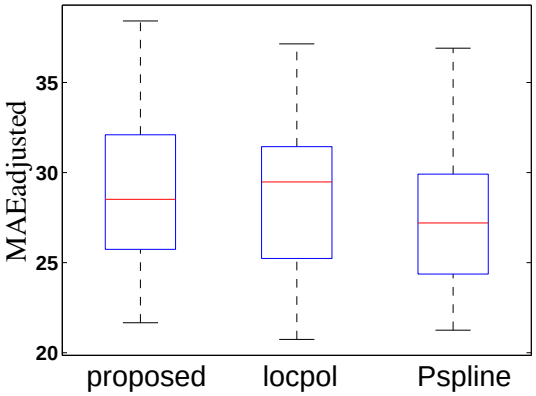


Figure 5: Result of the Monte Carlo study for the proposed method and two other well-known methods for second order derivative estimation.

6. Conclusion

In this paper we proposed a methodology to estimate derivatives nonparametrically without estimating the regression function. We derived L_1 and L_2 rates and established consistency of the estimator. The newly created data sets based on empirical derivatives are no longer independent and identically distributed (i.i.d.) random variables. In order to effectively deal with the non-i.i.d. nature of

the data, we proposed a simple factor method, based on bimodal kernels, for the local polynomial regression framework. Further, we showed that the order bias of the empirical derivative does not depend on the order of the derivative q and that slower rates of convergence are to be expected for increasing orders of derivatives q. However, our technique has also a drawback w.r.t. the design assumptions. All our results have been derived for equispaced design. In many practical applications and data coming from industrial sensors (e.g., process industry, robotics, nanoparticles, growth data) equispaced data is often available since sensors are measuring at predefined times, see, for example, Charnigo et al. (2007) and Patan (2008). However, our approach does not cover all possible applications, that is, application with inherent random design. In this case the weight sequence would depend on the design density, which in practice has to be estimated.

Acknowledgments

Kris De Brabanter is a postdoctoral researcher supported by an FWO fellowship grant. BDM is full professor at the Katholieke Universiteit Leuven, Belgium. Research supported by Onderzoeksfonds KU Leuven/Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet , CoE EF/05/006 Optimization in Engineering (OPTEC) en PFV/10/002 (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government:FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC) IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011);IBBT EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGHWIND (259 166) Contract Research: AMINAL Other: Helmholtz: viCERP ACCM. IG is a full professor at the Katholieke Universiteit Leuven, Belgium. GOA/07/04 en GOA/12/014, IUAP: P6/03, FWO-project G.0328.08N. Interreg IVa 07-022-BE i-MOCCA. The scientific responsibility is assumed by its authors.

Appendix A. Proof Of Proposition 1

Using the fact that $x_{i+j} - x_{i-j} = 2j(n-1)^{-1}d(\mathcal{X})$, where $d(\mathcal{X}) := \sup\mathcal{X} - \inf\mathcal{X}$, yields

$$\begin{aligned}
\mathbf{Var}(Y_i^{(1)}) &= \mathbf{Var}\left(\sum_{j=1}^{k} w_j \cdot \left(\frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}}\right)\right) \\
&= \mathbf{Var}\left(\left(1 - \sum_{j=2}^{k} w_j\right)\frac{Y_{i+1} - Y_{i-1}}{x_{i+1} - x_{i-1}} + \sum_{j=2}^{k} w_j \cdot \left(\frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}}\right)\right) \\
&= \frac{\sigma^2(n-1)^2}{2\,d(\mathcal{X})^2}\left\{\left(1 - \sum_{j=2}^{k} w_j\right)^2 + \sum_{j=2}^{k}\frac{w_j^2}{j^2}\right\}.
\end{aligned}$$

Setting the partial derivatives to zero gives

$$2\left(1 - \sum_{j=2}^{k} w_j\right) = \frac{2w_j}{j^2}, \quad j = 2, \ldots, k,$$

and hence $j^2 w_1 = w_j$. Normalizing such that the weights sum up to one yields

$$w_j = \frac{j^2}{\sum_{i=1}^{k} i^2} = \frac{6j^2}{k(k+1)(2k+1)} \quad j = 1, \ldots, k.$$

Appendix B. Proof Of Theorem 2

Since m is twice continuously differentiable, the following Taylor expansions are valid for $m(x_{i+j})$ and $m(x_{i-j})$ round x_i:

$$m(x_{i+j}) = m(x_i) + (x_{i+j} - x_i)m'(x_i) + \frac{(x_{i+j} - x_i)^2}{2} m^{(2)}(\zeta_{i,i+j})$$

and

$$m(x_{i-j}) = m(x_i) + (x_{i-j} - x_i)m'(x_i) + \frac{(x_{i-j} - x_i)^2}{2} m^{(2)}(\zeta_{i-j,i}),$$

where $\zeta_{i,i+j} \in]x_i, x_{i+j}[$ and $\zeta_{i-j,i} \in]x_{i-j}, x_i[$. Using the above Taylor series and the fact that $x_{i+j} - x_{i-j} = 2j(n-1)^{-1} d(\mathcal{X})$ and $(x_{i+j} - x_i) = \frac{1}{2}(x_{i+j} - x_{i-j})$, it follows that the absolute value of the bias of $Y_i^{(1)}$ is given by

$$\begin{aligned}
\left| \sum_{j=1}^{k} w_j \frac{m(x_{i+j}) - m(x_{i-j})}{x_{i+j} - x_{i-j}} - m'(x_i) \right| &= \left| \sum_{j=1}^{k} w_j \frac{(x_{i+j} - x_{i-j})[m^{(2)}(\zeta_{i,i+j}) - m^{(2)}(\zeta_{i-j,i})]}{8} \right| \\
&\leq \sup_{x \in \mathcal{X}} |m^{(2)}(x)| \left| \sum_{j=1}^{k} w_j \frac{(x_{i+j} - x_{i-j})}{4} \right| \\
&= \frac{\sup_{x \in \mathcal{X}} |m^{(2)}(x)| (n-1)^{-1} d(\mathcal{X})}{2} \sum_{j=1}^{k} \frac{j^3}{\sum_{i=1}^{k} i^2} \\
&= \frac{3k(k+1) \sup_{x \in \mathcal{X}} |m^{(2)}(x)| d(\mathcal{X})}{4(n-1)(2k+1)} \\
&= O(kn^{-1})
\end{aligned}$$

uniformly over i. Using Proposition 1, the variance of $Y_i^{(1)}$ yields

$$\begin{aligned}
\mathbf{Var}(Y_i^{(1)}) &= \frac{\sigma^2 (n-1)^2}{2 d(\mathcal{X})^2} \left\{ \left(1 - \sum_{j=2}^{k} w_j \right)^2 + \sum_{j=2}^{k} \frac{w_j^2}{j^2} \right\} \\
&= \frac{\sigma^2 (n-1)^2}{2 d(\mathcal{X})^2} \sum_{j=1}^{k} \frac{w_j^2}{j^2} \\
&= \frac{\sigma^2 (n-1)^2}{2 d(\mathcal{X})^2} \sum_{j=1}^{k} \frac{36 j^2}{k^2 (k+1)^2 (2k+1)^2} \\
&= \frac{3\sigma^2 (n-1)^2}{k(k+1)(2k+1) d(\mathcal{X})^2} = O(n^2 k^{-3})
\end{aligned}$$

uniformly over i.

Appendix C. Proof of Theorem 3

Due to Chebyshev's inequality, it suffices to show that the mean squared error (MSE) goes to zero, that is,

$$\lim_{n\to\infty} \mathrm{MSE}(Y_i^{(1)}) \to 0. \tag{13}$$

Under the conditions $k \to \infty$ as $n \to \infty$ such that $n^{-1}k \to 0$ and $nk^{-3/2} \to 0$, the bias and variance go to zero (see Theorem 2). Hence, condition (13) is fulfilled.

Appendix D. Proof Of Theorem 4

The first step is to notice that there exist $\lambda \in (0,1)$ and $c_1 \in (0,\infty)$ (see Theorem 3) so that the bias and variance of the first order empirical derivative can be written as $\mathrm{bias}(Y_i^{(1)}) = O(n^{\lambda-1})$ and $\mathbf{Var}(Y_i^{(1)}) = O(n^{2-3\lambda})$ uniformly over i for $k_1 n^{-\lambda} \to c_1$ as $n \to \infty$. Next, we continue the proof by induction. For the bias, assume that the first $(q+1)$ derivatives of m are continuous on the compact interval $\mathcal{X}$. Hence, all $O(\cdot)$-terms are uniformly over i. For any $l \in \{0,1,\ldots,q\}$, a Taylor series yields

$$m^{(l)}(x_{i\pm j}) = m^{(l)}(x_i) + \sum_{p=1}^{q-l} \frac{\left(\pm\frac{jd(\mathcal{X})}{n-1}\right)^p}{p!} m^{(p+l)}(x_i) + O\left((j/n)^{q-l+1}\right). \tag{14}$$

The expected value of the first order empirical derivative is given by (see Section 2)

$$\mathbf{E}(Y_i^{(1)}) = m'(x_i) + \sum_{p=3,5,\ldots}^{q} m^{(p)}(x_i) \sum_{j=1}^{k_1} \frac{w_{j,1}}{p!} \frac{j^{p-1} d(\mathcal{X})^{p-1}}{(n-1)^{p-1}} + O\left(n^{q(\lambda-1)}\right),$$

with

$$\theta_{p,1} = \sum_{j=1}^{k_1} \frac{w_{j,1}}{p!} \frac{j^{p-1} d(\mathcal{X})^{p-1}}{(n-1)^{p-1}} = O\left(n^{(p-1)(\lambda-1)}\right),$$

for $k_1 n^{-\lambda} \to c_1$ as $n \to \infty$. Suppose that for $l \in \{2,\ldots,q\}$ and $k_l n^{-\lambda} \to c_l$, where $c_l \in (0,\infty)$, as $n \to \infty$

$$\mathbf{E}(Y_i^{(l-1)}) = m^{(l-1)}(x_i) + \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} m^{(p)}(x_i) + O\left(n^{(q-l+2)(\lambda-1)}\right) \tag{15}$$

for $\theta_{p,l-1} = O\left(n^{(p-l+1)(\lambda-1)}\right)$. We now prove that

$$\mathbf{E}(Y_i^{(l)}) = m^{(l)}(x_i) + \sum_{p=l+2,l+4,\ldots}^{q} \theta_{p,l} m^{(p)}(x_i) + O\left(n^{(q-l+1)(\lambda-1)}\right)$$

for $\theta_{p,l} = O\left(n^{(p-l)(\lambda-1)}\right)$. Using (14) and (15) yields for $\Delta = \mathbf{E}(Y_{i+j}^{(l-1)}) - \mathbf{E}(Y_{i-j}^{(l-1)})$

$$\begin{aligned}
\Delta &= m^{(l-1)}(x_{i+j}) + \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} m^{(p)}(x_{i+j}) - m^{(l-1)}(x_{i-j}) - \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} m^{(p)}(x_{i-j}) + O\left(n^{(q-l+2)(\lambda-1)}\right) \\
&= \sum_{p=1}^{q-l+1} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^p}{p!} m^{(p+l-1)}(x_i) + O\left((j/n)^{q-l+2}\right) \\
&\quad + \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} \left[m^{(p)}(x_i) + \sum_{s=1}^{q-p} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^s}{s!} m^{(p+s)}(x_i) + O\left((j/n)^{q-p+1}\right) \right] \\
&\quad - \sum_{p=1}^{q-l+1} \frac{\left(\frac{-jd(\mathcal{X})}{n-1}\right)^p}{p!} m^{(p+l-1)}(x_i) + O\left((j/n)^{q-l+2}\right) \\
&\quad - \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} \left[m^{(p)}(x_i) + \sum_{s=1}^{q-p} \frac{\left(\frac{-jd(\mathcal{X})}{n-1}\right)^s}{s!} m^{(p+s)}(x_i) + O\left((j/n)^{q-p+1}\right) \right] + O\left(n^{(q-l+2)(\lambda-1)}\right).
\end{aligned}$$

Rearranging and grouping term gives

$$\begin{aligned}
\frac{\Delta}{x_{i+j} - x_{i-j}} &= m^{(l)}(x_i) + \sum_{p=3,5,\ldots}^{q-l+1} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^{p-1}}{p!} m^{(p+l-1)}(x_i) + O\left((j/n)^{q-l+1}\right) \\
&\quad + \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} \left[\sum_{s=1,3,\ldots}^{q-p} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^{s-1}}{s!} m^{(p+s)}(x_i) + O\left((j/n)^{q-p}\right) \right] \\
&\quad + \frac{n-1}{2jd(\mathcal{X})} O\left(n^{(q-l+2)(\lambda-1)}\right).
\end{aligned}$$

Multiplying all the above terms by $w_{j,l} = \dfrac{j}{\sum_{i=1}^{k_l} i}$ and summing over $j = 1, 2, \ldots, k_l$ results in

$$\begin{aligned}
\mathbf{E}(Y_i^{(l)}) &= m^{(l)}(x_i) \\
&\quad + \sum_{j=1}^{k_l} \frac{j}{\sum_{i=1}^{k_l} i} \sum_{p=3,5,\ldots}^{q-l+1} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^{p-1}}{p!} m^{(p+l-1)}(x_i) && (16) \\
&\quad + \sum_{j=1}^{k_l} \frac{j}{\sum_{i=1}^{k_l} i} O\left((j/n)^{q-l+1}\right) && (17) \\
&\quad + \sum_{j=1}^{k_l} \frac{j}{\sum_{i=1}^{k_l} i} \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} \sum_{s=1,3,\ldots}^{q-p} \frac{\left(\frac{jd(\mathcal{X})}{n-1}\right)^{s-1}}{s!} m^{(p+s)}(x_i) && (18) \\
&\quad + \sum_{j=1}^{k_l} \frac{j}{\sum_{i=1}^{k_l} i} \sum_{p=l+1,l+3,\ldots}^{q} \theta_{p,l-1} O\left((j/n)^{q-p}\right) && (19) \\
&\quad + \sum_{j=1}^{k_l} \frac{j}{\sum_{i=1}^{k_l} i} \frac{n-1}{2jd(\mathcal{X})} O\left(n^{(q-l+2)(\lambda-1)}\right). && (20)
\end{aligned}$$

The terms (17), (19) and (20) all yield $O(n^{(q-l+1)(\lambda-1)})$ for $\theta_{p,l-1} = O(n^{(p-l+1)(\lambda-1)})$. Similar, the terms (16) and (18) yield $\sum_{p=l+2,l+4,\ldots}^{q} \theta_{p,l} m^{(p)}(x_i)$ for $\theta_{p,l} = O\left(n^{(p-l)(\lambda-1)}\right)$ for $k_l n^{-\lambda} \to c_l$ as $n \to \infty$. As a consequence, the bias of $Y_i^{(l)}$ is given by

$$\text{bias}(Y_i^{(l)}) = \mathbf{E}(Y_i^{(l)}) - m^{(l)}(x_i) = \sum_{p=l+2,l+4,\ldots}^{q} \theta_{p,l} m^{(p)}(x_i) + O(n^{(\lambda-1)}) = O(n^{\lambda-1}).$$

For the variance, we proceed in a similar way. Note that $\mathbf{Var}(Y_i^{(1)}) = O(n^{2-3\lambda})$ uniformly over i. Assume that $\mathbf{Var}(Y_i^{(l-1)}) = O(n^{2(l-1)-2\lambda(l-1/2)})$ uniformly over i for $l \in \{2, 3, \ldots, q\}$. The proof will be complete if we show that $\mathbf{Var}(Y_i^{(l)}) = O(n^{2l-2\lambda(l+1/2)})$. The variance of $Y_i^{(l)}$ is given by

$$\begin{aligned}
\mathbf{Var}(Y_i^{(l)}) &= \frac{(n-1)^2}{4d(\mathcal{X})^2} \mathbf{Var}\left(\sum_{j=1}^{k_l} \frac{w_{j,l}}{j}\left(Y_{i+j}^{(l-1)} - Y_{i-j}^{(l-1)}\right)\right) \\
&\leq \frac{(n-1)^2}{2d(\mathcal{X})^2}\left[\mathbf{Var}\left(\sum_{j=1}^{k_l} \frac{w_{j,l}}{j} Y_{i+j}^{(l-1)}\right) + \mathbf{Var}\left(\sum_{j=1}^{k_l} \frac{w_{j,l}}{j} Y_{i-j}^{(l-1)}\right)\right].
\end{aligned}$$

For $a_j \in \mathbb{N} \setminus \{0\}, j = 1, \ldots, k_l$, the variance is upperbounded by

$$\mathbf{Var}(Y_i^{(l)}) \leq \frac{(n-1)^2}{d(\mathcal{X})^2}\left(\sum_{j=1}^{k_l} a_j \frac{w_{j,l}^2}{j^2}\right) O(n^{2(l-1)-2\lambda(l-1/2)}).$$

As in the proof of the bias, the choice of the weights become clear. If we choose $w_{j,l} = \frac{j}{\sum_{i=1}^{k_l} i}$ for $l \geq 2$ then $\sum_{j=1}^{k_l} a_j \frac{w_{j,l}^2}{j^2} = O(n^{-2\lambda})$. Then, for $k_l n^{-\lambda} \to c_l$ as $n \to \infty$, it readily follows that $\mathbf{Var}(Y_i^{(l)}) = O(n^{2l-2\lambda(l+1/2)})$.

References

J.L.O. Cabrera. *locpol: Kernel Local Polynomial Regression*, 2009. URL `http://CRAN.R-project.org/package=locpol`. R package version 0.4-0.

R. Charnigo, M. Francoeur, M.P. Mengüç, A. Brock, M. Leichter, and C. Srinivasan. Derivatives of scattering profiles: tools for nanoparticle characterization. *J. Opt. Soc. Am. A*, 24(9):2578–2589, 2007.

P. Chaudhuri and J.S. Marron. SiZer for exploration of structures in curves. *J. Amer. Statist. Assoc.*, 94(447):807–823, 1999.

C.K. Chu and J.S. Marron. Comparison of two bandwidth selectors with dependent errors. *Ann. Statist.*, 19(4):1906–1918, 1991.

K. De Brabanter, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Kernel regression in the presence of correlated errors. *J. Mach. Learn. Res.*, 12:1955–1976, 2011.

M. Delecroix and A.C. Rosa. Nonparametric estimation of a regression function and its derivatives under an ergodic hypothesis. *J. Nonparametr. Stat.*, 6(4):367–382, 2007.

R.L. Eubank and P.L. Speckman. Confidence bands in nonparametric regression. *J. Amer. Statist. Assoc.*, 88(424):1287–1301, 1993.

J. Fan and I. Gijbels. Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation. *J. R. Stat. Soc. Ser. B*, 57(2):371–394, 1995.

J. Fan and I. Gijbels. *Local Polynomial Modeling and Its Applications*. Chapman & Hall, 1996.

T. Gasser and H.-G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scand. J. Statist.*, 11(3):171–185, 1984.

I. Gijbels and A.-C. Goderniaux. Data-driven discontinuity detection in derivatives of a regression function. *Communications in Statistics–Theory and Methods*, 33:851–871, 2004.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2002.

P. Hall, J.W. Kay, and D.M. Titterington. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3):521–528, 1990.

W. Härdle. *Applied Nonparametric Regression*. Cambridge University Press, 1990.

W. Härdle and T. Gasser. On robust kernel estimation of derivatives of regression functions. *Scand. J. Statist.*, 12(3):233–240, 1985.

A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 1996.

R. Jarrow, D. Ruppert, and Y. Yu. Estimating the term structure of corporate debt with a semiparametric penalized spline model. *J. Amer. Statist. Assoc.*, 99(465):57–66, 2004.

H.-G. Müller. *Nonparametric Regression Analysis of Longitudinal Data*. Springer-Verlag, 1988.

H.-G. Müller, U. Stadtmüller, and T. Schmitt. Bandwidth choice and confidence intervals for derivatives of noisy data. *Biometrika*, 74(4):743–749, 1987.

J. Newell and J. Einbeck. A comparative study of nonparametric derivative estimators. *Proc. of the 22nd International Workshop on Statistical Modelling*, 2007.

J. Opsomer, Y. Wang, and Y. Yang. Nonparametric regression with correlated errors. *Statist. Sci.*, 16(2):134–153, 2001.

C. Park and K.-H. Kang. SiZer analysis for the comparison of regression curves. *Comput. Statist. Data Anal.*, 52(8):3954–3970, 2008.

K. Patan. *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*. Springer-Verlag, 2008.

J. Ramsay. Derivative estimation. StatLib – S-News, Thursday, March 12, 1998: `http://www.math.yorku.ca/Who/Faculty/Monette/S-news/0556.html`, 1998.

J.O. Ramsay and B.W. Silverman. *Applied Functional Data Analysis*. Springer-Verlag, 2002.

J. Ramsey and B. Ripley. *pspline: Penalized Smoothing Splines*, 2010. URL `http://CRAN.R-project.org/package=pspline`. R package version 1.0-14.

V. Rondonotti, J.S. Marron, and C. Park. SiZer for time series: A new approach to the analysis of trends. *Electron. J. Stat.*, 1:268–289, 2007.

D. Ruppert and M.P. Wand. Multivariate locally weighted least squares regression. *Ann. Statist.*, 22 (3):1346–1370, 1994.

J.S. Simonoff. *Smoothing Methods in Statistics*. Springer-Verlag, 1996.

C. Stone. Additive regression and other nonparametric models. *Ann. Statist.*, 13(2):689–705, 1985.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

G. Wahba and Y. Wang. When is the optimal regularization parameter insensitive to the choice of loss function? *Comm. Statist. Theory Methods*, 19(5):1685–1700, 1990.

M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.

S. Zhou and D.A. Wolfe. On derivative estimation in spline regression. *Statist. Sinica*, 10(1): 93–108, 2000.

Journal of Machine Learning Research 14 (2013) 303-331 Submitted 1/12; Revised 10/12; Published 2/13

Using Symmetry and Evolutionary Search to Minimize Sorting Networks

Vinod K. Valsalam VKV@CS.UTEXAS.EDU
Risto Miikkulainen RISTO@CS.UTEXAS.EDU
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712, USA

Editor: Una-May O'Reilly

Abstract

Sorting networks are an interesting class of parallel sorting algorithms with applications in multiprocessor computers and switching networks. They are built by cascading a series of comparison-exchange units called comparators. Minimizing the number of comparators for a given number of inputs is a challenging optimization problem. This paper presents a two-pronged approach called Symmetry and Evolution based Network Sort Optimization (SENSO) that makes it possible to scale the solutions to networks with a larger number of inputs than previously possible. First, it uses the symmetry of the problem to decompose the minimization goal into subgoals that are easier to solve. Second, it minimizes the resulting greedy solutions further by using an evolutionary algorithm to learn the statistical distribution of comparators in minimal networks. The final solutions improve upon half-century of results published in patents, books, and peer-reviewed literature, demonstrating the potential of the SENSO approach for solving difficult combinatorial problems.

Keywords: symmetry, evolution, estimation of distribution algorithms, sorting networks, combinatorial optimization

1. Introduction

A sorting network of n inputs is a fixed sequence of comparison-exchange operations (comparators) that sorts all inputs of size n (Knuth, 1998). Since the same fixed sequence of comparators can sort any input, it represents an oblivious or data-independent sorting algorithm, that is, the sequence of comparisons does not depend on the input data. The resulting fixed pattern of communication makes them desirable in parallel implementations of sorting, such as those on graphics processing units (Kipfer et al., 2004). For the same reason, they are simple to implement in hardware and are useful as switching networks in multiprocessor computers (Batcher, 1968; Kannan and Ray, 2001; Baddar, 2009).

Driven by such applications, sorting networks have been the subject of active research since the 1950's (Knuth, 1998). Of particular interest are minimal-size networks that use a minimal number of comparators. Designing such networks is a hard combinatorial optimization problem, first investigated in a U.S. Patent by O'Connor and Nelson (1962) for $4 \leq n \leq 8$. Their networks had the minimal number of comparators for 4, 5, 6, and 8 inputs, but required two extra comparators for 7 inputs. This result was improved by Batcher (1968), whose algorithmic construction produces provably minimal networks for $n \leq 8$ (Knuth, 1998).

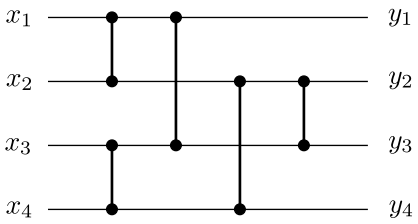


Figure 1: A 4-input sorting network. The input values $\{x_1, x_2, x_3, x_4\}$ at the left side of the horizontal lines pass through a sequence of comparison-exchange operations, represented by vertical lines connecting pairs of horizontal lines. Each such comparator sorts its two values, resulting in the horizontal lines containing the sorted output values $\{y_1 \leq y_2 \leq y_3 \leq y_4\}$ at right. This network is minimal in terms of the number of comparators. Such minimal networks are not known in general for input sizes larger than 8 and designing them is a challenging optimization problem.

Still today, provably minimal networks are known only for $n \leq 8$. Finding the minimum number of comparators for $n > 8$ is thus a challenging open problem. It has been studied by various researchers using specialized techniques, often separately for each value of n (Knuth, 1998; Koza et al., 1999). Their efforts during the last few decades have improved the size of the networks for $9 \leq n \leq 16$. For larger values of n, all best known solutions are simply merges of smaller networks; the problem is so difficult that it has not been possible to improve on these straightforward constructions (Baddar, 2009).

This paper presents a two-pronged approach to this problem, using symmetry and evolutionary search, which makes it possible to scale the problem to larger number of inputs. This approach, called Symmetry and Evolution based Network Sort Optimization (SENSO), learns the comparator distribution of minimal networks from a population of candidate solutions and improves them iteratively through evolution. Each such solution is generated by sampling comparators from the previous distribution such that the required network symmetry is built step-by-step, thereby focusing evolution on more likely candidates and making search more effective. This approach was able to discover new minimal networks for 17, 18, 19, 20, 21, and 22 inputs. Moreover, for the other $n \leq 23$, it discovered networks that have the same size as the best known networks. These results demonstrate that the approach makes the problem more tractable and suggests ways in which it can be scaled further and applied to other similarly difficult combinatorial problems.

This paper is organized as follows. Section 2 begins by describing the problem of finding minimal sorting networks in more detail and reviews previous research on solving it. Section 3 presents the SENSO approach, based on symmetry and evolution. Section 4 discusses the experimental setup for evaluating the approach and presents the results. Section 5 concludes with an analysis of the results and discussion of ways to make the approach even more effective and general in the future.

2. Background

Figure 1 illustrates a 4-input sorting network. The horizontal lines of the network receive the input values $\{x_1, x_2, x_3, x_4\}$ at left. Each vertical line represents a comparison-exchange operation that

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Batcher	**0**	**1**	**3**	**5**	**9**	**12**	**16**	**19**	26	31	37	41	48	53	59	63
Best	**0**	**1**	**3**	**5**	**9**	**12**	**16**	**19**	25	29	35	39	45	51	56	60

Table 1: The fewest number of comparators known to date for sorting networks of input sizes $n \leq 16$. These networks have been studied extensively, but the best results have been proven to be minimal only for $n \leq 8$ (shown in bold; Knuth, 1998). Such small networks are interesting because they optimize hardware resources in implementations such as multiprocessor switching networks.

takes two values and exchanges them if necessary such that the larger value is on the lower line. As a result of these comparison-exchanges, the output values appear at the right side of the network in sorted order: $\{y_1 \leq y_2 \leq y_3 \leq y_4\}$.

Sorting networks with $n \leq 16$ have been studied extensively with the goal of minimizing their sizes. The smallest sizes of such networks known to date are listed in Table 1 (Knuth, 1998). The number of comparators has been proven to be minimal only for $n \leq 8$ (Knuth, 1998). These networks can be constructed using Batcher's algorithm for odd-even merging networks (Batcher, 1968). The odd-even merge builds larger networks iteratively from smaller networks by merging two sorted lists. The odd and even indexed values of these two lists are first merged separately using small merging networks. Comparison-exchange operations are then applied to the corresponding values of the resulting small sorted lists to obtain the full sorted list.

Finding the minimum number of comparators required for $n > 8$ remains an open problem. The results in Table 1, for these values of n, improve on the number of comparators used by Batcher's method. For example, the 16-input case, for which Batcher's method requires 63 comparators, was improved by Shapiro who found a 62-comparator network in 1969. Soon afterwards, Green (1972) found a network with 60 comparators (Figure 2), which still remains the best in terms of the number of comparators.

In Green's construction, comparisons made after the first four levels (i.e., the first 32 comparators) are difficult to understand, making his method hard to generalize to larger values of n. For such values, Batcher's method can be extended with more complex merging strategies to produce significant savings in the number of comparators (Van Voorhis, 1974; Drysdale and Young, 1975). For example, the best known 256-input sorting network due to Van Voorhis requires only 3651 comparators, compared to 3839 comparators required by Batcher's method (Drysdale and Young, 1975; Knuth, 1998). Asymptotically, the methods based on merging require $O(n \log^2 n)$ comparators (Van Voorhis, 1974). In comparison, the *AKS network* by Ajtai et al. (1983) produces better upper bounds, requiring only $O(n \log n)$ comparators. However, the constants hidden in its asymptotic notation are so large that these networks are impractical. Although still not practical, Leighton and Plaxton (1990) showed that small constants are actually possible in networks that sort all but a superpolynomially small fraction of the $n!$ input permutations.

Since better algorithms are not known for constructing networks that sort all $n!$ input permutations, Batcher's or Van Voorhis' algorithms are often used in practice for large values of n, despite their non-optimality. For example, these algorithms were used to obtain the networks for $17 \leq n \leq 32$ listed in Table 2 by merging the outputs of smaller networks from Table 1 (Van Voorhis, 1971; Baddar, 2009).

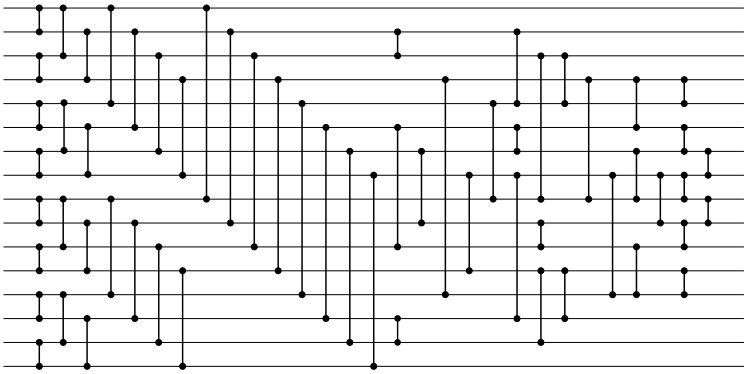


Figure 2: The 16-input sorting network found by Green. This network has 60 comparators, which is the fewest known for 16 inputs (Green, 1972; Knuth, 1998). The comparators in such hand-designed networks are often symmetrically arranged about a horizontal axis through the middle of the network. This observation has been used by some researchers to bias evolutionary search on this problem (Graham and Oppacher, 2006) and is also used as a heuristic to augment the symmetry-building approach described in Section 3.

n	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Best	73	79	88	93	103	110	118	123	133	140	150	156	166	172	180	185

Table 2: The fewest number of comparators known to date for sorting networks of input sizes $17 \leq n \leq 32$. Networks for these values of n were obtained by merging the outputs of smaller networks from Table 1 using the non-optimal Batcher's or Van Voorhis' algorithms (Van Voorhis, 1971; Baddar, 2009). The methods used to optimize networks for $n \leq 16$ are intractable for these larger values of n because of the explosive growth in the size of the search space. The approach presented in this paper mitigates this problem by constraining search to promising solutions and improves these results for input sizes 17, 18, 19, 20, 21, and 22.

The difficulty of finding such minimal sorting networks prompted researchers to attack the problem using evolutionary techniques. In one such study by Hillis (1991), a 16-input network having 61 comparators was evolved. He facilitated the evolutionary search by initializing the population with the first four levels of Green's network, so that evolution would need to discover only the remaining comparators. This (host) population of sorting networks was co-evolved with a (parasite) population of test cases that were scored based on how well they made the sorting networks fail. The purpose of the parasitic test cases is to nudge the solutions away from getting stuck on local optima.

Juillé (1995) improved on Hillis' results by evolving 16-input networks that are as good as Green's network (60 comparators), from scratch without specifying the first 32 comparators. Moreover, Juillé's method discovered 45-comparator networks for the 13-input problem, which was an improvement of one comparator over the previously known best result. His method, based on the Evolving Non-Determinism (END) model, constructs solutions incrementally as paths in a search

tree whose leaves represent valid sorting networks. The individuals in the evolving population are internal nodes of this search tree. The search proceeds in a way similar to beam search by assigning a fitness score to internal nodes and selecting nodes that are the most promising. The fitness of an internal node is estimated by constructing a path incrementally and randomly to a leaf node. This method found good networks with the same number of comparators as in Table 1 for all $9 \leq n \leq 16$.

Motivated by observations of symmetric arrangement of comparators in many sorting networks (Figure 2), Graham and Oppacher (2006) used symmetry explicitly to bias evolutionary search. They compared evolutionary runs on populations initialized randomly with either symmetric or asymmetric networks for the 10-input sorting problem. The symmetric networks were produced using symmetric comparator pairs, that is, pairs of comparators that are vertical mirror images of each other. Although evolution was allowed to disrupt the initial symmetry through variation operators, symmetric initialization resulted in higher success rates compared to asymmetric initialization. A similar heuristic is used to augment the SENSO approach discussed in this paper.

Evolutionary approaches must verify that the solution network sorts all possible inputs correctly. A naive approach is to test the network on all $n!$ permutations of n distinct numbers. A better approach requiring far fewer tests uses the *zero-one principle* (Knuth, 1998) to reduce the number of test cases to 2^n binary sequences. According to this principle, if a network with n input lines sorts all 2^n binary sequences correctly, then it will also sort any arbitrary sequence of n non-binary numbers correctly. However, the increase in the number of test cases remains exponential and is a bottleneck in fitness evaluations. Therefore, some researchers have used FPGAs to mitigate this problem by performing the fitness evaluations on a massively parallel scale (Koza et al., 1998; Korenek and Sekanina, 2005). In contrast, this paper develops a Boolean function representation of the zero-one principle for fitness evaluation, as discussed next.

3. Approach

This section presents the new SENSO approach based on symmetry and evolutionary search to minimize the number of comparators in sorting networks. It begins with a description of how the sorting network outputs can be represented as monotone Boolean functions, exposing the symmetries of the network. This representation makes it possible to decompose the problem into subgoals, which are easier to solve. Each subgoal constitutes a step in building the symmetries of the network with as few comparators as possible. The resulting greedy solutions are optimized further by using an evolutionary algorithm to learn the distribution of comparators that produce minimal networks.

3.1 Boolean Function Representation

The zero-one principle (Section 2) can be used to express the inputs of a sorting network as Boolean variables and its outputs as functions of those variables. It simplifies the sorting problem to counting the number of inputs that have the value 1 and setting that many of the lowermost outputs to 1 and the remaining outputs to 0. In particular, the function $f_i(x_i, \ldots, x_n)$ at output i takes the value 1 if and only if at least $n+1-i$ inputs are 1. That is, f_i is the disjunction of all conjunctive terms with exactly $n+1-i$ variables.

Since these functions are implemented by the comparators in the network, the problem of designing a sorting network can be restated as the problem of finding a sequence of comparators that compute its output functions. Each comparator computes the conjunction (upper line) and disjunction (lower line) of their inputs. As a result, a sequence of comparators computes Boolean functions

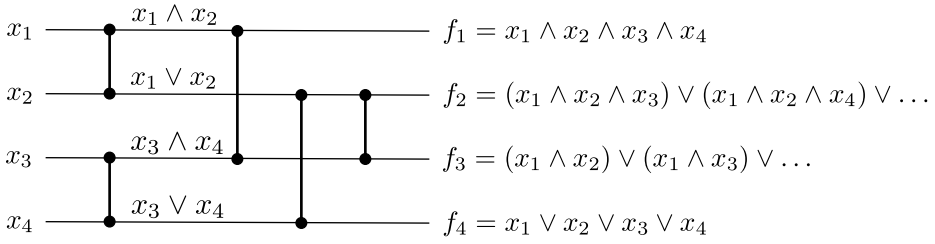
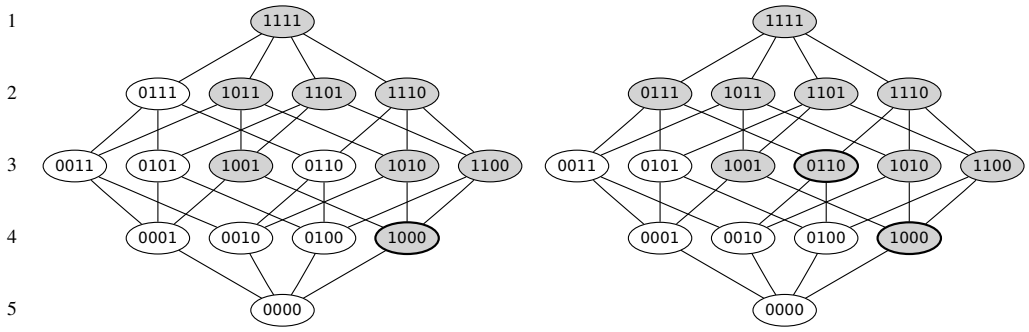


Figure 3: Boolean output functions of a 4-input sorting network. The zero-one principle can be used to represent the inputs of the network as Boolean variables. Each comparator produces the conjunction of its inputs on its upper line and their disjunction on its lower line. As a result, the functions at the outputs of the network are compositions of conjunctions and disjunctions of the input variables, that is, they are monotone Boolean functions. In particular, the output function f_i at line i is the disjunction of all conjunctive terms with exactly $n+1-i$ variables. Therefore, a sorting network is a sequence of comparators that compute all its output functions from its input variables. This representation makes it possible to express network symmetry, which turns out to be useful in constructing minimal networks.

that are compositions of conjunctions and disjunctions of the input variables (Figure 3). Since the number of terms in these functions can grow combinatorially as comparators are added, it is necessary to use a representation that makes it efficient to compute them and to determine whether all output functions have been computed.

Such functions computed using only conjunctions and disjunctions without any negations are called *monotone Boolean functions* (Korshunov, 2003). For example, the functions for the 4-input sorting network in Figure 3 are all monotone Boolean functions. Such a function f on n binary variables has the property that $f(\mathbf{a}) \leq f(\mathbf{b})$ for any distinct binary n-tuples $\mathbf{a} = a_1, \ldots, a_n$ and $\mathbf{b} = b_1, \ldots, b_n$ such that $\mathbf{a} \prec \mathbf{b}$, where $\mathbf{a} \prec \mathbf{b}$ if $a_i \leq b_i$ for $1 \leq i \leq n$. The set of all 2^n binary n-tuples ordered by $\prec$ is a partially ordered set called a *Boolean lattice*, which makes it possible to represent monotone Boolean functions conveniently. The Boolean lattice for $n = 4$ is illustrated in Figure 4 as an undirected graph (Hasse diagram) of $2^4 = 16$ nodes. Any two nodes in the lattice are *comparable* and are connected by a path if they can be ordered by $\prec$. A subset of nodes that are pair-wise incomparable is called an *antichain*. A subset X of nodes is said to be *bounded above* by the node $\mathbf{y}$ if $\mathbf{x} \prec \mathbf{y}$ for all $\mathbf{x} \in X$. The term *bounded below* is defined in a similar manner. These concepts are used to characterize monotone Boolean functions in sorting networks.

For any monotone Boolean function f, the subset of lattice nodes at which it takes the value 1 are bounded above by the topmost node in the lattice and are bounded below by an antichain of nodes corresponding to the conjunctive terms in its disjunctive normal form. That is, the nodes in this antichain form a boundary in the lattice, separating the nodes at which f takes the value 1 from those at which it takes the value 0. Therefore, it is sufficient to specify the antichain of boundary nodes to define a monotone Boolean function. Moreover, nodes in the same level i (numbered from the top of the lattice) form an antichain of boundary nodes because they all have the same number $n+1-i$ of 1s in their binary representations and are therefore incomparable. In fact, they are the boundary nodes of function f_i at output i of the sorting network since they correspond to the



(a) $f(x_1,x_2,x_3,x_4) = x_1$ (b) $f(x_1,x_2,x_3,x_4) = x_1 \vee (x_2 \wedge x_3)$

Figure 4: Representation of monotone Boolean functions on four variables in the Boolean lattice. The $2^4 = 16$ nodes of the lattice are organized in levels (numbered on the left), each containing the binary value assignments to the 4-tuple $x_1x_2x_3x_4$ with the same number of 1s. The truth table for any Boolean function $f(x_1,x_2,x_3,x_4)$ can be represented in this lattice by shading all the nodes for which f takes the value 1. Furthermore, a node $b_1b_2b_3b_4$ has a path to a lower node $a_1a_2a_3a_4$ if $a_i \leq b_i$ for $1 \leq i \leq 4$. As a result, if node $a_1a_2a_3a_4$ is shaded for a monotone function f, then all higher nodes reachable from it are also shaded, that is, f is defined completely by the nodes in the lower boundary of its shaded region. This set of nodes (shown by bold outline) corresponds to the conjunctive terms in the disjunctive normal form of f. For example, it contains just the node 1000 for $f = x_1$ and two nodes 1000 and 0110 for $f = x_1 \vee (x_2 \wedge x_3)$. This representation makes it possible to compute monotone Boolean functions more efficiently.

disjunction of all conjunctive terms with exactly $n+1-i$ variables. Thus, levels 1 to n of the lattice have a one-to-one correspondence with the output functions of the n-input network. Moreover, it is possible to efficiently determine whether f_i has been computed at output i just by verifying whether it takes the value 1 at all level i boundary nodes.

Monotone Boolean functions can thus be represented by their antichain of boundary nodes in the Boolean lattice. In a lattice of size 2^n, the maximum size of this representation is equal to the size of the longest antichain, which is only $\binom{n}{\lceil n/2 \rceil}$ nodes (by Stirling's approximation, $\binom{n}{\lceil n/2 \rceil} = \mathrm{O}\left(\frac{2^n}{\sqrt{n}}\right)$). However, computing conjunctions and disjunctions using this representation produces combinatorially more redundant, non-boundary nodes that have to be removed (Gunter et al., 1996). A more efficient representation is based on storing the values of the function in its entire truth table as a bit-vector of length 2^n. Its values are grouped according to the levels in the Boolean lattice so that values for any level can be retrieved easily. This representation also allows computing conjunctions and disjunctions efficiently as the bitwise AND and bitwise OR of the bit-vectors, respectively. Moreover, efficient algorithms for bit-counting can be used to determine if a given sorting network is valid by checking if its function at output i has the value 1 at all level i nodes for $1 \leq i \leq n$, which is the case when all output functions f_i are computed correctly.

	DNF	CNF
f_1	$x_1 \wedge x_2 \wedge x_3 \wedge x_4$	$x_1 \wedge x_2 \wedge x_3 \wedge x_4$
f_2	$(x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee \ldots$	$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge \ldots$
f_3	$(x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee \ldots$	$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge \ldots$
f_4	$x_1 \vee x_2 \vee x_3 \vee x_4$	$x_1 \vee x_2 \vee x_3 \vee x_4$

Table 3: Symmetries of the 4-input sorting network in terms of its output functions. Writing the Boolean output functions of the sorting network in both the disjunctive normal form (DNF) and in the conjunctive normal form (CNF) is a good way to visualize the symmetries of the output functions. For example, swapping the conjunctions $\wedge$ and disjunctions $\vee$ in the DNF form of either function f_2 or f_3 yields the CNF form of the other function. Therefore, for the operation of swapping $\wedge$ and $\vee$ in both f_2 and f_3 and also swapping their row positions in the table, the resulting table of functions remains the same as the original table. Moreover, this assertion holds for any pair of functions f_i and f_{5-i}, not just for f_2 and f_3. Such an operation that preserves the output functions of the network is called a symmetry. These symmetries can be used to minimize the number of comparators in the network.

Finding a minimum sequence of comparators that computes all the output functions is a challenging combinatorial problem. It can be made more tractable by using the symmetries of the network, represented in terms of the symmetries of its output functions, as will be described next.

3.2 Sorting Network Symmetries

A sorting network *symmetry* is an operation on the ordered set of network output functions that leaves the functions invariant, that is, the resulting network outputs remain unchanged. For example, swapping the outputs of all comparators of a network to reverse its sorting order and then flipping the network vertically to restore its original sorting order is a symmetry operation. Swapping the comparator outputs swaps the conjunctions $\wedge$ and disjunctions $\vee$ in the output functions. The resulting reversal of the network sorting order can be expressed as $f_i(x_i,\ldots,x_n,\wedge,\vee) = f_{n+1-i}(x_i,\ldots,x_n,\vee,\wedge)$ for all $1 \leq i \leq n$, that is, the output function f_{n+1-i} can be obtained from f_i by swapping its $\wedge$ and $\vee$, and vice versa. Therefore, in addition to swapping $\wedge$ and $\vee$, if the *dual* functions f_i and f_{n+1-i} are also swapped, then the network outputs remain the same. This type of symmetry is illustrated in Table 3 for the 4-input sorting network.

It is thus possible to define symmetry operations σ_i for $1 \leq i \leq \lceil \frac{n}{2} \rceil$ that act on the ordered set of network output functions by swapping the function f_i and its dual f_{n+1-i} and swapping their $\wedge$ and $\vee$. The compositions of these symmetry operations are also symmetries because σ_i and σ_j operate independently on different pairs of output functions. That is, this set of operations are closed under composition, and they are associative. Moreover, each operation is its own inverse, producing the identity when applied twice in a row. Thus they satisfy all the axioms of a *group* for representing symmetries mathematically. Since every element of this group can be expressed as the composition of finitely many elements of the set $\Sigma = \{\sigma_1,\ldots,\sigma_{\lceil \frac{n}{2} \rceil}\}$, the group is said to be *generated* by Σ and is denoted $\langle\Sigma\rangle$.

Similarly, the *subgroups* of $\langle\Sigma\rangle$, that is, subsets that satisfy the group axioms, can be used to represent the symmetries of partial networks created in the process of constructing a full sorting network. In particular, computing pairs of dual output functions produces symmetries corresponding to a subgroup of $\langle\Sigma\rangle$ (Figure 5). Since each symmetry element in Σ operates on disjoint pairs of dual functions, any such subgroup can be written as $\langle\Gamma\rangle$, where Γ is a subset of Σ.

Initially, before any comparators have been added, each line i in the network has the trivial monotone Boolean function x_i. As a result, the network does not have any symmetries, that is, $\Gamma = \{\}$. Adding comparators to compute the output function f_i and its dual f_{n+1-i} yields $\Gamma = \{\sigma_i\}$ for the resulting partial network. Adding more comparators to compute both f_j and its dual f_{n+1-j} creates a new partial network with $\Gamma = \{\sigma_i, \sigma_j\}$, that is, the new partial network is more symmetric. Continuing to add comparators until all output functions have been constructed produces a complete sorting network with $\Gamma = \Sigma$.

Thus adding comparators to the network in a particular sequence builds its symmetry in a corresponding sequence of increasingly larger subgroups. Conversely, building symmetry in a particular sequence constrains the comparator sequences that are possible. Symmetry can therefore be used to constrain the search space for designing networks with desired properties. In particular, a sequence of subgroups can represent a sequence of subgoals for minimizing the number of comparators in the network. Each subgoal in this sequence is defined as the subgroup that can be produced from the previous subgoal by adding the fewest number of comparators.

Applying this heuristic to the initial network with symmetry $\Gamma = \{\}$, the first subgoal is defined as the symmetry that can be produced from the input variables by computing a pair of dual output functions with the fewest number of comparators (Figure 6). The functions $f_1 = x_1 \wedge \ldots \wedge x_n$ and $f_n = x_1 \vee \ldots \vee x_n$ have the fewest number of variable combinations and can therefore be computed by adding fewer comparators than any other pair of dual output functions. Thus the first subgoal is to produce the symmetry $\Gamma = \{\sigma_1\}$ using as few comparators as possible.

After computing f_1 and f_n, the next pair of dual output functions with the fewest number of variable combinations are f_2 and f_{n-1}. Therefore, the second subgoal is to compute them and produce the symmetry $\Gamma = \{\sigma_1, \sigma_2\}$. In this way, the number of variable combinations in the output functions continues to increase from the outer lines to the middle lines of the network. Therefore, from any subgoal that adds the symmetry σ_k to Γ, the next subgoal adds the symmetry σ_{k+1} to Γ. This sequence of subgoals continues until all the output functions are computed, producing the final goal symmetry $\Gamma = \{\sigma_1, \ldots, \sigma_{\lceil \frac{n}{2} \rceil}\}$.

Although this subgoal sequence specifies the order in which to compute the output functions, it does not specify an optimal combination of comparators for each subgoal. However, it is easier to minimize the number of comparators required for each subgoal than for the entire network, as will be described next.

3.3 Minimizing Comparator Requirement

In order to reach the first subgoal, the same comparator can compute a conjunction for f_1 and also a disjunction for f_n simultaneously (Figure 6). Sharing the same comparator to compute dual functions in this manner reduces the number of comparators required in the network. However, such sharing between dual functions of the same subgoal is possible only in some cases. In other cases, it may still be possible to share a comparator with the dual function of a later subgoal. Thus,

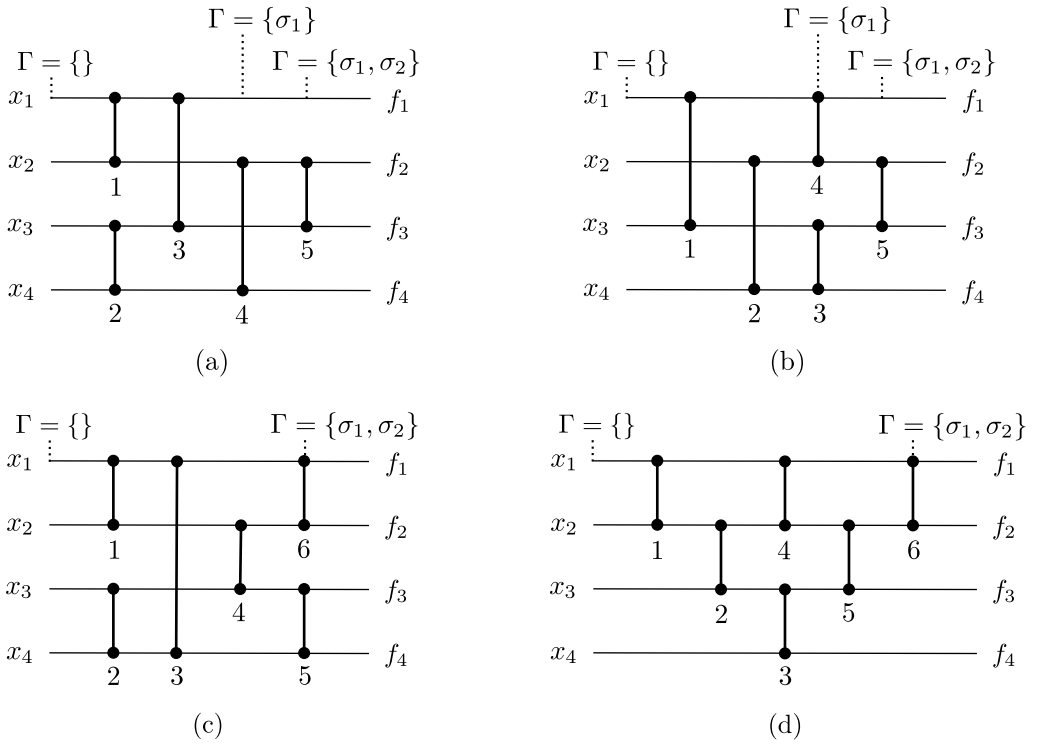


Figure 5: Symmetries of 4-input sorting networks. The numbers below the comparators indicate the sequence in which the comparators are added during network construction. The last comparator touching horizontal line i completes computing the output function f_i for that line. Functions f_i and f_{n+1-i} form a dual, and computing them both gives the network the symmetry σ_i. In network (a), adding comparator 3 completes computing f_1 and when comparator 4 is added to complete computing its dual f_4, the network gets the symmetry σ_1. Adding comparator 5 then completes computing both f_2 and its dual f_3, giving the network its second symmetry σ_2. Network (b) also produces the same sequence of symmetries and has the same number of comparators. In network (c), adding comparator 5 completes computing both f_3 and f_4, but not their duals f_1 and f_2. Only when comparator 6 is added to complete computing f_1 and f_2 does it get both its symmetries σ_1 and σ_2. Network (d) is similar to (c), and they both require one more comparator than networks (a) and (b). Thus the sequence in which the comparators are added determines the sequence in which the network gets its symmetries. Conversely, a preferred sequence of symmetries can be specified to constrain the sequence in which comparators are added and to minimize the number of comparators required.

minimizing the number of comparators requires determining which comparators can be shared and then adding those comparators that maximize sharing.

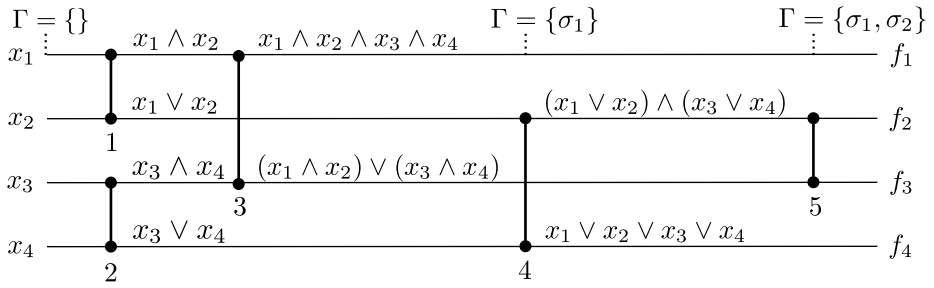


Figure 6: Subgoals for constructing a minimal 4-input sorting network. The final goal is to produce the symmetry $\Gamma = \{\sigma_1, \sigma_2\}$ by computing all four output functions f_i while using the minimum number of comparators. This goal can be decomposed into a sequence of subgoals specified as subgroups of the final symmetry group $\langle\Gamma\rangle$. At any stage in the construction, the next subgoal is the subgroup that can be produced by adding the fewest number of comparators. Initially, the network does not have any symmetries, that is, $\Gamma = \{\}$. The dual functions f_1 and f_4 are the easiest to compute, having fewer variable combinations and therefore requiring fewer comparators than f_2 and f_3. Hence the first subgoal is to produce the symmetry $\Gamma = \{\sigma_1\}$. Notice that comparators 1 and 2 compute parts of both f_1 and f_4 to achieve this subgoal with the minimum number of comparators. The second subgoal is to produce the symmetry $\Gamma = \{\sigma_1, \sigma_2\}$ by computing the functions f_2 and f_3. Adding comparator 5 completes this subgoal since comparators 3 and 4 have already computed f_2 and f_3 partially. Optimizing the number of comparators required to reach each subgoal separately in this way makes it possible to scale the approach to networks with more inputs.

The Boolean lattice representation of functions discussed in Section 3.1 can be used to determine whether or not sharing a comparator for computing parts of two functions simultaneously is possible (Figure 7). Assume that the current subgoal is to compute the output function f_i and its dual f_{n+1-i}. That is, functions for outputs less than i and greater than $n+1-i$ have already been fully computed, implying that each of these functions f_j has the value 1 at all nodes in levels less than or equal to j and the value 0 everywhere else. Moreover, the functions for the remaining outputs have been partially computed. In particular, each of these intermediate functions are guaranteed to have the value 1 at all nodes in levels less than or equal to i and the value 0 at all nodes in levels greater than $n+1-i$. If that was not the case, it will be impossible to compute at least one of the remaining output functions by adding more comparators since conjunctions preserve 0s and disjunctions preserve 1s of the intermediate functions they combine.

The current subgoal of computing function f_i requires setting its value at all nodes in level i to 1 and its value at all nodes in level $i+1$ to 0, thus defining its node boundary in the lattice. Its monotonicity then implies that it has the value 1 at all nodes in levels less than i and the value 0 at all nodes in levels greater than $i+1$. Moreover, since the intermediate functions f'_j on lines $i \leq j \leq n+1-i$ already have the value 1 at all nodes in levels less than or equal to i, computing f_i from them will retain that value at those nodes automatically. Therefore, f_i can be computed just by setting its value at all nodes in level $i+1$ to 0.

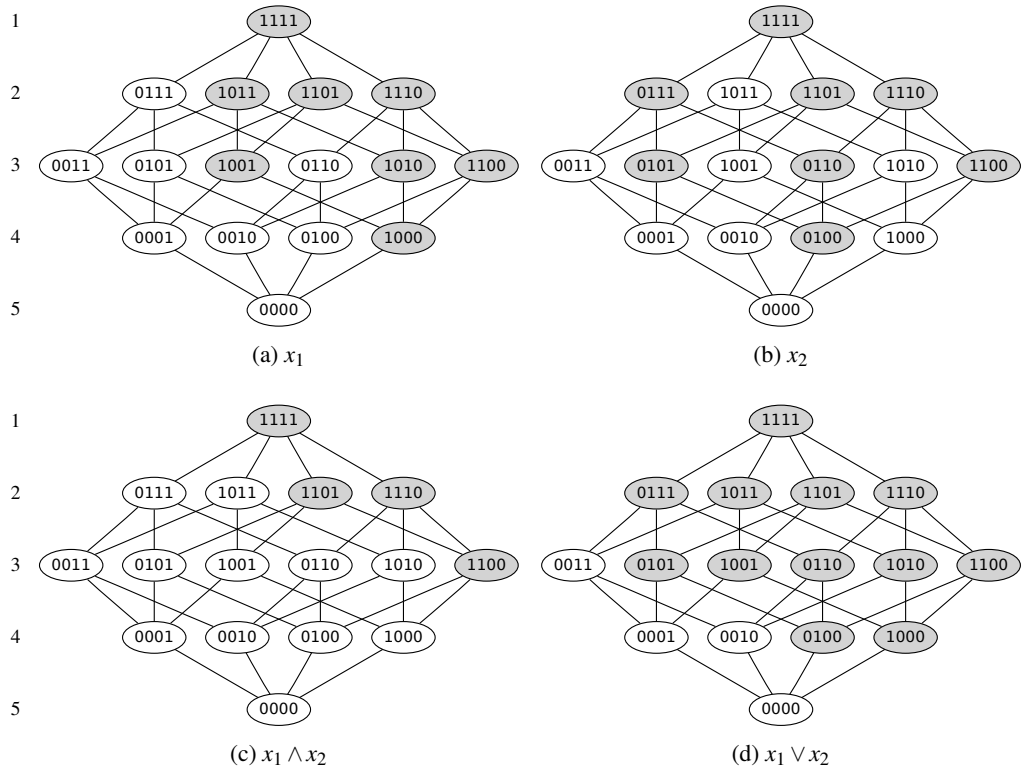


Figure 7: Comparator sharing to compute dual output functions in a 4-input sorting network. This figure illustrates the Boolean lattice representation of the functions computed by comparator 1 in Figure 6. The levels of the lattices are numbered on the left and the nodes at which the function takes the value 1 are shaded. Comparator 1 computes the conjunction (c) and the disjunction (d) of the functions (a) and (b) for the subgoal of computing the output functions $f_1 = x_1 \wedge x_2 \wedge x_3 \wedge x_4$ and $f_4 = x_1 \vee x_2 \vee x_3 \vee x_4$. Function f_1 can be computed by using conjunctions to set its value at all nodes in level 2 of the lattice to 0. Similarly, f_4 can be computed by using disjunctions to set its value at all nodes in level 4 to 1. Thus, comparator 1 contributes to computing both f_1 and f_4 by setting the values at two nodes in level 2 of its conjunction to 0 and the values at two nodes in level 4 of its disjunction to 1. Sharing comparators in this manner reduces the number of comparators required to construct the sorting network.

The value of f_i at a node in level $i+1$ can be set to 0 by adding a comparator that computes its conjunction with another function that already has the value 0 at that node, thus increasing the number of 0-valued nodes. The disjunction that this comparator also computes has fewer 0-valued nodes than either of its input functions and is therefore not useful for computing f_i. However, the disjunction will be used to compute the other remaining output functions, implying that it has the value 1 at all nodes in level $i+1$ as required by those functions. Since the disjunction does not have any 0-valued nodes in level $i+1$, its inputs do not have any common 0-valued nodes in that level. That is, exactly one of the intermediate functions f'_j has the value 0 for any particular node in level

$i+1$. Adding a comparator between a pair of such functions collects the 0-valued nodes from both functions in their conjunction. Repeating this process recursively collects the 0-valued nodes in level $i+1$ from all functions to the function on line i, thus producing f_i. Similarly, its dual function f_{n+1-i} can be computed from the functions f'_j by using disjunctions instead of conjunctions to set its values at all nodes in level $n+1-i$ to 1.

The leaves of the resulting binary recursion tree for f_i are the functions f'_j that have 0-valued nodes in level $i+1$ and its internal nodes are the conjunctive comparator outputs. Since the number of nodes of degree two in a binary tree is one less than the number of leaves (Mehta and Sahni, 2005), the number of comparators required depends only on the number of functions with which the recursion starts, that is, it is invariant to the order in which the recursion pairs the leaves. However, the recursion trees for f_i and f_{n+1-i} may have common leaves, making it possible to use the same comparator to compute a conjunction for f_i and a disjunction for f_{n+1-i}. Maximizing such sharing of comparators between the two recursion trees minimizes the number of comparators required for the current subgoal.

It may also be possible to share a comparator with a later subgoal, for example, when it computes a conjunction for f_i and a disjunction for f_{n+1-k}, where $i < k \leq \lceil \frac{n}{2} \rceil$. In order to prioritize subgoals and determine which comparators maximize sharing, each pair of lines where a comparator can potentially be added is assigned a utility. Comparators that contribute to both f_i and f_{n+1-i} for the current subgoal get the highest utility. Comparators that contribute to an output function for the current subgoal and an output function for the next subgoal get the next highest utility. Similarly, other comparators are also assigned utilities based on the output functions to which they contribute and the subgoals to which those output functions belong. Many comparators may have the same highest utility; therefore, one comparator is chosen randomly from that set and it is added to the network. Repeating this process produces a sequence of comparators that optimizes sharing within the current subgoal and between the current subgoal and later subgoals.

Optimizing for each subgoal separately in this manner constitutes a greedy algorithm that produces minimal-size networks with high probability for $n \leq 8$. However, for larger values of n, the search space is too large for this greedy approach to find a global optimum reliably. In such cases, stochastic search such as evolution can be used to explore the neighborhood of the greedy solutions for further optimization, as will be described next.

3.4 Evolving Minimal-Size Networks

The most straightforward approach is to initialize evolution with a population of solutions that the greedy algorithm produces. The fitness of each solution is the negative of its number of comparators so that improving fitness will minimize the number of comparators. In each generation, two-way tournament selection based on this fitness measure is used to select the best individuals in the population for reproduction. Reproduction mutates the parent network, creating an offspring network in two steps: (1) a comparator is chosen from the network randomly and the network is truncated after it, discarding all later comparators, and (2) the greedy algorithm is used to add comparators again, reconstructing a new offspring network. Since the greedy algorithm chooses a comparator with the highest utility randomly, this mutation explores a new combination of comparators that might be more optimal than the parent.

This straightforward approach restricts the search to the space of comparator combinations suggested by the greedy algorithm and assumes that it contains a globally minimal network. In some

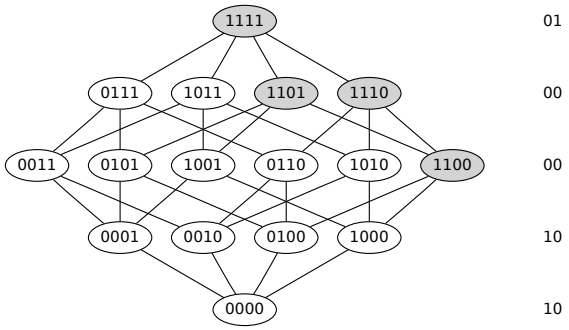


Figure 8: State representation of the function $x_1 \wedge x_2$ used in the EDA. The state (shown on the right) is a bit-string with two bits for each level of the Boolean lattice. The first bit is 1 only if the value of the function for all nodes in that level is 0 and the second bit is 1 only if its value for all nodes in that level is 1. This condensed representation of the function is based on the information used by the symmetry-building greedy algorithm and it is therefore useful for constructing minimal-size sorting networks.

cases, however, the globally minimal networks may use comparators that are different from those suggested by the greedy algorithm. Therefore, a more powerful (but still brute force) approach is to let evolution use such comparators as well: with a probability determined empirically, the suggestions of the greedy algorithm are ignored and instead the next comparator to be added to the network is selected randomly from the set of all potential comparators.

A more effective way to combine evolution with such departures from the greedy algorithm is to use an Estimation of Distribution Algorithm (EDA) (Bengoetxea et al., 2001; Alden, 2007; Mühlenbein and Höns, 2005). The idea is to estimate the probability distribution of comparator combinations in the smallest networks evolved thus far and to use this distribution to generate comparator suggestions for the next generation. The EDA is initialized as before with a population of networks generated by the greedy algorithm. In each generation, a set of networks with the highest fitness are selected from the population. These networks are used in three ways: (1) to estimate the distribution of comparators for a generative model of small networks, (2) as elite networks, passed unmodified to the next generation, and (3) as parent networks, from which new offspring networks are created for the next generation.

The generative model of the EDA specifies the probability $P(C|S)$ of adding a comparator C to an n-input partial network with state S. The state of a partial network is defined in terms of the n Boolean functions that its comparators compute. These functions determine the remaining comparators that are needed to finish computing the output functions, making them a good representation of the partial network. However, storing the state as the concatenation of the n functions is computationally intractable since each function is represented as a vector of 2^n bits. Therefore, a condensed state representation is computed based on the observation that the greedy algorithm does not use the actual function values for the nodes in the Boolean lattice; it only checks whether the values in a given level are all 0s or all 1s. This information, encoded as $2(n+1)$ bits (Figure 8), is suitable as the state representation for the model as well.

Since the model is estimated from the set of the smallest networks in the population, it is likely to generate small networks as well. Although it can generate new networks from scratch, it is used as part of the above reproduction mechanism to reconstruct a new offspring network from the truncated parent network, that is, it is used in Step 2 of reproduction instead of the greedy algorithm. In this step, some comparators are also chosen randomly from the set of all potential comparators to encourage exploration of comparator combinations outside the model. Moreover, if the model does not generate any comparators for the current state, then the reconstruction step falls back to the greedy algorithm for adding a comparator.

As discussed in Section 3.3, the greedy algorithm chooses the comparator to be added to the network randomly from those that have the highest utility (Variant 1). This random choice can be modified slightly to prefer comparators that are symmetric with respect to another comparator that is already in the network (Variant 2). Doing so makes the arrangement of comparators more bilaterally symmetric about a horizontal axis through the middle of the network. This heuristic was motivated by Graham and Oppacher (2006), who found that biasing evolutionary search using such symmetric comparator pairs was beneficial. The EDA works well with both of these variants of the greedy algorithm, learning to find smaller sorting networks than previous results, as demonstrated next.

4. Results

SENSO was run with a population size of 200 for 500 generations to evolve minimal-size networks for different input sizes. In each generation, the top half of the population (i.e., 100 networks with the fewest number of comparators) was selected for estimating the model. The same set of networks was copied to the next generation without modification. Each of them also produced an offspring network to replace those in the bottom half of the population. A Gaussian probability distribution was used to select the comparator from which to truncate the parent network. This Gaussian distribution was centered at the middle of its comparator sequence with a standard deviation of one-fourth of its number of comparators. As a result, parent networks were more likely to be truncated near the middle than near the ends. When reconstructing the truncated network, the next comparator to be added to the network was generated either by the estimated model (with probability 0.5) or was selected randomly from the set of all potential comparators (with probability 0.5). Results were insensitive to small changes in these probabilities. The SENSO source code to run this experiment is available from the website `http://nn.cs.utexas.edu/?sorting-code`.

The above experiment was repeated 20 times for each variant of the greedy algorithm and for each input size $n \leq 23$, each time with a different random number seed. The smallest network found in each set of 20 runs was recorded as the result for that particular combination of algorithmic variant and input size. This procedure was repeated 25 times for each set of 20 runs to determine which of the two variants produced smaller networks. According to the Mann-Whitney U-test, the median number of comparators in the smallest networks found by variant 2 was significantly fewer for input sizes 13, 15, 18, 20, 22 ($p < 0.02$, one-tailed). There was no significant difference between the two variants for the other input sizes. That is, the symmetry heuristic used in variant 2 makes it better or as good as variant 1 for finding small networks.

The fewest number of comparators found for each input size is listed in Table 4. For input sizes $n \leq 11$, the initial population of SENSO already contained networks with the smallest-known sizes, that is, the greedy algorithm was sufficient to find the smallest-known networks. For input sizes 12

n	12	13	14	15	16	17	18	19	20	21	22	23
Previous best	Hand-design and END					Batcher's and Van Voorhis' merge						
	39	45	51	56	60	73	79	88	93	103	110	118
SENSO	*39*	*45*	*51*	*56*	*60*	**71**	**78**	**86**	**92**	**102**	**108**	*118*

Table 4: Sizes of the smallest networks for different input sizes found by SENSO. For input sizes $n \leq 11$, networks with the smallest-known sizes (Section 2) were already found in the initial population of SENSO, that is, the greedy algorithm using symmetry was sufficient. These sizes are therefore omitted from this table. For larger input sizes, evolution found networks that matched previous best results (indicated in *italics*) or improved them (indicated in **bold**). Appendix A lists examples of these networks. These results demonstrate that the SENSO approach is effective at designing minimal-size sorting networks. Prospects of extending these results to input sizes greater than 23 will be discussed in Section 5.

to 16, and 23, SENSO evolved networks that have the same size as the best known networks. For 15 inputs, networks matching previous best results were obtained indirectly by removing the bottom line of the evolved 16-input networks and all comparators touching that line (Knuth, 1998). Most importantly, SENSO improved the previous best results for input sizes 17, 18, 19, 20, 21, and 22 by one or two comparators. Examples of these networks are listed in Appendix A.

For 23 inputs, SENSO required about 4GB of memory and 46 hours to complete 500 generations on a Xeon X5440 processor running at 2.83GHz. These requirements approximately double for every unit increase in the number of inputs due to the $O(2^n)$ complexity of the algorithm. Prospects for mitigating the effects of this exponential growth and for extending the results to $n > 23$, including to larger power-of-two networks, will be discussed in Section 5.

The previous best results for input sizes 12 through 16 were obtained either by hand design or by the END evolutionary algorithm (Knuth, 1998; Juillé, 1995; Van Voorhis, 1971; Baddar, 2009). The END algorithm improved a 25-year old result for the 13-input case by one comparator and matched the best known results for other input sizes up to 16. However, it is a massively parallel search algorithm, requiring very large computational resources, for example, a population size of 65,536 on 4096 processors to find minimal-size networks for 13 and 16 inputs (Table 5). In contrast, the SENSO approach finds such networks with much less resources (e.g., population size of 200 on a single processor in a similar number of generations), making it promising for larger problems, as will be discussed in the next section.

5. Discussion and Future Work

Previous results on designing minimal-size networks automatically by search have been limited to small input sizes ($n \leq 16$) because the number of valid sorting networks near the optimal size is very small compared to the combinatorially large space that has to be searched (Juillé, 1995). The symmetry-building approach presented in Section 3 mitigates this problem by using symmetry to focus the search on the space of networks near the optimal size. As a result, it was possible to search for minimal-size networks with more inputs ($n \leq 23$), improving the previous best results in five cases.

	END	SENSO (variant 2)
Processor family	MasPar MP-2	Xeon X5440
Number of processors used	4,096 @ 17Gop/s	1 @ 2.83GHz
Memory used	unknown	37MB
Population size	65,536	200
Runs that produced 60 comparators	2 out of 3	18 out of 20
Number of generations	300 to 500	500
Execution time for each run	48 to 72 hours	15 min

Table 5: Performance metrics of END and SENSO for the 16-input problem. This table compares the performance of SENSO with the END algorithm (Juillé, 1995) for finding 16-input networks with 60 comparators, which is the best known result for that input size. In contrast to the massively parallel END algorithm, SENSO finds such networks using much less computational resources.

These improvements can be transferred to larger values of n by using Batcher's or Van Voorhis' merge to construct such larger networks from the improved smaller networks (Knuth, 1998). The resulting networks accumulate the combined improvement of the smaller networks. For example, since the 22-input network has been improved by two comparators, two copies of it can be merged to construct a 44-input network with four fewer comparators than previous results. This merging procedure can be repeated to construct even larger networks, doubling the improvement in each step. Such networks are useful in massively parallel applications such as sorting in GPUs with hundreds of cores (Kipfer and Westermann, 2005).

It should be possible to improve these results further by extending SENSO in the following ways. First, the greedy algorithm for adding comparators can be improved by evaluating the sharing utility of groups of one or more comparators instead of single comparators. Such groups that have the highest average utility will then be preferred.

Second, the greedy algorithm can be made less greedy by considering the impact of current comparator choices on the number of comparators that will be required for later subgoals. This analysis will make it possible to optimize across subgoals, potentially producing smaller networks at the cost of additional computations.

Third, the state representation that the EDA algorithm uses contains only sparse information about the functions computed by the comparators. Extending it to include more relevant information should make it possible for the EDA to disambiguate overlapping states and therefore to model comparator distribution more accurately.

Fourth, in some cases, good n-input networks can be obtained from $n+1$-input networks by simply removing its bottom line and all comparators touching that line (Knuth, 1998), as was done in the 15-input case in this paper. This observation suggests that a potentially more powerful approach is to augment the information contained in the state representation of the EDA with the comparator distribution for multiple input sizes.

Fifth, the EDA generates comparators to add to the network only if the state of the network matches a state in the generative model exactly. Making this match graded, based on some similarity measure, may produce better results by exploring similar states when an exact match is not found.

Sixth, evolutionary search can be parallelized, for example, using the massively parallel END algorithm that has been shown to evolve the best known network sizes for $n \leq 16$ (Juillé, 1995). Conversely, using the symmetry-building approach to constrain the search space should make it possible to run the END algorithm on networks with more inputs.

Seventh, the symmetry-building approach itself can be improved. For example, it uses only the symmetries resulting from the duality of the output functions. It may be possible to extend this approach by also using the symmetries resulting from the permutations of the input variables.

Eighth, large networks can be constructed from smaller networks by merging the outputs of the smaller networks. Since smaller networks are easier to optimize, they can be evolved first and then merged by continuing evolution to add more comparators. This approach is similar to constructing minimal networks for $n > 16$ by merging smaller networks (Batcher, 1968; Van Voorhis, 1974).

In addition to finding minimal-size networks, the SENSO approach can also be used to find minimal-delay networks. Instead of minimizing the number of comparators, it would now minimize the number of parallel steps into which the comparators are grouped. Both these objectives can be optimized simultaneously as well, either by preferring one objective over the other in the fitness function or by using a multi-objective optimization algorithm such as NSGA-II (Deb et al., 2000).

Moreover, this approach can potentially be extended to design comparator networks for other related problems such as rank-order filters (Chakrabarti and Wang, 1994; Hiasat and Hasan, 2003; Chung and Lin, 1997). A rank order filter with rank r selects the r^{th} largest element from an input set of n elements. Such filters are widely used in image and signal processing applications, for example, to reduce high-frequency noise while preserving edge information. Since these filters are often implemented in hardware, minimizing their comparator requirement is necessary to minimize their chip area. More generally, similar symmetry-based approaches may be useful for designing stack filters, that is, circuits implementing monotone Boolean functions, which are also popular in signal processing applications (Hiasat and Hasan, 2003; Shmulevich et al., 1995). Furthermore, such approaches can potentially be used to design rearrangeable networks for switching applications (Seo et al., 1993; Yeh and Feng, 1992).

6. Conclusion

Minimizing the number of comparators in a sorting network is a challenging optimization problem. This paper presented an approach called SENSO that simplifies it by converting it into the problem of building the symmetry of the network optimally. The resulting structure makes it possible to construct the network in steps and to minimize the number of comparators required for each step separately. However, the networks constructed in this manner may be sub-optimal greedy solutions, and they are optimized further by an evolutionary algorithm that learns to anticipate the distribution of comparators in minimal networks. This approach focuses the solutions on promising regions of the search space, thus finding smaller networks more effectively than previous methods.

Acknowledgments

We would like to thank Greg Plaxton for useful suggestions on formalizing the problem. This research was supported in part by the National Science Foundation under grants IIS-0915038, IIS-

0757479, and EIA-0303609; the Texas Higher Education Coordinating Board under grant 003658-0036-2007; and the College of Natural Sciences.

Appendix A. Evolved Minimal-Size Sorting Networks

This appendix lists examples of minimal-size sorting networks evolved by SENSO. For each example, the sequence of comparators is illustrated in a figure and also listed as pairs of horizontal lines numbered from top to bottom.

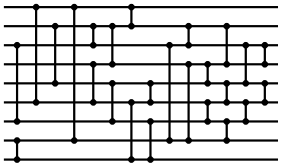


Figure 9: Evolved 9-input network with 25 comparators: [3, 7], [1, 6], [2, 5], [8, 9], [1, 8], [2, 3], [4, 6], [5, 7], [6, 9], [2, 4], [7, 9], [1, 2], [5, 6], [3, 8], [4, 8], [4, 5], [6, 7], [2, 3], [2, 4], [7, 8], [5, 6], [3, 5], [6, 7], [3, 4], [5, 6].

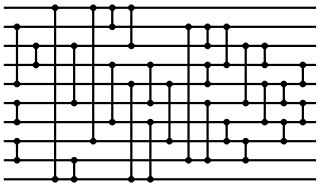


Figure 10: Evolved 10-input network with 29 comparators: [2, 5], [8, 9], [3, 4], [6, 7], [1, 10], [3, 6], [1, 8], [9, 10], [4, 7], [5, 10], [1, 2], [1, 3], [7, 10], [4, 6], [5, 8], [2, 9], [4, 5], [6, 9], [7, 8], [2, 3], [8, 9], [2, 4], [3, 6], [5, 7], [3, 4], [7, 8], [5, 6], [4, 5], [6, 7].

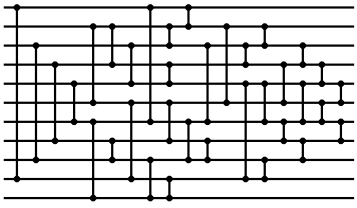


Figure 11: Evolved 11-input network with 35 comparators: [1, 10], [3, 9], [4, 8], [5, 7], [2, 6], [2, 4], [3, 5], [7, 11], [8, 9], [6, 10], [1, 7], [2, 3], [9, 11], [10, 11], [1, 2], [6, 8], [4, 5], [7, 9], [3, 7], [2, 6], [8, 9], [5, 10], [3, 4], [9, 10], [2, 3], [5, 7], [4, 6], [7, 8], [8, 9], [3, 4], [5, 7], [6, 7], [4, 5], [7, 8], [5, 6].

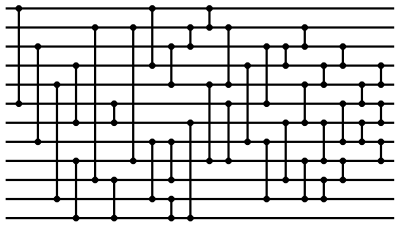


Figure 12: Evolved 12-input network with 39 comparators: [1, 6], [3, 8], [5, 11], [4, 7], [9, 12], [2, 10], [6, 7], [2, 9], [1, 4], [3, 5], [10, 12], [8, 11], [8, 10], [11, 12], [2, 3], [7, 12], [1, 2], [5, 9], [6, 9], [2, 5], [4, 8], [3, 6], [8, 11], [7, 10], [3, 4], [5, 7], [9, 11], [2, 3], [10, 11], [7, 9], [4, 5], [9, 10], [3, 4], [6, 8], [5, 6], [7, 8], [8, 9], [6, 7], [4, 5].

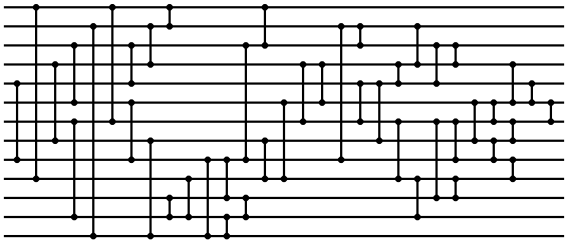


Figure 13: Evolved 13-input network with 45 comparators: [5, 9], [1, 10], [4, 8], [3, 6], [7, 12], [2, 13], [1, 7], [3, 5], [6, 9], [8, 13], [2, 4], [11, 12], [10, 12], [1, 2], [9, 13], [9, 11], [3, 9], [12, 13], [1, 3], [8, 10], [6, 10], [4, 7], [4, 6], [2, 9], [5, 7], [5, 8], [11, 12], [7, 10], [4, 5], [2, 3], [10, 12], [2, 4], [7, 11], [3, 5], [3, 4], [10, 11], [7, 9], [6, 8], [6, 7], [8, 9], [4, 6], [9, 10], [5, 6], [7, 8], [6, 7].

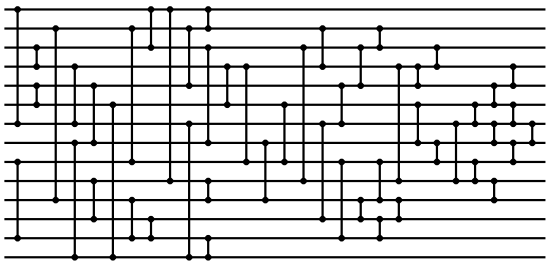


Figure 14: Evolved 14-input network with 51 comparators: [1, 7], [3, 4], [9, 13], [5, 6], [2, 11], [8, 14], [10, 12], [4, 7], [5, 8], [6, 14], [2, 9], [11, 13], [1, 3], [12, 13], [1, 10], [2, 5], [7, 14], [13, 14], [1, 2], [3, 8], [4, 6], [10, 11], [4, 9], [8, 11], [6, 9], [3, 10], [7, 12], [5, 7], [9, 13], [2, 4], [11, 12], [3, 5], [12, 13], [2, 3], [9, 11], [4, 10], [4, 5], [3, 4], [11, 12], [6, 8], [8, 9], [7, 10], [6, 7], [5, 6], [9, 10], [7, 8], [10, 11], [4, 5], [6, 7], [8, 9], [7, 8].

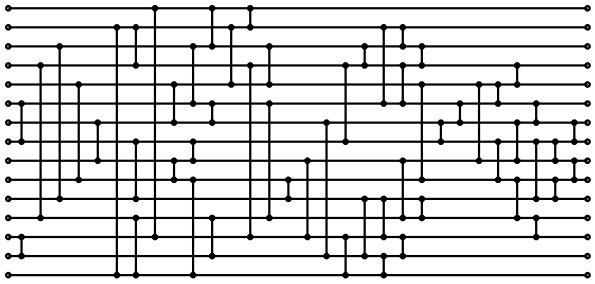


Figure 15: Evolved 15-input network with 56 comparators: [13, 14], [6, 8], [4, 12], [3, 11], [5, 10], [7, 9], [2, 15], [12, 15], [2, 4], [8, 11], [1, 13], [5, 7], [3, 6], [9, 10], [1, 3], [10, 15], [2, 5], [1, 2], [6, 7], [8, 9], [12, 14], [4, 13], [6, 12], [10, 11], [9, 13], [3, 5], [7, 14], [4, 8], [3, 4], [13, 15], [11, 14], [2, 6], [14, 15], [2, 3], [4, 6], [11, 13], [13, 14], [3, 4], [9, 12], [5, 10], [11, 12], [7, 8], [6, 7], [5, 9], [8, 10], [5, 6], [10, 12], [12, 13], [4, 5], [7, 9], [8, 11], [10, 11], [6, 7], [8, 9], [9, 10], [7, 8].

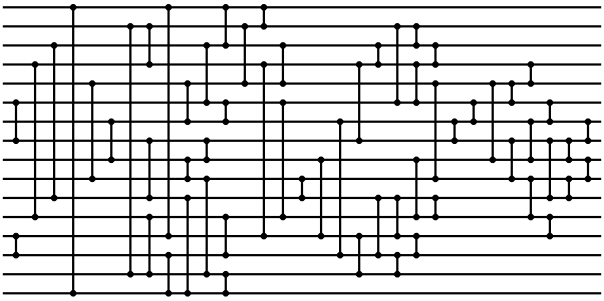


Figure 16: Evolved 16-input network with 60 comparators: [13, 14], [6, 8], [4, 12], [3, 11], [1, 16], [5, 10], [7, 9], [2, 15], [12, 15], [2, 4], [8, 11], [1, 13], [5, 7], [3, 6], [9, 10], [14, 16], [11, 16], [1, 3], [10, 15], [2, 5], [1, 2], [15, 16], [6, 7], [8, 9], [12, 14], [4, 13], [6, 12], [10, 11], [9, 13], [3, 5], [7, 14], [4, 8], [3, 4], [13, 15], [11, 14], [2, 6], [14, 15], [2, 3], [4, 6], [11, 13], [13, 14], [3, 4], [9, 12], [5, 10], [11, 12], [7, 8], [6, 7], [5, 9], [8, 10], [5, 6], [10, 12], [12, 13], [4, 5], [7, 9], [8, 11], [10, 11], [6, 7], [8, 9], [9, 10], [7, 8].

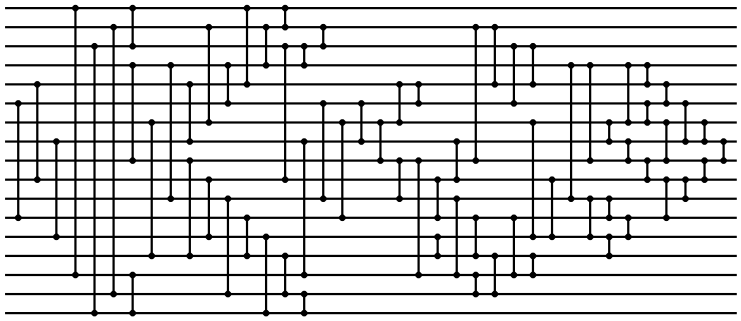


Figure 17: Evolved 17-input network with 71 comparators: [6, 12], [5, 10], [8, 13], [1, 15], [3, 17], [2, 16], [4, 9], [7, 14], [4, 11], [9, 14], [5, 8], [10, 13], [1, 3], [15, 17], [2, 7], [11, 16], [4, 6], [12, 14], [1, 5], [13, 17], [2, 4], [14, 16], [1, 2], [16, 17], [3, 10], [8, 15], [6, 11], [7, 12], [6, 8], [7, 9], [9, 11], [3, 4], [9, 15], [10, 12], [13, 14], [5, 7], [11, 15], [5, 6], [8, 10], [12, 14], [2, 3], [15, 16], [2, 9], [14, 16], [2, 5], [3, 6], [12, 15], [14, 15], [3, 5], [7, 13], [10, 13], [4, 11], [4, 9], [7, 8], [11, 13], [4, 7], [4, 5], [13, 14], [11, 12], [6, 7], [12, 13], [5, 6], [8, 9], [9, 10], [7, 9], [10, 12], [6, 8], [7, 8], [10, 11], [9, 10], [8, 9].

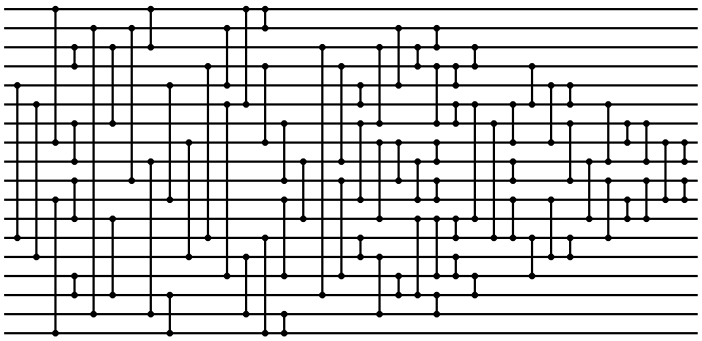


Figure 18: Evolved 18-input network with 78 comparators: [5, 13], [6, 14], [1, 8], [11, 18], [3, 4], [15, 16], [7, 9], [10, 12], [2, 17], [3, 7], [12, 16], [2, 10], [9, 17], [5, 11], [8, 14], [4, 13], [6, 15], [1, 3], [16, 18], [2, 5], [14, 17], [1, 6], [13, 18], [1, 2], [17, 18], [4, 8], [11, 15], [7, 10], [9, 12], [3, 16], [4, 9], [10, 15], [5, 6], [13, 14], [7, 11], [3, 7], [8, 12], [2, 5], [14, 17], [15, 16], [3, 4], [12, 16], [16, 17], [2, 3], [12, 15], [4, 7], [14, 15], [4, 5], [15, 16], [3, 4], [6, 7], [12, 13], [8, 10], [9, 11], [10, 11], [8, 9], [6, 12], [7, 13], [11, 13], [6, 8], [13, 15], [4, 6], [11, 14], [5, 8], [13, 14], [5, 6], [9, 10], [7, 10], [9, 12], [10, 13], [6, 9], [7, 8], [11, 12], [7, 9], [10, 12], [8, 11], [10, 11], [8, 9].

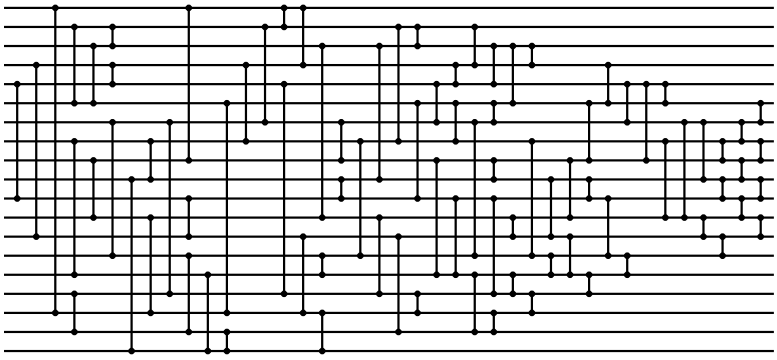


Figure 19: Evolved 19-input network with 86 comparators: [5, 11], [4, 13], [1, 17], [8, 15], [9, 12], [7, 14], [16, 18], [2, 6], [10, 19], [3, 6], [12, 17], [8, 10], [2, 3], [7, 16], [11, 13], [4, 5], [14, 18], [1, 9], [15, 19], [6, 17], [4, 8], [18, 19], [2, 7], [5, 16], [1, 2], [13, 17], [1, 4], [17, 19], [3, 12], [10, 11], [14, 15], [7, 9], [8, 14], [3, 10], [12, 16], [2, 8], [6, 11], [13, 18], [9, 15], [5, 7], [11, 15], [4, 5], [16, 17], [2, 3], [15, 18], [2, 4], [17, 18], [6, 8], [7, 14], [6, 7], [11, 16], [3, 5], [15, 16], [3, 6], [12, 13], [16, 17], [3, 4], [9, 10], [8, 14], [10, 13], [9, 12], [10, 11], [14, 15], [6, 9], [13, 15], [15, 16], [4, 6], [5, 7], [11, 14], [5, 9], [5, 6], [14, 15], [8, 12], [7, 12], [7, 10], [8, 9], [12, 13], [7, 8], [13, 14], [6, 7], [10, 11], [11, 12], [12, 13], [9, 10], [8, 9], [10, 11].

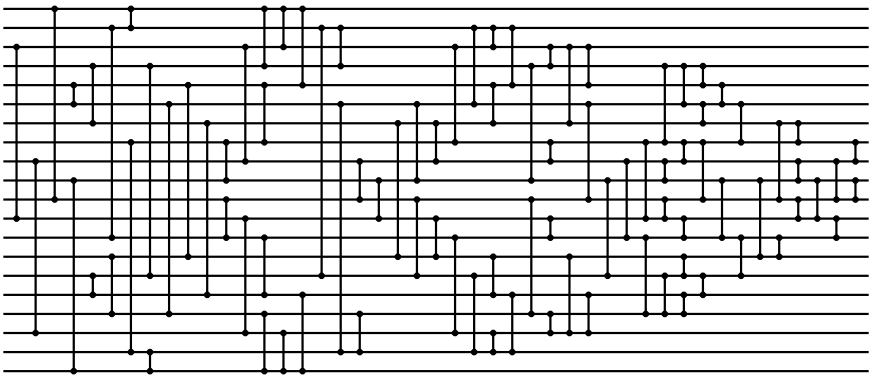


Figure 20: Evolved 20-input network with 92 comparators: [3, 12], [9, 18], [1, 11], [10, 20], [5, 6], [15, 16], [4, 7], [14, 17], [2, 13], [8, 19], [4, 15], [6, 17], [1, 2], [19, 20], [5, 14], [7, 16], [8, 10], [11, 13], [3, 9], [12, 18], [5, 8], [13, 16], [1, 4], [17, 20], [1, 3], [18, 20], [1, 5], [16, 20], [2, 15], [6, 19], [9, 11], [10, 12], [7, 14], [6, 10], [11, 15], [2, 4], [17, 19], [7, 9], [12, 14], [3, 8], [13, 18], [2, 6], [2, 3], [15, 19], [5, 7], [14, 16], [18, 19], [16, 19], [2, 5], [4, 10], [11, 17], [3, 4], [17, 18], [14, 18], [3, 7], [16, 18], [3, 5], [8, 9], [12, 13], [6, 11], [10, 15], [9, 13], [8, 12], [4, 8], [13, 17], [4, 6], [15, 17], [16, 17], [4, 5], [6, 7], [14, 15], [15, 16], [5, 6], [11, 12], [9, 10], [12, 13], [8, 9], [8, 11], [10, 13], [6, 8], [13, 15], [10, 14], [7, 11], [7, 8], [11, 12], [13, 14], [9, 10], [10, 12], [12, 13], [9, 11], [8, 9], [10, 11].

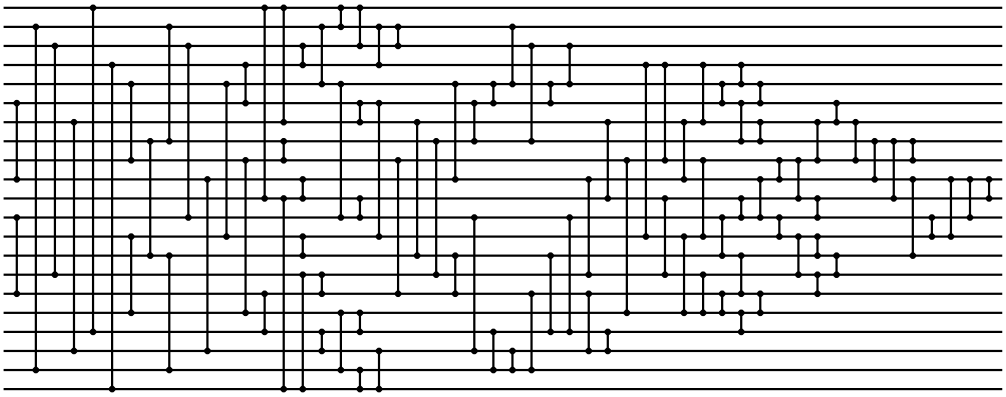


Figure 21: Evolved 21-input network with 102 comparators: [6, 10], [12, 16], [2, 20], [3, 15], [7, 19], [1, 18], [4, 21], [5, 9], [13, 17], [8, 14], [2, 8], [14, 20], [3, 12], [10, 19], [5, 13], [9, 17], [4, 6], [16, 18], [1, 11], [11, 21], [1, 7], [15, 21], [3, 4], [18, 19], [2, 5], [17, 20], [1, 2], [20, 21], [1, 3], [19, 21], [8, 9], [13, 14], [10, 11], [5, 12], [6, 7], [15, 16], [11, 12], [6, 13], [9, 16], [7, 14], [8, 15], [17, 18], [2, 4], [5, 10], [6, 8], [14, 16], [12, 19], [18, 20], [2, 3], [19, 20], [5, 6], [2, 5], [16, 20], [14, 18], [3, 8], [12, 18], [10, 15], [5, 6], [16, 19], [18, 19], [3, 5], [7, 11], [9, 17], [4, 13], [11, 15], [13, 17], [4, 9], [7, 10], [15, 17], [9, 13], [4, 7], [5, 6], [16, 17], [17, 18], [4, 5], [12, 14], [6, 8], [14, 16], [7, 8], [16, 17], [5, 6], [11, 12], [10, 12], [9, 10], [12, 13], [13, 15], [9, 11], [7, 9], [15, 16], [6, 7], [13, 14], [14, 15], [7, 9], [8, 10], [11, 12], [8, 11], [8, 9], [10, 14], [12, 13], [10, 13], [10, 12], [10, 11].

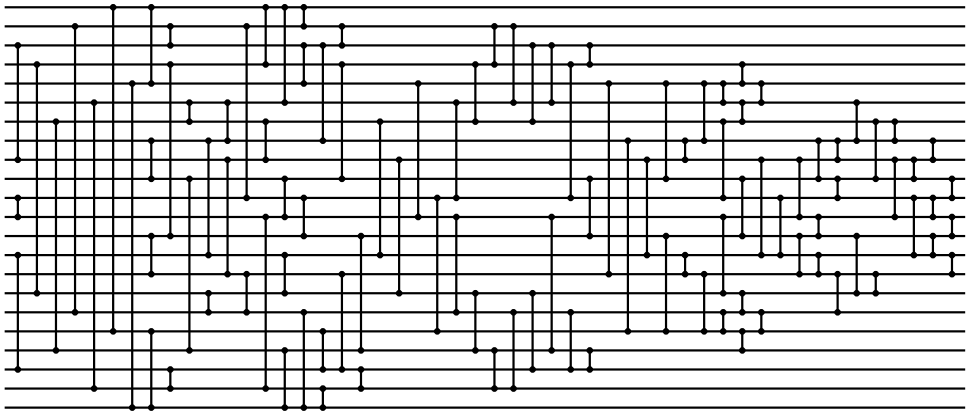


Figure 22: Evolved 22-input network with 108 comparators: [11, 12], [3, 9], [14, 20], [4, 16], [7, 19], [2, 17], [6, 21], [1, 18], [5, 22], [8, 10], [13, 15], [1, 5], [18, 22], [4, 13], [10, 19], [2, 3], [20, 21], [8, 14], [9, 15], [6, 7], [16, 17], [6, 8], [15, 17], [2, 11], [12, 21], [1, 4], [19, 22], [1, 6], [17, 22], [1, 2], [21, 22], [7, 9], [14, 16], [3, 5], [18, 20], [10, 12], [11, 13], [3, 8], [15, 20], [4, 10], [13, 19], [7, 14], [9, 16], [5, 12], [11, 18], [6, 11], [12, 17], [4, 7], [16, 19], [2, 3], [20, 21], [2, 4], [19, 21], [2, 6], [17, 21], [3, 7], [16, 20], [12, 19], [3, 6], [17, 20], [4, 11], [3, 4], [19, 20], [10, 13], [5, 15], [8, 18], [9, 14], [13, 18], [5, 10], [14, 15], [8, 9], [5, 8], [15, 18], [5, 6], [17, 18], [18, 19], [4, 5], [7, 11], [12, 16], [6, 7], [16, 17], [5, 6], [17, 18], [10, 13], [9, 14], [11, 14], [9, 12], [8, 10], [13, 15], [8, 9], [14, 15], [15, 17], [6, 8], [10, 11], [12, 13], [7, 10], [13, 16], [15, 16], [7, 8], [9, 12], [11, 14], [9, 10], [13, 14], [8, 9], [14, 15], [11, 12], [12, 13], [10, 11].

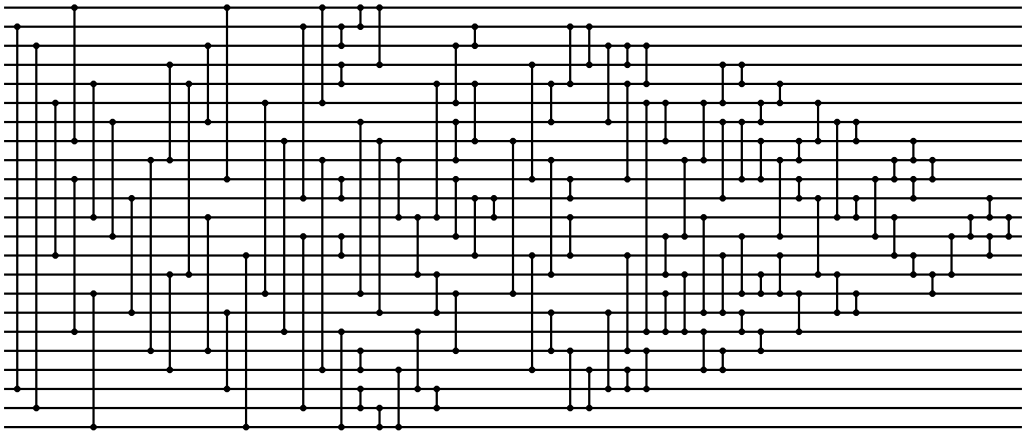


Figure 23: Evolved 23-input network with 118 comparators: [2, 21], [3, 22], [6, 14], [10, 18], [1, 8], [16, 23], [5, 12], [7, 13], [11, 17], [9, 19], [15, 20], [4, 9], [5, 15], [12, 19], [3, 7], [17, 21], [1, 10], [14, 23], [6, 16], [8, 18], [2, 11], [13, 22], [9, 20], [18, 23], [1, 6], [21, 22], [2, 3], [19, 20], [4, 5], [22, 23], [1, 2], [20, 23], [1, 4], [13, 14], [10, 11], [7, 16], [8, 17], [9, 12], [12, 15], [5, 12], [7, 9], [15, 17], [18, 21], [3, 6], [10, 13], [11, 14], [16, 19], [11, 12], [5, 8], [21, 22], [2, 3], [8, 16], [4, 10], [14, 20], [17, 19], [9, 15], [5, 7], [19, 22], [2, 5], [20, 22], [2, 4], [10, 11], [12, 14], [3, 7], [17, 21], [5, 10], [14, 19], [20, 21], [3, 4], [19, 21], [3, 5], [6, 18], [13, 15], [9, 13], [6, 8], [16, 18], [6, 9], [15, 18], [4, 6], [18, 20], [4, 5], [19, 20], [7, 11], [12, 17], [14, 17], [7, 10], [17, 18], [6, 7], [5, 6], [8, 10], [18, 19], [13, 16], [15, 16], [9, 13], [8, 9], [14, 16], [16, 18], [6, 8], [10, 11], [11, 15], [7, 12], [15, 17], [16, 17], [7, 8], [11, 12], [10, 13], [12, 14], [14, 15], [9, 10], [8, 9], [15, 16], [10, 11], [9, 10], [13, 15], [12, 13], [13, 14], [11, 12], [12, 13].

References

M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983. ISSN 0209-9683. doi: http://dx.doi.org.ezproxy.lib.utexas.edu/10.1007/BF02579338.

M. E. Alden. *MARLEDA: Effective Distribution Estimation Through Markov Random Fields*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, 2007. URL `http://nn.cs.utexas.edu/keyword?alden:phd07`. Technical Report AI07-349.

S. W. A. Baddar. *Finding Better Sorting Networks*. PhD thesis, Kent State University, 2009. URL `http://rave.ohiolink.edu/etdc/view?acc_num`.

K. E. Batcher. Sorting networks and their applications. In *AFIPS Spring Joint Computing Conference*, pages 307–314, 1968.

E. Bengoetxea, P. Larranaga, I. Bloch, and A. Perchant. Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 454–469. Springer, 2001. URL `http://dx.doi.org.ezproxy.lib.utexas.edu/10.1007/3-540-44745-8_30`.

C. Chakrabarti and L.-Y. Wang. Novel sorting network-based architectures for rank order filters. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):502–507, 1994. ISSN 1063-8210. doi: http://dx.doi.org.ezproxy.lib.utexas.edu/10.1109/92.335027.

K.-L. Chung and Y.-K. Lin. A generalized pipelined median filter network. *Signal Processing*, 63(1):101 – 106, 1997. ISSN 0165-1684. doi: DOI:10.1016/S0165-1684(97)00144-8. URL `http://www.sciencedirect.com/science/article/B6V18-3SNYT5C-1B/2/38110bb682311c8cc6169b64b233dac5`.

K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *PPSN VI*, pages 849–858, 2000.

R. L. Drysdale and F. H. Young. Improved divide/sort/merge sorting networks. *SIAM Journal on Computing*, 4(3):264–270, 1975.

L. Graham and F. Oppacher. Symmetric comparator pairs in the initialization of genetic algorithm populations for sorting networks. *IEEE Congress on Evolutionary Computation, 2006 (CEC 2006)*, pages 2845–2850, 2006. doi: 10.1109/CEC.2006.1688666.

M. W. Green. Some improvements in non-adaptive sorting algorithms. In *Proceedings of the Sixth Annual Princeton Conference on Information Sciences and Systems*, pages 387–391, 1972.

C. A. Gunter, T.-H. Ngair, and D. Subramanian. Sets as anti-chains. In *ASIAN '96: Proceedings of the Second Asian Computing Science Conference on Concurrency and Parallelism, Programming, Networking, and Security*, pages 116–128, London, UK, 1996. Springer-Verlag. ISBN 3-540-62031-1.

A. Hiasat and O. Hasan. Bit-serial architecture for rank order and stack filters. *Integration, the VLSI Journal*, 36(1-2):3 – 12, 2003. ISSN 0167-9260. doi: DOI:10.1016/S0167-9260(03)00017-8. URL `http://www.sciencedirect.com/science/article/B6V1M-48NKGY5-1/2/53c0e6c9dfb4ef44292e616c4eab3356`.

W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In J. D. Farmer, C. Langton, S. Rasmussen, and C. Taylor, editors, *Artificial Life II*. Addison-Wesley, Reading, MA, 1991.

H. Juillé. Evolution of non-deterministic incremental algorithms as a new approach for search in state spaces. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 351–358, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

R. Kannan and S. Ray. Sorting networks with applications to hierarchical optical interconnects. In *2001 International Conference on Parallel Processing Workshops*, pages 327–332. IEEE Computer Society, 2001. ISBN 0-7695-1260-7. doi: http://doi.ieeecomputersociety.org/10.1109/ICPPW.2001.951969.

P. Kipfer and R. Westermann. Improved GPU sorting. In M. Pharr, editor, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, chapter 46. Addison-Wesley, 2005.

P. Kipfer, M. Segal, and R. Westermann. Uberflow: A gpu-based particle engine. In *HWWS '04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 115–122, New York, NY, USA, 2004. ACM. ISBN 3-905673-15-0. doi: http://doi.acm.org.ezproxy.lib.utexas.edu/10.1145/1058129.1058146.

D. E. Knuth. *Art of Computer Programming: Sorting and Searching*, volume 3, chapter 5, pages 219–229. Addison-Wesley Professional, 2 edition, April 1998.

J. Korenek and L. Sekanina. Intrinsic evolution of sorting networks: A novel complete hardware implementation for FPGAs. In *Evolvable Systems: From Biology to Hardware*, pages 46–55. Springer, 2005. URL `http://dx.doi.org.ezproxy.lib.utexas.edu/10.1007/11549703_5`.

A. D. Korshunov. Monotone boolean functions. *Russian Mathematical Surveys*, 58(5):929, 2003. URL `http://stacks.iop.org/0036-0279/58/i=5/a=R02`.

J. R. Koza, J. R. Koza, I. Forest H. Bennett, I. Forest H. Bennett, J. L. Hutchings, S. L. Bade, M. A. Keane, and D. Andre. Evolving computer programs using rapidly reconfigurable field-programmable gate arrays and genetic programming. In *FPGA '98: Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, pages 209–219, New York, NY, USA, 1998. ACM. doi: 10.1145/275107.275141.

J. R. Koza, D. Andre, F. H. Bennett, and M. A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*, chapter 21, pages 335–348. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. ISBN 1558605436.

T. Leighton and C. G. Plaxton. A (fairly) simple circuit that (usually) sorts. In *SFCS '90: Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 264–274 vol.1, Washington, DC, USA, 1990. IEEE Computer Society. ISBN 0-8186-2082-X. doi: http://dx.doi.org.ezproxy.lib.utexas.edu/10.1109/FSCS.1990.89545.

D. P. Mehta and S. Sahni. *Handbook of Data Structures and Applications*, chapter 3, pages 3.4–3.7. CRC Press, 2005. ISBN 9781584884354.

H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

D. G. O'Connor and R. J. Nelson. Sorting system with n-line sorting switch. United States Patent number 3,029,413, April 1962. Filed Feb 21, 1957. Issued Apr 10, 1962.

S.-W. Seo, T. yun Feng, and Y. Kim. A simulation scheme in rearrangeable networks. In *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, pages 177 – 180 vol. 1, Aug 1993. doi: 10.1109/MWSCAS.1993.343100.

I. Shmulevich, T. M. Sellke, M. Gabbouj, and E. J. Coyle. Stack filters and free distributive lattices. In *Proceedings of the 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pages 927–930. IEEE Computer Society, 1995.

D. C. Van Voorhis. A generalization of the divide-sort-merge strategy for sorting networks. Technical Report 16, Digital Systems Laboratory, Stanford University, Stanford, California, August 1971.

D. C. Van Voorhis. An economical construction for sorting networks. In *Proceedings of AFIPS National Computer Conference*, pages 921–927, New York, NY, USA, 1974. ACM. doi: http://doi.acm.org/10.1145/1500175.1500347. URL `http://doi.acm.org/10.1145/1500175.1500347`.

Y.-M. Yeh and T.-y. Feng. On a class of rearrangeable networks. *IEEE Transactions on Computers*, 41(11):1361–1379, 1992. ISSN 0018-9340. doi: http://dx.doi.org.ezproxy.lib.utexas.edu/10.1109/12.177307.

Journal of Machine Learning Research 14 (2013) 333-350 Submitted 11/11; Revised 6/12, 11/12; Published 2/13

A Framework for Evaluating Approximation Methods for Gaussian Process Regression

Krzysztof Chalupka[*] KJCHALUP@CALTECH.EDU
Computation and Neural Systems
California Institute of Technology
1200 E. California Boulevard
Pasadena, CA 91125, USA

Christopher K. I. Williams C.K.I.WILLIAMS@ED.AC.UK
Iain Murray I.MURRAY@ED.AC.UK
School of Informatics
University of Edinburgh
10 Crichton St
Edinburgh EH8 9AB, UK

Editor: Neil Lawrence

Abstract

Gaussian process (GP) predictors are an important component of many Bayesian approaches to machine learning. However, even a straightforward implementation of Gaussian process regression (GPR) requires $O(n^2)$ space and $O(n^3)$ time for a data set of n examples. Several approximation methods have been proposed, but there is a lack of understanding of the relative merits of the different approximations, and in what situations they are most useful. We recommend assessing the quality of the predictions obtained as a function of the compute time taken, and comparing to standard baselines (e.g., Subset of Data and FITC). We empirically investigate four different approximation algorithms on four different prediction problems, and make our code available to encourage future comparisons.

Keywords: Gaussian process regression, subset of data, FITC, local GP

1. Introduction

Gaussian process (GP) predictors are widely used in non-parametric Bayesian approaches to supervised learning problems (Rasmussen and Williams, 2006). They can also be used as components for other tasks including unsupervised learning (Lawrence, 2004), and dependent processes for a variety of applications (e.g., Sudderth and Jordan 2009; Adams et al. 2010). The basic model on which these are based is Gaussian process regression (GPR), for which a standard implementation requires $O(n^2)$ space and $O(n^3)$ time for a data set of n examples (e.g., Rasmussen and Williams, 2006, Chapter 2). Several approximation methods have now been proposed, as detailed below. Typically the approximation methods are compared to the basic GPR algorithm. However, as there are now a range of different approximations, the user is faced with the problem of understanding their relative merits, and in what situations they are most useful.

∗. This research was carried out when KC was a student at the University of Edinburgh.

Most approximation algorithms have a tunable complexity parameter, which we denote as m. Our key recommendation is to study the quality of the predictions obtained as a function of the *compute time* taken as m is varied, as times can be compared across different methods. New approximation methods should be compared against current baselines like Subset of Data and FITC (described in Sections 2.1–2.2). The time decomposes into that needed for training the predictor (including setting hyperparameters), and test time; the user needs to understand which will dominate in their application. We illustrate this process by studying four different approximation algorithms on four different prediction problems. We have published our code in order to encourage comparisons of other methods against these baselines.

The structure of the paper is as follows: In Section 2 we outline the complexity of the full GP algorithm and various approximations, and give some specific details needed to apply them in practice. Section 3 outlines issues that should be considered when selecting or developing a GP approximation algorithm. Section 4 describes the experimental setup for comparisons, and the results of these experiments. We conclude with future directions and a discussion.

2. Approximation Algorithms for Gaussian Process Regression (GPR)

A regression task has a training set $\mathcal{D}=\{\mathbf{x}_i, y_i\}_{i=1}^n$ with D-dimensional inputs $\mathbf{x}_i$ and scalar outputs y_i. Assuming that the outputs are noisy observations of a latent function f at values $f_i = f(\mathbf{x}_i)$, the goal is to compute a predictive distribution over the latent function value f_* at a test location $\mathbf{x}_*$.

Assuming a Gaussian process prior over functions f with zero mean, and covariance or kernel function $k(\cdot,\cdot)$, and Gaussian observations, $y_i = f_i + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0,\sigma^2)$, gives Gaussian predictions $p(f_* \,|\, \mathbf{x}_*, \mathcal{D}) = \mathcal{N}(\overline{f}_*, \mathbb{V}[f_*])$, with predictive mean and variance (see, e.g., Rasmussen and Williams, 2006, Section 2.2):

$$\overline{f}_* = \mathbf{k}^\top(\mathbf{x}_*)(K+\sigma^2 I)^{-1}\mathbf{y} \stackrel{def}{=} \mathbf{k}^\top(\mathbf{x}_*)\boldsymbol{\alpha}, \tag{1}$$

$$\mathbb{V}[f_*] = k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}^\top(\mathbf{x}_*)(K+\sigma^2 I)^{-1}\mathbf{k}(\mathbf{x}_*), \tag{2}$$

where K is the $n \times n$ matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}(\mathbf{x}_*)$ is the $n \times 1$ column vector with the ith entry being $k(\mathbf{x}_*, \mathbf{x}_i)$, $\mathbf{y}$ is the column vector of the n target values, and $\boldsymbol{\alpha} = (K+\sigma^2 I)^{-1}\mathbf{y}$.

The log marginal likelihood of the GPR model is also available in closed form:

$$L = \log p(\mathbf{y}|X) = -\tfrac{1}{2}\,\mathbf{y}^\top(K+\sigma_n^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K+\sigma^2 I| - \tfrac{n}{2}\log 2\pi. \tag{3}$$

Typically L is viewed as a function of a set of parameters θ that specify the kernel. Below we assume that θ is set by numerically maximizing L with a routine like conjugate gradients. Computation of L and the gradient $\nabla_\theta L$ can be carried out in $O(n^3)$. Optimizing L is a maximum-likelihood type II or ML-II procedure for θ; alternatively one might sample over $p(\theta|\mathcal{D})$ using, for example, MCMC. Equations 1–3 form the basis of GPR prediction.

We identify three computational phases in carrying out GPR:

hyperparameter learning: The hyperparameters are learned, by for example maximizing the log marginal likelihood. This is often the most computationally expensive phase.

training: Given the hyperparameters, all computations that do not involve test inputs are performed, such as computing $\boldsymbol{\alpha}$ above, and/or computing the Cholesky decomposition of $K+\sigma_n^2 I$. This phase was called "precomputation" by Quiñonero-Candela et al. (2007, Section 9.6).

Method	Storage	Training	Mean	Variance
Full	$O(n^2)$	$O(n^3)$	$O(n)$	$O(n^2)$
SoD	$O(m^2)$	$O(m^3)$	$O(m)$	$O(m^2)$
FITC	$O(mn)$	$O(m^2n)$	$O(m)$	$O(m^2)$
Local	$O(mn)$	$O(m^2n)$	$O(m)$	$O(m^2)$

Table 1: A comparison of the space and time complexity of the Full, SoD, FITC and Local methods, ignoring the time taken to select the m subset/inducing points/clusters from the n datapoints. Training: the time required for preliminary computations before the test point $\mathbf{x}_*$ is known, for each hyperparameter setting considered. Mean (resp. variance): the time needed to compute the predictive mean (variance) at test point $\mathbf{x}_*$.

testing: Only the computations involving the test inputs are carried out, those which could not have been done previously. This phase may be significant if there is a very large test set, or if deploying a trained model on a machine with limited resources.

Table 1 lists the computational complexity of training and testing full GPR as a function of n. Evaluating the marginal likelihood L and its gradient takes more operations than 'training' (i.e., computing the parts of (1) and (2) that do not depend on $\mathbf{x}_*$), but has the same scaling with n. Hyperparameter learning involves evaluating L for all values of the hyperparameters θ that are searched over, and so is more expensive than training for fixed hyperparameters.

These complexities can be reduced in special cases, for example, for stationary covariance functions and grid designs, as may be found, for example, in geoscience problems. In this case the eigenvectors of K are the Fourier basis, and matrix inversions etc can be computed analytically. See, for example, Wikle et al. (2001), Paciorek (2007) and Fritz et al. (2009) for more details.

Common methods for approximate GPR include Subset of Data (SoD), where data points are simply thrown away; inducing point methods (Quiñonero-Candela and Rasmussen, 2005), where K is approximated by a low-rank plus diagonal form; Local methods where nearby data is used to make predictions in a given region of space; and fast matrix-vector multiplication (MVM) methods, which can be used with iterative methods to speed up the solution of linear systems. We discuss these in turn, so as to give coverage to the wide variety methods that have been proposed. We use the Fully Independent Training Conditional (FITC) method as it is recommended over other inducing point methods by Quiñonero-Candela et al. (2007), and the Improved Fast Gauss Transform (IFGT) of Yang et al. (2005) as a representative of fast MVM methods.

2.1 Subset of Data

The simplest way of dealing with large amounts of data is simply to ignore some or most of it. The 'Subset of Data (SoD) approximation' simply applies the full GP prediction method to a subset of size $m < n$. Therefore the computational complexities of SoD result from replacing n with m in the expressions for the full method (Table 1). Despite the 'obvious' nature of SoD, most papers on approximate GP methods only compare to a GP applied to the full data set of size n.

To complete the description of the SoD method we must also specify how the subset is selected. We consider two of the possible alternatives: 1) Selecting m points randomly costs $O(m)$ if we need

not look at the other points. 2) We select m cluster centres from a Farthest Point Clustering (FPC, Gonzales 1985) of the data set; using the algorithm proposed by Gonzales this has computational complexity of $O(mn)$. In theory, FPC can be sped up to $O(n\log m)$ using suitable data structures (Feder and Greene, 1988), although in practice the original algorithm can be faster for machine learning problems of moderate dimensionality. FPC has a random aspect as the first point can be chosen randomly. Our SoD implementation is based on `gp.m` in the MATLAB `gpml` toolbox: `http://www.gaussianprocess.org/gpml/code/matlab/doc/`.

Rather than selecting the subset randomly, it is also possible to make a more informed choice. For example Lawrence et al. (2003) came up with a fast selection scheme (the "informative vector machine") that takes only $O(m^2n)$. Keerthi and Chu (2006) also proposed a matching pursuit approach which has similar asymptotic complexity, although the associated constant is larger.

2.2 Inducing Point Methods: FITC

A number of GP approximation algorithms use alternative kernel matrices based on *inducing points*, $\mathbf{u}$, in the D-dimensional input space (Quiñonero-Candela and Rasmussen, 2005). Here we restrict the m inducing points to be a subset of the training inputs. The Subset of Regressors (SoR) kernel function is given by $k_{SoR}(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{k}(\mathbf{x}_i,\mathbf{u})K_{\mathbf{uu}}^{-1}\mathbf{k}(\mathbf{u},\mathbf{x}_j)$, and the Fully Independent Training Conditional (FITC) method uses

$$k_{FITC}(\mathbf{x}_i,\mathbf{x}_j) = k_{SoR}(\mathbf{x}_i,\mathbf{x}_j) + \delta_{ij}[k(\mathbf{x}_i,\mathbf{x}_j) - k_{SoR}(\mathbf{x}_i,\mathbf{x}_j)].$$

FITC approximates the matrix K as a rank-m plus diagonal matrix. An attractive property of FITC, not shared by all approximations, is that it corresponds to exact inference for a GP with the given k_{FITC} kernel (Quiñonero-Candela et al., 2007). Other inducing point approximations (e.g., SoR, deterministic training conditionals) have similar complexity but Quiñonero-Candela et al. (2007) recommend FITC over them. Since then there have been further developments (Titsias, 2009; Lázaro-Gredilla et al., 2010), which would also be interesting to compare.

To make predictions with FITC, and to evaluate its marginal likelihood, simply substitute k_{FITC} for the original kernel in Equations 1–3. This substitution gives a mean predictor of the form $\overline{f}_* = \sum_{i=1}^m \beta_i k(\mathbf{x}_*,\mathbf{x}_i)$, where $i = 1,\ldots,m$ indexes the selected subset of training points, and the βs are obtained by solving a linear system. Snelson (2007, pp 60-62) showed that in the limit of zero noise FITC reduces to SoD.

We again choose a set of inducing points of size m from the training inputs either randomly or using FPC, and use the FITC implementation from the `gpml` toolbox.

It is possible to "mix and match" the SoD and FITC methods, adapting the hyperparameters to optimize the SoD approximation to the marginal likelihood, then using the FITC algorithm to make predictions using the same data subset and the SoD-trained hyperparameters. We refer to this procedure as the Hybrid method.[1] We expect that saving time on the hyperparameter learning phase, $O(m^3)$ instead of $O(m^2n)$, will come at the cost of reducing the predictive performance of FITC for a given m.

2.3 Local GPR

The basic idea here is of divide-and-conquer, although without any guarantees of correctness. We divide the n training points into $k = \lceil \frac{n}{m} \rceil$ clusters each of size m, and run GPR in each cluster,

1. We thank one of the anonymous reviewers for suggesting this method.

ignoring the training data outside of the given cluster. At test time we assign a test input $\mathbf{x}_*$ to the closest cluster. This method has been discussed by Snelson and Ghahramani (2007). The hard cluster boundaries can lead to ugly discontinuities in the predictions, which are unacceptable if a smooth surface is required, for example in some physical simulations.

One important issue is how the clustering is done. We found that FPC tended to produce clusters of very unequal size, which limited the speedups obtained by Local GPR. Thus we devised a method we call Recursive Projection Clustering (RPC), which works as follows. We start off with all the data in one cluster C. Choose two data points at random from C, draw a line through these points and calculate the orthogonal projection of all points from C onto the line. Split C into two equal-sized subsets C_L and C_R depending on whether points are to the left or right of the median. Now repeat recursively in each cluster until the cluster size is no larger than m. In our implementation we make use of MATLAB's `sort` function to find the median value, taking time $O(n \log n)$ for n datapoints, although it is possible to reduce median finding to $O(n)$ (Blum et al., 1973). Thus overall the complexity of RPC is $O(ns \log n)$, where $s = \lceil \log_2(n/m) \rceil$. A test point $\mathbf{x}_*$ is assigned to the appropriate cluster by descending the tree of splits constructed by RPC.

Another issue concerns hyperparameter learning. L is approximated by the sum of terms like Equation 3 over all clusters. Hyperparameters can either be tied across all clusters ("joint" training), or unique to each cluster ("separate" training). Joint training is likely to be useful for small m. We implemented Local GPR using the `gpml` toolbox with small modifications to sum gradients for joint training.

2.4 Iterative Methods and IFGT Matrix-Vector Multiplies

The Conjugate Gradients (CG) method (e.g., Golub and Van Loan 1996) can be used at training time to solve the linear system $(K + \sigma^2 I)\boldsymbol{\alpha} = \mathbf{y}$. Indeed, all GPR computations can be based on iterative methods (Gibbs, 1997). CG and several other iterative methods (e.g., Li et al. 2007; Liberty et al. 2007) for solving linear systems require the ability to multiply a matrix of kernel values with an arbitrary vector.

Standard dense matrix-vector multiplication (MVM) costs $O(n^2)$. It has been argued (e.g., Gibbs 1997; Li et al. 2007) that iterative methods alone provide a cost saving if terminated after $k \ll n$ matrix-vector multiplies. Papers often do not state how CG was terminated (e.g., Shen et al., 2006; Freitas et al., 2006), although some are explicit about using a small fixed number of iterations based on preliminary runs (e.g., Gray, 2004). Ad-hoc termination rules, or those using the 'relative residual' (Golub and Van Loan, 1996) (see Section 4.1) do not necessarily give the best trade-off between time and test-error. In Section 4.1 we examine the progression of test error throughout training, to see what error/time trade-offs might be achieved by different termination rules.

Iterative methods are not used routinely for dense linear system solving, they are usually only recommended when the cost of MVMs is reduced by exploiting sparsity or other matrix structure. Whether iterative methods can provide a speedup for GPR or not, fast MVM methods will certainly be required to scale to huge data sets. Firstly, while other methods can be made linear in the size of the data set size ($O(m^2 n)$, see Table 1), a standard MVM costs $O(n^2)$. Most importantly, explicitly constructing the K matrix uses $O(n^2)$ memory, which sets a hard ceiling on data set size. Storing the kernel elements on disk, or reproducing the kernel computations on the fly, is prohibitively expensive. Fast MVM methods potentially reduce the storage required, as well as the computation time of the standard dense implementation.

We have previously demonstrated some negative results concerning speeding up MVMs (Murray, 2009): 1) if the kernel matrix were approximately sparse (i.e., many entries near zero) it would be possible to speed up MVMs using sparse matrix techniques, but in the hyperparameter regimes identified in practice this does not usually occur; 2) the piecewise constant approximations used by simple kd-tree approximations to GPR (Shen et al., 2006; Gray, 2004; Freitas et al., 2006) cannot safely provide meaningful speedups.

The Improved Fast Gauss Transform (IFGT) is a MVM method that can be applied when using a squared-exponential kernel. The IFGT is based on a truncated multivariate Taylor series around a number of cluster centres. It has been applied to kernel machines in a number of publications, for example, Yang et al. (2005); Morariu et al. (2009). Our experiments use the IFGT implementation from the Figtree C++ package with MATLAB wrappers available from `http://www.umiacs.umd.edu/~morariu/figtree/`. This software provides automatic choices for a number of parameters within IFGT. The time complexity of IFGT depends on a number of factors as described in Morariu et al. (2009), and we focus below on empirical results.

There are open problems with making iterative methods and fast MVMs for GPR work routinely. Firstly, unlike standard dense linear algebra routines, the number of operations depends on the hyperparameter settings. Sometimes the programs can take a very long time, or even crash due to numerical problems. Methods to diagnose and handle these situations automatically are required. Secondly, iterative methods for GPR are usually only applied to mean prediction, Equation 1; finding variances $\mathbb{V}[f_*]$ would require solving a new linear system for each $\mathbf{k}(\mathbf{x}_*)$. In principle, an iterative method could approximately factorize $(K+\sigma^2 I)$ for variance prediction. To our knowledge, no one has demonstrated the use of such a method for GPR with good scaling in practice.

2.5 Comparing the Approximation Methods

Above we have reviewed the SoD, FITC, Hybrid, Local and Iterative MVM methods for speeding up GP regression for large n. The space and time complexities for the SoD, FITC, and Local methods are given in Table 1; as explained above there are open problems with making iterative methods and fast MVMs work routinely for GPR, see also Sections 4.1 and 4.2.

Comparing FITC to SoD, we note that the mean predictor contains the same basis functions as the SoD predictor, but that the coefficients are (in general) different as FITC has "absorbed" the effect of the remaining $n-m$ datapoints. Hence for fixed m we might expect FITC to obtain better results. Comparing Local to SoD, we might expect that using training points lying nearer to the test point would help, so that for fixed m Local would beat SoD. However, both FITC and Local have $O(m^2n)$ training times (although the associated constants may differ), compared to $O(m^3)$ for SoD. So if equal training time was allowed, a larger m could be afforded for SoD than the others. This is the key to the comparisons in Section 4.3 below. The Hybrid method has the same hyperparameter learning time as SoD by definition, but the training phase will take longer than SoD with the same m, because of the need for a final $O(m^2n)$ phase of FITC training, as compared to the $O(m^3)$ for SoD. However, as per the argument above, we would expect the FITC predictions to be superior to the SoD ones, even if the hyperparameters have not been optimized explicitly for FITC prediction; this is explored experimentally in Section 4.3.

At test time Table 1 shows that the SoD, FITC, Hybrid and Local approximations are $O(m)$ for mean prediction, and $O(m^2)$ for predictive variances. This means that the method which has obtained the best "m-size" predictor will win on test-time performance.

3. A Basis for Comparing Approximations

For fixed hyperparameters, comparing an approximate method to the full GPR is relatively straightforward: we can evaluate the predictive error made by the approximate method, and compare that against the "gold standard" of full GPR. The 'best' method could be the approximation with best predictions for a given computational cost, or alternatively the smallest computational cost for a given predictive performance. However, there are still some options, for example, different performance criteria to choose from (mean squared error, mean predictive log likelihood). Also there are different possible relevant computational costs (hyperparameter learning, training, testing) and definitions of cost itself (CPU time, 'flops' or other operation counts). It should also be borne in mind that any error measure compresses the predictive mean and variance functions into a single number; for low-dimensional problems visualizing these functions can illustrate the differences between approximations (e.g., Quiñonero-Candela et al., 2007, Figure 9.4).

It is rare that the appropriate hyperparameters are known for a given problem, unless it is a synthetic problem drawn from a GP. For real-world data we are faced with two alternatives: (i) compare approximate methods using the same set of hyperparameters as obtained by full GPR, or (ii) allow the approximate methods freedom to determine their own hyperparameters, for example, by using approximate marginal likelihoods consistent with the approximations. Below we follow the second approach as it is more realistic, although it does complicate comparisons by changing both the approximation method and the hyperparameters.

In terms of computational cost we use the CPU time in seconds, based on MATLAB implementations of the algorithms (except for the IFGT where the Figtree C++ code is used with MATLAB wrappers). The core GPR calculations are well suited to efficient implementation in MATLAB. Our SoD, FITC, Hybrid and Local GP implementations are all derived from the standard `gpml` toolbox of Rasmussen and Nickisch.

Before making empirical comparisons on particular data sets, we identify aspects of regression problems, models and approximations that affect the appropriateness of using a particular method:

The nature of the underlying problem: We usually standardize the inputs to have zero mean and unit variance on each dimension. Then clearly we would expect to require more datapoints to pin down accurately a higher frequency (more "wiggly") function than a lower frequency one.

For multivariate input spaces there will also be issues of dimensionality, either wrt the intrinsic dimensionality of $\mathbf{x}$ (for example if the data lies on a manifold of lower dimensionality) or the apparent dimensionality. Note that if there are irrelevant inputs these can potentially be detected by a kernel equipped with "Automatic Relevance Determination" (ARD) (Neal, 1996; Rasmussen and Williams, 2006, p. 106).

Another factor is the noise level on the data. An eigenanalysis of the problem (see, e.g., Rasmussen and Williams 2006, Section 2.6) shows that it is more difficult to discover low-amplitude components in the underlying function if there is high noise. It is relatively easy to get an upper bound on the noise level by computing the variance of the y's around a given $\mathbf{x}$ location (or an average of such calculations), particularly if the lengthscale of variation of function is much larger than inter-datapoint distances (i.e., high sampling density); this provides a useful sanity check on the noise level returned during hyperparameter optimization.

The choice of kernel function: Selecting an appropriate family of kernel functions is an important part of modelling a particular problem. For example, poor results can be obtained when using an isotropic kernel on a problem where there are irrelevant input dimensions, while an ARD param-

eterization would be a better choice. Some approximation methods (e.g., the IFGT) have only been derived for particular kernel functions. For simplicity of comparison we consider only the SE-ARD kernel (Rasmussen and Williams, 2006, p. 106), as that is the kernel most widely used in practice.

The practical usability of a method: Finally, some more mundane issues contribute significantly to the usability of a method, such as: (a) Is the method numerically robust? If there are problems it should be clear how to diagnose and deal with them. (b) Is it clear how to set tweak parameters, for example, termination criteria? Difficulties with these issues do not just make it difficult to make fair comparisons, but reflect real difficulties with using the methods. (c) Does the method work efficiently for a wide range of hyperparameter settings? If not, hyperparameter searching must be performed much more carefully and one has to ask if the method will work well on good hyperparameter settings.

4. Experiments

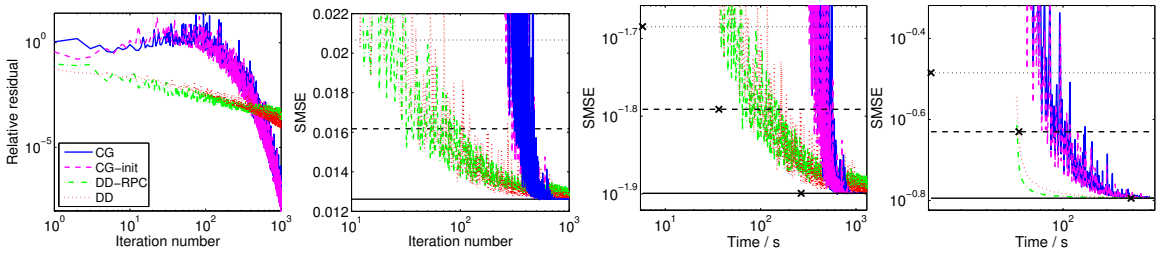
Data sets: We use four data sets for comparison. The first two are synthetic data sets, SYNTH2 and SYNTH8, with $D = 2$ and $D = 8$ input dimensions. The inputs were drawn from a $N(0, I)$ Gaussian, and the function was drawn from a GP with zero mean and isotropic SE kernel with unit lengthscale. There are 30,543 training points and 30,544 test points in each data set.[2] The noise variance is 10^{-6} for SYNTH2, and 10^{-3} for SYNTH8. The CHEM data set is derived from physical simulations relating to electron energies in molecules (Malshe et al., 2007).[3] The input dimensionality is 15, and the data is split into 31,535 training cases and 31,536 test cases. Additional results on this data set have been reported by Manzhos and Carrington Jr. (2008). The SARCOS data set concerns the inverse kinematics of a robot arm, and is used, for example, in Rasmussen and Williams (2006, Section 2.5). It has 21 input dimensions, 44,484 training cases and 4,449 test cases (the split used by Rasmussen and Williams 2006). The SARCOS data set is already publicly available from `http://www.gaussianprocess.org`. All four data sets are included in the code and data tarfile associated with this paper.

Error measures: We measured the accuracy of the methods' predictions on the test sets using the Standardized Mean Squared Error (SMSE), and Mean Standardized Log Loss (MSLL), as defined in (Rasmussen and Williams, 2006, Section 2.5). The SMSE is the mean squared error normalized by the MSE of the dumb predictor that always predicts the mean of the training set. The MSLL is obtained by averaging $-\log p(y_*|\mathcal{D}, \mathbf{x}_*)$ over the test set and subtracting the same score for a trivial model which always predicts the mean and variance of the training set. Notice that MSLL involves the predictive variances while SMSE does not.

Each experiment was carried out on a 3.47 GHz core with at least 10 GB available memory, except for Section 4.1 which used 3 GHz cores with 12 GB memory. Approximate log marginal likelihoods were optimized wrt θ using Carl Rasmussen's `minimize.m` routine from the `gpml` toolbox, using a maximum of 100 iterations. The code and data used to run the experiments is available from `http://homepages.inf.ed.ac.uk/ckiw/code/gpr_approx.html` .

In Section 4.1 we provide results investigating the efficacy of iterative methods for GPR. In Section 4.2 we investigate the utility of IFGT to speed up MVMs. Section 4.3 compares the SoD, FITC and Local approximations on the four data sets, and Section 4.4 compares predictions made with the learned hyperparameters and the generative hyperparameters on the synthetic data sets.

2. We thank Carl Rasmussen for providing these data sets.
3. We thank Prof. Lionel Raff of Oklahoma State University and colleagues for permission to distribute this data.



a) SARCOS b) SARCOS c) SARCOS d) SYNTH8

Figure 1: Experiments with 16,384 training points. Legend abbreviations: CG: conjugate gradients; DD: 'domain decomposition' with 16 randomly chosen clusters; CG-init: CG initialized with one iteration of DD (CG's starting point of zero is not responsible for bad early behaviour); DD-RPC: clusters were chosen with recursive projection clustering (Section 2.3). The horizontal lines give test performance for SoD with 4,096, 8,192 and 16,384 training points. Crosses on these lines also show the time taken.

4.1 Results for Iterative Methods

Most attempts to use iterative methods for Gaussian processes have used conjugate gradient (CG) methods (Gibbs, 1997; Gray, 2004; Shen et al., 2006; Freitas et al., 2006). However, Li et al. (2007) introduced a method, which they called Domain Decomposition (DD), that over 50 iterations appeared to converge faster than CG. We have compared CG and DD for training a GP mean predictor based on 16,384 points from the SARCOS data, with the same fixed hyperparameters used by Rasmussen and Williams (2006).

Figure 1a) plots the 'relative residual', $\|(K+\sigma^2 I)\boldsymbol{\alpha}_t - \mathbf{y}\|/\|\mathbf{y}\|$, the convergence diagnostic used by Li et al. (2007, Figure 2), against iteration number for both their method and CG, where $\boldsymbol{\alpha}_t$ is the approximation to $\boldsymbol{\alpha}$ obtained at iteration t. We reproduce the result that CG gives higher and fluctuating residuals for early iterations. However, by running the simulation for longer, and plotting on a log scale, we see that CG converges, according to this measure, much faster at later iterations. Figure 1a) is not directly useful for choosing between the methods however, because we do not know how many iterations are required for a competitive test-error.

Figure 1b) instead plots test-set SMSE, and adds reference lines for the SMSEs obtained by subsets with 4,096, 8,192 and 16,384 training points. We now see that 50 iterations are insufficient for meaningful convergence on this problem. Figure 1c) plots the SMSE against computer time taken on our machine.[4] SoD performs better than the iterative methods.

These results depend on the data set and hyperparameters. Figure 1d) shows the test-set SMSE progression against time for 16,384 points from SYNTH8 using the true hyperparameters. Here CG takes a similar time to direct Cholesky solving. However, there is now a part of the error-time plot where the DD approach has better SMSEs at smaller times than either CG or SoD.

The timing results are heavily implementation and architecture dependent. For example, the results reported so far were run on a single 3 GHz core. On our machines, the iterative methods scale less well when deployed on multiple CPU cores. Increasing the number of cores to four (using

4. The time per iteration was measured on a separate run that was not slowed down by storing the intermediate results required for these plots.

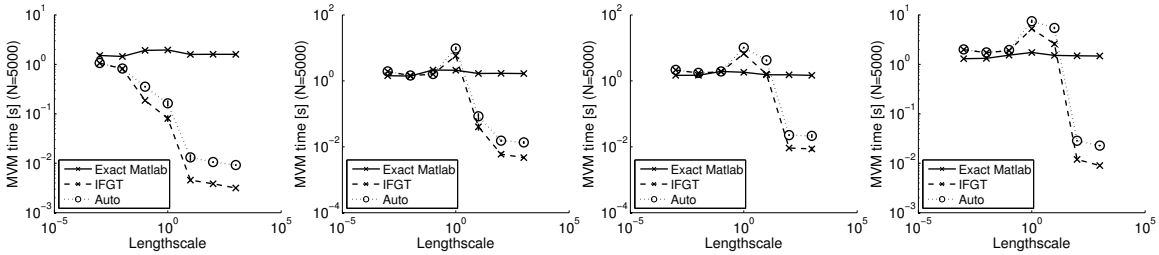


Figure 2: Plot of time vs lengthscale using IFGT for matrix-vector multiplication (MVM) on the four data sets. The Auto method was introduced in Raykar and Duraiswami (2007) as a way to speed up IFGT in some regimes.

MATLAB, which uses Intel's MKL), the time to perform a 16384×16384 Cholesky decomposition decreased by a factor of 3.1, whereas a matrix vector multiply improved by only a factor of 1.7.

4.2 Results for IFGT

We focus here on whether the IFGT provides fast MVMs for the data sets in our comparison. We used the isotropic squared-exponential kernel (which has one lengthscale parameter shared over all dimensions). For each of the four data sets we randomly chose 5000 datapoints to construct a kernel matrix, and a 5000-element random vector (with elements sampled from $U[0,1]$). Figure 2 shows the MVM time as a function of lengthscale. For SYNTH2 and SYNTH8 the known lengthscale is 1. For the two other problems, and indeed many standardized regression problems, lengthscales of ≈ 1 (the width of the input distribution) are also appropriate. Figure 2 shows that useful MVM speedups over a direct implementation are only obtained for SYNTH2. The result on SARCOS is consistent with Raykar and Duraiswami (2007)'s result that IFGT does not accelerate GPR on this data set.

4.3 Comparison of SoD, FITC, Hybrid and Local GPR

All of the experiments below used the squared exponential kernel with ARD parameterization (Rasmussen and Williams, 2006, p. 106). The test times given below include computation of the predictive variances.

SoD was run with m ascending in powers of 2 from $32, 64 \ldots$ up to 4096. FITC was run with m ranging from 8 to 512 in powers of two; this is smaller than for SoD as FITC is much more memory intensive. Local was run with m ranging from 16 to 2048 in powers of two. For all experiments the selection of the subset/inducing points/clusters has a random aspect, and we performed five runs.

In Figure 3 we plot the test set SMSE against hyperparameter training time (left column), and test time (right column) for the four methods on the four data sets. Figure 4 shows similar plots for the test set MSLL. When there are further choices to be made (e.g., subset selection methods, joint/separate estimation of hyperparameters), we generally present the best results obtained by the method; these choices are detailed at the end of this section for each data set individually. Further details including tables of learned hyperparameters are provided by Chalupka (2011), although the experiments were re-run for this paper, so there are some differences between the two.

The empirical times deviate from theory (Table 1) most for the Local method for small m. There is overhead due to the creation of many small matrices in MATLAB, so that (for example) $m = 32$ is always slower (on our four data sets) than $m = 64$ and $m = 128$. This effect has been demonstrated explicitly by Chalupka (2011, Figure 4.1), and accounts for the bending back observed in the plots for Local. (The effect is present with all four data sets, but can be difficult to see in some of the plots.)

Looking at the hyperparameter training plots (left column), it is noticeable that SoD and FITC reduce monotonically with increasing time, and that SoD outperforms FITC on all data sets (i.e., for the same amount of time, the SoD performance is better). On the test time plots (right column) the pattern between SoD and FITC is reversed, with FITC being superior. These results are consistent with theoretical scalings (Table 1): at training time FITC has worse scaling, at test time its scaling is the same,[5] and it turns out that its more sophisticated approximation does give better results.

Comparing Hybrid to SoD for hyperparameter learning, we note a general improvement in performance for very similar time; this is because the additional cost of one FITC training step at the end is small relative to the time taken to optimize the hyperparameters using the SoD approximation of the marginal likelihood. At test time the Hybrid results are inferior to FITC for the same m as expected, but the faster hyperparameter learning time means that larger subset sizes can be used with Hybrid.

For Local, the most noticeable pattern is that the run time does not change monotonically with m. We also note that for small m the other methods can make faster approximations than Local can for any value of m. For Local there is a general trend for larger m to produce better results, although on SARCOS the error actually increases with m, and for SYNTH2 the SMSE error rises for $m = 1024,\ 2048$. However, Local often gives better performance than the other methods in the time regimes where it operates.

We now comment on the specific data sets:

SYNTH2: This function was fairly easy to learn and all methods were able to obtain good performance (with SMSE close to the noise level of 10^{-6}) for sufficiently large m. For SoD and FITC, it turned out that FPC gave significantly better results than random subset selection. FPC distributes the inducing points in a more regular fashion in the space, instead of having multiple close by in regions of high density. For Local, the joint estimation of hyperparameters was found to be significantly better than separate; this result makes sense as the target function is actually drawn from a single GP. For FITC and Hybrid the plots are cut off at $m = 128$ and $m = 256$ respectively, as numerical instabilities in the `gpml` FITC code for larger m values gave larger errors.

SYNTH8: This function was difficult for all methods to learn, notice the slow decrease in error as a function of time. The SMSE obtained is far above the noise level of 10^{-3}. Both SoD and FITC did slightly better when selecting the inducing points randomly. For the Local method, again joint estimation of hyperparameters was found to be superior, as for SYNTH2. For both SYNTH2 and SYNTH8 we note that the lengthscales learned by the FITC approximation did not converge to the true values even for the largest m, while convergence was observed for SoD and Local; full details are available (Chalupka, 2011, Appendix 1).

5. In fact, careful comparison of the test time plots show that FITC takes longer than SoD; this constant-factor performance difference is due to an implementation detail in `gpml`, which represents the FITC and SoD predictors differently, although they could be manipulated into the same form.

CHEM: Both SoD and FITC did slightly better when selecting the inducing points randomly. Local with joint and separate hyperparameter training gave similar results. We report results on the joint method, for consistency with the other data sets.

SARCOS: For SoD and FITC, FPC gave very slightly better results than random. Local with joint hyperparameter training did better than separate training.

4.4 Comparison with Prediction using the Generative Hyperparameters

For the SYNTH2 and SYNTH8 data sets it is possible to compare the results with learned hyperparameters against those obtained with hyperparameters fixed to the true generative values. We refer to these as the learned and fixed hyperparameter settings.

For the SoD and Local methods there is good agreement between the learned and fixed settings, although for SoD the learned setting generally performs worse on both SMSE and MSLL for small m, as would be expected given the small data sizes. The learned and fixed settings are noticeably different for SoD for $m \leq 128$ on SYNTH2, and $m \leq 512$ on SYNTH8.

For FITC there is also good agreement between the learned and fixed settings, although on SYNTH8 we observed that the learned model slightly outperformed the fixed model by around 0.05 nats for MSLL, and by up to 0.05 for SMSE. This may suggest that for FITC the hyperparameters that produce optimal performance may not be the generative ones.

5. Future Directions

We have seen that Local GPR can sometimes make better predictions than the other methods for some ranges of available computer time. However, our implementation suffers from unusual scaling behaviour at small m due to the book-keeping overhead required to keep track of thousands of small matrices. More careful, lower-level programming than our MATLAB code might reduce these problems.

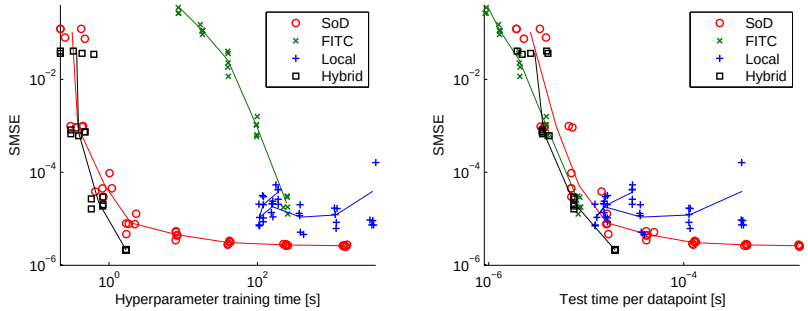
It is possible to combine the SoD with other methods. As a data set's size tends to infinity, SoD (with random selection) will always beat the other approximations that we have considered, as SoD is the only method with no n-dependence (Table 1). Of course the other approximate methods, such as FITC, could also be run on a subset. Investigating how to simultaneously choose the data set size to consider, n, and the control parameter of an approximation, m, has received no attention in the literature to our knowledge.

Some methods will have more choices than a single control parameter m. For example, Snelson and Ghahramani (2006) optimized the locations of the m inducing points, potentially improving test-time performance at the expense of a longer training time. A potential future area of research is working out how to intelligently balance the computer time spent on selecting and moving inducing points, while performing hyperparameter training, and choosing a subset size. Developing methods that work well in a wide variety of contexts without tweaking might be challenging, but success could be measured using the framework of this paper.

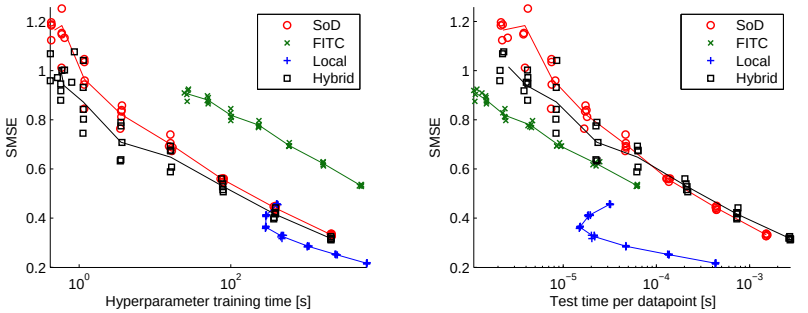
6. Conclusions

We have advocated the comparison of GPR approximation methods on the basis of prediction quality obtained vs compute time. We have explored the times required for the hyperparameter learning, training and testing phases, and also addressed other factors that are relevant for comparing approx-

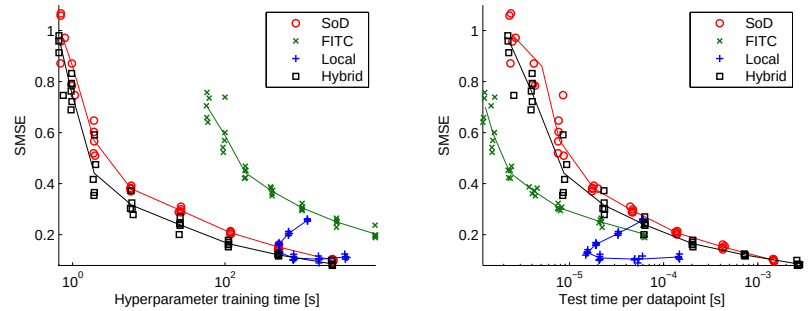
SYNTH2



SYNTH8



CHEM



SARCOS

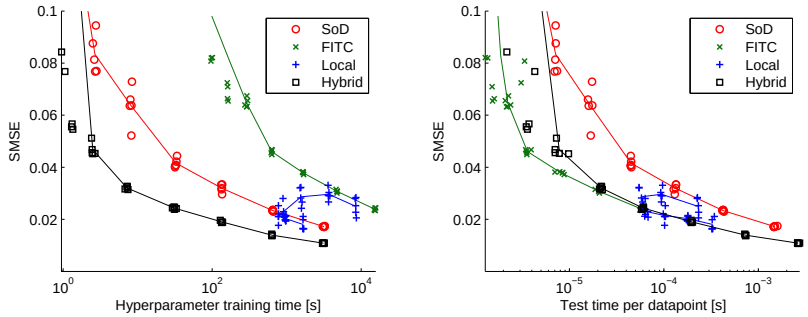
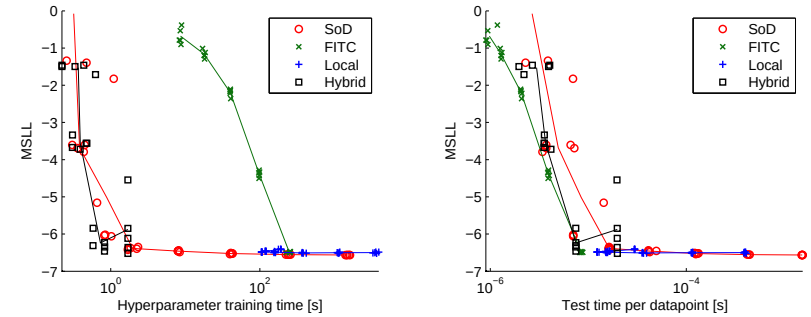
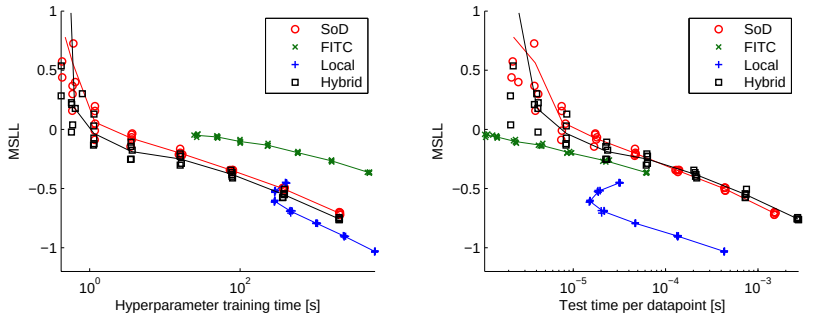


Figure 3: SMSE (log scale) as a function of time (log scale) for the four data sets. Left: hyperparameter training time. Right: test time per test point (including variance computations, despite not being needed to report SMSE). Points give the result for each run; lines connect the means of the 5 runs at each m.

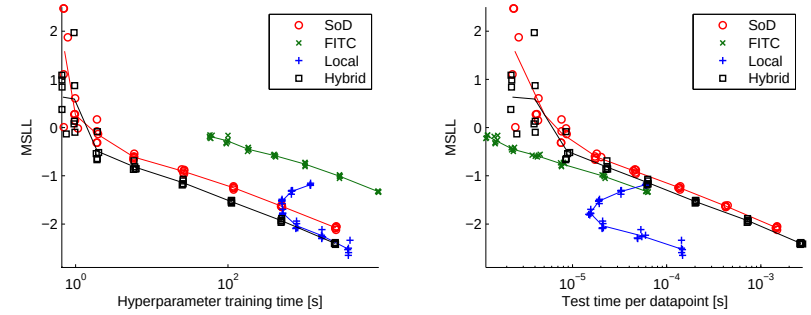
SYNTH2



SYNTH8



CHEM



SARCOS

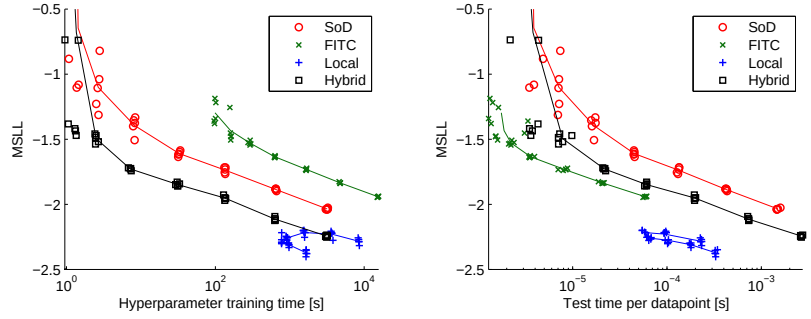


Figure 4: MSLL as a function of time (log scale) for the four data sets. Left: hyperparameter training time. Right: test time per test point. Points give the result for each run; lines connect the means of the 5 runs at each m.

imations. We believe that future evaluations of GP approximations should consider these factors (Section 3), and compare error-time curves with standard approximations such as SoD and FITC. To this end we have made our data and code available to facilitate comparisons. Most papers that have proposed GP approximations have not compared to SoD, and on trying the methods it is often difficult to get appreciably below SoD's error-time curve for the learning phase. Yet these methods are often more difficult to run and more limited in applicability than SoD.

On the data sets we considered, SoD and Hybrid dominate FITC in terms of hyperparameter learning. However, FITC (for as long as we ran it) gave better accuracy for a given test time. SoD, Hybrid and FITC behaved monotonically with subset/inducing-set size m, making m a useful control parameter. The Local method produces more varied results, but can provide a win for some problems and cluster sizes. Comparison of the iterative methods, CG and DD, to SoD revealed that they should not be run for a small fixed number of iterations, and that performance can be comparable with simpler, more stable approaches. Faster MVM methods might make iterative methods more compelling, although the IFGT method only provided a speedup on the SYNTH2 problem out of our data sets. Assuming that hyperparameter learning is the dominant factor in computation time, the results presented above point to the very simple Subset of Data method (or the Hybrid variant) as being the leading contender. We hope this will act as a rallying cry to those working on GP approximations to beat this "dumb" method. This can be addressed both by empirical evaluations (as presented here), and by theoretical work.

Many approximate methods require choosing subsets of partitions of the data. Although farthest point clustering (FPC) improved SoD and FITC on the low-dimensional (easiest) problem, simple random subset selection worked similarly or better on all other data sets. Random selection also has better scaling (no n-dependence) for the largest-scale problems. The choice of partitioning scheme was important for Local regression: Our preliminary experiments showed that performance was severely hampered by many small clusters produced by FPC; we recommend our recursive partitioning scheme (RPC).

Acknowledgments

We thank the anonymous referees whose comments helped improve the paper. We also thank Carl Rasmussen, Ed Snelson and Joaquin Quiñinero-Candela for many discussions on the comparison of GP approximation methods.

This work is supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

References

R. P. Adams, G. E. Dahl, and I. Murray. Incorporating side information into probabilistic matrix factorization using Gaussian processes. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 1–9. AUAI Press, 2010.

M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7:448–461, 1973.

K. Chalupka. Empirical evaluation of Gaussian process approximation algorithms. Master's thesis, School of Informatics, University of Edinburgh, 2011. http://homepages.inf.ed.ac.uk/ckiw/postscript/Chalupka2011diss.pdf.

T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 434–444. ACM Press, New York, USA, 1988. ISBN 0-89791-264-0. doi: http://doi.acm.org/10.1145/62212.62255.

N. De Freitas, Y. Wang, M. Mahdaviani, and D. Lang. Fast Krylov methods for N-body learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 251–258. MIT Press, 2006.

J. Fritz, I. Neuweiler, and W. Nowak. Application of FFT-based algorithms for large-scale universal Kriging problems. *Mathematical Geosciences*, 41:509–533, 2009.

M. Gibbs. *Bayesian Gaussian processes for Classification and Regression*. PhD thesis, University of Cambridge, 1997.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

T. F. Gonzales. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, 1985.

A. Gray. Fast kernel matrix-vector multiplication with application to Gaussian process learning. Technical Report CMU-CS-04-110, School of Computer Science, Carnegie Mellon University, 2004.

S. Keerthi and W. Chu. A matching pursuit approach to sparse Gaussian process regression. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 643–650. MIT Press, Cambridge, MA, 2006.

N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 625–632. MIT Press, 2003.

N. D. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 329–336. MIT Press, 2004.

M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

W. Li, K-H. Lee, and K-S. Leung. Large-scale RLSC learning without agony. In *Proceedings of the 24th International Conference on Machine learning*, pages 529–536. ACM Press New York, NY, USA, 2007.

E. Liberty, F. Woolfe, P-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51): 20167–72, 2007.

M. Malshe, L. M. Raff, M. G. Rockey, M. Hagan, P. M. Agrawal, and R. Komanduri. Theoretical investigation of the dissociation dynamics of vibrationally excited vinyl bromide on an ab initio potential-energy surface obtained using modified novelty sampling and feedforward neural networks. II. Numerical application of the method. *The Journal of Chemical Physics*, 127(13): 134105, 2007.

S. Manzhos and T. Carrington Jr. Using neural networks, optimized coordinates, and high-dimensional model representations to obtain a vinyl bromide potential surface. *The Journal of Chemical Physics*, 129:224104–1–224104–8, 2008.

V. I. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. S. Davis. Automatic online tuning for fast Gaussian summation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1113–1120, 2009.

I. Murray. Gaussian processes and fast matrix-vector multiplies, 2009. Presented at the Numerical Mathematics in Machine Learning workshop at the 26th International Conference on Machine Learning (ICML 2009), Montreal, Canada. URL `http://www.cs.toronto.edu/~murray/pub/09gp_eval/` (as of March 2011).

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, New York, 1996. Lecture Notes in Statistics 118.

C. J. Paciorek. Bayesian smoothing with Gaussian processes using Fourier basis functions in the spectralGP package. *Journal of Statistical Software*, 19(2):1–38, 2007. URL `http://www.jstatsoft.org/v19/i02`.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

J. Quiñonero-Candela, C. E. Rasmussen, and C. K. I. Williams. Approximation methods for Gaussian process regression. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Learning Machines*, pages 203–223. MIT Press, 2007.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.

V. C. Raykar and R. Duraiswami. Fast large scale Gaussian process regression using approximate matrix-vector products. In *Learning Workshop 2007*, 2007. Available from: `http://www.umiacs.umd.edu/~vikas/publications/raykar_learning_workshop_2007_full_paper.pdf`.

Y. Shen, A. Ng, and M. Seeger. Fast Gaussian process regression using KD-trees. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1225–1232. MIT Press, 2006.

E. Snelson. *Flexible and Efficient Gaussian Process Models for Machine Learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2007.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264, 2006.

E. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximations. In M. Meila and X. Shen, editors, *Artificial Intelligence and Statistics 11*. Omnipress, 2007.

E. Sudderth and M. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1585–1592, 2009.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics 12*, volume 5, pages 567–574. JMLR: W&CP, 2009.

C. K. Wikle, R. F. Milliff, D. Nychka, and L. M. Berliner. Spatiotemporal hierarchical Bayesian modeling: tropical ocean surface winds. *Journal of the American Statistical Association*, 96 (454):382–397, 2001.

C. Yang, R. Duraiswami, and L. Davis. Efficient kernel machines using the improved fast Gauss transform. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1561–1568. MIT Press, 2005.

Journal of Machine Learning Research 14 (2013) 351-376 Submitted 11/11; Revised 11/12; Published 2/13

Risk Bounds of Learning Processes for Lévy Processes

Chao Zhang[*] ZHANGCHAO1015@GMAIL.COM
Center for Evolutionary Medicine and Informatics
Biodesign Institute, Arizona State University
Tempe, AZ 85287, U.S.A.

Dacheng Tao DACHENG.TAO@GMAIL.COM
Centre for Quantum Computation & Intelligent Systems
FEIT, University of Technology, Sydney
NSW 2007, Australia

Editor: Mehryar Mohri

Abstract

Lévy processes refer to a class of stochastic processes, for example, Poisson processes and Brownian motions, and play an important role in stochastic processes and machine learning. Therefore, it is essential to study risk bounds of the learning process for time-dependent samples drawn from a Lévy process (or briefly called learning process for Lévy process). It is noteworthy that samples in this learning process are not independently and identically distributed (i.i.d.). Therefore, results in traditional statistical learning theory are not applicable (or at least cannot be applied directly), because they are obtained under the sample-i.i.d. assumption. In this paper, we study risk bounds of the learning process for time-dependent samples drawn from a Lévy process, and then analyze the asymptotical behavior of the learning process. In particular, we first develop the deviation inequalities and the symmetrization inequality for the learning process. By using the resultant inequalities, we then obtain the risk bounds based on the covering number. Finally, based on the resulting risk bounds, we study the asymptotic convergence and the rate of convergence of the learning process for Lévy process. Meanwhile, we also give a comparison to the related results under the sample-i.i.d. assumption.

Keywords: Lévy process, risk bound, deviation inequality, symmetrization inequality, statistical learning theory, time-dependent

1. Introduction

In statistical learning theory, one of the major concerns is the risk bound, which explains the asymptotic behavior of the probability that a function produced by an algorithm has a sufficiently small error. Generally, there are three essential parts in the process of obtaining risk bounds: deviation or concentration inequalities, symmetrization inequalities and complexity measures of function classes. For example, Van der Vaart and Wellner (1996) showed risk bounds based on the Rademacher complexity and the covering number by using Hoeffding's inequality. Vapnik (1998) gave risk bounds based on the annealed Vapnik-Chervonenkis (VC) entropy and the VC dimension, respectively. In Vapnik (1998), Vapnik applied some classical inequalities, for example, Chernoff's inequality and Hoeffding's inequality, but also developed specific concentration inequalities

∗. This work was partly completed when the author was with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore.

to study the asymptotic behavior of the i.i.d. empirical process. Bartlett et al. (2005) proposed the local Rademacher complexity and obtained a sharp risk bound for a particular function class $\{f \in \mathcal{F} | \mathrm{E}f^2 < \beta \mathrm{E}f, \beta > 0\}$ by using Talagrand's inequality. Moreover, there are also other investigations to statistical learning theory (see Cesa-Bianchi and Gentile, 2008; Mendelson, 2008, 2002; Koltchinskii, 2001). However, all of these results are built under the sample-i.i.d. assumption.

Samples are not always i.i.d. in practice, for example, some financial and physical behaviors are temporally dependent, and the aforementioned research results are not applicable (or at least cannot be applied directly) to most cases. Thus, it is essential to study the risk bounds in the scenario of non-i.i.d. samples. The scenario of non-i.i.d. samples contains a wide variety of cases and it is impossible to find a unified form to cover all the cases. Instead, one feasible scheme is to find some representative processes, such as the Lévy process and the mixing process that cover several useful cases in the scenario of non-i.i.d. samples, and then we study the theoretical properties of each process individually.

Recently, Mohri and Rostamizadeh (2010) obtained risk bounds for stationary β-mixing sequences based on the Rademacher complexity. Mixing sequences can be deemed as a transition between the i.i.d. scenario and the non-i.i.d. scenario, where the dependence between samples diminishes along time. Especially, by adopting a technique of independent blocks (Yu, 1994), samples drawn from a β-mixing sequence can be transformed to an i.i.d. scenario and thus some classical results under the sample-i.i.d. assumption can be applied to obtain the risk bounds. Jiang (2009) extended Hoeffding's inequality to handle the situations with unbounded loss and dependent data, and then provided probability bounds for uniform deviations in a general framework involving discrete decision rules, unbounded loss and a dependence structure. Moreover, there are also some works about the uniform laws for dependent processes (Nobel and Dembo, 1993).

Lévy processes are the stochastic processes with stationary and independent increments and cover a large class of stochastic processes, for example, Brownian motions, Poisson processes, stable processes and subordinators (see Kyprianou, 2006). Moreover, Lévy processes have been regarded as prototypes of semimartingales and Feller-Markov processes (Applebaum, 2004b; Sato, 2004). Lévy processes have been successfully applied to practical applications in finance (Cont and Tankov, 2006), physics (Applebaum, 2004a), signal processing (Duncan, 2009), image processing (Pedersen et al., 2005) and actuarial science (Barndorff-Nielsen et al., 2001). Figueroa-López and Houdré (2006) used projection estimators to estimate the Lévy density, and then gave a bound to exhibit the discrepancy between a projection estimator and the orthogonal projection by using the concentration inequalities for functionals of Poisson integrals. In this paper, we extend the existing works on the infinitely divisible distribution (see Houdré, 2002; Houdré et al., 1998) to develop the deviation inequalities for Lévy processes and then obtain the risk bounds by using the resulted deviation inequalities. Next, we summarize the main results of this paper.

1.1 Overview of Main Results

This paper is mainly concerned with the theoretical analysis of the learning process for the time-dependent samples drawn from a Lévy process. There are four major concerns in this paper: the deviation inequality for Lévy process; the symmetrization inequality of the learning process; the risk bounds and the asymptotical behavior of the learning process.

Generally, in order to obtain the risk bounds of a certain learning process, it is necessary to first obtain the corresponding concentration (or deviation) inequalities for the learning process. Thus, we

extend the previous works (Houdré, 2002; Houdré et al., 1998) to develop the deviation inequalities for the Lévy process, which are suitable for the sequence of random variables at different time points. We then present the symmetrization inequality of the learning process for Lévy process. By applying the derived deviation and symmetrization inequalities, we obtain the risk bounds of the learning process, which is based on the covering number. Finally, we use the resulted risk bounds to analyze the asymptotical convergence and the rate of convergence of the learning process for Lévy process, respectively. Meanwhile, we also give a comparison with the learning process for i.i.d. samples.

Zhang and Tao (2010) discussed risk bounds for Lévy process with *zero* Gaussian component, but their results are based on some specific assumptions to function classes. The current results do not require any conditions of function classes except the boundedness and the Lipschitz continuity and are valid for a more general scenario where the considered Lévy process has non-zero Gaussian component, so they are more general than the previous results.

1.2 Organization of the Paper

The rest of this paper is organized as follows. In Section 2, we formalize the main research of this paper. Section 3 introduces some preliminaries of the infinitely divisible (ID) distribution and the Lévy process. We present the deviation inequalities and the symmetrization inequality of the learning process for Lévy process in Section 4. Section 5 gives the risk bounds of the learning process. In Section 6, we analyze the asymptotic behavior of the learning process and the last section concludes the paper. The proofs of main results are given in the appendices including Theorem 8, Theorem 11, Theorem 12 and Theorem 15.

2. Problem Setup

Denote $\mathcal{X} \subset \mathbb{R}^I$ as an input space and $\mathcal{Y} \subset \mathbb{R}^J$ as its corresponding output space. Let $\mathbf{Z} = (\mathcal{X}, \mathcal{Y}) \subset \mathbb{R}^K$ $(K = I + J)$ and $\{Z_t\}_{t\geq 0}$ be an undetermined Lévy process. Assume that $\mathbf{Z} = \{Z_t\}_{t\geq 0}$ with $Z_t = (\mathbf{x}_t, \mathbf{y}_t)$. Let $\mathcal{G} \subset \mathcal{Y}^{\mathcal{X}}$ be a function class with the domain $\mathcal{X}$ and the range $\mathcal{Y}$. Given a loss function $\ell : \mathcal{Y}^2 \to \mathbb{R}$ and a time interval $[T_1, T_2]$, it is expected to find a function $g^* \in \mathcal{G} : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected risk

$$\mathrm{E}(\ell \circ g) := \frac{1}{T} \int_{T_1}^{T_2} \int \ell(g(\mathbf{x}_t), \mathbf{y}_t) dP_t dt, \;\; g \in \mathcal{G}, \tag{1}$$

where $T = T_2 - T_1$, P_t stands for the distribution of $Z_t = (\mathbf{x}_t, \mathbf{y_t})$ at time t and $\ell(g(x), y)$ is denoted as $(\ell \circ g)(x, y)$.

Generally, if P_t $(t \in [T_1, T_2])$ are unknown, the target function g^* usually cannot be directly obtained by minimizing (1). Instead, we can apply the empirical risk minimization (ERM) principle to handle this issue. Given a function class $\mathcal{G}$ and a sample set $\mathbf{Z}_1^N := \{Z_{t_n}\}_{n=1}^N$ drawn from $\mathbf{Z}$ in the time interval $[T_1, T_2]$ with $T_1 \leq t_1 < \cdots < t_N \leq T_2$, we define the empirical risk of $g \in \mathcal{G}$ as

$$\mathrm{E}_N(\ell \circ g) := \frac{1}{N} \sum_{n=1}^{N} \ell(g(\mathbf{x}_{t_n}), \mathbf{y}_{t_n}), \tag{2}$$

which is considered as an approximation to the expected risk (1). Let $g_N \in \mathcal{G}$ be the function that minimizes the empirical risk (2) over $\mathcal{G}$ and we deem g_N as an estimate to g^* with respect to $\mathbf{Z}_1^N$.

It is noteworthy that such learning process covers many kinds of practical applications, for example, the predicting for time series (Mukherjee et al., 1997; Kim, 2003) and the estimation of channel state information (Biguesh and Gershman, 2006; Love et al., 2008; Tulino et al., 2005). We take the estimation of channel state information for example.

In the estimation of channel state information, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^I$ and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^J$ are regarded as the transmit and the receive vectors, respectively. The following are the reasons why we suppose that $\mathbf{Z}$ is a segment of an undetermined Lévy process:

- In fact, the tasks of the estimation of channel state information are time-dependent and can be regarded as the approximation to unknown stochastic processes.
- The Lévy process is one of representative processes and covers a large body of stochastic processes, that is, Brownian motions, Poisson processes, compound Poisson processes, Gamma processes and inverse Gaussian processes (see Kyprianou, 2006).
- Many kinds of signals have the Poisson property, the martingale property or both of them.

One of the most frequently used models is $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where $\mathbf{H}$ and $\mathbf{n}$ are the channel matrix and the noise vector, respectively (see Love et al., 2008; Tulino et al., 2005). The corresponding function class $\mathcal{G}$ can be formalized as $\mathcal{G} := \{\mathbf{x} \mapsto \mathbf{H}\mathbf{x} + \mathbf{n} : \mathbf{H} \in \mathbb{R}^J \times \mathbb{R}^I,\ \mathbf{n} \in \mathbb{R}^J\}$. The loss function ℓ is selected as the mean square error function, and then the least-square estimation is used to find the function that minimizes the empirical risk (2). Moreover, there are also other ERM-based methods proposed for the estimation of channel state information (see Sanchez-Fernandez et al., 2004; Sutivong et al., 2005).

In the aforementioned learning process, we are mainly interested in the asymptotic behavior of the quantity $(\mathrm{E}(\ell \circ g^*) - \mathrm{E}_N(\ell \circ g_N))$, when the sample number N goes to the *infinity*. Since $\mathrm{E}_N(\ell \circ g^*) - \mathrm{E}_N(\ell \circ g_N) \geq 0$, we have

$$\begin{aligned}
\mathrm{E}(\ell \circ g_N) =& \mathrm{E}(\ell \circ g_N) - \mathrm{E}(\ell \circ g^*) + \mathrm{E}(\ell \circ g^*) \\
\leq& \mathrm{E}_N(\ell \circ g^*) - \mathrm{E}_N(\ell \circ g_N) + \mathrm{E}(\ell \circ g_N) - \mathrm{E}(\ell \circ g^*) + \mathrm{E}(\ell \circ g^*) \\
\leq& 2 \sup_{g \in \mathcal{G}} \left|\mathrm{E}(\ell \circ g) - \mathrm{E}_N(\ell \circ g)\right| + \mathrm{E}(\ell \circ g^*),
\end{aligned}$$

and thus

$$0 \leq \mathrm{E}(\ell \circ g_N) - \mathrm{E}(\ell \circ g^*) \leq 2 \sup_{g \in \mathcal{G}} \left|\mathrm{E}(\ell \circ g) - \mathrm{E}_N(\ell \circ g)\right|.$$

The supremum

$$\sup_{g \in \mathcal{G}} \left|\mathrm{E}(\ell \circ g) - \mathrm{E}_N(\ell \circ g)\right| \tag{3}$$

is the so-called risk bound of the learning process for a Lévy process $\{Z_t\}_{t \geq 0}$.

Then, we define the loss function class

$$\mathcal{F} := \{Z \mapsto \ell(g(\mathbf{x}), \mathbf{y}) : g \in \mathcal{G}\}, \tag{4}$$

and call $\mathcal{F}$ the function class in the rest of this paper. Given a sample set $\{Z_{t_n}\}_{n=1}^N$ drawn from $\{Z_t\}_{t \geq 0}$, we shortly denote for any $f \in \mathcal{F}$,

$$\mathrm{E}_t f := \int f(Z) dP_t\ ,\ \ t > 0, \tag{5}$$

and

$$\mathrm{E}_N f := \frac{1}{N}\sum_{n=1}^{N} f(Z_{t_n}), \tag{6}$$

where E_t stands for the expectation taken with respect to Z_t.

According to (3), (4), (5) and (6), we have

$$\begin{aligned}
&\sup_{f\in\mathcal{F}} |\mathrm{E}f - \mathrm{E}_N f| \\
=&\sup_{g\in\mathcal{G}} \left|\mathrm{E}(\ell\circ g) - \mathrm{E}_N(\ell\circ g)\right| \\
=&\sup_{g\in\mathcal{G}} \left|\frac{1}{T}\int_{T_1}^{T_2}\int \ell(g(\mathbf{x}_t),\mathbf{y}_t)dP_t dt - \frac{1}{N}\sum_{n=1}^{N}\ell(g(\mathbf{x}_{t_n}),\mathbf{y}_{t_n})\right| \\
\leq& 2 \sup_{\substack{g\in\mathcal{G}\\ t\in[T_1,T_2]}} \left|\int \ell(g(\mathbf{x}_t),\mathbf{y}_t)dP_t - \frac{1}{N}\sum_{n=1}^{N}\ell(g(\mathbf{x}_{t_n}),\mathbf{y}_{t_n})\right| \\
=& 2 \sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \left|\mathrm{E}_t f - \mathrm{E}_N f\right|.
\end{aligned}$$

Therefore, the supremum

$$\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \left|\mathrm{E}_t f - \mathrm{E}_N f\right|$$

plays an important role in studying the risk bound $\sup_{f\in\mathcal{F}}|\mathrm{E}f - \mathrm{E}_N f|$ of the learning process for Lévy process.

3. Infinitely Divisible Distributions and Lévy Processes

Since the infinitely divisible (ID) distribution is strongly related to the Lévy process, this section first introduces the ID distribution and then briefs the Lévy process for the subsequent discussion.

3.1 ID Distributions

A probability distribution is said to be infinitely divisible if and only if it can be represented by the distribution of the sum of an arbitrary number of i.i.d. random variables. Many probability distributions have the infinite divisibility, for example, Poisson, geometric, lognormal, noncentral chi-square, exponential, Gamma, Pareto and Cauchy (see Bose et al., 2002). The ID distribution can be defined based on the characteristic function.

Definition 1 *Let $\phi(\theta)$ be the characteristic function of a random variable Z, that is*

$$\phi(\theta) := \mathrm{E}\left\{\mathrm{e}^{i\theta Z}\right\} = \int_{-\infty}^{+\infty}\mathrm{e}^{i\theta Z}dP(Z). \tag{7}$$

Then, the distribution of Z is infinitely divisible if and only if for any $N\in\mathbb{N}$, there exists a characteristic function $\phi_N(\theta)$ such that

$$\phi(\theta) = \underbrace{\phi_N(\theta) * \cdots * \phi_N(\theta)}_{N},$$

where "$*$" stands for multiplication.

By (7), given a characteristic function $\phi(\theta)$, we define the corresponding characteristic exponent as

$$\psi(\theta) := \ln \phi(\theta) = \ln \left(\mathrm{E} e^{i\theta Z} \right).$$

Afterward, we will show that the characteristic exponent of any ID distribution has a unified form (see Sato, 2004). Before the formal presentation, we need to give a definition of the Lévy measure (see Applebaum, 2004a).

Definition 2 *Let ν be a Borel measure defined on $\mathbb{R}^K \setminus \{0\}$. This ν will be a Lévy measure if*

$$\int_{\mathbb{R}^K \setminus \{0\}} \min\{\|u\|^2, 1\} \nu(du) < \infty,$$

and $\nu(\{0\}) = 0$.

The Lévy measure describes the expected number of a certain height jump in a time interval of the unit length. Define the indicator function for the event $\mathcal{E}$ as

$$\mathbf{1}_{\mathcal{E}} = \begin{cases} 1, & \text{the event } \mathcal{E} \text{ appears;} \\ 0, & \text{otherwise,} \end{cases}$$

and for any ID random variable, its characteristic exponent takes the following form (see Sato, 2004).

Theorem 3 (Lévy-Khintchine) *A Borel probability measure μ of a random variable $Z \in \mathbb{R}^K$ is infinitely divisible if and only if there exists a triplet $(\mathbf{a}, \mathbf{A}, \nu)$ such that for all $\theta \in \mathbb{R}^K$, the characteristic exponent ψ_μ is of the form*

$$\psi_\mu(\theta) = i\langle \mathbf{a}, \theta \rangle - \frac{1}{2} \langle \theta, \mathbf{A}\theta \rangle + \int_{\mathbb{R}^K \setminus \{0\}} \left(e^{i\langle \theta, u \rangle} - 1 - i\langle \theta, u \rangle \mathbf{1}_{\|u\| \le 1} \right) \nu(du), \tag{8}$$

where $\mathbf{a} \in \mathbb{R}^K$, $\mathbf{A}$ is a $K \times K$ positive-definite symmetric matrix, ν is a Lévy measure on $\mathbb{R}^K \setminus \{0\}$, and $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ stand for the inner product and the norm in $\mathbb{R}^K$, respectively.

Theorem 3 shows that an ID distribution can be completely determined by a triplet $(\mathbf{a}, \mathbf{A}, \nu)$, where $\mathbf{a}$ is a drift, $\mathbf{A}$ is a Gaussian component and ν is a Lévy measure. Thus, we call $(\mathbf{a}, \mathbf{A}, \nu)$ the generating triplet of an ID distribution.

3.2 Lévy Processes

First, we give a rigorous definition of Lévy processes.

Definition 4 *A stochastic process $\{Z_t\}_{t \ge 0}$ on $\mathbb{R}^K$ is a Lévy process if it satisfies the following conditions:*

1. *$Z_0 = 0$, almost surely.*

2. *For any $n \geq 1$ and $0 \leq t_0 \leq t_1 \leq \cdots \leq t_n$, the random variables*

$$Z_{t_0}, Z_{t_1} - Z_{t_0}, \cdots, Z_{t_n} - Z_{t_{n-1}}$$

are independent.

3. *The increments are stationary, that is, the distribution of $Z_{s+t} - Z_s$ is independent of s.*

4. *The process is right continuous, that is, for any $0 \leq t \leq s$ and $\varepsilon > 0$, we have*

$$\lim_{s \to t} \Pr\Big\{ |Z_t - Z_s| > \varepsilon \Big\} = 0.$$

According to Theorem 7.10 of Sato (2004), a Lévy process $\{Z_t\}_{t \geq 0}$ can be distinguished by the distribution of Z_1, which has an ID distribution with the generating triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$, and at any time $t > 0$, $Z_t \in \{Z_t\}_{t \geq 0}$ has an ID distribution with the generating triplet $(\mathbf{a}_t, \mathbf{A}_t, \nu_t)$. Therefore, we call $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$ the characteristic triplet of the Lévy process $\{Z_t\}_{t \geq 0}$. For any $t > 0$, there also holds that

$$(\mathbf{a}_t, \mathbf{A}_t, \nu_t) := (\mathbf{a}_1 t, \mathbf{A}_1 t, \nu_1 t).$$

Next, we introduce the Lévy-Ito decomposition to discuss the relationship between the path of a Lévy process and its characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$. The details are referred to Kyprianou (2006); Sato (2004).

Theorem 5 (Lévy-Ito Decomposition) *Consider a triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$ where $\mathbf{a}_1 \in \mathbb{R}^K$, $\mathbf{A}_1$ is a $K \times K$ positive-definite symmetric matrix, ν_1 is a Lévy measure on $\mathbb{R}^K \setminus \{0\}$. Then, there exist four independent Lévy processes, $L^{(1)}$, $L^{(2)}$, $L^{(3)}$ and $L^{(4)}$, where $L^{(1)}$ is a constant drift, $L^{(2)}$ is a Brownian motion, $L^{(3)}$ is a compound Poisson process and $L^{(4)}$ is a square integrable (pure jump) martingale with an a.s. countable number of jumps of magnitude less than 1 on each finite time interval. Taking $L = L^{(1)} + L^{(2)} + L^{(3)} + L^{(4)}$, there then exists a Lévy process $L = \{Z_t\}_{0 \leq t \leq T}$ with characteristic exponent in the form of* (8).

The proof of this theorem has been given by Chapter 4 in Sato (2004) or Chapter 2 in Kyprianou (2006), so we omit it here. We only go through some steps of the proof to reveal the relationship between the path of a Lévy process and its characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$. Recalling (8), we can split the characteristic exponent ψ into four parts:

$$\psi = \psi^{(1)} + \psi^{(2)} + \psi^{(3)} + \psi^{(4)}$$

with

$$\begin{aligned}
&\psi^{(1)}(\theta) = i\langle \mathbf{a}_1, \theta \rangle; \quad \psi^{(2)}(\theta) = -\frac{1}{2}\langle \theta, \mathbf{A}_1 \theta \rangle; \\
&\psi^{(3)}(\theta) = \int_{\|u\| \geq 1} \big(e^{i\langle \theta, u \rangle} - 1 \big) \nu_1(du); \\
&\psi^{(4)}(\theta) = \int_{\|u\| < 1} \big(e^{i\langle \theta, u \rangle} - 1 - i\langle \theta, u \rangle \big) \nu_1(du),
\end{aligned}$$

which correspond to $L^{(1)}$, $L^{(2)}$, $L^{(3)}$ and $L^{(4)}$, respectively. We also refer to Jacobsen (2005) for the knowledge on jump processes as well as Lévy processes. At the end of this section, we give two examples of Lévy processes in addition to the corresponding Lévy-Khintchine representations and Lévy-Ito decompositions:

- A Poisson process $\{N_t\}_{t\geq 0}$ is a Lévy process that has a Poisson distribution with parameter pt at any time $t > 0$. In the Lévy-Khintchine representation, we find that $\mathbf{a}_1$ and $\mathbf{A}_1$ are both *zero* and $\nu_1 = p\delta_1$, where δ_1 is the Dirac measure supported on $\{1\}$. In the Lévy-Ito decomposition, its characteristic exponent is expressed as $\psi(\theta) = \psi^{(3)}(\theta) = p(e^{i\theta} - 1)$.

- A scaled Brownian motion with linear drift is also a Lévy process with the characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, 0)$ in the Lévy-Khintchine representation. In the Lévy-Ito decomposition, its characteristic exponent is expressed as $\psi(\theta) = \psi^{(1)}(\theta) + \psi^{(2)}(\theta)$ with $\psi^{(1)}(\theta) = i\langle \mathbf{a}_1, \theta\rangle$ and $\psi^{(2)}(\theta) = -\frac{1}{2}\langle \theta, \mathbf{A}_1\theta\rangle$.

4. Deviation Inequalities and Symmetrization Inequalities

In this section, we present the deviation inequalities and symmetrization inequality of the learning process for Lévy process.

4.1 Preliminaries

Firstly, we need to introduce some notations and conditions for the following discussion.

4.1.1 NOTATIONS

Assume that $\mathcal{F}$ is a function class consisting of λ-Lipschitz functions and $\{Z_t\}_{t\geq 0} \subset \mathbb{R}^K$ is a Lévy process with the characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$. Let $\mathbf{Z}_1^N = \{Z_{t_n}\}_{n=1}^N$ be a sample set drawn from $\{Z_t\}_{t\geq 0}$ in the time interval $[T_1, T_2]$. For any $t \in [T_1, T_2]$, we give the following definitions:

$$
\begin{aligned}
&(D1) \quad \Sigma_N^{(*)} := \sup_{t\in[T_1,T_2]} \frac{1}{N}\sum_{n=1}^{N} \sup_{f\in\mathcal{F}} |\mathrm{E}_{t_n} f - \mathrm{E}_t f|;\\
&(D2) \quad \varphi(\alpha) := \sum_{n=1}^{N} \lambda^2 \pi K^2 \alpha t_n + \int_{\mathbb{R}^K} \lambda \|u\| \left(e^{\lambda\alpha\|u\|} - 1\right) \nu_{t_n}(du);\\
&(D3) \quad V^{(n)} := \int_{\mathbb{R}^K} \|u\|^2 \nu_{t_n}(du) = t_n \int_{\mathbb{R}^K} \|u\|^2 \nu_1(du);\\
&(D4) \quad \Gamma(x) := x - (x+1)\ln(x+1).
\end{aligned}
$$

Note that the quantity $\sup_{f\in\mathcal{F}} |\mathrm{E}_{t_n} f - \mathrm{E}_t f|$ is called the integral probability metric and has been widely used to measure the difference between two probability distributions (see Zolotarev, 1984; Rachev, 1991; Müller, 1997; Reid and Williamson, 2011). Recently, Sriperumbudur et al. (2012) gave the further investigation and proposed the empirical method to compute the integral probability metric. As mentioned by Müller (1997), the quantity $\sup_{f\in\mathcal{F}} |\mathrm{E}_{t_n} f - \mathrm{E}_t f|$ is a (semi)metric to measure the difference between the distributions of $\{Z_t\}_{t\geq 0}$ at two time points t and t_n. In fact, given a non-trivial function class $\mathcal{F}$, the quantity $\sup_{f\in\mathcal{F}} |\mathrm{E}_{t_n} f - \mathrm{E}_t f|$ is equal to *zero* if the distributions at the two time points match or the two time points coincide, that is, $t = t_n$.

4.1.2 CONDITIONS

In order to achieve the desired risk bounds, some necessary conditions need to be introduced to specify the behavior of Lévy processes.

$(C1)$ The f is a partially differentiable function on $\mathbb{R}^K$ and there exists a constant $\lambda > 0$ such that for any $Z = (z_1, \cdots, z_K)^T \in \mathbb{R}^K$,

$$\max_{1 \le k \le K} \left| \frac{\partial f(Z)}{\partial z_k} \right| \le \lambda.$$

$(C2)$ Denoting $\mathbf{A}_1 = \{a_{ij}\}_{K \times K}$, there exists a constant $\pi > 0$ such that

$$\max_{1 \le i,j \le K} |a_{ij}| \le \pi.$$

$(C3)$ The ν_1 has a bounded support with

$$R = \inf\{\rho > 0 : \nu_1(\{u : \|u\| > \rho\}) = 0\}.$$

Condition $(C1)$ implies that f has bounded partial derivatives and holds for many kinds of functions, for example, quadratic functions with bounded domains and trigonometric functions. The constant λ is determined by the selected function and thus it is manipulatable. Condition $(C2)$ implies that all entries of $\mathbf{A}_1$ are bounded. Condition $(C3)$ implies the Lévy measure ν_1 has a bounded support. To take an example of Conditions $(C2)$-$(C3)$, we refer to Poisson processes whose characteristic triplet is $(0,0,\nu_1)$ with ν_1 supporting on $\{1\}$. Afterwards, we come up with the deviation inequalities of the learning process for Lévy process.

4.2 Deviation Inequalities

Deviation inequalities play an essential role in obtaining risk bounds of a certain learning process. Generally, specific deviation inequalities need to be developed for different learning processes. There are a lot of popular concentration inequalities and deviation inequalities, for example, Hoeffding's inequality, McDiarmid's inequality, Bennett's inequality, Bernstein's inequality and Talagrand's inequality, which are all valid under the sample-i.i.d. assumption. Moreover, there have been the deviation inequalities for ID distributions and Lévy processes both with *zero* Gaussian components proposed by Houdré (2002); Houdré and Marchal (2008), respectively. Here, we extend the deviation results in Houdré (2002) to develop the deviation inequalities of the learning process for Lévy process, which is related to a sequence of random variables taking values from a Lévy process at different time points.

Based on a fact that the vector formed by N independent ID random vectors is itself infinitely divisible, the following theorem and corollary can be derived from Theorem 1 & Corollary 1 of Houdré (2002) and Proposition 2 of Houdré et al. (1998). We also refer to Zhang and Tao (2011a,b) for the related discussions.

Theorem 6 *Assume that f is a function satisfying Condition $(C1)$ and $\{Z_t\}_{t \ge 0} \subset \mathbb{R}^K$ is a Lévy process with the characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$ satisfying Condition $(C2)$. Let $\mathbf{Z}_1^N = \{Z_{t_n}\}_{n=1}^N$ $(t_1 < t_2 < \cdots < t_N)$ be a set of time-dependent samples drawn from $\{Z_t\}_{t \ge 0}$ in the time interval $[T_1, T_2]$. Define a function $F : \mathbb{R}^{NK} \to \mathbb{R}$ as*

$$F\left(\mathbf{Z}_1^N\right) := \sum_{n=1}^{N} f(Z_{t_n}). \tag{9}$$

If Condition $(C1)$ *is valid and* $\mathrm{E}e^{\alpha\|\mathbf{z}_t\|}\big|_{t=1} < +\infty$ *holds for some* $\alpha > 0$*, then we have for any* $0 < \xi < \varphi((M/\lambda)^-)$*,*

$$\Pr\left\{\left|F\left(\mathbf{Z}_1^N\right) - \overline{\mathrm{E}}F\right| > \xi\right\} \leq 2\exp\left\{-\int_0^{\xi} \varphi^{-1}(s)ds\right\}, \tag{10}$$

where the expectation $\overline{\mathrm{E}}$ *is taken on all* $\{Z_{t_1}, \cdots, Z_{t_N}\}$*,* φ *is given in Definition* $(D2)$*,* $\varphi(a^-)$ *is the left-hand limit of* φ *at* a*,* $M = \sup\left\{\alpha > 0 : \mathrm{E}e^{\alpha\|\mathbf{z}_t\|}\big|_{t=1} < +\infty\right\}$ *and* φ^{-1} *is the inverse of* $\varphi(\alpha)$ *with the domain of* $0 < \alpha < M/\lambda$*.*

In Theorem 6, we present a deviation inequality of the learning process for the Lévy process satisfying Condition $(C2)$. However, there are two drawbacks of this result that will bring some difficulties to the future theoretical analysis of asymptotic behavior.

- The deviation inequality (10) is represented by the integral of φ^{-1}, and thus the inequality cannot explicitly reflect the asymptotic behavior as N goes to the *infinity*.
- Recalling Definition $(D2)$, there is an integral term in the expression of the function φ. Thus, given a certain $\xi > 0$, it may be difficult to justify whether the ξ satisfies the condition $\xi < \varphi((M/\lambda)^-)$.

In order to overcome these drawbacks, we add Condition $(C3)$ to achieve another deviation inequality for the learning process.

Corollary 7 *Follow notations in Theorem 6. If Conditions* $(C1)$*-*$(C3)$ *are all valid, then we have for any* $\xi > 0$*,*

$$\begin{aligned}
&\Pr\left\{\left|F\left(\mathbf{Z}_1^N\right) - \overline{\mathrm{E}}F\right| > \xi\right\} \\
\leq& 2\exp\left\{\frac{\sum_{n=1}^N (\lambda^2\pi K^2\alpha t_n + V^{(n)})}{(\lambda R)^2}\Gamma\left(\frac{\lambda R\xi}{\sum_{n=1}^N (\lambda^2\pi K^2\alpha t_n + V^{(n)})}\right)\right\} \\
\leq& 2\exp\left\{\frac{NT_1(\lambda^2\pi K^2\alpha + V)}{(\lambda R)^2}\Gamma\left(\frac{\lambda R\xi}{NT_2(\lambda^2\pi K^2\alpha + V)}\right)\right\},
\end{aligned} \tag{11}$$

where Γ *is given in Definition* $(D4)$ *and*

$$V := \int_{\mathbb{R}^K} \|u\|^2 \nu_1(du).$$

The second inequality of the above result is derived from the facts that there holds that $V^{(n)} \leq T_2 V$ for any $1 \leq n \leq N$ and the function $\Gamma(x)$ is a monotonically decreasing function when $x > 0$ as shown in Figure 1.

Compared to the result (10), the deviation inequality (11) holds for any $\xi > 0$ and its right-hand-side is represented by using the function $\Gamma(x)$ $(x > 0)$. Therefore, we can directly analyze the asymptotic behavior as N goes to the *infinity*. In fact, since the function $\Gamma(x)$ is smaller than *zero* when $x > 0$, the right-hand-side of (11) will go to *zero* for any $\xi > 0$ when N approaches to the *infinity*. Next, we present the symmetrization inequality of the learning process for Lévy process.

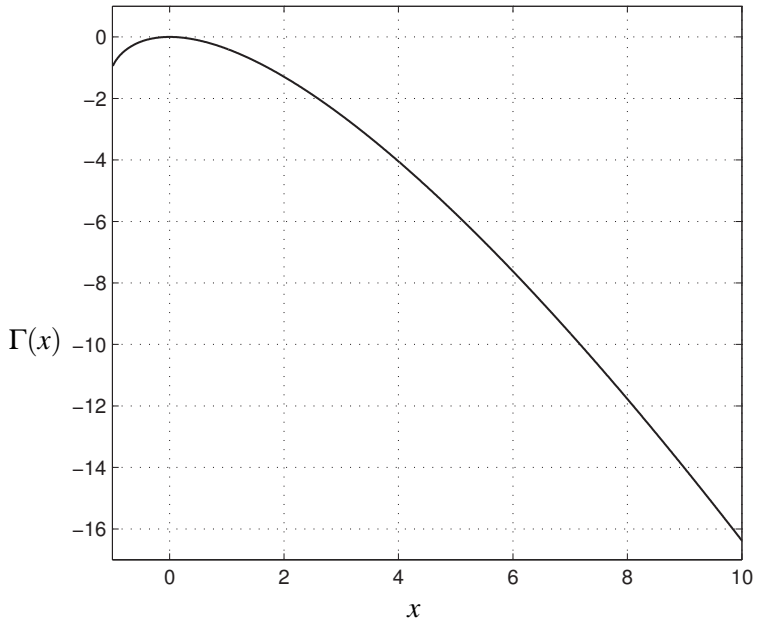


Figure 1: The Function Curve of $\Gamma(x)$

4.3 Symmetrization Inequality

Symmetrization inequalities are mainly used to replace the expected risk by an empirical risk computed on another sample set that is independent of the given sample set but has the identical distribution. In this manner, risk bounds can be achieved by using some kinds of complexity measures, for example, the covering number and the VC dimension. However, the classical symmetrization results (see Bousquet et al., 2004) are only valid under the sample-i.i.d. assumption. Afterward, we propose the symmetrization inequality of the learning process for Lévy process.

For clarity of presentation, we give a notation that will be used in the rest of the paper. Given a sample set $\mathbf{Z}_1^N = \{Z_{t_n}\}_{n=1}^N$ $(t_1 < t_2 < \cdots < t_N)$, we denote $\mathbf{Z}'^N_1 := \{Z'_{t_n}\}_{n=1}^N$ as the ghost sample set of $\mathbf{Z}_1^N$, where Z'_{t_n} has the same distribution as Z_{t_n} for any $1 \le n \le N$. Then, the following theorem presents the symmetrization inequality of the learning process.

Theorem 8 *Assume that $\mathcal{F}$ is a function class with the range $[a,b]$ and $\{Z_t\}_{t\ge 0} \subset \mathbb{R}^K$ is a Lévy process. Let $\mathbf{Z}_1^N$ and $\mathbf{Z}'^N_1$ be drawn from $\{Z_t\}_{t\ge 0}$ in the time interval $[T_1,T_2]$. Then, given any $\xi > \Sigma_N^{(*)}$, we have for any $N \ge \frac{8(b-a)^2}{(\xi')^2}$ with $\xi' = \xi - \Sigma_N^{(*)}$,*

$$\Pr\left\{\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \left|\mathrm{E}_t f - \mathrm{E}_N f\right| > \xi\right\} \le 2\Pr\left\{\sup_{f\in\mathcal{F}} \left|\mathrm{E}'_N f - \mathrm{E}_N f\right| > \frac{\xi'}{2}\right\}. \tag{12}$$

This theorem shows that given $\xi > 0$, the probability of the event:

$$\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \left|\mathrm{E}_t f - \mathrm{E}_N f\right| > \xi$$

can be bounded by using the probability of the event:

$$\left|\mathrm{E}'_N f - \mathrm{E}_N f\right| > \frac{\xi'}{2}$$

that is only determined by the characteristics of the sample sets $\mathbf{Z}_1^N$ and $\mathbf{Z}'^N_1$, when $N \geq \frac{8(b-a)^2}{(\xi')^2}$ for any given $\xi' > 0$ with $\xi' = \xi - \Sigma_N^{(*)}$. Compared with the classical symmetrization result under the sample-i.i.d. assumption (see Bousquet et al., 2004), the derived symmetrization inequality (12) incorporated a discrepancy term $\Sigma_N^{(*)}$ and the two results coincide when the time interval $[T_1, T_2]$ shrinks to one time point that match to t, that is, $T_1 = T_2 = t$ that results in $\Sigma_N^{(*)} = 0$.

In the next section, we use the resulted deviation inequalities and symmetrization inequality to achieve the risk bounds of the learning process for Lévy process.

5. Risk Bounds of Learning Processes for Lévy Processes

In this section, we present the risk bounds of the learning process for Lévy process. Since the resulting bounds are based on the covering number, we first introduce the definition of the cover and then present the definition of the covering number of $\mathcal{F}$.

Definition 9 *Let $\mathcal{N}$ be a collection of sets. Then, the collection $\mathcal{N}$ is said to be a cover of a given set Ω, if for any $\mathbf{x} \in \Omega$, there always exists an element of $\mathcal{N}$ that contains the point $\mathbf{x}$.*

Next, we define the the covering number of $\mathcal{F}$ as follows.

Definition 10 *Let $\mathbf{Z}_1^N$ be a sample set drawn from a distribution $\mathbf{Z}$. For any $1 \leq p \leq \infty$ and $\xi > 0$, the covering number of $\mathcal{F}$ at radius ξ, with respect to $\ell_p(\mathbf{Z}_1^N)$, denoted by $\mathcal{N}(\mathcal{F}, \xi, \ell_p(\mathbf{Z}_1^N))$ is the minimum size of a cover of radius ξ.*

Subsequently, we come up with the main results of this paper.

Theorem 11 *Assume that $\mathcal{F}$ is a function class composed of functions satisfying Condition $(C1)$ with the range $[a,b]$ and $\{Z_t\}_{t\geq 0} \subset \mathbb{R}^K$ is a Lévy process with the characteristic triplet $(\mathbf{a}_1, \mathbf{A}_1, \nu_1)$ satisfying Condition $(C2)$. Let $\mathbf{Z}_1^N$ and $\mathbf{Z}'^N_1$ be drawn from $\{Z_t\}_{t\geq 0}$ in the time interval $[T_1, T_2]$, and denote $\mathbf{Z}_1^{2N} := \{\mathbf{Z}_1^N, \mathbf{Z}'^N_1\}$. Given any $\Sigma_N^{(*)} < \xi < \Sigma_N^{(*)} + \frac{8\varphi((M/\lambda)^-)}{N}$, if $\mathrm{E}e^{\alpha\|\mathbf{z}_t\|}\big|_{t=1} < +\infty$ holds for*

some $\alpha > 0$*, then we have for any* $N \geq \frac{8(b-a)^2}{(\xi')^2}$ *with* $\xi' = \xi - \Sigma_N^{(*)}$,

$$\begin{aligned}
&\Pr\left\{\sup_{f\in\mathcal{F}}\frac{1}{2}\left|\mathrm{E}f-\mathrm{E}_N f\right|>\xi\right\}\\
\leq&\Pr\left\{\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}}\left|\mathrm{E}_t f-\mathrm{E}_N f\right|>\xi\right\}\\
\leq&8\mathrm{E}\mathcal{N}\left(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N})\right)\exp\left\{-\int_0^{\frac{N\xi'}{8}}\varphi^{-1}(s)ds\right\},
\end{aligned}\tag{13}$$

where $\varphi(a^-)$ *denotes the left-hand limit of* φ *at the point a,* $M = \sup\left\{\alpha>0 : \mathrm{E}e^{\alpha\|\mathbf{z}_t\|}\big|_{t=1} < +\infty\right\}$ *and* φ^{-1} *is the inverse of* $\varphi(\alpha)$ *with the domain of* $0<\alpha<M/\lambda$.

The result shown in this theorem has the same drawbacks as those of Theorem 6. The right-hand-side of the inequality (13) is represented by using the integrals of φ^{-1}, so it is difficult to find the asymptotic behavior of the risk bound as N goes to the *infinity*. The range $0<\xi'<\frac{8\varphi((M/\lambda)^-)}{N}$ of ξ' is expressed by incorporating the function φ that contains an integral term [see Definition $(D2)$]. These will bring difficulties to the future theoretical analysis of asymptotic convergence. To overcome the two drawbacks, we develop another risk bound of the learning process for Lévy process by adding a mild condition $(C3)$ that requires that the Lévy measure ν have a bounded support.

Theorem 12 *Follow notations in Theorem 11. Given any* $\xi > \Sigma_N^{(*)}$*, if Conditions* $(C1)$*-*$(C3)$ *are valid, then we have for any* $N \geq \frac{8(b-a)^2}{(\xi')^2}$ *with* $\xi' = \xi - \Sigma_N^{(*)}$,

$$\begin{aligned}
&\Pr\left\{\sup_{f\in\mathcal{F}}\frac{1}{2}\left|\mathrm{E}f-\mathrm{E}_N f\right|>\xi\right\}\\
\leq&\Pr\left\{\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}}\left|\mathrm{E}_t f-\mathrm{E}_N f\right|>\xi\right\}\\
\leq&8\mathrm{E}\mathcal{N}\left(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N})\right)\exp\left\{\frac{NT_1(\lambda^2\pi K^2\alpha+V)}{(\lambda R)^2}\Gamma\left(\frac{\lambda R(\xi-\Sigma_N^{(*)})}{8T_2(\lambda^2\pi K^2\alpha+V)}\right)\right\},
\end{aligned}\tag{14}$$

where Γ *is given in Definition* $(D4)$.

This theorem shows that under Conditions $(C1)$*-*$(C3)$, given any $\xi > \Sigma_N^{(*)}$, the probability of the event that for any $N \geq \frac{8(b-a)^2}{(\xi')^2}$ with $\xi' = \xi - \Sigma_N^{(*)}$,

$$\sup_{f\in\mathcal{F}}\left|\mathrm{E}f-\mathrm{E}_N f\right|>2\xi$$

can be bounded by the last term of (14). Until now, we have achieved the risk bound (3) and the result (14) can explicitly reflect the asymptotic behavior as N goes to the *infinity*. Following this result, the next section will discuss the asymptotical behavior of the learning process for Lévy process.

6. Convergence Analysis

Based on the risk bound (14), this section presents a detailed theoretical analysis to asymptotic convergence and the rate of convergence of the learning process for Lévy process. Meanwhile, we also give a comparison with the related results of the learning process for i.i.d. samples.

6.1 Asymptotic Convergence

In statistical learning theory, it is well-known that the complexity of function classes is the main factor to the asymptotic convergence of the learning process for i.i.d. samples (see Vapnik, 1998; Van der Vaart and Wellner, 1996; Mendelson, 2003).

Based on Theorem 12, we show that the asymptotic convergence of the learning process for Lévy process is affected by two factors: the covering number $\mathcal{N}(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N}))$ and the quantity $\Sigma_N^{(*)}$.

Recalling Definition $(D4)$, it is noteworthy that there is only one solution $x=0$ to the equation $\Gamma(x)=0$ and $\Gamma(x)$ is monotonically decreasing when $x\geq 0$ (see Figure 1). Thus, according to Theorem 12, we can obtain the following result that describes the asymptotic convergence of the learning process for Lévy process.

Theorem 13 *Assume that $\mathcal{F}$ is a function class composed of functions satisfying Condition $(C1)$ with the range $[a,b]$ and $\{Z_t\}_{t\geq 0}\subset\mathbb{R}^K$ is a Lévy process with the characteristic triplet $(\mathbf{a}_1,\mathbf{A}_1,\nu_1)$ satisfying Conditions $(C2)$ and $(C3)$. Let $\mathbf{Z}_1^N=\{Z_{t_n}\}_{n=1}^N$ and $\mathbf{Z}'^N_1=\{Z'_{t_n}\}_{n=1}^N$ be drawn from $\{Z_t\}_{t\geq 0}$ in the time interval $[T_1,T_2]$, and denote $\mathbf{Z}_1^{2N}:=\{\mathbf{Z}_1^N,\mathbf{Z}'^N_1\}$. If the following condition holds:*

$$\lim_{N\to+\infty}\frac{\ln \mathrm{E}\mathcal{N}\left(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N})\right)}{N}<+\infty, \tag{15}$$

then we have for any $\xi>\Sigma_N^{(*)}$ with $\xi'=\xi-\Sigma_N^{(*)}$,

$$\lim_{N\to+\infty}\Pr\left\{\sup_{f\in\mathcal{F}}\left|\mathrm{E}f-\mathrm{E}_Nf\right|>2\xi\right\}=0, \tag{16}$$

where $\mathrm{E}f$ and E_Nf are defined in (1) *and* (2)*, respectively.*

As shown in Theorem 13, if the covering number $\mathcal{N}(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N}))$ satisfies the condition (15), the probability of the event

$$\sup_{f\in\mathcal{F}}\left|\mathrm{E}f-\mathrm{E}_Nf\right|>2\xi$$

will converge to *zero* for any $\xi>\Sigma_N^{(*)}$, when the sample number N goes to the *infinity*. This is partially in accordance with the classical result given by Theorem 2.3 of Mendelson (2003): the probability of the event

$$\sup_{f\in\mathcal{F}}\left|\mathrm{E}f-\mathrm{E}_Nf\right|>\xi \tag{17}$$

will converge to *zero* for any $\xi > 0$, if the covering number $\mathcal{N}\left(\mathcal{F}, \xi, \ell_1(\mathbf{Z}_1^N)\right)$ satisfies the following condition:

$$\lim_{N \to +\infty} \frac{\ln \mathrm{E}\left\{\mathcal{N}\left(\mathcal{F}, \xi, \ell_1(\mathbf{Z}_1^N)\right)\right\}}{N} < +\infty. \tag{18}$$

Note that in the learning process for Lévy process, the uniform convergence of the empirical risk $\mathrm{E}_N f$ to the expected risk $\mathrm{E}f$ may not hold, because the limit (16) does not hold for any $\xi > 0$ but for any $\xi > \Sigma_N^{(*)}$. By contrast, the inequality (17) holds for all $\xi > 0$ in the learning process for i.i.d. samples, if the condition (18) is satisfied. Again, these two results coincide when the time interval $[T_1, T_2]$ shrinks to one single time point that matches to t, that is, $T_1 = T_2 = t$ that results in $\Sigma_N^{(*)} = 0$.

Interestingly, we show below that by ignoring the quantity $\Sigma_N^{(*)}$, the learning process for Lévy process has a faster rate of convergence than the classical result (see Mendelson, 2003, Theorem 2.3) in the large-derivation case.

6.2 Rate of Convergence

The classical result (see Mendelson, 2003, Theorem 2.3) is actually derived from Hoeffding's inequality. Thus, it is said to be of Hoeffding-type and can directly lead to its alternative expression as follows:

$$\sup_{f \in \mathcal{F}} \left|\mathrm{E}_N f - \mathrm{E}f\right| \leq O\left(\left(\frac{\ln \mathrm{E}\left\{\mathcal{N}\left(\mathcal{F}, \xi, \ell_1(\mathbf{Z}_1^N)\right)\right\} - \ln(\varepsilon/8)}{N}\right)^{\frac{1}{2}}\right), \tag{19}$$

which implies that the rate of convergence of the i.i.d. learning process is up to $O(1/\sqrt{N})$.

Recalling the classical Bennett's inequality (Bennett, 1962; Bousquet, 2002), we can find that the expression of the risk bound (14) is similar to that of Bennett's inequality, that is, both of them are in the form of $\mathrm{e}^{\Gamma(x)}$ with $\Gamma(x) = x - (x+1)\ln(x+1)$. For convenience, this form is said to be of Bennett-type. Differing from the Hoeffding-type result (see Mendelson, 2003, Theorem 2.3), it is difficult to directly achieve the alternative expression of the Bennett-type result (14), because it is difficult to obtain the analytical expression of the inverse function of $\Gamma(x)$. Instead, one generally uses the term $\frac{-x^2}{2+(2x/3)}$ to approximate the function $\Gamma(x)$ and then get the so-called Bernstein's inequality. In this way, we can obtain the following alternative expression of the Bennett-type result (14):[1]

$$\begin{aligned}\sup_{f \in \mathcal{F}} \left|\mathrm{E}_N f - \mathrm{E}f\right| \leq & 2\Sigma_N^{(*)} + \frac{64\lambda R T_2 \left(\ln \mathrm{E}\left\{\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^N)\right)\right\} - \ln(\varepsilon/8)\right)}{3NT_1} \\ & + \frac{16 T_2 \sqrt{2(\lambda^2 \pi K^2 \alpha + V)\left(\ln \mathrm{E}\left\{\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^N)\right)\right\} - \ln(\varepsilon/8)\right)}}{\sqrt{NT_1}},\end{aligned} \tag{20}$$

which implies that the rate of convergence of the learning process for Lévy process is also up to $O(1/\sqrt{N})$, which is in accordance with the classical result (19), if the discrepancy term $\Sigma_N^{(*)}$ is ignored.

1. The details are referred to `http://ocw.mit.edu/courses/mathematics/18-465-topics-in-statistics-statistical-learning-theory-spring-2007/lecture-notes/l6.pdf`.

Here, we adopt a new method to obtain another alternative expression of the Bennett-type risk bound (14) and show that the rate of convergence of the learning process can be up to $o(1/N^{\frac{1}{1.3}})$ in the large-deviation case.

Remark 14 *Here, "large-deviation" means that the discrepancy between the empirical risk and the expected risk is large (or not small). Given any $\xi > \Sigma_N^{(*)}$ with $\xi' := \xi - \Sigma_N^{(*)}$, one of our major concerns is the probability $\Pr\left\{\sup_{f\in\mathcal{F}}\left|\mathrm{E}_N f - \mathrm{E}f\right| > \xi\right\}$, and then we say that the case that $\frac{\lambda R\xi'}{8T_2(\lambda^2\pi K^2\alpha+V)} > 1.719$ is of large-deviation, that is, $\xi > \frac{13.752T_2(\lambda^2\pi K^2\alpha+V)}{\lambda R} + \Sigma_N^{(*)}$.*

Theorem 15 *Follow the notations and conditions of Theorem 12. Then, given any $\xi > \frac{13.752T_2(\lambda^2\pi K^2\alpha+V)}{\lambda R} + \Sigma_N^{(*)}$ and for any $N \geq \frac{8(b-a)^2}{(\xi')^2}$ with $\xi' = \xi - \Sigma_N^{(*)}$, we have with probability at least $1-\varepsilon$,*

$$\sup_{f\in\mathcal{F}}\left|\mathrm{E}_N f - \mathrm{E}f\right| \leq 2\Sigma_N^{(*)} + \frac{27.504T_2(\lambda^2\pi K^2\alpha+V)}{\lambda R} + \frac{16T_2(\lambda^2\pi K^2\alpha+V)}{\lambda R} \times \left(\frac{(\lambda R)^2\left(\ln \mathrm{E}\left\{\mathcal{N}\left(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N})\right)\right\} - \ln(\varepsilon/8)\right)}{NT_1(\lambda^2\pi K^2\alpha+V)}\right)^{\frac{1}{\gamma}},$$

where

$$\varepsilon := 8\mathrm{E}\mathcal{N}\left(\mathcal{F},\xi'/8,\ell_1(\mathbf{Z}_1^{2N})\right)\exp\left\{\frac{NT_1(\lambda^2\pi K^2\alpha+V)}{(\lambda R)^2}\Gamma\left(\frac{\lambda R(\xi-\Sigma_N^{(*)})}{8T_2(\lambda^2\pi K^2\alpha+V)}\right)\right\},$$

and $0 < \gamma \leq \gamma\left(\frac{\lambda R\xi'}{8T_2(\lambda^2\pi K^2\alpha+V)}\right) < 1.3$ with

$$\gamma(x) := \frac{\ln\left((x+1)\ln(x+1)-x\right)}{\ln x}.$$

The above theorem provides another upper bound of the risk bound $\sup_{f\in\mathcal{F}}\left|\mathrm{E}_N f - \mathrm{E}f\right|$ in the large-deviation case, where 1.719 is the numerical solution to the equation $\gamma(x) = 0$. Compared to the classical result (19), there is a discrepancy quantity $\Sigma_N^{(*)}$ that also appears in the Bernstein-type result (20). Interestingly, in the large-deviation case, the risk bound (14) can provide a faster rate $o\left(\frac{1}{N^{1/1.3}}\right)$ of convergence than the rate $O\left(\frac{1}{N^{1/2}}\right)$ of the classical result (19) and the Bernstein-type result (20). Note that the rate $o\left(\frac{1}{N^{1/1.3}}\right)$ will not hold if the large-deviation case is not valid (that is, $0 < \frac{\lambda R\xi'}{8T_2(\lambda^2\pi K^2\alpha+V)} \leq 1.719$), while the Bernstein-type result (20) for the learning process performs well and provides the rate $O\left(\frac{1}{N^{1/2}}\right)$ regardless of whether the large-deviation case is valid.

7. Conclusion

In this paper, we study the risk bounds of the learning process for time-dependent samples drawn from a Lévy process. We first provide the deviation inequalities and the symmetrization inequality of the learning process, respectively. We then use the resulted deviation inequalities and symmetrization inequality to derive the risk bounds based on the covering number.

By using the risk bound shown in Theorem 12, we analyze the asymptotic convergence and the rate of convergence of the learning process for Lévy process. We point out that the asymptotic convergence of such learning process is affected by two factors: the complexities of the function class $\mathcal{F}$ measured by the covering number and the quantity $\Sigma_N^{(*)}$. This is partially in accordance with the classical result on the asymptotic convergence of the learning process for i.i.d. samples (see Mendelson, 2003). Due to the quantity $\Sigma_N^{(*)}$, the uniform convergence of the learning process for Lévy process may not be valid. We also show that the rate of convergence of the learning process is up to $O(1/\sqrt{N})$, which matches with the the classical result under the sample-i.i.d. assumption. Furthermore, we adopt a new method to obtain another alternative expression of the risk bound (14) and then find that the rate of convergence of the learning process can reach $o(1/N^{\frac{1}{1.3}})$ in the large-deviation case. Note that as stated in Sections 3 & 5, the faster rate of convergence is actually provided by the specific deviation inequality (17) which is of Bennett-type (that is, its expression is similar to that of Bennett's inequality), while the classical result (19) is derived from Hoeffding's inequality (see Mendelson, 2003).

In our future work, we will attempt to study risk bounds for other stochastic processes via some specific concentration or deviation inequalities, for example, stochastic processes with exchangeable increments that are a well-known generalization of stochastic processes with independent increments (Kallenberg, 1973; Kallenberg et al., 1975). Then, we will develop the risk bounds of the learning process for Lévy process by using other complexity measures, for example, the Rademacher complexity and the fat-shattering dimension.

Acknowledgments

We are grateful to the anonymous reviewers and the editors for their valuable comments and suggestions. This project was supported by Australian Research Council Discovery Project with number ARC DP-120103730.

Appendix A. Proof of Theorem 8

Proof of Theorem 8. Let $\widehat{f}$ and $\widehat{t}$ be the function and the time achieving the supremum

$$\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \left|\mathrm{E}_t f - \mathrm{E}_N f\right|$$

with respect to $\mathbf{Z}_1^N$, respectively. According to Definition (D1), we have

$$\begin{aligned}\left|\mathrm{E}_{\widehat{t}}\widehat{f} - \mathrm{E}_N\widehat{f}\right| =& \left|\mathrm{E}_{\widehat{t}}\widehat{f} - \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f} + \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f} - \mathrm{E}_N\widehat{f}\right| \\ \leq& \Sigma_N^{(*)} + \left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f} - \mathrm{E}_N\widehat{f}\right|,\end{aligned}$$

which can lead to for any $\xi > \Sigma_N^{(*)}$ with $\xi' = \xi - \Sigma_N^{(*)}$,

$$\Pr\left\{\left|\mathrm{E}_{\widehat{t}}\widehat{f} - \mathrm{E}_N\widehat{f}\right| > \xi\right\} \leq \Pr\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f} - \mathrm{E}_N\widehat{f}\right| > \xi'\right\}.$$

According to the triangle inequality, we have

$$\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|-\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}'_N\widehat{f}\right|\leq\left|\mathrm{E}'_N\widehat{f}-\mathrm{E}_N\widehat{f}\right|. \tag{21}$$

Let $\mathcal{A}$ stand for an event and denote the indicator function of the event $\mathcal{A}$ as

$$\mathbf{1}_{\mathcal{A}}=\begin{cases}1, & \text{if } \mathcal{A} \text{ occurs;}\\ 0, & \text{otherwise.}\end{cases}$$

By denoting $\wedge$ as the conjunction of two events, it is followed from (21) that

$$\begin{aligned}&\left(\mathbf{1}_{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\xi'}\right)\left(\mathbf{1}_{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}'_N\widehat{f}\right|<\frac{\xi'}{2}}\right)\\ =&\mathbf{1}_{\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\xi'\right\}\wedge\left\{\left|\mathrm{E}'_N\widehat{f}-\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}\right|<\frac{\xi'}{2}\right\}}\\ \leq&\mathbf{1}_{\left|\mathrm{E}'_N\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\frac{\xi'}{2}}.\end{aligned}$$

Then, taking the expectation with respect to $\mathbf{Z}'^{N}_{1}$ gives

$$\begin{aligned}&\left(\mathbf{1}_{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\xi'}\right)\Pr'\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}'_N\widehat{f}\right|<\frac{\xi'}{2}\right\}\\ \leq&\Pr'\left\{\left|\mathrm{E}'_N\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\frac{\xi'}{2}\right\}.\end{aligned} \tag{22}$$

By Chebyshev's inequality, we have for any $\xi'>0$,

$$\begin{aligned}\Pr'\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}'_N\widehat{f}\right|\geq\frac{\xi'}{2}\right\}=&\Pr'\left\{\left|\sum_{n=1}^{N}(\mathrm{E}_{t_n}\widehat{f}-\widehat{f}(Z'_{t_n})\right|\geq\frac{N\xi'}{2}\right\}\\ \leq&\frac{4\mathrm{E}\left\{\sum_{n=1}^{N}\left(\mathrm{E}_{t_n}\widehat{f}-\widehat{f}(Z'_{t_n})\right)^2\right\}}{N^2(\xi')^2}\\ \leq&\frac{4N(b-a)^2}{N^2(\xi')^2}=\frac{4(b-a)^2}{N(\xi')^2}.\end{aligned} \tag{23}$$

Subsequently, according to (22) and (23), we have for any $\xi'>0$,

$$\Pr'\left\{\left|\mathrm{E}'_N\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\frac{\xi'}{2}\right\}\geq\left(\mathbf{1}_{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\xi'}\right)\left(1-\frac{4(b-a)^2}{N(\xi')^2}\right).$$

Let

$$\frac{4(b-a)^2}{N(\xi')^2}\leq\frac{1}{2}$$

and take the expectation with respect to $\mathbf{Z}_1^N$. Given any $\xi>\Sigma_N^{(*)}$, we then have for any $N\geq\frac{8(b-a)^2}{(\xi')^2}$ with $\xi'=\xi-\Sigma_N^{(*)}$,

$$\begin{aligned}\Pr\left\{\left|\mathrm{E}_{\widehat{t}}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\xi\right\}\leq&2\Pr\left\{\left|\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}_{t_n}\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\frac{\xi'}{2}\right\}\\ \leq&2\Pr\left\{\left|\mathrm{E}'_N\widehat{f}-\mathrm{E}_N\widehat{f}\right|>\frac{\xi'}{2}\right\}.\end{aligned}$$

This completes the proof. ■

Appendix B. Proofs of Theorems 11 & 12

Proof of Theorem 11. Consider $\{\varepsilon_n\}_{n=1}^N$ as independent Rademacher random variables, that is, independent $\{-1,1\}$-valued random variables with equal probability of taking either value. Given $\{\varepsilon_n\}_{n=1}^N$ and $\mathbf{Z}_1^{2N}$, denote

$$\overrightarrow{\varepsilon} := (\varepsilon_1, \cdots, \varepsilon_N, -\varepsilon_1, \cdots, -\varepsilon_N)^T, \tag{24}$$

and for any $f \in \mathcal{F}$,

$$\overrightarrow{f}(\mathbf{Z}_1^{2N}) := \big(f(Z'_{t_1}), \cdots, f(Z'_{t_N}), f(Z_{t_1}), \cdots, f(Z_{t_N})\big)^T. \tag{25}$$

According to (6) and Theorem 8, given any $\xi > \Sigma_N^{(*)}$, we have for any $N \geq \frac{8(b-a)^2}{(\xi')^2}$ with $\xi' = \xi - \Sigma_N^{(*)}$,

$$\begin{aligned}
&\Pr\left\{\sup_{\substack{f\in\mathcal{F}\\ t\in[T_1,T_2]}} \big|\mathrm{E}_t f - \mathrm{E}_N f\big| > \xi\right\}\\
\leq& 2\Pr\left\{\sup_{f\in\mathcal{F}} \big|\mathrm{E}'_N f - \mathrm{E}_N f\big| > \frac{\xi'}{2}\right\} \qquad \text{(by Theorem 8)}\\
=& 2\Pr\left\{\sup_{f\in\mathcal{F}} \Big|\frac{1}{N}\sum_{n=1}^N \big(f(Z'_{t_n}) - f(Z_{t_n})\big)\Big| > \frac{\xi'}{2}\right\}\\
=& 2\Pr\left\{\sup_{f\in\mathcal{F}} \Big|\frac{1}{N}\sum_{n=1}^N \varepsilon_n\big(f(Z'_{t_n}) - f(Z_{t_n})\big)\Big| > \frac{\xi'}{2}\right\} \qquad \text{(since } Z'_{t_n} \text{ and } Z_{t_n} \text{ are i.i.d.)}\\
=& 2\Pr\left\{\sup_{f\in\mathcal{F}} \Big|\frac{1}{2N}\big\langle\overrightarrow{\varepsilon}, \overrightarrow{f}(\mathbf{Z}_1^{2N})\big\rangle\Big| > \frac{\xi'}{4}\right\}. \qquad \text{(by (24) and (25))}
\end{aligned} \tag{26}$$

Fix a realization of $\mathbf{Z}_1^{2N}$ and let Λ be a $\xi'/8$-radius cover of $\mathcal{F}$ with respect to the $\ell_1(\mathbf{Z}_1^{2N})$ norm. Since $\mathcal{F}$ is composed of the λ-Lipschitz functions with the range $[a,b]$, we assume that the same holds for any $h \in \Lambda$. If $\widehat{f}$ is the function that achieves $\sup_{f\in\mathcal{F}} \frac{1}{2N}\big|\big\langle\overrightarrow{\varepsilon}, \overrightarrow{f}(\mathbf{Z}_1^{2N})\big\rangle\big| > \frac{\xi'}{4}$, there must be an $\widehat{h} \in \Lambda$ that satisfies that

$$\frac{1}{2N}\sum_{n=1}^N \Big(|\widehat{f}(Z'_{t_n}) - \widehat{h}(Z'_{t_n})| + |\widehat{f}(Z_{t_n}) - \widehat{h}(Z_{t_n})|\Big) < \frac{\xi'}{8},$$

and meanwhile,

$$\sup_{h\in\Lambda} \frac{1}{2N}\big|\big\langle\overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N})\big\rangle\big| > \frac{\xi'}{8}.$$

Therefore, for the realization of $\mathbf{Z}_1^{2N}$, we arrive at

$$\Pr\left\{\sup_{f\in\mathcal{F}} \Big|\frac{1}{2N}\big\langle\overrightarrow{\varepsilon}, \overrightarrow{f}(\mathbf{Z}_1^{2N})\big\rangle\Big| > \frac{\xi'}{4}\right\} \leq \Pr\left\{\sup_{h\in\Lambda} \Big|\frac{1}{2N}\big\langle\overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N})\big\rangle\Big| > \frac{\xi'}{8}\right\}. \tag{27}$$

Moreover, we denote the event

$$A := \left\{ \Pr\left\{ \sup_{h\in\Lambda} \left| \frac{1}{2N} \langle \overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N}) \rangle \right| > \frac{\xi'}{8} \right\} \right\},$$

and let $\mathbf{1}_A$ be the characteristic function of the event A. By Fubini's Theorem, we have

$$\Pr\{A\} = \mathrm{E}\left\{ \mathrm{E}_{\overrightarrow{\varepsilon}}\{\mathbf{1}_A\} \,\middle|\, \mathbf{Z}_1^{2N} \right\} = \mathrm{E}\left\{ \Pr\left\{ \sup_{h\in\Lambda} \left| \frac{1}{2N} \langle \overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N}) \rangle \right| > \frac{\xi'}{8} \right\} \,\middle|\, \mathbf{Z}_1^{2N} \right\}. \tag{28}$$

Fix a realization of $\mathbf{Z}_1^{2N}$ again. According to (24), (25) and Theorem 6, we have

$$\begin{aligned}
&\Pr\left\{ \sup_{h\in\Lambda} \left| \frac{1}{2N} \langle \overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N}) \rangle \right| > \frac{\xi'}{8} \right\} \\
\leq & |\Lambda| \max_{h\in\Lambda} \Pr\left\{ \left| \frac{1}{2N} \langle \overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N}) \rangle \right| > \frac{\xi'}{8} \right\} \\
= & \mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \max_{h\in\Lambda} \Pr\left\{ \left| \mathrm{E'}_N h - \mathrm{E}_N h \right| > \frac{\xi'}{4} \right\} \\
\leq & \mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \max_{h\in\Lambda} \Pr\left\{ \left| \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}_{t_n} h - \mathrm{E'}_N h \right| + \left| \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}_{t_n} h - \mathrm{E}_N h \right| > \frac{\xi'}{4} \right\} \\
\leq & 2\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \max_{h\in\Lambda} \Pr\left\{ \left| \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}_{t_n} h - \mathrm{E}_N h \right| > \frac{\xi'}{8} \right\} \\
\leq & 4\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \exp\left\{ -\int_0^{\frac{N\xi'}{8}} \varphi^{-1}(s) ds \right\}.
\end{aligned} \tag{29}$$

The combination of (26), (27), (28) and (29) leads to the result (13). This completes the proof. ■

In the similar way, we can also prove Theorem 12.

Proof of Theorem 12. Similarly, by (11), we have

$$\begin{aligned}
&\Pr\left\{ \sup_{h\in\Lambda} \left| \frac{1}{2N} \langle \overrightarrow{\varepsilon}, \overrightarrow{h}(\mathbf{Z}_1^{2N}) \rangle \right| > \frac{\xi'}{8} \right\} \\
\leq & 2\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \max_{h\in\Lambda} \Pr\left\{ \left| \frac{1}{N} \sum_{n=1}^{N} \mathrm{E}_{t_n} h - \mathrm{E}_N h \right| > \frac{\xi'}{8} \right\} \\
\leq & 4\mathcal{N}\left(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\right) \exp\left\{ \frac{NT_1(\lambda^2\pi K^2\alpha + V)}{(\lambda R)^2} \Gamma\left(\frac{\lambda R(\xi - \Sigma_N^{(*)})}{8T_2(\lambda^2\pi K^2\alpha + V)} \right) \right\}.
\end{aligned} \tag{30}$$

Then, the combination of (26), (27), (28) and (30) can lead to the result (14). This completes the proof. ■

B.1 Proof of Theorem 15

Proof of Theorem 15. Given any $x > 1$, consider the following equation with respect to $\gamma > 0$

$$x - (x+1)\ln(x+1) = -x^{\gamma}, \tag{31}$$

and denote its solution as

$$\gamma(x) := \frac{\ln\big((x+1)\ln(x+1)-x\big)}{\ln(x)}. \tag{32}$$

It is evident that $\gamma(x)$ is a continuously differentiable function with respect to $x > 1$ and there is only one solution to the equation $\gamma(x) = 0$. Its numerical solution is $\bar{x} \approx 1.719$ and $\gamma(x) > 0$ holds for all $x > \bar{x} \approx 1.719$. Then, given any $x > 1.719$, we have for any $0 < \widetilde{\gamma} \le \gamma(x)$,

$$x - (x+1)\ln(x+1) \le -x^{\widetilde{\gamma}}. \tag{33}$$

By combining Theorem 12, (31), (32) and (33), we can straightforwardly show an upper bound of the risk bound $\sup_{f\in\mathcal{F}} \big|\mathrm{E}_N f - \mathrm{E}f\big|$ in the large-deviation case: letting

$$\varepsilon := 8\mathrm{E}\mathcal{N}\big(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\big) \exp\left\{ \frac{NT_1(\lambda^2\pi K^2\alpha + V)}{(\lambda R)^2} \Gamma\left(\frac{\lambda R(\xi - \Sigma_N^{(*)})}{8T_2(\lambda^2\pi K^2\alpha + V)} \right) \right\}.$$

and with probability at least $1-\varepsilon$,

$$\begin{aligned} \sup_{f\in\mathcal{F}} \big|\mathrm{E}_N f - \mathrm{E}f\big| \le{}& 2\Sigma_N^{(*)} + \frac{27.504T_2(\lambda^2\pi K^2\alpha + V)}{\lambda R} + \frac{16T_2(\lambda^2\pi K^2\alpha + V)}{\lambda R} \\ & \times \left(\frac{(\lambda R)^2 \left(\ln \mathrm{E}\left\{\mathcal{N}\big(\mathcal{F}, \xi'/8, \ell_1(\mathbf{Z}_1^{2N})\big)\right\} - \ln(\varepsilon/8)\right)}{NT_1(\lambda^2\pi K^2\alpha + V)} \right)^{\frac{1}{\gamma}}, \end{aligned}$$

where $0 < \gamma \le \gamma\left(\frac{\lambda R\xi'}{8T_2(\lambda^2\pi K^2\alpha+V)}\right)$ with $\frac{\lambda R\xi'}{8T_2(\lambda^2\pi K^2\alpha+V)} > 1.719$. Thus, we only need to find the upper bound of the function $\gamma(x)$ when $x > 1.719$.

According to (32), for any $x > 1.719$, we consider the derivative of $\gamma(x)$

$$\gamma'(x) = \frac{\ln(x+1)}{\ln(x)\big((x+1)\ln(x+1)-x\big)} - \frac{\ln\big((x+1)\ln(x+1)-x\big)}{x(\ln x)^2}, \tag{34}$$

and draw the function curve of $\gamma'(x)$ in Figure 2.

Figure 2 shows that there is only one solution to the equation $\gamma'(x) = 0$ $(x > 1.719)$. Letting the solution be $\widehat{x}$, we then have $\gamma'(x) > 0$ $(1.719 < x < \widehat{x})$ and $\gamma'(x) < 0$ $(x > \widehat{x})$, that is, $\gamma(x)$ is monotonically decreasing when $x > \widehat{x}$. Meanwhile, by (34), there holds that

$$\lim_{x\to+\infty} \gamma'(x) = 0. \tag{35}$$

Furthermore, we study the second derivative of $\gamma''(x)$

$$\begin{aligned} \gamma''(x) = {}& \frac{\ln\big((x+1)\ln(x+1)-x\big)}{x^2(\ln x)^2} - \frac{1}{(x+1)(x-(x+1)\ln(x+1))\ln x} \\ & + \frac{2\ln\big((x+1)\ln(x+1)-x\big)}{x^2(\ln x)^3} - \frac{(\ln(x+1))^2}{(x-(x+1)\ln(x+1))^2\ln x} \\ & + \frac{2\ln(x+1)}{x(\ln x)^2(x-(x+1)\ln(x+1))}, \end{aligned} \tag{36}$$

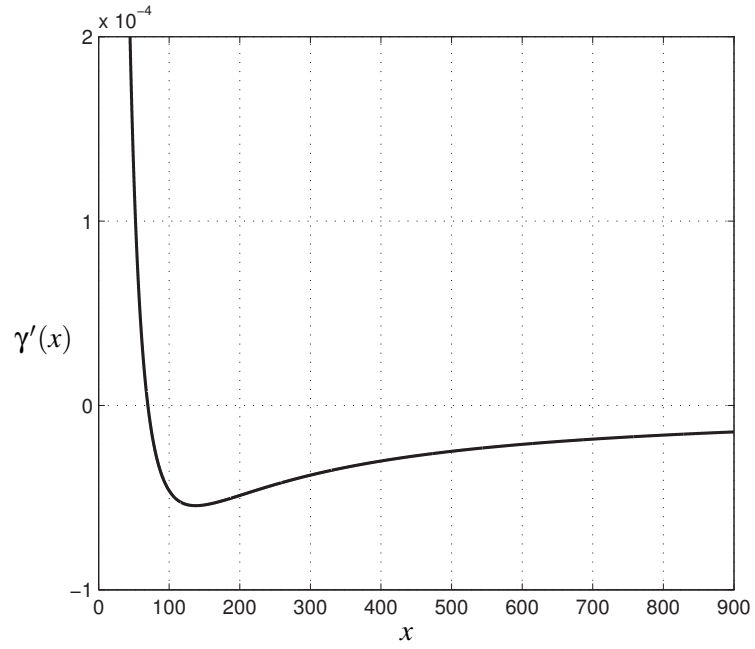


Figure 2: The Function Curve of $\gamma'(x)$

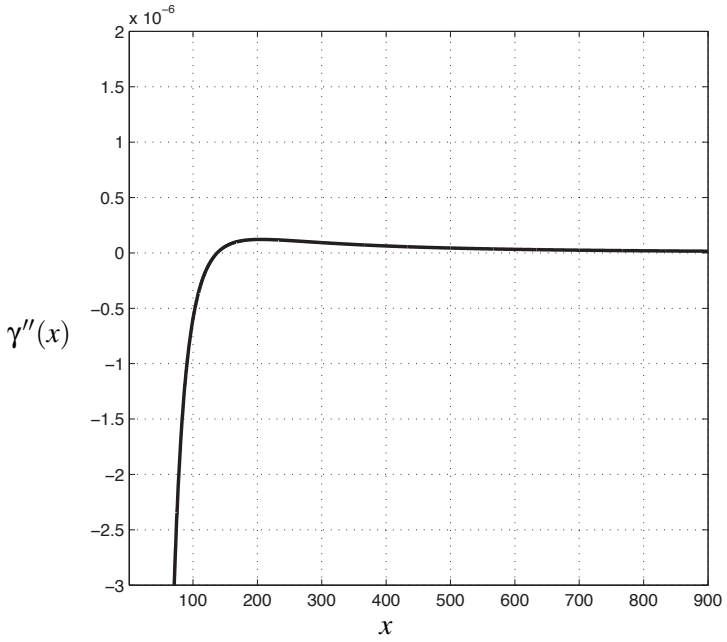


Figure 3: The Function Curve of $\gamma''(x)$

and draw the function curve of $\gamma''(x)$ in Figure 3. This figure shows that there is a solution to the equation $\gamma''(x) = 0$ and its value approximately equals to 137.67. Moreover, according to (36), we arrive at

$$\lim_{x \to +\infty} \gamma''(x) = 0. \tag{37}$$

Therefore, by combining (34), (35), (36) and (37), we obtain that $\gamma(x)$ has only one global maximum point when $x > 1.719$ and thus the solution $\widehat{x}$ to the equation $\gamma'(x) = 0$ also achieves

$$\widehat{x} = \arg \max_{x > 1.719} \gamma(x).$$

Our further numerical experiment shows that the value of $\widehat{x}$ approximately equals to 69.85 and the maximum of $\gamma(x)$ $(x > 1.719)$ is not larger than 1.3 (see Figure 4). This completes the proof. ■

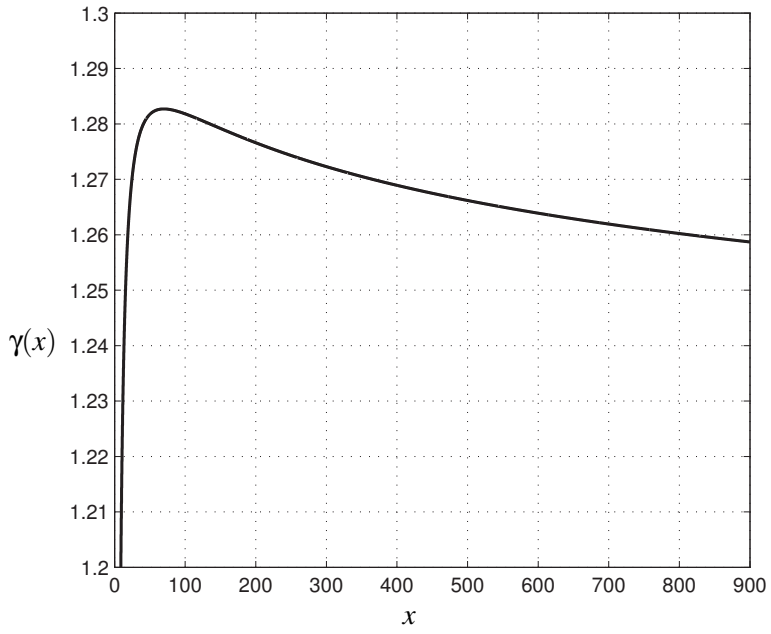


Figure 4: The Function Curve of $\gamma(x)$

References

D. Applebaum. *Lévy Processes and Stochastic Calculus*. Cambridge: Cambridge Press, 2004a.

D. Applebaum. Lévy processes-from probability to finance and quantum groups. *Notices of the American Mathematical Society*, 51:1336–1347, 2004b.

O.E. Barndorff-Nielsen, T. Mikosch, and S.I. Resnick. *Lévy Processes: Theory and Applications*. Birkhauser, 2001.

P.L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

G. Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962.

M. Biguesh and A.B. Gershman. Training-based mimo channel estimation: a study of estimator tradeoffs and optimal training signals. *IEEE Transactions on Signal Processing*, 54(3):884–893, 2006.

A. Bose, A. Dasgupta, and H. Rubin. A contemporary review and bibliography of infinitely divisible distributions and processes. *The Indian Journal of Statistics, Series A*, 64:763–819, 2002.

O. Bousquet. A bennett concentration inequality and its application to suprema of empirical processes. *Comptes Rendus Mathematique*, 334(6):495–500, 2002.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. *Advanced Lectures on Machine Learning*, pages 169–207, 2004.

N. Cesa-Bianchi and C. Gentile. Improved risk tail bounds for on-line algorithms. *IEEE Transactions on Information Theory*, 54(1):386–390, 2008.

R. Cont and P. Tankov. Retrieving lévy processes from option prices: Regularization of an ill-posed inverse problem. *SIAM Journal on Control and Optimization*, 45(1):1–25, 2006.

T.E. Duncan. Mutual information for stochastic signals and lévy processes. *IEEE Transactions on Information Theory*, 56(1):18–24, 2009.

J.E. Figueroa-López and C. Houdré. Risk bounds for the non-parametric estimation of lévy processes. *Lecture Notes-Monograph Series*, pages 96–116, 2006.

C. Houdré. Remarks on deviation inequalities for functions of infinitely divisible random vectors. *Annals of probability*, pages 1223–1237, 2002.

C. Houdré and P. Marchal. Median, concentration and fluctuations for lévy processes. *Stochastic Processes and their Applications*, 118(5):852–863, 2008.

C. Houdré, V. Pérez-Abreu, and D. Surgailis. Interpolation, correlation identities, and inequalities for infinitely divisible variables. *Journal of Fourier Analysis and Applications*, 4(6):651–668, 1998.

M. Jacobsen. *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes*. Birkhäuser Boston, 2005.

W. Jiang. On the uniform deviations of general empirical risks with unboundedness, dependence, and high dimensionality. *Journal of Machine Learning Research*, 10:977–996, 2009.

O. Kallenberg. Canonical representations and convergence criteria for processes with interchangeable increments. *Probability Theory and Related Fields*, 27(1):23–36, 1973.

O. Kallenberg et al. On symmetrically distributed random measures. *Trans. Amer. Math. Soc*, 202: 105–121, 1975.

K. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1): 307–319, 2003.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

A. Kyprianou. *Introductory Lectures on Fluctuations of Lévy Processes with Applications (Universitext)*. Springer, 2006.

D.J. Love, R.W. Heath, V.K.N. Lau, D. Gesbert, B.D. Rao, and M. Andrews. An overview of limited feedback in wireless communication systems. *IEEE Journal on Selected Areas Communications*, 26(8):1341–1365, 2008.

S. Mendelson. Rademacher averages and phase transitions in glivenko-cantelli classes. *IEEE Transactions on Information Theory*, 48(1):251–263, 2002.

S. Mendelson. A few notes on statistical learning theory. *Advanced Lectures on Machine Learning*, pages 1–40, 2003.

S. Mendelson. Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory*, 54(8):3797–3803, 2008.

M. Mohri and A. Rostamizadeh. Stability bounds for stationary ϕ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11:798–814, 2010.

S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using support vector machines. In *IEEE Workshop on Neural Networks for Signal Processing VII*, pages 511–520, 1997.

A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

A.B. Nobel and A. Dembo. A note on uniform laws of averages for dependent processes. *Statistics & Probability Letters*, 17(3):169–172, 1993.

K.S. Pedersen, R. Duits, and M. Nielsen. On α kernels, lévy processes, and natural image statistics. In Kimmel, Sochen, and Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, pages 468–479, 2005.

S.T. Rachev. *Probability Metrics and the Stability of Stochastic Models*. New York: Wiley, 1991.

M. Reid and B. Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12:731–817, 2011.

M. Sanchez-Fernandez, M. de Prado-Cumplido, J. Arenas-Garcia, and F. Perez-Cruz. Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems. *IEEE Transactions on Signal Processing*, 52(8):2298–2307, 2004.

K. Sato. *Lévy Processes and Infinite Divisible Distributions (Cambridge Studies in Advanced Mathematics)*. USA: Cambridge University Press, 2004.

B.K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G.R.G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

A. Sutivong, M. Chiang, T.M. Cover, and Y.H. Kim. Channel capacity and state estimation for state-dependent gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1486–1495, 2005.

A.M. Tulino, A. Lozano, and S. Verdú. Impact of antenna correlation on the capacity of multiantenna channels. *IEEE Transactions on Information Theory*, 51(7):2491–2509, 2005.

A. Van der Vaart and J. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, 1996.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *Annals of Probability*, 22(1):94–116, 1994.

C. Zhang and D. Tao. Risk bounds for lévy processes in the pac-learning framework. *Journal of Machine Learning Research-Proceedings Track*, 9:948–955, 2010.

C. Zhang and D. Tao. Generalization bound for infinitely divisible empirical process. *J. Mach. Learn. Res.-Proc. Track*, 15:864–872, 2011a.

C. Zhang and D. Tao. Risk bounds for infinitely divisible distribution. In *The 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, 2011b.

V.M. Zolotarev. Probability metrics. *Theory of Probability and its Application*, 28(1):278–302, 1984.

Journal of Machine Learning Research 14 (2013) 377-397 Submitted 2/12; Revised 8/12; Published 2/13

Learning Theory Approach to Minimum Error Entropy Criterion

Ting Hu TINGHU@WHU.EDU.CN
School of Mathematics and Statistics
Wuhan University
Wuhan 430072, China

Jun Fan JUNFAN2@STUDENT.CITYU.EDU.HK
Department of Mathematics
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong, China

Qiang Wu QWU@MTSU.EDU
Department of Mathematical Sciences
Middle Tennessee State University
Murfreesboro, TN 37132, USA

Ding-Xuan Zhou MAZHOU@CITYU.EDU.HK
Department of Mathematics
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong, China

Editor: Gabor Lugosi

Abstract

We consider the minimum error entropy (MEE) criterion and an empirical risk minimization learning algorithm when an approximation of Rényi's entropy (of order 2) by Parzen windowing is minimized. This learning algorithm involves a Parzen windowing scaling parameter. We present a learning theory approach for this MEE algorithm in a regression setting when the scaling parameter is large. Consistency and explicit convergence rates are provided in terms of the approximation ability and capacity of the involved hypothesis space. Novel analysis is carried out for the generalization error associated with Rényi's entropy and a Parzen windowing function, to overcome technical difficulties arising from the essential differences between the classical least squares problems and the MEE setting. An involved symmetrized least squares error is introduced and analyzed, which is related to some ranking algorithms.

Keywords: minimum error entropy, learning theory, Rényi's entropy, empirical risk minimization, approximation error

1. Introduction

Information theoretical learning is inspired by introducing information theory into a machine learning paradigm. Within this framework algorithms have been developed for several learning tasks, including regression, classification, and unsupervised learning. It attracts more and more attention because of its successful applications in signal processing, system engineering, and data mining.

A systematic treatment and recent development of this area can be found in Principe (2010) and references therein.

Minimum error entropy (MEE) is a principle of information theoretical learning and provides a family of supervised learning algorithms. It was introduced for adaptive system training in Erdogmus and Principe (2002) and has been applied to blind source separation, maximally informative subspace projections, clustering, feature selection, blind deconvolution, and some other topics (Erdogmus and Principe, 2003; Principe, 2010; Silva et al., 2010). The idea of MEE is to extract from data as much information as possible about the data generating systems by minimizing error entropies in various ways. In information theory, entropies are used to measure average information quantitatively. For a random variable E with probability density function p_E, Shannon's entropy of E is defined as

$$H_S(E) = -\mathbb{E}[\log p_E] = -\int p_E(e)\log p_E(e)de$$

while Rényi's entropy of order α ($\alpha > 0$ but $\alpha \neq 1$) is defined as

$$H_{R,\alpha}(E) = \frac{1}{1-\alpha}\log \mathbb{E}[p_E^{\alpha-1}] = \frac{1}{1-\alpha}\log\left(\int (p_E(e))^{\alpha} de\right)$$

satisfying $\lim_{\alpha\to 1} H_{R,\alpha}(E) = H_S(E)$. In supervised learning our target is to predict the response variable Y from the explanatory variable X. Then the random variable E becomes the error variable $E = Y - f(X)$ when a predictor $f(X)$ is used and the MEE principle aims at searching for a predictor $f(X)$ that contains the most information of the response variable by minimizing information entropies of the error variable $E = Y - f(X)$. This principle is a substitution of the classical least squares method when the noise is non-Gaussian. Note that $\mathbb{E}[Y - f(X)]^2 = \int e^2 p_E(e)de$. The least squares method minimizes the variance of the error variable E and is perfect to deal with problems involving Gaussian noise (such as some from linear signal processing). But it only puts the first two moments into consideration, and does not work very well for problems involving heavy tailed non-Gaussian noise. For such problems, MEE might still perform very well in principle since moments of all orders of the error variable are taken into account by entropies. Here we only consider Rényi's entropy of order $\alpha = 2$: $H_R(E) = H_{R,2}(E) = -\log\int (p_E(e))^2 de$. Our analysis does not apply to Rényi's entropy of order $\alpha \neq 2$.

In most real applications, neither the explanatory variable X nor the response variable Y is explicitly known. Instead, in supervised learning, a sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$ is available which reflects the distribution of the explanatory variable X and the functional relation between X and the response variable Y. With this sample, information entropies of the error variable $E = Y - f(X)$ can be approximated by estimating its probability density function p_E by Parzen (1962) windowing $\widehat{p}_E(e) = \frac{1}{mh}\sum_{i=1}^m G(\frac{(e-e_i)^2}{2h^2})$, where $e_i = y_i - f(x_i)$, $h > 0$ is an MEE scaling parameter, and G is a windowing function. A typical choice for the windowing function $G(t) = \exp\{-t\}$ corresponds to Gaussian windowing. Then approximations of Shannon's entropy and Rényi's entropy of order 2 are given by their empirical versions $-\frac{1}{m}\sum_{i=1}^m \log \widehat{p}_E(e_i)$ and $-\log(\frac{1}{m}\sum_{i=1}^m \widehat{p}_E(e_i))$ as

$$\widehat{H_S} = -\frac{1}{m}\sum_{i=1}^m \log\left[\frac{1}{mh}\sum_{j=1}^m G\left(\frac{(e_i - e_j)^2}{2h^2}\right)\right]$$

and

$$\widehat{H_R} = -\log\frac{1}{m^2 h}\sum_{i=1}^m\sum_{j=1}^m G\left(\frac{(e_i - e_j)^2}{2h^2}\right),$$

respectively. The empirical MEE is implemented by minimizing these computable quantities.

Though the MEE principle has been proposed for a decade and MEE algorithms have been shown to be effective in various applications, its theoretical foundation for mathematical error analysis is not well understood yet. There is even no consistency result in the literature. It has been observed in applications that the scaling parameter h should be large enough for MEE algorithms to work well before smaller values are tuned. However, it is well known that the convergence of Parzen windowing requires h to converge to 0. We believe this contradiction imposes difficulty for rigorous mathematical analysis of MEE algorithms. Another technical barrier for mathematical analysis of MEE algorithms for regression is the possibility that the regression function may not be a minimizer of the associated generalization error, as described in detail in Section 3 below. The main contribution of this paper is a consistency result for an MEE algorithm for regression. It does require h to be large and explains the effectiveness of the MEE principle in applications.

In the sequel of this paper, we consider an MEE learning algorithm that minimizes the empirical Rényi's entropy $\widehat{H_R}$ and focus on the regression problem. We will take a learning theory approach and analyze this algorithm in an *empirical risk minimization* (ERM) setting. Assume ρ is a probability measure on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a separable metric space (input space for learning) and $\mathcal{Y} = \mathbb{R}$ (output space). Let ρ_X be its marginal distribution on $\mathcal{X}$ (for the explanatory variable X) and $\rho(\cdot|x)$ be the conditional distribution of Y for given $X = x$. The sample $\mathbf{z}$ is assumed to be drawn from ρ independently and identically distributed. The aim of the regression problem is to predict the conditional mean of Y for given X by learning the regression function defined by

$$f_\rho(x) = \mathbb{E}(Y|X = x) = \int_{\mathcal{X}} y d\rho(y|x), \qquad x \in \mathcal{X}.$$

The minimization of empirical Rényi's entropy cannot be done over all possible measurable functions which would lead to overfitting. A suitable hypothesis space should be chosen appropriately in the ERM setting. The ERM framework for MEE learning is defined as follows. Recall $e_i = y_i - f(x_i)$.

Definition 1 *Let G be a continuous function defined on $[0,\infty)$ and $h > 0$. Let $\mathcal{H}$ be a compact subset of $C(\mathcal{X})$. Then the MEE learning algorithm associated with $\mathcal{H}$ is defined by*

$$f_{\mathbf{z}} = \arg\min_{f \in \mathcal{H}} \left\{ -\log \frac{1}{m^2 h} \sum_{i=1}^{m} \sum_{j=1}^{m} G\left(\frac{[(y_i - f(x_i)) - (y_j - f(x_j))]^2}{2h^2} \right) \right\}. \tag{1}$$

The set $\mathcal{H}$ is called the hypothesis space for learning. Its compactness ensures the existence of a minimizer $f_{\mathbf{z}}$. Computational methods for solving optimization problem (1) and its applications in signal processing have been described in a vast MEE literature (Principe, 2010; Erdogmus and Principe, 2002, 2003; Silva et al., 2010). For different purposes the MEE scaling parameter h may be chosen to be large or small. It has been observed empirically that the MEE criterion has nice convergence properties when the MEE scaling parameter h is large. The main purpose of this paper is to verify this observation in the ERM setting and show that $f_{\mathbf{z}}$ with a suitable constant adjustment approximates the regression function well with confidence. Note that the requirement of a constant adjustment is natural because any translate $f_{\mathbf{z}} + c$ of a solution $f_{\mathbf{z}}$ to (1) with a constant $c \in \mathbb{R}$ is another solution to (1). So our consistency result for MEE algorithm (1) will be stated in terms of the variance $\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)]$ of the error function $f_{\mathbf{z}} - f_\rho$. Here we use $\mathbf{var}$ to denote the variance of a random variable.

2. Main Results on Consistency and Convergence Rates

Throughout the paper, we assume $h \geq 1$ and that

$$\mathbb{E}[|Y|^q] < \infty \text{ for some } q > 2, \text{ and } f_\rho \in L^\infty_{\rho_X}. \quad \text{Denote } q^* = \min\{q-2, 2\}. \tag{2}$$

We also assume that the windowing function G satisfies

$$G \in C^2[0,\infty),\ G'_+(0) = -1, \text{ and } C_G := \sup_{t\in(0,\infty)} \left\{|(1+t)G'(t)| + |(1+t)G''(t)|\right\} < \infty. \tag{3}$$

The special example $G(t) = \exp\{-t\}$ for the Gaussian windowing satisfies (3).

Consistency analysis for regression algorithms is often carried out in the literature under a decay assumption for Y such as uniform boundedness and exponential decays. A recent study (Audibert and Catoni, 2011) was made under the assumption $\mathbb{E}[|Y|^4] < \infty$. Our assumption (2) is weaker since q may be arbitrarily close to 2. Note that (2) obviously holds when $|Y| \leq M$ almost surely for some constant $M > 0$, in which case we shall denote $q^* = 2$.

Our consistency result, to be proved in Section 5, asserts that when h and m are large enough, the error $\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)]$ of MEE algorithm (1) can be arbitrarily close to the approximation error (Smale and Zhou, 2003) of the hypothesis space $\mathcal{H}$ with respect to the regression function f_ρ.

Definition 2 *The approximation error of the pair $(\mathcal{H}, \rho)$ is defined by*

$$\mathcal{D}_{\mathcal{H}}(f_\rho) = \inf_{f\in\mathcal{H}} \mathbf{var}[f(X) - f_\rho(X)].$$

Theorem 3 *Under assumptions (2) and (3), for any $0 < \varepsilon \leq 1$ and $0 < \delta < 1$, there exist $h_{\varepsilon,\delta} \geq 1$ and $m_{\varepsilon,\delta}(h) \geq 1$ both depending on $\mathcal{H}, G, \rho, \varepsilon, \delta$ such that for $h \geq h_{\varepsilon,\delta}$ and $m \geq m_{\varepsilon,\delta}(h)$, with confidence $1-\delta$, we have*

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \leq \mathcal{D}_{\mathcal{H}}(f_\rho) + \varepsilon. \tag{4}$$

Our convergence rates will be stated in terms of the approximation error and the capacity of the hypothesis space $\mathcal{H}$ measured by covering numbers in this paper.

Definition 4 *For $\varepsilon > 0$, the covering number $\mathcal{N}(\mathcal{H}, \varepsilon)$ is defined to be the smallest integer $l \in \mathbb{N}$ such that there exist l disks in $C(\mathcal{X})$ with radius ε and centers in $\mathcal{H}$ covering the set $\mathcal{H}$. We shall assume that for some constants $p > 0$ and $A_p > 0$, there holds*

$$\log \mathcal{N}(\mathcal{H}, \varepsilon) \leq A_p \varepsilon^{-p}, \qquad \forall \varepsilon > 0. \tag{5}$$

The behavior (5) of the covering numbers is typical in learning theory. It is satisfied by balls of Sobolev spaces on $\mathcal{X} \subset \mathbb{R}^n$ and reproducing kernel Hilbert spaces associated with Sobolev smooth kernels. See Anthony and Bartlett (1999), Zhou (2002), Zhou (2003) and Yao (2010). We remark that empirical covering numbers might be used together with concentration inequalities to provide shaper error estimates. This is however beyond our scope and for simplicity we adopt the the covering number in $C(\mathcal{X})$ throughout this paper.

The following convergence rates for (1) with large h will be proved in Section 5.

Theorem 5 *Assume (2), (3) and covering number condition (5) for some $p > 0$. Then for any $0 < \eta \leq 1$ and $0 < \delta < 1$, with confidence $1-\delta$ we have*

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \leq \widetilde{C}_{\mathcal{H}} \eta^{(2-q)/2} \left(h^{-\min\{q-2,2\}} + h m^{-\frac{1}{1+p}} \right) \log \frac{2}{\delta} + (1+\eta) \mathcal{D}_{\mathcal{H}}(f_\rho). \tag{6}$$

If $|Y| \leq M$ almost surely for some $M > 0$, then with confidence $1-\delta$ we have

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \leq \frac{\widetilde{C}_{\mathcal{H}}}{\eta} \left(h^{-2} + m^{-\frac{1}{1+p}} \right) \log \frac{2}{\delta} + (1+\eta) \mathcal{D}_{\mathcal{H}}(f_\rho). \tag{7}$$

Here $\widetilde{C}_{\mathcal{H}}$ is a constant independent of m, δ, η or h (depending on $\mathcal{H}, G, \rho$ given explicitly in the proof).

Remark 6 *In Theorem 5, we use a parameter $\eta > 0$ in error bounds (6) and (7) to show that the bounds consist of two terms, one of which is essentially the approximation error $\mathcal{D}_{\mathcal{H}}(f_\rho)$ since η can be arbitrarily small. The reader can simply set $\eta = 1$ to get the main ideas of our analysis.*

If moment condition (2) with $q \geq 4$ is satisfied and $\eta = 1$, then by taking $h = m^{\frac{1}{3(1+p)}}$, (6) becomes

$$\mathbf{var}[(f_{\mathbf{z}}(X) - f_\rho(X)] \leq 2\widetilde{C}_{\mathcal{H}} \left(\frac{1}{m} \right)^{\frac{2}{3(1+p)}} \log \frac{2}{\delta} + 2\mathcal{D}_{\mathcal{H}}(f_\rho). \tag{8}$$

If $|Y| \leq M$ almost surely, then by taking $h = m^{\frac{1}{2(1+p)}}$ and $\eta = 1$, error bound (7) becomes

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \leq 2\widetilde{C}_{\mathcal{H}} m^{-\frac{1}{1+p}} \log \frac{2}{\delta} + 2\mathcal{D}_{\mathcal{H}}(f_\rho). \tag{9}$$

Remark 7 *When the index p in covering number condition (5) is small enough (the case when $\mathcal{H}$ is a finite ball of a reproducing kernel Hilbert space with a smooth kernel), we see that the power indices for the sample error terms of convergence rates (8) and (9) can be arbitrarily close to $2/3$ and 1, respectively. There is a gap in the rates between the case of (2) with large q and the uniform bounded case. This gap is caused by the Parzen windowing process for which our method does not lead to better estimates when $q > 4$. It would be interesting to know whether the gap can be narrowed.*

Note the result in Theorem 5 does not guarantee that $f_{\mathbf{z}}$ itself approximates f_ρ well when the bounds are small. Instead a constant adjustment is required. Theoretically the best constant is $\mathbb{E}[f_{\mathbf{z}}(X) - f_\rho(X)]$. In practice it is usually approximated by the sample mean $\frac{1}{m}\sum_{i=1}^m (f_{\mathbf{z}}(x_i) - y_i)$ in the case of uniformly bounded noise and the approximation can be easily handled. To deal with heavy tailed noise, we project the output values onto the closed interval $[-\sqrt{m}, \sqrt{m}]$ by the projection $\pi_{\sqrt{m}} : \mathbb{R} \to \mathbb{R}$ defined by

$$\pi_{\sqrt{m}}(y) = \begin{cases} y, & \text{if } y \in [-\sqrt{m}, \sqrt{m}], \\ \sqrt{m}, & \text{if } y > \sqrt{m}, \\ -\sqrt{m}, & \text{if } y < -\sqrt{m}, \end{cases}$$

and then approximate $\mathbb{E}[f_{\mathbf{z}}(X) - f_\rho(X)]$ by the computable quantity

$$\frac{1}{m} \sum_{i=1}^m \left[f_{\mathbf{z}}(x_i) - \pi_{\sqrt{m}}(y_i) \right]. \tag{10}$$

The following quantitative result, to be proved in Section 5, tells us that this is a good approximation.

Theorem 8 *Assume $\mathbb{E}[|Y|^2] < \infty$ and covering number condition (5) for some $p > 0$. Then for any $0 < \delta < 1$, with confidence $1-\delta$ we have*

$$\sup_{f\in\mathcal{H}} \left| \frac{1}{m}\sum_{i=1}^{m} \left[f(x_i) - \pi_{\sqrt{m}}(y_i)\right] - \mathbb{E}[f(X) - f_\rho(X)] \right| \le \widetilde{C}'_{\mathcal{H}} m^{-\frac{1}{2+p}} \log\frac{2}{\delta} \tag{11}$$

which implies in particular that

$$\left| \frac{1}{m}\sum_{i=1}^{m} \left[f_{\mathbf{z}}(x_i) - \pi_{\sqrt{m}}(y_i)\right] - \mathbb{E}[f_{\mathbf{z}}(X) - f_\rho(X)] \right| \le \widetilde{C}'_{\mathcal{H}} m^{-\frac{1}{2+p}} \log\frac{2}{\delta}, \tag{12}$$

where $\widetilde{C}'_{\mathcal{H}}$ is the constant given by

$$\widetilde{C}'_{\mathcal{H}} = 7\sup_{f\in\mathcal{H}} \|f\|_\infty + 4 + 7\sqrt{\mathbb{E}[|Y|^2]} + \mathbb{E}[|Y|^2] + A_p^{\frac{1}{2+p}}.$$

Replacing the mean $\mathbb{E}[f_{\mathbf{z}}(X) - f_\rho(X)]$ by the quantity (10), we define an estimator of f_ρ as

$$\widetilde{f}_{\mathbf{z}} = f_{\mathbf{z}} - \frac{1}{m}\sum_{i=1}^{m} \left[f_{\mathbf{z}}(x_i) - \pi_{\sqrt{m}}(y_i)\right].$$

Putting (12) and the bounds from Theorem 5 into the obvious error expression

$$\left\|\widetilde{f}_{\mathbf{z}} - f_\rho\right\|_{L^2_{\rho_X}} \le \left| \frac{1}{m}\sum_{i=1}^{m} \left[f_{\mathbf{z}}(x_i) - \pi_{\sqrt{m}}(y_i)\right] - \mathbb{E}[f_{\mathbf{z}}(X) - f_\rho(X)] \right| + \sqrt{\mathbf{var}[(f_{\mathbf{z}}(X) - f_\rho(X)]}, \tag{13}$$

we see that $\widetilde{f}_{\mathbf{z}}$ is a good estimator of f_ρ: the power index $\frac{1}{2+p}$ in (12) is greater than $\frac{1}{2(1+p)}$, the power index appearing in the last term of (13) when the variance term is bounded by (9), even in the uniformly bounded case.

To interpret our main results better we present a corollary and an example below.

If there is a constant c_ρ such that $f_\rho + c_\rho \in \mathcal{H}$, we have $\mathcal{D}_{\mathcal{H}}(f_\rho) = 0$. In this case, the choice $\eta = 1$ in Theorem 5 yields the following learning rate. Note that (2) implies $\mathbb{E}[|Y|^2] < \infty$.

Corollary 9 *Assume (5) with some $p > 0$ and $f_\rho + c_\rho \in \mathcal{H}$ for some constant $c_\rho \in \mathbb{R}$. Under conditions (2) and (3), by taking $h = m^{\frac{1}{(1+p)\min\{q-1,3\}}}$, we have with confidence $1-\delta$,*

$$\left\|\widetilde{f}_{\mathbf{z}} - f_\rho\right\|_{L^2_{\rho_X}} \le \left(\widetilde{C}'_{\mathcal{H}} + \sqrt{2\widetilde{C}_{\mathcal{H}}}\right) m^{-\frac{\min\{q-2,2\}}{2(1+p)\min\{q-1,3\}}} \log\frac{2}{\delta}.$$

If $|Y| \le M$ almost surely, then by taking $h = m^{\frac{1}{2(1+p)}}$, we have with confidence $1-\delta$,

$$\left\|\widetilde{f}_{\mathbf{z}} - f_\rho\right\|_{L^2_{\rho_X}} \le \left(\widetilde{C}'_{\mathcal{H}} + \sqrt{2\widetilde{C}_{\mathcal{H}}}\right) m^{-\frac{1}{2(1+p)}} \log\frac{2}{\delta}.$$

This corollary states that $\widetilde{f}_{\mathbf{z}}$ can approximate the regression function very well. Note, however, this happens when the hypothesis space is chosen appropriately and the parameter h tends to infinity.

A special example of the hypothesis space is a ball of a Sobolev space $H^s(\mathcal{X})$ with index $s > \frac{n}{2}$ on a domain $\mathcal{X} \subset \mathbb{R}^n$ which satisfies (5) with $p = \frac{n}{s}$. When s is large enough, the positive index $\frac{n}{s}$ can be arbitrarily small. Then the power exponent of the following convergence rate can be arbitrarily close to $\frac{1}{3}$ when $\mathbb{E}[|Y|^4] < \infty$, and $\frac{1}{2}$ when $|Y| \le M$ almost surely.

Example 1 *Let $\mathcal{X}$ be a bounded domain of $\mathbb{R}^n$ with Lipschitz boundary. Assume $f_\rho \in H^s(X)$ for some $s > \frac{n}{2}$ and take $\mathcal{H} = \{f \in H^s(X) : \|f\|_{H^s(X)} \le R\}$ with $R \ge \|f_\rho\|_{H^s(X)}$ and $R \ge 1$. If $\mathbb{E}[|Y|^4] < \infty$, then by taking $h = m^{\frac{1}{3(1+n/s)}}$, we have with confidence $1-\delta$,*

$$\left\| \widetilde{f}_{\mathbf{z}} - f_\rho \right\|_{L^2_{\rho_X}} \le C_{s,n,\rho} R^{\frac{n}{2(s+n)}} m^{-\frac{1}{3(1+n/s)}} \log \frac{2}{\delta}.$$

If $|Y| \le M$ almost surely, then by taking $h = m^{\frac{1}{2(1+n/s)}}$, with confidence $1-\delta$,

$$\left\| \widetilde{f}_{\mathbf{z}} - f_\rho \right\|_{L^2_{\rho_X}} \le C_{s,n,\rho} R^{\frac{n}{2(s+n)}} m^{-\frac{1}{2+2n/s}} \log \frac{2}{\delta}.$$

Here the constant $C_{s,n,\rho}$ is independent of R.

Compared to the analysis of least squares methods, our consistency results for the MEE algorithm require a weaker condition by allowing heavy tailed noise, while the convergence rates are comparable but slightly worse than the optimal one $O(m^{-\frac{1}{2+n/s}})$. Further investigation of error analysis for the MEE algorithm is required to achieve the optimal rate, which is beyond the scope of this paper.

3. Technical Difficulties in MEE and Novelties

The MEE algorithm (1) involving sample pairs like quadratic forms is different from most classical ERM learning algorithms (Vapnik, 1998; Anthony and Bartlett, 1999) constructed by sums of independent random variables. But as done for some ranking algorithms (Agarwal and Niyogi, 2009; Clemencon et al., 2005), one can still follow the same line to define a functional called generalization error or *information error* (related to information potential defined on page 88 of Principe, 2010) associated with the windowing function G over the space of measurable functions on $\mathcal{X}$ as

$$\mathcal{E}^{(h)}(f) = \int_{\mathcal{Z}} \int_{\mathcal{Z}} -h^2 G\left(\frac{[(y-f(x)) - (y'-f(x'))]^2}{2h^2} \right) d\rho(x,y) d\rho(x',y').$$

An essential barrier for our consistency analysis is an observation made by numerical simulations (Erdogmus and Principe, 2003; Silva et al., 2010) and verified mathematically for Shannon's entropy in Chen and Principe (2012) that the regression function f_ρ may not be a minimizer of $\mathcal{E}^{(h)}$. It is totally different from the classical least squares generalization error $\mathcal{E}^{ls}(f) = \int_{\mathcal{Z}} (f(x)-y)^2 d\rho$ which satisfies a nice identity $\mathcal{E}^{ls}(f) - \mathcal{E}^{ls}(f_\rho) = \|f - f_\rho\|^2_{L^2_{\rho_X}} \ge 0$. This barrier leads to three technical difficulties in our error analysis which will be overcome by our novel approaches making full use of the special feature that the MEE scaling parameter h is large in this paper.

3.1 Approximation of Information Error

The first technical difficulty we meet in our mathematical analysis for MEE algorithm (1) is the varying form depending on the windowing function G. Our novel approach here is an approximation of the information error in terms of the variance $\mathbf{var}[f(X) - f_\rho(X)]$ when h is large. This is achieved by showing that $\mathcal{E}^{(h)}$ is closely related to the following *symmetrized least squares error* which has appeared in the literature of ranking algorithms (Clemencon et al., 2005; Agarwal and Niyogi, 2009).

Definition 10 *The symmetrized least squares error is defined on the space $L^2_{\rho_X}$ by*

$$\mathcal{E}^{sls}(f) = \int_Z \int_Z \left[(y - f(x)) - \big(y' - f(x')\big)\right]^2 d\rho(x,y) d\rho(x',y'), \qquad f \in L^2_{\rho_X}.$$

To give the approximation of $\mathcal{E}^{(h)}$, we need a simpler form of $\mathcal{E}^{sls}$.

Lemma 11 *If $\mathbb{E}[Y^2] < \infty$, then by denoting $C_\rho = \int_Z \left[y - f_\rho(x)\right]^2 d\rho$, we have*

$$\mathcal{E}^{sls}(f) = 2\mathbf{var}[f(X) - f_\rho(X)] + 2C_\rho, \qquad \forall f \in L^2_{\rho_X}.$$

Proof Recall that for two independent and identically distributed samples ξ and ξ' of a random variable, one has the identity

$$\mathbb{E}[(\xi - \xi')^2] = 2[\mathbb{E}(\xi - \mathbb{E}\xi)^2] = 2\mathbf{var}(\xi).$$

Then we have

$$\mathcal{E}^{sls}(f) = \mathbb{E}\left[\Big((y - f(x)) - \big(y' - f(x')\big)\Big)^2\right] = 2\mathbf{var}[Y - f(X)].$$

By the definition $\mathbb{E}[Y|X] = f_\rho(X)$, it is easy to see that $C_\rho = \mathbf{var}(Y - f_\rho(X))$ and the covariance between $Y - f_\rho(X)$ and $f_\rho(X) - f(X)$ vanishes. So $\mathbf{var}[Y - f(X)] = \mathbf{var}[Y - f_\rho(X)] + \mathbf{var}[f(X) - f_\rho(X)]$. This proves the desired identity. ■

We are in a position to present the approximation of $\mathcal{E}^{(h)}$ for which a large scaling parameter h plays an important role. Since $\mathcal{H}$ is a compact subset of $C(X)$, we know that the number $\sup_{f\in\mathcal{H}} \|f\|_\infty$ is finite.

Lemma 12 *Under assumptions (2) and (3), for any essentially bounded measurable function f on X, we have*

$$\left|\mathcal{E}^{(h)}(f) + h^2 G(0) - C_\rho - \mathbf{var}[f(X) - f_\rho(X)]\right| \le 5 \cdot 2^7 C_G \left(\left(\mathbb{E}[|Y|^q]\right)^{\frac{q^*+2}{q}} + \|f\|_\infty^{q^*+2} \right) h^{-q^*}.$$

In particular,

$$\left|\mathcal{E}^{(h)}(f) + h^2 G(0) - C_\rho - \mathbf{var}[f(X) - f_\rho(X)]\right| \le C'_{\mathcal{H}} h^{-q^*}, \qquad \forall f \in \mathcal{H},$$

where $C'_{\mathcal{H}}$ is the constant depending on ρ, G, q and $\mathcal{H}$ given by

$$C'_{\mathcal{H}} = 5 \cdot 2^7 C_G \left(\left(\mathbb{E}[|Y|^q]\right)^{(q^*+2)/q} + \left(\sup_{f\in\mathcal{H}} \|f\|_\infty \right)^{q^*+2} \right).$$

Proof Observe that $q^*+2=\min\{q,4\}\in(2,4]$. By the Taylor expansion and the mean value theorem, we have

$$|G(t)-G(0)-G'_+(0)t|\le\begin{cases}\frac{\|G''\|_\infty}{2}t^2\le\frac{\|G''\|_\infty}{2}t^{(q^*+2)/2}, & \text{if } 0\le t\le 1,\\ 2\|G'\|_\infty t\le 2\|G'\|_\infty t^{(q^*+2)/2}, & \text{if } t>1.\end{cases}$$

So $|G(t)-G(0)-G'_+(0)t|\le\left(\frac{\|G''\|_\infty}{2}+2\|G'\|_\infty\right)t^{(q^*+2)/2}$ for all $t\ge 0$, and by setting $t=\frac{[(y-f(x))-(y'-f(x'))]^2}{2h^2}$, we know that

$$\begin{aligned}&\left|\mathcal{E}^{(h)}(f)+h^2G(0)+\int_{\mathcal{Z}}\int_{\mathcal{Z}}G'_+(0)\frac{[(y-f(x))-(y'-f(x'))]^2}{2}d\rho(x,y)d\rho(x',y')\right|\\ &\le\left(\frac{\|G''\|_\infty}{2}+2\|G'\|_\infty\right)h^{-q^*}2^{-\frac{q^*+2}{2}}\int_{\mathcal{Z}}\int_{\mathcal{Z}}\left|(y-f(x))-\left(y'-f(x')\right)\right|^{q^*+2}d\rho(x,y)d\rho(x',y')\\ &\le\left(\frac{\|G''\|_\infty}{2}+2\|G'\|_\infty\right)h^{-q^*}2^8\left\{\int_{\mathcal{Z}}|y|^{q^*+2}d\rho+\|f\|_\infty^{q^*+2}\right\}.\end{aligned}$$

This together with Lemma 11, the normalization assumption $G'_+(0)=-1$ and Hölder's inequality applied when $q>4$ proves the desired bound and hence our conclusion. ■

Applying Lemma 12 to a function $f\in\mathcal{H}$ and $f_\rho\in L^\infty_{\rho_X}$ yields the following fact on the excess generalization error $\mathcal{E}^{(h)}(f)-\mathcal{E}^{(h)}(f_\rho)$.

Theorem 13 *Under assumptions (2) and (3), we have*

$$\left|\mathcal{E}^{(h)}(f)-\mathcal{E}^{(h)}(f_\rho)-\mathbf{var}[f(X)-f_\rho(X)]\right|\le C''_{\mathcal{H}}h^{-q^*},\qquad\forall f\in\mathcal{H},$$

where $C''_{\mathcal{H}}$ is the constant depending on ρ,G,q and $\mathcal{H}$ given by

$$C''_{\mathcal{H}}=5\cdot 2^8C_G\left((\mathbb{E}[|Y|^q])^{(q^*+2)/q}+\left(\sup_{f\in\mathcal{H}}\|f\|_\infty\right)^{q^*+2}+\|f_\rho\|_\infty^{q^*+2}\right).$$

3.2 Functional Minimizer and Best Approximation

As f_ρ may not be a minimizer of $\mathcal{E}^{(h)}$, the second technical difficulty in our error analysis is the diversity of two ways to define a *target function* in $\mathcal{H}$, one to minimize the information error and the other to minimize the variance $\mathbf{var}[f(X)-f_\rho(X)]$. These possible candidates for the target function are defined as

$$\begin{aligned}f_{\mathcal{H}}&:=\arg\min_{f\in\mathcal{H}}\mathcal{E}^{(h)}(f),\\ f_{approx}&:=\arg\min_{f\in\mathcal{H}}\mathbf{var}[f(X)-f_\rho(X)].\end{aligned}$$

Our novelty to overcome the technical difficulty is to show that when the MEE scaling parameter h is large, these two functions are actually very close.

Theorem 14 *Under assumptions (2) and (3), we have*

$$\mathcal{E}^{(h)}(f_{approx}) \le \mathcal{E}^{(h)}(f_{\mathcal{H}}) + 2C''_{\mathcal{H}} h^{-q^*}$$

and

$$\mathbf{var}[f_{\mathcal{H}}(X) - f_\rho(X)] \le \mathbf{var}[f_{approx}(X) - f_\rho(X)] + 2C''_{\mathcal{H}} h^{-q^*}.$$

Proof By Theorem 13 and the definitions of $f_{\mathcal{H}}$ and f_{approx}, we have

$$\begin{aligned}
&\mathcal{E}^{(h)}(f_{\mathcal{H}}) - \mathcal{E}^{(h)}(f_\rho) \le \mathcal{E}^{(h)}(f_{approx}) - \mathcal{E}^{(h)}(f_\rho) \le \mathbf{var}[f_{approx}(X) - f_\rho(X)] + C''_{\mathcal{H}} h^{-q^*} \\
&\le \mathbf{var}[f_{\mathcal{H}}(X) - f_\rho(X)] + C''_{\mathcal{H}} h^{-q^*} \le \mathcal{E}^{(h)}(f_{\mathcal{H}}) - \mathcal{E}^{(h)}(f_\rho) + 2C''_{\mathcal{H}} h^{-q^*} \\
&\le \mathbf{var}[f_{approx}(X) - f_\rho(X)] + 3C''_{\mathcal{H}} h^{-q^*}.
\end{aligned}$$

Then the desired inequalities follow. ■

Moreover, Theorem 13 yields the following error decomposition for our algorithm.

Lemma 15 *Under assumptions (2) and (3), we have*

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \le \left\{ \mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}}) \right\} + \mathbf{var}[f_{approx}(X) - f_\rho(X)] + 2C''_{\mathcal{H}} h^{-q^*}. \tag{14}$$

Proof By Theorem 13,

$$\begin{aligned}
\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] &\le \mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_\rho) + C''_{\mathcal{H}} h^{-q^*} \\
&\le \left\{ \mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}}) \right\} + \mathcal{E}^{(h)}(f_{\mathcal{H}}) - \mathcal{E}^{(h)}(f_\rho) + C''_{\mathcal{H}} h^{-q^*}.
\end{aligned}$$

Since $f_{approx} \in \mathcal{H}$, the definition of $f_{\mathcal{H}}$ tells us that

$$\mathcal{E}^{(h)}(f_{\mathcal{H}}) - \mathcal{E}^{(h)}(f_\rho) \le \mathcal{E}^{(h)}(f_{approx}) - \mathcal{E}^{(h)}(f_\rho).$$

Applying Theorem 13 to the above bound implies

$$\mathbf{var}[f_{\mathbf{z}}(X) - f_\rho(X)] \le \left\{ \mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}}) \right\} + \mathbf{var}[f_{approx}(X) - f_\rho(X)] + 2C''_{\mathcal{H}} h^{-q^*}.$$

Then desired error decomposition (14) follows. ■

Error decomposition has been a standard technique to analyze least squares ERM regression algorithms (Anthony and Bartlett, 1999; Cucker and Zhou, 2007; Smale and Zhou, 2009; Ying, 2007). In error decomposition (14) for MEE learning algorithm (1), the first term on the right side is the sample error, the second term $\mathbf{var}[f_{approx}(X) - f_\rho(X)]$ is the approximation error, while the last extra term $2C''_{\mathcal{H}} h^{-q^*}$ is caused by the Parzen windowing and is small when h is large. The quantity $\mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}})$ of the sample error term will be bounded in the following discussion.

3.3 Error Decomposition by U-statistics and Special Properties

We shall decompose the sample error term $\mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}})$ further by means of U-statistics defined for $f \in \mathcal{H}$ and the sample $\mathbf{z}$ as

$$V_f(\mathbf{z}) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i} U_f(z_i, z_j),$$

where U_f is a kernel given with $z = (x,y), z' = (x',y') \in \mathcal{Z}$ by

$$U_f(z,z') = -h^2 G\left(\frac{[(y-f(x))-(y'-f(x'))]^2}{2h^2}\right) + h^2 G\left(\frac{\left[\left(y-f_\rho(x)\right)-\left(y'-f_\rho(x')\right)\right]^2}{2h^2}\right). \quad (15)$$

It is easy to see that $\mathbb{E}[V_f] = \mathcal{E}^{(h)}(f) - \mathcal{E}^{(h)}(f_\rho)$ and $U_f(z,z) = 0$. Then

$$\mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}}) = \mathbb{E}\left[V_{f_{\mathbf{z}}}\right] - \mathbb{E}\left[V_{f_{\mathcal{H}}}\right] = \mathbb{E}\left[V_{f_{\mathbf{z}}}\right] - V_{f_{\mathbf{z}}} + V_{f_{\mathbf{z}}} - V_{f_{\mathcal{H}}} + V_{f_{\mathcal{H}}} - \mathbb{E}\left[V_{f_{\mathcal{H}}}\right].$$

By the definition of $f_{\mathbf{z}}$, we have $V_{f_{\mathbf{z}}} - V_{f_{\mathcal{H}}} \leq 0$. Hence

$$\mathcal{E}^{(h)}(f_{\mathbf{z}}) - \mathcal{E}^{(h)}(f_{\mathcal{H}}) \leq \mathbb{E}\left[V_{f_{\mathbf{z}}}\right] - V_{f_{\mathbf{z}}} + V_{f_{\mathcal{H}}} - \mathbb{E}\left[V_{f_{\mathcal{H}}}\right]. \quad (16)$$

The above bound will be estimated by a uniform ratio probability inequality. A technical difficulty we meet here is the possibility that $\mathbb{E}[V_f] = \mathcal{E}^{(h)}(f) - \mathcal{E}^{(h)}(f_\rho)$ might be negative since f_ρ may not be a minimizer of $\mathcal{E}^{(h)}$. It is overcome by the following novel observation which is an immediate consequence of Theorem 13.

Lemma 16 *Under assumptions (2) and (3), if* $\varepsilon \geq C''_{\mathcal{H}} h^{-q^*}$, *then*

$$\mathbb{E}[V_f] + 2\varepsilon \geq \mathbb{E}[V_f] + C''_{\mathcal{H}} h^{-q^*} + \varepsilon \geq \mathbf{var}[f(X) - f_\rho(X)] + \varepsilon \geq \varepsilon, \qquad \forall f \in \mathcal{H}. \quad (17)$$

4. Sample Error Estimates

In this section, we follow (16) and estimate the sample error by a uniform ratio probability inequality based on the following Hoeffding's probability inequality for U-statistics (Hoeffding, 1963).

Lemma 17 *If U is a symmetric real-valued function on $\mathcal{Z} \times \mathcal{Z}$ satisfying $a \leq U(z,z') \leq b$ almost surely and* $\mathrm{var}[U] = \sigma^2$, *then for any* $\varepsilon > 0$,

$$\mathrm{Prob}\left\{\left|\frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i} U(z_i, z_j) - \mathbb{E}[U]\right| \geq \varepsilon\right\} \leq 2\exp\left\{-\frac{(m-1)\varepsilon^2}{4\sigma^2 + (4/3)(b-a)\varepsilon}\right\}.$$

To apply Lemma 17 we need to bound σ^2 and $b-a$ for the kernel U_f defined by (15). Our novelty for getting sharp bounds is to use a Taylor expansion involving a C^2 function $\widetilde{G}$ on $\mathbb{R}$:

$$\widetilde{G}(w) = \widetilde{G}(0) + \widetilde{G}'(0)w + \int_0^w (w-t)\widetilde{G}''(t)dt, \qquad \forall w \in \mathbb{R}. \quad (18)$$

Denote a constant $A_{\mathcal{H}}$ depending on ρ, G, q and $\mathcal{H}$ as

$$A_{\mathcal{H}} = 9 \cdot 2^8 C_G^2 \sup_{f \in \mathcal{H}} \|f - f_\rho\|_\infty^{\frac{4}{q}} \left((\mathbb{E}[|Y|^q])^{\frac{2}{q}} + \|f_\rho\|_\infty^2 + \sup_{f \in \mathcal{H}} \|f - f_\rho\|_\infty^2 \right).$$

Lemma 18 *Assume (2) and (3).*

(a) For any $f, g \in \mathcal{H}$, we have

$$\left|U_f\right| \le 4C_G\|f-f_\rho\|_\infty h \quad \text{and} \quad \left|U_f-U_g\right| \le 4C_G\|f-g\|_\infty h$$

and

$$\mathrm{var}[U_f] \le A_{\mathcal{H}}\left(\mathrm{var}[f(X)-f_\rho(X)]\right)^{(q-2)/q}.$$

(b) If $|Y| \le M$ almost surely for some constant $M > 0$, then we have almost surely

$$\left|U_f\right| \le A'_{\mathcal{H}}\left|(f(x)-f_\rho(x))-(f(x')-f_\rho(x'))\right|, \quad \forall f \in \mathcal{H} \tag{19}$$

and

$$\left|U_f-U_g\right| \le A'_{\mathcal{H}}\left|(f(x)-g(x))-(f(x')-g(x'))\right|, \quad \forall f,g \in \mathcal{H}, \tag{20}$$

where $A'_{\mathcal{H}}$ is a constant depending on ρ, G and $\mathcal{H}$ given by

$$A'_{\mathcal{H}} = 36C_G\left(M+\sup_{f\in\mathcal{H}}\|f\|_\infty\right).$$

Proof Define a function $\widetilde{G}$ on $\mathbb{R}$ by

$$\widetilde{G}(t) = G(t^2/2), \qquad t \in \mathbb{R}.$$

We see that $\widetilde{G} \in C^2(\mathbb{R})$, $\widetilde{G}(0) = G(0)$, $\widetilde{G}'(0) = 0$, $\widetilde{G}'(t) = tG'(t^2/2)$ and $\widetilde{G}''(t) = G'(t^2/2)+t^2G''(t^2/2)$. Moreover,

$$U_f(z,z') = -h^2\widetilde{G}\left(\frac{(y-f(x))-(y'-f(x'))}{h}\right)+h^2\widetilde{G}\left(\frac{\big(y-f_\rho(x)\big)-\big(y'-f_\rho(x')\big)}{h}\right).$$

(a) We apply the mean value theorem and see that $|U_f(z,z')| \le 2h\|\widetilde{G}'\|_\infty\|f-f_\rho\|_\infty$. The inequality for $|U_f-U_g|$ is obtained when f_ρ is replaced by g. Note that $\|\widetilde{G}'\|_\infty = \|tG'(t^2/2)\|_\infty$. Then the bounds for U_f and U_f-U_g are verified by noting $\|tG'(t^2/2)\|_\infty \le 2C_G$.

To bound the variance, we apply (18) to the two points $w_1 = \frac{(y-f(x))-(y'-f(x'))}{h}$ and $w_2 = \frac{\big(y-f_\rho(x)\big)-\big(y'-f_\rho(x')\big)}{h}$. Writing w_2-t as $w_2-w_1+w_1-t$, we see from $\widetilde{G}'(0)=0$ that

$$\begin{aligned} U_f(z,z') &= h^2\left(\widetilde{G}(w_2)-\widetilde{G}(w_1)\right) = h^2\widetilde{G}'(0)(w_2-w_1) \\ &\quad +h^2\int_0^{w_2}(w_2-t)\widetilde{G}''(t)dt - h^2\int_0^{w_1}(w_1-t)\widetilde{G}''(t)dt \\ &= h^2\int_0^{w_2}(w_2-w_1)\widetilde{G}''(t)dt + h^2\int_{w_1}^{w_2}(w_1-t)\widetilde{G}''(t)dt. \end{aligned}$$

It follows that

$$\begin{aligned} \left|U_f(z,z')\right| &\le \|\widetilde{G}''\|_\infty\left|\big(y-f_\rho(x)\big)-\big(y'-f_\rho(x')\big)\right|\ \left|\big(f(x)-f_\rho(x)\big)-\big(f(x')-f_\rho(x')\big)\right| \\ &\quad +\|\widetilde{G}''\|_\infty\left|\big(f(x)-f_\rho(x)\big)-\big(f(x')-f_\rho(x')\big)\right|^2. \end{aligned} \tag{21}$$

Since $\mathbb{E}[|Y|^q] < \infty$, we apply Hölder's inequality and see that

$$\begin{aligned}
&\int_{\mathcal{Z}}\int_{\mathcal{Z}} \left|\big(y-f_\rho(x)\big)-\big(y'-f_\rho(x')\big)\right|^2 \left|\big(f(x)-f_\rho(x)\big)-\big(f(x')-f_\rho(x')\big)\right|^2 d\rho(z)d\rho(z') \\
&\le \left\{\int_{\mathcal{Z}}\int_{\mathcal{Z}} \left|\big(y-f_\rho(x)\big)-\big(y'-f_\rho(x')\big)\right|^q d\rho(z)d\rho(z')\right\}^{2/q} \\
&\qquad \left\{\int_{\mathcal{Z}}\int_{\mathcal{Z}} \left|\big(f(x)-f_\rho(x)\big)-\big(f(x')-f_\rho(x')\big)\right|^{2q/(q-2)} d\rho(z)d\rho(z')\right\}^{1-2/q} \\
&\le \left\{4^{q+1}(\mathbb{E}[|Y|^q]+\|f_\rho\|_\infty^q)\right\}^{2/q} \left\{\|f-f_\rho\|_\infty^{4/(q-2)} 2\mathrm{var}[f(X)-f_\rho(X)]\right\}^{(q-2)/q}.
\end{aligned}$$

Here we have separated the power index $2q/(q-2)$ into the sum of $4/(q-2)$ and 2. Then

$$\begin{aligned}
\mathrm{var}[U_f] \quad &\le \quad \mathbb{E}[U_f^2] \le 2\|\widetilde{G}''\|_\infty^2 2^{\frac{5q+3}{q}} (\mathbb{E}[|Y|^q]+\|f_\rho\|_\infty^q)^{\frac{2}{q}} \|f-f_\rho\|_\infty^{\frac{4}{q}} \left(\mathrm{var}[f(X)-f_\rho(X)]\right)^{\frac{q-2}{q}} \\
&\quad +2\|\widetilde{G}''\|_\infty^2 4\|f-f_\rho\|_\infty^2 2\mathrm{var}[f(X)-f_\rho(X)].
\end{aligned}$$

Hence the desired inequality holds true since $\|\widetilde{G}''\|_\infty \le \|G'\|_\infty + \|t^2 G''(t^2/2)\|_\infty \le 3C_G$ and $\mathrm{var}[f(X)-f_\rho(X)] \le \|f-f_\rho\|_\infty^2$.

(b) If $|Y| \le M$ almost surely for some constant $M > 0$, then we see from (21) that almost surely $\left|U_f(z,z')\right| \le 4\|\widetilde{G}''\|_\infty (M+\|f_\rho\|_\infty+\|f-f_\rho\|_\infty)\left|\big(f(x)-f_\rho(x)\big)-\big(f(x')-f_\rho(x')\big)\right|$. Hence (19) holds true almost surely. Replacing f_ρ by g in (21), we see immediately inequality (20). The proof of Lemma 18 is complete. ■

With the above preparation, we can now give the uniform ratio probability inequality for U-statistics to estimate the sample error, following methods in the learning theory literature (Haussler et al., 1994; Koltchinskii, 2006; Cucker and Zhou, 2007).

Lemma 19 *Assume (2), (3) and $\varepsilon \ge C''_{\mathcal{H}} h^{-q^*}$. Then we have*

$$\mathrm{Prob}\left\{\sup_{f\in\mathcal{H}} \frac{\left|V_f-\mathbb{E}[V_f]\right|}{(\mathbb{E}[V_f]+2\varepsilon)^{(q-2)/q}} > 4\varepsilon^{2/q}\right\} \le 2\mathcal{N}\left(\mathcal{H}, \frac{\varepsilon}{4C_G h}\right) \exp\left\{-\frac{(m-1)\varepsilon}{A''_{\mathcal{H}} h}\right\},$$

where $A''_{\mathcal{H}}$ is the constant given by

$$A''_{\mathcal{H}} = 4A_{\mathcal{H}}(C''_{\mathcal{H}})^{-2/q} + 12C_G \sup_{f\in\mathcal{H}} \|f-f_\rho\|_\infty.$$

If $|Y| \le M$ almost surely for some constant $M > 0$, then we have

$$\mathrm{Prob}\left\{\sup_{f\in\mathcal{H}} \frac{\left|V_f-\mathbb{E}[V_f]\right|}{\sqrt{\mathbb{E}[V_f]+2\varepsilon}} > 4\sqrt{\varepsilon}\right\} \le 2\mathcal{N}\left(\mathcal{H}, \frac{\varepsilon}{2A'_{\mathcal{H}}}\right) \exp\left\{-\frac{(m-1)\varepsilon}{A''_{\mathcal{H}}}\right\},$$

where $A''_{\mathcal{H}}$ is the constant given by

$$A''_{\mathcal{H}} = 8A'_{\mathcal{H}} + 6A'_{\mathcal{H}} \sup_{f\in\mathcal{H}} \|f-f_\rho\|_\infty.$$

Proof If $\|f-f_j\|_\infty \le \frac{\varepsilon}{4C_G h}$, Lemma 18 (a) implies $|\mathbb{E}[V_f]-\mathbb{E}[V_{f_j}]| \le \varepsilon$ and $|V_f - V_{f_j}| \le \varepsilon$ almost surely. These in connection with Lemma 16 tell us that

$$\frac{\big|V_f-\mathbb{E}[V_f]\big|}{(\mathbb{E}[V_f]+2\varepsilon)^{(q-2)/q}} > 4\varepsilon^{2/q} \quad \Longrightarrow \quad \frac{\big|V_{f_j}-\mathbb{E}[V_{f_j}]\big|}{(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}} > \varepsilon^{2/q}.$$

Thus by taking $\{f_j\}_{j=1}^N$ to be an $\frac{\varepsilon}{4C_G h}$ net of the set $\mathcal{H}$ with N being the covering number $\mathcal{N}\left(\mathcal{H}, \frac{\varepsilon}{4C_G h}\right)$, we find

$$\begin{aligned}
&\text{Prob}\left\{\sup_{f\in\mathcal{H}} \frac{\big|V_f-\mathbb{E}[V_f]\big|}{(\mathbb{E}[V_f]+2\varepsilon)^{(q-2)/q}} > 4\varepsilon^{2/q}\right\} \le \text{Prob}\left\{\sup_{j=1,\ldots,N} \frac{\big|V_{f_j}-\mathbb{E}[V_{f_j}]\big|}{(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}} > \varepsilon^{2/q}\right\} \\
&\le \sum_{j=1,\ldots,N} \text{Prob}\left\{\frac{\big|V_{f_j}-\mathbb{E}[V_{f_j}]\big|}{(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}} > \varepsilon^{2/q}\right\}.
\end{aligned}$$

Fix $j \in \{1,\ldots,N\}$. Apply Lemma 17 to $U = U_{f_j}$ satisfying $\frac{1}{m(m-1)}\sum_{i=1}^m \sum_{j\neq i} U(z_i,z_j) - \mathbb{E}[U] = V_{f_j} - \mathbb{E}[V_{f_j}]$. By the bounds for $|U_{f_j}|$ and $\text{var}[U_{f_j}]$ from Part (b) of Lemma 18, we know by taking $\widetilde{\varepsilon} = \varepsilon^{2/q}(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}$ that

$$\begin{aligned}
&\text{Prob}\left\{\frac{\big|V_{f_j}-\mathbb{E}[V_{f_j}]\big|}{(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}} > \varepsilon^{2/q}\right\} = \text{Prob}\left\{\big|V_{f_j}-\mathbb{E}[V_{f_j}]\big| > \widetilde{\varepsilon}\right\} \\
&\le 2\exp\left\{-\frac{(m-1)\widetilde{\varepsilon}^2}{4A_{\mathcal{H}}\left(\text{var}[f_j(X)-f_\rho(X)]\right)^{(q-2)/q} + 12C_G\|f_j-f_\rho\|_\infty h\widetilde{\varepsilon}}\right\} \\
&\le 2\exp\left\{-\frac{(m-1)\varepsilon^{4/q}(\mathbb{E}[V_{f_j}]+2\varepsilon)^{(q-2)/q}}{4A_{\mathcal{H}}+12C_G\|f_j-f_\rho\|_\infty h\varepsilon^{2/q}}\right\},
\end{aligned}$$

where in the last step we have used the important relation (17) to the function $f = f_j$ and bounded $\left(\text{var}[f_j(X)-f_\rho(X)]\right)^{(q-2)/q}$ by $\left\{(\mathbb{E}[V_{f_j}]+2\varepsilon)\right\}^{(q-2)/q}$. This together with the notation $N = \mathcal{N}\left(\mathcal{H}, \frac{\varepsilon}{4C_G h}\right)$ and the inequality $\|f_j-f_\rho\|_\infty \le \sup_{f\in\mathcal{H}}\|f-f_\rho\|_\infty$ gives the first desired bound, where we have observed that $\varepsilon \ge C''_{\mathcal{H}} h^{-q^*}$ and $h \ge 1$ imply $\varepsilon^{-2/q} \le (C''_{\mathcal{H}})^{-2/q} h$.

If $|Y| \le M$ almost surely for some constant $M > 0$, then we follows the same line as in our above proof. According to Part (b) of Lemma 18, we should replace $4C_G h$ by $2A'_{\mathcal{H}}$, q by 4, and bound the variance $\text{var}[U_{f_j}]$ by $2A'_{\mathcal{H}}\text{var}[f_j(X)-f_\rho(X)] \le 2A'_{\mathcal{H}}(\mathbb{E}[V_{f_j}]+2\varepsilon)$. Then the desired estimate follows. The proof of Lemma 19 is complete. ■

We are in a position to bound the sample error. To unify the two estimates in Lemma 19, we denote $A'_{\mathcal{H}} = 2C_G$ in the general case. For $m \in \mathbb{N}, 0 < \delta < 1$, let $\varepsilon_{m,\delta}$ be the smallest positive solution to the inequality

$$\log \mathcal{N}\left(\mathcal{H}, \frac{\varepsilon}{2A'_{\mathcal{H}}}\right) - \frac{(m-1)\varepsilon}{A''_{\mathcal{H}}} \le \log\frac{\delta}{2}. \tag{22}$$

Proposition 20 *Let* $0<\delta<1, 0<\eta\leq 1$. *Under assumptions (2) and (3), we have with confidence of* $1-\delta$,

$$\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]\leq(1+\eta)\mathbf{var}[f_{approx}(X)-f_\rho(X)]+12\left(2+24^{\frac{q-2}{2}}\right)\eta^{\frac{2-q}{2}}(h\varepsilon_{m,\delta}+2C''_{\mathcal{H}}h^{-q^*}).$$

If $|Y|\leq M$ *almost surely for some* $M>0$, *then with confidence of* $1-\delta$, *we have*

$$\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]\leq(1+\eta)\mathbf{var}[f_{approx}(X)-f_\rho(X)]+\frac{278}{\eta}(\varepsilon_{m,\delta}+2C''_{\mathcal{H}}h^{-2}).$$

Proof Denote $\tau=(q-2)/q$ and $\varepsilon_{m,\delta,h}=\max\{h\varepsilon_{m,\delta},C''_{\mathcal{H}}h^{-q^*}\}$ in the general case with some $q>2$, while $\tau=1/2$ and $\varepsilon_{m,\delta,h}=\max\{\varepsilon_{m,\delta},C''_{\mathcal{H}}h^{-2}\}$ when $|Y|\leq M$ almost surely. Then by Lemma 19, we know that with confidence $1-\delta$, there holds

$$\sup_{f\in\mathcal{H}}\frac{\left|V_f-\mathbb{E}[V_f]\right|}{(\mathbb{E}[V_f]+2\varepsilon_{m,\delta,h})^\tau}\leq 4\varepsilon_{m,\delta,h}^{1-\tau}$$

which implies

$$\mathbb{E}\left[V_{f_{\mathbf{z}}}\right]-V_{f_{\mathbf{z}}}+V_{f_{\mathcal{H}}}-\mathbb{E}\left[V_{f_{\mathcal{H}}}\right]\leq 4\varepsilon_{m,\delta,h}^{1-\tau}(\mathbb{E}[V_{f_{\mathbf{z}}}]+2\varepsilon_{m,\delta,h})^\tau+4\varepsilon_{m,\delta,h}^{1-\tau}(\mathbb{E}[V_{f_{\mathcal{H}}}]+2\varepsilon_{m,\delta,h})^\tau.$$

This together with Lemma 15 and (16) yields

$$\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]\leq 4\mathcal{S}+16\varepsilon_{m,\delta,h}+\mathbf{var}[f_{approx}(X)-f_\rho(X)]+2C''_{\mathcal{H}}h^{-q^*},\tag{23}$$

where

$$\mathcal{S}:=\varepsilon_{m,\delta,h}^{1-\tau}(\mathbb{E}[V_{f_{\mathbf{z}}}])^\tau+\varepsilon_{m,\delta,h}^{1-\tau}(\mathbb{E}[V_{f_{\mathcal{H}}}])^\tau=(\frac{24}{\eta})^\tau\varepsilon_{m,\delta,h}^{1-\tau}\left(\frac{\eta}{24}\mathbb{E}[V_{f_{\mathbf{z}}}]\right)^\tau+(\frac{12}{\eta})^\tau\varepsilon_{m,\delta,h}^{1-\tau}\left(\frac{\eta}{12}\mathbb{E}[V_{f_{\mathcal{H}}}]\right)^\tau.$$

Now we apply Young's inequality

$$a\cdot b\leq(1-\tau)a^{1/(1-\tau)}+\tau b^{1/\tau},\qquad a,b\geq 0$$

and find

$$\mathcal{S}\leq\left(\frac{24}{\eta}\right)^{\tau/(1-\tau)}\varepsilon_{m,\delta,h}+\frac{\eta}{24}\mathbb{E}[V_{f_{\mathbf{z}}}]+\left(\frac{12}{\eta}\right)^{\tau/(1-\tau)}\varepsilon_{m,\delta,h}+\frac{\eta}{12}\mathbb{E}[V_{f_{\mathcal{H}}}].$$

Combining this with (23), Theorem 13 and the identity $\mathbb{E}[V_f]=\mathcal{E}^{(h)}(f)-\mathcal{E}^{(h)}(f_\rho)$ gives

$$\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]\leq\frac{\eta}{6}\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]+(1+\frac{\eta}{3})\mathbf{var}[f_{approx}(X)-f_\rho(X)]+\mathcal{S}',$$

where $\mathcal{S}':=(16+8(24/\eta)^{\tau/(1-\tau)})\varepsilon_{m,\delta,h}+3C''_{\mathcal{H}}h^{-q^*}$. Since $1/(1-\frac{\eta}{6})\leq 1+\frac{\eta}{3}$ and $(1+\frac{\eta}{3})^2\leq 1+\eta$, we see that

$$\mathbf{var}[f_{\mathbf{z}}(X)-f_\rho(X)]\leq(1+\eta)\mathbf{var}[f_{approx}(X)-f_\rho(X)]+\frac{4}{3}\mathcal{S}'.$$

Then the desired estimates follow, and the proposition is proved. ■

5. Proof of Main Results

We are now in a position to prove our main results stated in Section 2.

5.1 Proof of Theorem 3

Recall $\mathcal{D}_{\mathcal{H}}(f_\rho) = \mathbf{var}[f_{approx}(X) - f_\rho(X)]$. Take $\eta = \min\{\varepsilon/(3\mathcal{D}_{\mathcal{H}}(f_\rho)), 1\}$. Then

$$\eta \mathbf{var}[f_{approx}(X) - f_\rho(X)] \le \varepsilon/3.$$

Now we take

$$h_{\varepsilon,\delta} = \left(72\left(2+24^{(q-2)/2}\right)\eta^{(2-q)/2}C''_{\mathcal{H}}/\varepsilon\right)^{1/q^*}.$$

Set $\widetilde{\varepsilon} := \varepsilon/\left(36\left(2+24^{(q-2)/2}\right)\eta^{(2-q)/2}\right)$. We choose

$$m_{\varepsilon,\delta}(h) = \frac{hA''_{\mathcal{H}}}{\widetilde{\varepsilon}}\left(\log \mathcal{N}\left(\mathcal{H}, \frac{\widetilde{\varepsilon}}{2hA'_{\mathcal{H}}}\right) - \log\frac{\delta}{2}\right) + 1.$$

With this choice, we know that whenever $m \ge m_{\varepsilon,\delta}(h)$, the solution $\varepsilon_{m,\delta}$ to inequality (22) satisfies $\varepsilon_{m,\delta} \le \widetilde{\varepsilon}/h$. Combining all the above estimates and Proposition 20, we see that whenever $h \ge h_{\varepsilon,\delta}$ and $m \ge m_{\varepsilon,\delta}(h)$, error bound (4) holds true with confidence $1-\delta$. This proves Theorem 3. ■

5.2 Proof of Theorem 5

We apply Proposition 20. By covering number condition (5), we know that $\varepsilon_{m,\delta}$ is bounded by $\widetilde{\varepsilon}_{m,\delta}$, the smallest positive solution to the inequality

$$A_p\left(\frac{2A'_{\mathcal{H}}}{\varepsilon}\right)^p - \frac{(m-1)\varepsilon}{A''_{\mathcal{H}}} \le \log\frac{\delta}{2}.$$

This inequality written as $\varepsilon^{1+p} - \frac{A''_{\mathcal{H}}}{m-1}\log\frac{2}{\delta}\varepsilon^p - A_p\left(2A'_{\mathcal{H}}\right)^p\frac{A''_{\mathcal{H}}}{m-1} \ge 0$ is well understood in learning theory (e.g., Cucker and Zhou, 2007) and its solution can be bounded as

$$\widetilde{\varepsilon}_{m,\delta} \le \max\left\{2\frac{A''_{\mathcal{H}}}{m-1}\log\frac{2}{\delta}, \left(2A_pA''_{\mathcal{H}}(2A'_{\mathcal{H}})^p\right)^{1/(1+p)}(m-1)^{-\frac{1}{1+p}}\right\}.$$

If $\mathbb{E}[|Y|^q] < \infty$ for some $q > 2$, then the first part of Proposition 20 verifies (6) with the constant $\widetilde{C}_{\mathcal{H}}$ given by

$$\widetilde{C}_{\mathcal{H}} = 24\left(2+24^{(q-2)/2}\right)\left(2A''_{\mathcal{H}} + \left(2A_pA''_{\mathcal{H}}(2A'_{\mathcal{H}})^p\right)^{1/(1+p)} + 2C''_{\mathcal{H}}\right).$$

If $|Y| \le M$ almost surely for some $M > 0$, then the second part of Proposition 20 proves (7) with the constant $\widetilde{C}_{\mathcal{H}}$ given by

$$\widetilde{C}_{\mathcal{H}} = 278\left(2A''_{\mathcal{H}} + \left(2A_pA''_{\mathcal{H}}(2A'_{\mathcal{H}})^p\right)^{1/(1+p)} + 2C''_{\mathcal{H}}\right).$$

This completes the proof of Theorem 5. ■

5.3 Proof of Theorem 8

Note

$$\left|\frac{1}{m}\sum_{i=1}^{m}\left[f(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\frac{1}{m}\sum_{i=1}^{m}\left[g(x_i)-\pi_{\sqrt{m}}(y_i)\right]\right|\le\|f-g\|_\infty$$

and

$$\left|\mathbb{E}[f(X)-\pi_{\sqrt{m}}(Y)]-\mathbb{E}[g(X)-\pi_{\sqrt{m}}(Y)]\right|\le\|f-g\|_\infty.$$

So by taking $\{f_j\}_{j=1}^N$ to be an $\frac{\varepsilon}{4}$ net of the set $\mathcal{H}$ with $N=\mathcal{N}\left(\mathcal{H},\frac{\varepsilon}{4}\right)$, we know that for each $f\in\mathcal{H}$ there is some $j\in\{1,\ldots,N\}$ such that $\|f-f_j\|_\infty\le\frac{\varepsilon}{4}$. Hence

$$\begin{aligned}&\left|\frac{1}{m}\sum_{i=1}^{m}\left[f(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f(X)-\pi_{\sqrt{m}}(Y)]\right|>\varepsilon\\ \Longrightarrow\quad&\left|\frac{1}{m}\sum_{i=1}^{m}\left[f_j(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f_j(X)-\pi_{\sqrt{m}}(Y)]\right|>\frac{\varepsilon}{2}.\end{aligned}$$

It follows that

$$\begin{aligned}&\mathrm{Prob}\left\{\sup_{f\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\left[f(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f(X)-\pi_{\sqrt{m}}(Y)]\right|>\varepsilon\right\}\\ \le\quad&\mathrm{Prob}\left\{\sup_{j=1,\ldots,N}\left|\frac{1}{m}\sum_{i=1}^{m}\left[f_j(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f_j(X)-\pi_{\sqrt{m}}(Y)]\right|>\frac{\varepsilon}{2}\right\}\\ \le\quad&\sum_{j=1}^{N}\mathrm{Prob}\left\{\left|\frac{1}{m}\sum_{i=1}^{m}\left[f_j(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f_j(X)-\pi_{\sqrt{m}}(Y)]\right|>\frac{\varepsilon}{2}\right\}.\end{aligned}$$

For each fixed $j\in\{1,\ldots,N\}$, we apply the classical Bernstein probability inequality to the random variable $\xi=f_j(X)-\pi_{\sqrt{m}}(Y)$ on (Z,ρ) bounded by $\widetilde{M}=\sup_{f\in\mathcal{H}}\|f\|_\infty+\sqrt{m}$ with variance $\sigma^2(\xi)\le\mathbb{E}[|f_j(X)-\pi_{\sqrt{m}}(Y)|^2]\le 2\sup_{f\in\mathcal{H}}\|f\|_\infty^2+2\mathbb{E}[|Y|^2]=:\sigma_{\mathcal{H}}^2$ and know that

$$\begin{aligned}&\mathrm{Prob}\left\{\left|\frac{1}{m}\sum_{i=1}^{m}\left[f_j(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f_j(X)-\pi_{\sqrt{m}}(Y)]\right|>\frac{\varepsilon}{2}\right\}\\ &\le 2\exp\left\{-\frac{m(\varepsilon/2)^2}{\frac{2}{3}\widetilde{M}\varepsilon/2+2\sigma^2(\xi)}\right\}\le 2\exp\left\{-\frac{m\varepsilon^2}{\frac{4}{3}\widetilde{M}\varepsilon+8\sigma_{\mathcal{H}}^2}\right\}.\end{aligned}$$

The above argument together with covering number condition (5) yields

$$\begin{aligned}&\mathrm{Prob}\left\{\sup_{f\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\left[f(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f(X)-\pi_{\sqrt{m}}(Y)]\right|>\varepsilon\right\}\\ &\le 2N\exp\left\{-\frac{m\varepsilon^2}{\frac{4}{3}\widetilde{M}\varepsilon+8\sigma_{\mathcal{H}}^2}\right\}\le 2\exp\left\{A_p\left(\frac{4}{\varepsilon}\right)^p-\frac{m\varepsilon^2}{\frac{4}{3}\widetilde{M}\varepsilon+8\sigma_{\mathcal{H}}^2}\right\}.\end{aligned}$$

Bounding the right-hand side above by δ is equivalent to the inequality

$$\varepsilon^{2+p}-\frac{4}{3m}\widetilde{M}\log\frac{2}{\delta}\varepsilon^{1+p}-\frac{8}{m}\sigma_{\mathcal{H}}^2\log\frac{2}{\delta}\varepsilon^p-\frac{A_p4^p}{m}\ge 0.$$

By taking $\widetilde{\varepsilon}_{m,\delta}$ to be the smallest solution to the above inequality, we see from Cucker and Zhou (2007) as in the proof of Theorem 5 that with confidence at least $1-\delta$,

$$\begin{aligned}
&\sup_{f\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^{m}\left[f(x_i)-\pi_{\sqrt{m}}(y_i)\right]-\mathbb{E}[f(X)-\pi_{\sqrt{m}}(Y)]\right| \\
&\le\widetilde{\varepsilon}_{m,\delta}\le\max\left\{\frac{4\widetilde{M}}{m}\log\frac{2}{\delta},\sqrt{\frac{24\sigma^2_{\mathcal{H}}}{m}\log\frac{2}{\delta}},\left(\frac{A_p4^p}{m}\right)^{\frac{1}{2+p}}\right\} \\
&\le\left\{7\sup_{f\in\mathcal{H}}\|f\|_\infty+4+7\sqrt{\mathbb{E}[|Y|^2]}+4A_p^{\frac{1}{2+p}}\right\}m^{-\frac{1}{2+p}}\log\frac{2}{\delta}.
\end{aligned}$$

Moreover, since $\pi_{\sqrt{m}}(y)-y=0$ for $|y|\le\sqrt{m}$ while $|\pi_{\sqrt{m}}(y)-y|\le|y|\le\frac{|y|^2}{\sqrt{m}}$ for $|y|>\sqrt{m}$, we know that

$$\begin{aligned}
&\left|\mathbb{E}[\pi_{\sqrt{m}}(Y)]-\mathbb{E}[f_\rho(X)]\right|=\left|\int_X\int_Y\pi_{\sqrt{m}}(y)-yd\rho(y|x)d\rho_X(x)\right| \\
&=\left|\int_X\int_{|y|>\sqrt{m}}\pi_{\sqrt{m}}(y)-yd\rho(y|x)d\rho_X(x)\right|\le\int_X\int_{|y|>\sqrt{m}}\frac{|y|^2}{\sqrt{m}}d\rho(y|x)d\rho_X(x)\le\frac{\mathbb{E}[|Y|^2]}{\sqrt{m}}.
\end{aligned}$$

Therefore, (11) holds with confidence at least $1-\delta$. The proof of Theorem 8 is complete. ■

6. Conclusion and Discussion

In this paper we have proved the consistency of an MEE algorithm associated with Rényi's entropy of order 2 by letting the scaling parameter h in the kernel density estimator tends to infinity at an appropriate rate. This result explains the effectiveness of the MEE principle in empirical applications where the parameter h is required to be large enough before smaller values are tuned. However, the motivation of the MEE principle is to minimize error entropies approximately, and requires small h for the kernel density estimator to converge to the true probability density function. Therefore, our consistency result seems surprising.

As far as we know, our result is the first rigorous consistency result for MEE algorithms. There are many open questions in mathematical analysis of MEE algorithms. For instance, can MEE algorithm (1) be consistent by taking $h\to 0$? Can one carry out error analysis for the MEE algorithm if Shannon's entropy or Rényi's entropy of order $\alpha\neq 2$ is used? How can we establish error analysis for other learning settings such as those with non-identical sampling processes (Smale and Zhou, 2009; Hu, 2011)? These questions require further research and will be our future topics.

It might be helpful to understand our theoretical results by relating MEE algorithms to ranking algorithms. Note that MEE algorithm (1) essentially minimizes the empirical version of the information error which, according to our study in Section 2, differs from the symmetrized least squares error used in some ranking algorithms by an extra term which vanishes when $h\to\infty$. Our study may shed some light on analysis of some ranking algorithms.

Table 1: NOTATIONS

notation	meaning	pages
p_E	probability density function of a random variable E	378
$H_S(E)$	Shannon's entropy of a random variable E	378
$H_{R,\alpha}(E)$	Rényi's entropy of order α	378
X	explanatory variable for learning	378
Y	response variable for learning	378
$E = Y - f(X)$	error random variable associated with a predictor $f(X)$	378
$H_R(E)$	Rényi's entropy of order $\alpha = 2$	378
$\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$	a sample for learning	378
G	windowing function	378, 379, 380
h	MEE scaling parameter	378, 379
$\widehat{p_E}$	Parzen windowing approximation of p_E	378
$\widehat{H_S}$	empirical Shannon entropy	378
$\widehat{H_R}$	empirical Rényi's entropy of order 2	378
f_ρ	the regression function of ρ	379
$f_{\mathbf{z}}$	output function of the MEE learning algorithm (1)	379
$\mathcal{H}$	the hypothesis space for the ERM algorithm	379
var	the variance of a random variable	379
$q, q^* = \min\{q-2, 2\}$	power indices in condition (2) for $\mathbb{E}[\lvert Y\rvert^q] < \infty$	380
C_G	constant for decay condition (3) of G	380
$\mathcal{D}_{\mathcal{H}}(f_\rho)$	approximation error of the pair $(\mathcal{H}, \rho)$	380
$\mathcal{N}(\mathcal{H}, \varepsilon)$	covering number of the hypothesis space $\mathcal{H}$	380
p	power index for covering number condition (5)	380
$\pi_{\sqrt{m}}$	projection onto the closed interval $[-\sqrt{m}, \sqrt{m}]$	381
$\widetilde{f_{\mathbf{z}}}$	estimator of f_ρ	382
$\mathcal{E}^{(h)}(f)$	generalization error associated with G and h	383
$\mathcal{E}^{ls}(f)$	least squares generalization error $\mathcal{E}^{ls}(f) = \int_{\mathcal{Z}} (f(x) - y)^2 d\rho$	383
C_ρ	constant $C_\rho = \int_{\mathcal{Z}} \left[y - f_\rho(x)\right]^2 d\rho$ associated with ρ	384
$f_{\mathcal{H}}$	minimizer of $\mathcal{E}^{(h)}(f)$ in $\mathcal{H}$	385
f_{approx}	minimizer of $\mathbf{var}[f(X) - f_\rho(X)]$ in $\mathcal{H}$	385
U_f	kernel for the U statistics V_f	387
$\widetilde{G}$	an intermediate function defined by $\widetilde{G}(t) = G(t^2/2)$	388

Acknowledgments

We would like to thank the referees for their constructive suggestions and comments. The work described in this paper was supported by National Science Foundation of China under Grants (No. 11201348 and 11101403) and by a grant from the Research Grants Council of Hong Kong [Project No. CityU 103709]. The corresponding author is Ding-Xuan Zhou.

References

S. Agarwal and P. Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10:441–474, 2009.

M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

J. Y. Audibert and O. Catoni. Robust linear least squares regression. *Annals of Statistics*, 39:2766–2794, 2011.

B. Chen and J. C. Principe. Some further results on the minimum error entropy estimation. *Entropy*, 14:966–977, 2012.

S. Clemencon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. *Proceedings of COLT 2005, in LNCS Computational Learning Theory*, Springer-Verlag, Berlin, Heidelberg, 3559:1–15, 2005.

F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

D. Erdogmus and J. C. Principe. An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems. *IEEE Transactions on Signal Processing*, 50:1780–1786, 2002.

D. Erdogmus and J. C. Principe. Convergence properties and data efficiency of the minimum error entropy criterion in adaline training. *IEEE Transactions on Signal Processing*, 51:1966–1978, 2003.

D. Haussler, M. Kearns, and R. Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, 14:83–114, 1994.

T. Hu. Online regression with varying Gaussians and non-identical distributions. *Analysis and Applications*, 9:395–408, 2011.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *Annals of Statistics*, 34:2593–2656, 2006.

E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1049–1051, 1962.

J. C. Principe. *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*. Springer, New York, 2010.

L. M. Silva, J. M. de Sá, and L. A. Alexandre. The MEE principle in data classification: a perceptrop-based analysis. *Neural Computation*, 22:2698–2728, 2010.

S. Smale and D. X. Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1:17–41, 2003.

S. Smale and D. X. Zhou. Online learning with Markov sampling. *Analysis and Applications*, 7:87–113, 2009.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Y. Yao. On complexity issue of online learning algorithms. *IEEE Transactions on Information Theory*, 56:6470–6481, 2010.

Y. Ying. Convergence analysis of online algorithms. *Advances in Computational Mathematics*, 27:273–291, 2007.

D. X. Zhou. The covering number in learning theory. *Journal of Complexity*, 18:739–767, 2002.

D. X. Zhou. Capacity of reproducing kernel spaces in learning theory. *IEEE Transactions on Information Theory*, 49:1743-1752, 2003.

Journal of Machine Learning Research 14 (2013) 399-436 Submitted 10/10; Revised 9/12; Published 1/13

Ranked Bandits in Metric Spaces: Learning Diverse Rankings over Large Document Collections*

Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA

Filip Radlinski FILIPRAD@MICROSOFT.COM
Microsoft Research Cambridge
7 J.J. Thomson Ave.
Cambridge UK

Sreenivas Gollapudi SREENIG@MICROSOFT.COM
Microsoft Research Silicon Valley
1065 La Avenida
Mountain View, CA 94043, USA

Editor: Nicolo Cesa-Bianchi

Abstract

Most learning to rank research has assumed that the utility of different documents is independent, which results in learned ranking functions that return redundant results. The few approaches that avoid this have rather unsatisfyingly lacked theoretical foundations, or do not scale. We present a learning-to-rank formulation that optimizes the fraction of satisfied users, with several scalable algorithms that explicitly takes document similarity and ranking context into account. Our formulation is a non-trivial common generalization of two multi-armed bandit models from the literature: *ranked bandits* (Radlinski et al., 2008) and *Lipschitz bandits* (Kleinberg et al., 2008b). We present theoretical justifications for this approach, as well as a near-optimal algorithm. Our evaluation adds optimizations that improve empirical performance, and shows that our algorithms learn orders of magnitude more quickly than previous approaches.

Keywords: online learning, clickthrough data, diversity, multi-armed bandits, contextual bandits, regret, metric spaces

1. Introduction

Identifying the most relevant results to a query is a central problem in web search, hence learning ranking functions has received a lot of attention (e.g., Joachims, 2002; Burges et al., 2005; Chu and Ghahramani, 2005; Taylor et al., 2008). One increasingly important goal is to learn from user interactions with search engines, such as clicks. We address the task of learning a ranking function that minimizes the likelihood of *query abandonment*: the event that the user does not click on any of the search results for a given query. This objective is particularly interesting as query abandonment

*. Preliminary versions of this paper has been published as a conference paper in *ICML 2010* and as a technical report at `arxiv.org/abs/1005.5197` (May 2010). Compared to the conference version, this paper contains full proofs and a significantly revised presentation.

is a major challenge in today's search engines, and is also sensitive to the diversity and redundancy among documents presented.

We consider the Multi-Armed Bandit (MAB) setting (e.g., Cesa-Bianchi and Lugosi, 2006), which captures many online learning problems wherein an algorithm chooses sequentially among a fixed set of alternatives, traditionally called "arms". In each round an algorithm chooses an arm and collects the corresponding reward. Crucially, the algorithm receives limited feedback—only for the arm it has chosen, which gives rise to the tradeoff between *exploration* (acquiring new information) and *exploitation* (taking advantage of the information available so far).

While most of the literature on MAB corresponds to learning a single best alternative, MAB algorithms can also be extended to learning a ranking of documents that minimizes query abandonment (Radlinski et al., 2008; Streeter and Golovin, 2008). In this setting, called *Ranked Bandits*, in each round an algorithm chooses an *ordered list* of k documents from some fixed collection of documents, and receives clicks on some of the chosen documents. Crucially, the click probability for a given document may depend on the documents shown above: a user scrolls the list top-down and may leave as soon as she clicked on the first document. The goal is to minimize query abandonment.

Radlinski et al. (2008) and Streeter and Golovin (2008) propose a simple but effective approach: for each position in the ranking there is a separate instance bandit algorithm which is responsible for choosing a document for this position. However, the specific algorithms they considered are impractical at WWW scales.

Prior work on MAB algorithms has considered exploiting structure in the space of arms to improve convergence rates. One particular approach, articulated by Kleinberg et al. (2008b) is well suited to our scenario: when the arms form a metric space and the payoff function satisfies a Lipschitz condition with respect to this metric space. The metric space provides information about similarity between arms, which allows the algorithm to make inferences about similar arms without exploring them. Further, they propose a "zooming algorithm" which partitions the metric space into regions (and treats each region as a "meta-arm") so that the partition is adaptively refined over time and becomes finer in regions with higher payoffs.

In web search, a metric space directly models similarity between documents. (It is worth noting that most offline learning-to-rank approaches also rely on similarity between documents, at least implicitly.)

Our contributions. This paper initiates the study of bandit learning-to-rank with side information on similarity between documents. We adopt the Ranked bandits setup: a user scrolls the results top-down and may leave after a single click, the goal is to minimize query abandonment. The similarity information is expressed as a metric space.

In this paper we consider a "perfect world" scenario: there exists an informative distance function which meaningfully describes similarity between documents in a ranked setting, and an algorithm has access to such function. We focus on two high-level questions: How to represent the knowledge of document similarity, and how to use it algorithmically in a bandit setting. We believe that studying such "perfect world" scenario is useful, and perhaps necessary, to inform and guide the corresponding data-driven work.

We propose a simple bandit model which combines *Ranked bandits* (Radlinski et al., 2008) and *Lipschitz bandits* (Kleinberg et al., 2008b), and admits efficient bandit algorithms that, unlike those in prior work on bandit learning-to-rank, scale to large document collections. Our model is based on the new notion of "conditional Lipschitz continuity" which asserts that similar documents have similar click probabilities even conditional on the event that all documents in a given set

of documents are skipped (i.e., not clicked on) by the current user. We study this model both theoretically and empirically.

First, we validate the expressiveness of our model by providing an explicit construction for a wide family of plausible user distributions which provably fit the model. The analysis of this construction is perhaps the most technical contribution of this paper. We also use this construction in simulations.

Second, we put forth a battery of algorithms for our model. Some of these algorithms are straightforward combinations of ideas from prior work on Ranked bandits and Lipschitz bandits, and some are new.

A crucial insight in the new algorithms is that for each position i in the ranking there is a *context* that we can use, namely the set of documents chosen for the above positions in the same round. Indeed, since our objective is non-abandonment we only care about position i if all documents shown above i have been skipped in the present round. So the algorithm responsible for position i can simply *assume* that these documents have been skipped.

This interpretation of contexts allows us to cast the position-i problem as a *contextual bandit* problem. Moreover, we derive a Lipschitz condition on contexts (with respect to a suitably defined metric), which allows us to use the contextual Lipschitz MAB machinery from Slivkins (2009). We also exploit correlations between clicks: if a given document is included in the context—that is, if this document is skipped by the current user—then similar documents are likely to be skipped, too. More specifically, we propose two algorithms that use contexts: a "heavy-weight" algorithm which uses both the metric on contexts and correlated clicks, and a "light-weight" algorithm which uses correlated clicks but not the metric on contexts.

Third, we provide scalability guarantees for the heavy-weight contextual algorithm, proving that the convergence rate depends only on the dimensionality of the metric space but not on the number of documents. However, we argue that our provable guarantees do not fully reflect the power of the algorithm, and outline some directions for the follow-up theoretical work. In particular, we identify a stronger benchmark and discuss convergence to this benchmark. We provide an initial result: we prove, without any guarantees on the convergence rate, that the heavy-weight contextual algorithm indeed converges to this stonger benchmark. This theoretical discussion is one of the contributions.

Finally, we empirically study the performance of our algorithms. We run a large-scale simulation using the above-mentioned construction with realistic parameters. The main goal is to compare the convergence rates of the various approaches. In particular, we confirm that metric-aware algorithms significantly outperform the metric-oblivious ones, and that taking the context into account improves the convergence rate. Somewhat surprisingly, our light-weight contextual algorithm performs better than the heavy-weight one.

A secondary, smaller-scale experiment studies the limit behaviour of the algorithms, that is, the query abandonment probability that the algorithms converge to. Following the theoretical discussion mentioned above, we design a principled example on which different algorithms exhibit very different limit behaviour. Interestingly, the heavy-weight contextual algorithm is the only algorithm that achieves the optimal limit behaviour in this experiment.

Map of the paper. We start with a brief survey of related work (Section 2). We define our model in Section 3, and validate its expressiveness in Section 4. In-depth discussion of relevant approaches from prior work is in Section 5. Our new approach, ranked contextual bandits in metric spaces, is presented in Section 6. Scalability guarantees are discussed in Section 7. We present our simulations in Section 8.

To keep the flow of the paper, the lengthy proofs for the theoretical results in Section 4 are presented in Section A and Section B. Moreover, the background on instance-dependent regret bounds for UCB1-style algorithms is discussed in Appendix C.

2. Related Work on Multi-Armed Bandits

Multi-armed bandits have been studied for many decades as a simple yet expressive model for understanding exploration-exploitation tradeoffs. A thorough discussion of the literature on bandit problems is beyond the scope of this paper. For background, a reader can refer to a book (Cesa-Bianchi and Lugosi, 2006) and a recent survey (Bubeck and Cesa-Bianchi, 2012) on regret-minimizing bandits.[1] A somewhat different, Bayesian perspective can be found in surveys (Sundaram, 2005; Bergemann and Välimäki, 2006).

On a very high level, there is a crucial distinction between regret-minimizing formulations and Bayesian/MDP formulations (see the surveys mentioned above); this paper follows the former. Among regret-minimizing formulations, an important distinction is between stochastic rewards (Lai and Robbins, 1985; Auer et al., 2002a) and adversarial rewards (Auer et al., 2002b).

Below we survey several directions that are directly relevant to this paper.

Ranked bandits. A bandit model in which an algorithm learns a ranking of documents with a goal to minimize query abandonment has been introduced in Radlinski et al. (2008) under the name *ranked bandits*. A crucial feature in this setting is that the click probability for a given document may depend not only on the document and the position in which it is shown, but also the documents shown above. In particular, documents shown above can "steal" clicks from the documents shown below, in the sense that a user scrolls the list top-down and may leave as soon as she clicked on the first document.

Independently, Streeter and Golovin (2008) considered a more general model where the goal is to minimize an arbitrary (known) submodular set function, rather than query abandonment. A further generalization to submodular functions on ordered assignements (rather than on sets) was considered in (Golovin et al., 2009). The contributions of the three papers essentially coincide for the special case of ranked bandits.

Uchiya et al. (2010)[2] and Kale et al. (2010)[2] considered a related bandit model in which an algorithm selects a ranking of documents in each round, but the click probabilities for a given document do not depend on which other documents are shown to the same user.

Bandits with structure. Numerous papers enriched the basic MAB setting by assuming some structure on arms, typically in order to handle settings where the number of arms is very large or infinite. Most relevant to this paper is the model where arms lie in a metric space and their expected rewards satisfy the Lipschitz condition with respect to this metric space (see Section 3 for details). This model, for a general metric space, has been introduced in Kleinberg et al. (2008b) under the name *Lipschitz MAB*; the special case of unit interval has been studied in (Agrawal, 1995; Kleinberg, 2004; Auer et al., 2007) under the name *continuum-armed bandits*. Subsequent work on Lipschtz MAB includes Bubeck et al. (2011), Kleinberg and Slivkins (2010), Maillard and Munos (2010), Slivkins (2009) and Slivkins (2011). A closely related model posits that arms corresponds to leaves

1. Regret of an algorithm in T rounds, typically denoted $R(T)$, is the expected payoff of the benchmark in T rounds minus that of the algorithm. A standard benchmark is the best arm in hindsight.
2. This is either concurrent or subsequent work with respect to the conference publication of this paper.

on a tree, but no metric space is revealed to the algorithm (Kocsis and Szepesvari, 2006; Pandey et al., 2007; Munos and Coquelin, 2007; Slivkins, 2011).

Another commonly assumed structure is linear or convex payoffs (e.g., Awerbuch and Kleinberg, 2008; Flaxman et al., 2005; Dani et al., 2007; Abernethy et al., 2008; Hazan and Kale, 2009). Linear/convex payoffs is a much stronger assumption than similarity, essentially because it allows to make strong inferences about far-away arms. Other structural assumptions have been considered, for example, Wang et al. (2008) and Bubeck and Munos (2010) and Srinivas et al. (2010)[2].

The distinction between the various possible structural assumptions is orthogonal to the distinction between stochastic and adversarial rewards. With a few exceptions, papers on MAB with linear/convex payoffs allow adversarial payoffs, whereas papers on MAB with similarity information focus on stochastic payoffs

Contextual bandits. Here in each round the algorithm receives a *context*, chooses an arm, and the reward depends both on the arm and the context. The term "contextual bandits" was coined in Langford and Zhang (2007). The setting, with a number of different modifications, has been introduced independently in several papers; a possibly incomplete list is Woodroofe (1979), Auer et al. (2002b), Auer (2002), Wang et al. (2005), Langford and Zhang (2007), Hazan and Megiddo (2007) and Pandey et al. (2007).

There are several models for how contexts are related to rewards: rewards are linear in the context (e.g., Auer, 2002; Langford and Zhang, 2007) and Chu et al. (2011)[2], the context is a random variable correlated with rewards (Woodroofe, 1979; Wang et al., 2005; Rigollet and Zeevi, 2010); rewards are Lipschitz with respect to a metric space on contexts (Hazan and Megiddo, 2007; Slivkins, 2009) and Lu et al. (2010)[2].

Most work on contextual bandits has been theoretical in nature; experimental work on contextual MAB includes Pandey et al. (2007) and Li et al. (2010, 2011)[2].

3. Problem Formalization: Ranked Bandits in Metric Spaces

Let us introduce the online learning-to-rank problem that we study in this paper.

Ranked bandits. Following Radlinski et al. (2008), we are interested in learning an optimally diverse ranking of documents for a given query. We model it as a *ranked bandit* problem as follows. Let X be a set of documents ("arms"). Each 'user' is represented by a binary *relevance vector*: a function $\pi : X \to \{0,1\}$. A document $x \in X$ is called "relevant" to the user if and only if $\pi(x) = 1$. Let $\mathcal{F}_X$ be the set of all possible relevance vectors. Users come from a distribution $\mathcal{P}$ on $\mathcal{F}_X$ that is fixed but not revealed to an algorithm.[3] This $\mathcal{P}$ will henceforth be called the *user distribution*.

In each round, the following happens: a user arrives, sampled independently from $\mathcal{P}$; an algorithm outputs a list of k documents; the user scans this list top-down, and clicks on the first relevant document. The goal is to maximize the expected fraction of *satisfied users*: users who click on at least one document. Note that in contrast with prior work on diversifying existing rankings (e.g., Carbonell and Goldstein, 1998), the algorithm needs to directly learn a diverse ranking.

Since we count satisfied users rather than the clicks themselves, we can assume w.l.o.g. that a user leaves once she clicks once. (Alternatively, the algorithm does not record any subsequent clicks.) A user is satisfied or not satisfied independently of the order in which she scans the results. However, the assumption of the top-down scan determines the feedback received by the algorithm, that is, which document gets clicked.

3. This also models users for whom documents are probabilistically relevant (Radlinski et al., 2008).

We will say that there are *k slots* to be filled in each round, so that when the algorithm outputs the list of k documents, the i-th document in this list appears in slot i. Note that the standard model of MAB with stochastic rewards (e.g., Auer et al., 2002a) is a special case with a single slot ($k = 1$).

Click probabilities. Recall that $\mathcal{P}$ is a distribution over relevance vectors. The *pointwise mean* of $\mathcal{P}$ is a function $\mu : X \to [0,1]$ such that $\mu(x) \triangleq \mathbb{E}_{\pi\sim\mathcal{P}}[\pi(x)]$. Thus, $\mu(x)$ is the click probability for document x if it appears in the top slot.

Each slot $i > 1$ is examined by the user only in the event that all documents in the higher slots are not clicked, so the relevant click probabilities for this slot are conditional on this event. Formally, fix a subset of documents $S \subset X$ and let $Z_S \triangleq \{\pi(\cdot) = 0 \text{ on } S\}$ be the event that all documents in S are not relevant to the user. Let $(\mathcal{P}|Z_S)$ be the distribution of users obtained by conditioning $\mathcal{P}$ on this event, and let $\mu(\cdot\,|Z_S)$ be its pointwise mean. Then $\mu(x\,|Z_S)$ is the click probability for document x if S is the set of documents shown above x in the same round.

Metric spaces. Throughout the paper, let (X,D) be a *metric space*.That is, X is a set and D is a symmetric function on $X \times X \to [0,\infty]$ such that $D(x,y) = 0 \iff x = y$, and $D(x,y) + D(y,z) \geq D(x,z)$ (triangle inequality).

A function $\nu : X \to \mathbb{R}$ is said to be *Lipschitz-continuous* with respect to (X,D) if

$$|\nu(x) - \nu(y)| \leq D(x,y) \qquad \text{for all } x,y \in X. \tag{1}$$

Throughout the paper, we will write *L-continuous* for brevity.

A user distribution $\mathcal{P}$ is called L-continuous with respect to (X,D) if its pointwise mean μ is L-continuous with respect to (X,D).

Document similarity. To allow us to incorporate information about similarity between documents, we start with the model, called *Lipschitz MAB*, proposed by Kleinberg et al. (2008b) for the standard (single-slot) bandits. In this model, an algorithm is given a metric space (X,D) with respect to which the pointwise mean μ is L-continuous.[4]

While this model suffices for learning the document at the top slot (see Kleinberg et al., 2008b for details), it is not sufficiently informative for lower slots. This is because the relevant click probabilities $\mu(\cdot\,|Z_S)$ are conditional and therefore are not directly constrained by L-continuity. To enable efficient learning in all k slots, we will assume a stronger property called *conditional L-continuity*:

Definition 1 *$\mathcal{P}$ is called* conditionally *L-continuous w.r.t. (X,D) if the conditional pointwise mean $\mu(\cdot|Z_S)$ is L-continuous for all $S \subset X$.*

Now, a document x in slot $i > 1$ is examined only if event Z_S happens, where S is the set of documents in the higher slots: that is, if all documents in the higher slots are not relevant to the user. The document x has a conditional click probability $\mu(x|Z_S)$. The function $\mu(\cdot\,|Z_S)$ satisfies the Lipschitz condition (1), which will allow us to use the machinery from MAB problems on metric spaces.

Formally, we define the *k-slot Lipschitz MAB problem*, an instance of which consists of a triple $(X,D,\mathcal{P})$, where (X,D) is a metric space that is known to an algorithm, and $\mathcal{P}$ is a latent user distribution which is conditionally L-continuous w.r.t. (X,D).

4. One only needs to assume that similarity between any two documents x,y is summarized by a number $\delta_{x,y}$ such that $|\mu(x) - \mu(y)| \leq \delta_{x,y}$. Then one obtains a metric space by taking the shortest paths closure.

Note that the k-slot Lipschitz MAB problem subsumes the "metric-free" ranked bandit problem from Radlinski et al. (2008) (as a special case with a trivial metric space in which all distances are equal to 1) and the Lipschitz MAB problem from Kleinberg et al. (2008b) (as a special case with a single slot).

3.1 Metric Space: A Running Example

Web documents are often classified into hierarchies, where closer pairs are more similar.[5] For evaluation, we assume the documents X fall in such a tree, with each document $x \in X$ a leaf in the tree. On this tree, we consider a very natural metric: the distance between any two tree nodes u, v is exponential in the height (i.e., the hop-count distance to the root) of their least common ancestor:

$$D(u,v) = c \times \varepsilon^{\texttt{height}(\texttt{LCA}(u,v))},$$

for some constant c and base $\varepsilon \in (0,1)$. We call this the ε-*exponential tree metric* (with constant c). However, our algorithms and analyses extend to arbitrary metric spaces.

3.2 Alternative Notion of Document Similarity

An alternative notion of document similarity focuses on *correlated relevance*: correlation between the relevance of two documents to a given user. We express "similarity" by bounding the probability of the "discorrelation event" $\{\pi(x) \neq \pi(y)\}$. Specifically, we consider *conditional L-correlation*, defined as follows:

Definition 2 *Call* $\mathcal{P}$ L-correlated *w.r.t.* (X,D) *if*

$$\Pr_{\pi\sim\mathcal{P}}[\pi(x) \neq \pi(y)] \leq D(x,y) \quad \forall x,y \in X. \tag{2}$$

Call $\mathcal{P}$ conditionally L-correlated *w.r.t.* (X,D) *if (2) holds conditional on* Z_S *for any* $S \subset X$*, that is,*

$$\Pr_{\pi\sim(\mathcal{P}|Z_S)}[\pi(x) \neq \pi(y)] \leq D(x,y) \quad \forall x,y \in X, S \subset X.$$

It is easy to see that conditional L-correlation implies conditional L-continuity. In fact, we show that the two notions are essentially equivalent. Namely, we prove that conditional L-continuity w.r.t. (X,D) implies conditional L-correlation w.r.t. $(X,2D)$.

Lemma 3 *Consider an instance* $(X,D,\mathcal{P})$ *of the k-slot Lipschitz MAB problem. Then the user distribution* $\mathcal{P}$ *is conditionally L-correlated w.r.t.* $(X,2D)$.

Proof Fix documents $x,y \in X$ and a subset $S \subset X$. For brevity, write "$x=1$" to mean "$\pi(x)=1$", etc. We claim that

$$\Pr[x=1 \wedge y=0 \,|Z_S] \leq D(x,y). \tag{3}$$

Indeed, consider the event $Z = Z_{S+\{y\}}$. Applying the Bayes theorem to $(\mathcal{P}|Z_S)$, we obtain that

$$\begin{aligned} \mu(x|Z) &= \Pr[x=1 \,|\, \{y=0\} \wedge Z_S] \\ &= \frac{\Pr[x=1 \wedge y=0 \,|Z_S]}{\Pr[y=0 \,|Z_S]}. \end{aligned} \tag{4}$$

5. One example of such hierarchical classification is the Open Directory Project (`http://dmoz.org`).

On the other hand, since $\mu(y|Z) = 0$, by conditional L-continuity it holds that

$$\mu(x|Z) = |\mu(x|Z) - \mu(y|Z)| \leq D(x,y), \tag{5}$$

so claim (3) follows from Equation (4) and Equation (5).

Likewise, $\Pr[x = 0 \wedge y = 1 \,|Z_S] \leq D(x,y)$. Since

$$\{\pi(x) \neq \pi(y)\} = \{x = 1 \wedge y = 0\} \cup \{x = 0 \wedge y = 1\},$$

it follows that $\Pr[\pi(x) \neq \pi(y) \,|Z_S] \leq 2D(x,y)$. ■

4. Expressiveness of the Model

Our approach relies on the conditional L-continuity (equivalently, conditional L-correlation) of the user distribution. How "expressive" is this assumption, that is, how rich and "interesting" is the collection of problem instances that satisfy it? While the unconditional L-continuity assumption is usually considered reasonable from the expressiveness point of view, even the unconditional L-correlation (let alone the conditional L-correlation) is a very non-trivial property about correlated relevance, and thus potentially problematic. A related concern is how to generate a suitable collection of problem instances for simulation experiments.

We address both concerns by defining a natural (albeit highly stylized) generative model for the user distribution, which we then use in the experiments in Section 8. We start with a tree metric space (X,D) and the desired pointwise mean $\mu : X \to (0, \frac{1}{2}]$ that is L-continuous w.r.t. (X,D). The generative model provides a rich family of user distributions that are conditionally L-continuous w.r.t. (X, cD), for some small c. This result is a key theoretical contribution of this paper (and by far the most technical one).

We develop the generative model in Section 4.1. We extend this result to arbitrary metric spaces in Section 4.2, and to distributions over conditionally L-continuous user distributions in Section 4.3. To keep the flow of the paper, the detailed analysis is deferred to Section A and Section B.

4.1 Bayesian Tree Network

The generative model is a tree-shaped Bayesian network with 0-1 "relevance values" $\pi(\cdot)$ on nodes, where leaves correspond to documents. The tree is essentially a topical taxonomy on documents: subtopics correspond to subtrees. The relevance value on each sub-topic is obtained from that on the parent topic via a low-probability mutation.

The mutation probabilities need to be chosen so as to guarantee conditional L-continuity and the desired pointwise mean μ. It is fairly easy to derive a necessary and sufficient condition for the pointwise mean, and a necessary condition for conditional L-continuity. The latter condition states that the mutation probabilities need to be bounded in terms of the distance between the child and the parent. The hard part is to prove that this condition is *sufficient*.

Let us describe our Bayesian tree network in detail. The network inputs a tree metric space (X,D) and the desired pointwise mean μ, and outputs a relevance vector $\pi : X \to \{0,1\}$. Specifically, we assume that documents are leaves of a finite rooted edge-weighted tree, which we denote τ_{d}, with node set V and leaf set $X \subset V$, so that D is a (weighted) shortest-paths metric on V.

Algorithm 1 User distribution for tree metrics

Input: Tree (root r, node set V); $\mu(r) \in [0,1]$
mutation probabilities $q_0, q_1 : V \to [0,1]$
Output: relevance vector $\pi : V \to \{0,1\}$

function AssignClicks(tree node v)
$b \leftarrow \pi(v)$
for each child u of v **do**
$$\pi(u) \leftarrow \begin{cases} 1-b & \text{w/prob } q_b(u) \\ b & \text{otherwise} \end{cases}$$
AssignClicks(u)

Pick $\pi(r) \in \{0,1\}$ at random with expectation $\mu(r)$
AssignClicks(r)

Recall that μ is L-continuous w.r.t. (X,D). We assume that μ takes values in the interval $[\alpha, \frac{1}{2}]$, for some constant parameter $\alpha > 0$. We show that μ can be extended from X to V preserving the range and L-continuity (see Section A for the proof).

Lemma 4 *μ can be extended to V so that $\mu : V \to [\alpha, \frac{1}{2}]$ is L-continuous w.r.t. (V,D).*

In what follows, by a slight abuse of notation we will assume that the domain of μ is V, with the same range $[\alpha, \frac{1}{2}]$, and that μ is L-continuous w.r.t. (V,D). Also, we redefine the relevance vectors to be functions $V \to \{0,1\}$ rather than $X \to \{0,1\}$.

The Bayesian network itself is very intuitive. We pick $\pi(\text{root}) \in \{0,1\}$ at random with a suitable expectation $\mu(\text{root})$, and then proceed top-down so that the child's bit is obtained from the parent's bit via a low-probability mutation. The mutation is parameterized by functions $q_0, q_1 : V \to [0,1]$, as described in Algorithm 1: for each node u, if the parent's bit is set to b then the mutation $\{\pi(u) = 1-b\}$ happens with probability $q_b(u)$. These parameters let us vary the degree of independence between each child and its parent, resulting in a rich family of user distributions.

To complete the construction, it remains to define the mutation probabilities q_0, q_1. Let $\mathcal{P}$ be the resulting user distribution. It is easy to see that μ is the pointwise mean of $\mathcal{P}$ on V if and only if

$$\mu(u) = (1-\mu(v))\, q_0(u) + \mu(v)(1-q_1(u)) \tag{6}$$

whenever u is a child of v. (For sufficiency, use induction on the tree.) Further, letting $q_b = q_b(u)$ for each bit $b \in \{0,1\}$, note that

$$\begin{aligned} \Pr[\pi(u) \neq \pi(v)] &= \mu(v)\, q_1 + (1-\mu(v))\, q_0 \\ &= \mu(v)(q_0+q_1) + (1-2\mu(v))\, q_0 \\ &\geq \mu(v)(q_0+q_1). \end{aligned}$$

Thus, if $\mathcal{P}$ is L-correlated w.r.t. (X,D) then

$$q_0(u) + q_1(u) \leq D(u,v)/\mu(v). \tag{7}$$

We show that (6-7) suffices to guarantee conditional L-continuity.

For a concrete example, one could define

$$(q_0(u), q_1(u)) = \begin{cases} \left(0, \frac{\mu(v)-\mu(u)}{\mu(v)}\right) & \text{if } \mu(v) \geq \mu(u) \\ \left(\frac{\mu(u)-\mu(v)}{1-\mu(v)}, 0\right) & \text{otherwise.} \end{cases} \tag{8}$$

The q_0, q_1 defined as above satisfy (6-7) for any μ that is L-continuous on (V, D).

The provable properties of Algorithm 1 are summarized in the theorem below. It is technically more convenient to state this theorem in terms of L-correlation rather than L-continuity.

Theorem 5 *Let D be the shortest-paths metric of an edge-weighted rooted tree with a finite leaf set X. Let $\mu : X \to [\alpha, \frac{1}{2}]$, $\alpha > 0$ be L-continuous w.r.t. (X, D). Suppose $q_0, q_1 : V \to [0,1]$ satisfy (6-7).*

Let $\mathcal{P}$ be the user distribution constructed by Algorithm 1. Then $\mathcal{P}$ has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3D_\mu)$ where

$$D_\mu(x,y) \triangleq D(x,y) \min\left(\frac{1}{\alpha}, \frac{3}{\mu(x)+\mu(y)}\right).$$

Remark. The theorem can be strengthened by replacing D_μ with the shortest-paths metric induced by D_μ.

Below we provide a proof sketch. The detailed proof is presented in Section B.

Proof Sketch As we noted above, the statement about the pointwise mean trivially follows from Equation (6) using induction on the tree. In what follows we focus on conditional L-correlation.

Fix leaves $x, y \in X$ and a subset $S \subset X$. Let z be the least common ancestor of x, y. Recall that in Algorithm 1 the bit $\pi(\cdot)$ at each node is a random mutation of that of its parent. We focus on the event $\mathcal{E}$ that no mutation happened on the $z \to x$ and $z \to y$ paths. Note that $\mathcal{E}$ implies $\pi(x) = \pi(y) = \pi(z)$. Therefore

$$\Pr[\pi(x) \neq \pi(y) \,|Z_S] \leq \Pr[\bar{\mathcal{E}} \,|Z_S],$$

where $\bar{\mathcal{E}}$ is the negation of $\mathcal{E}$. Intuitively, $\bar{\mathcal{E}}$ is a low-probability "failure event". The rest of the proof is concerned with showing that $\Pr[\bar{\mathcal{E}} \,|Z_S] \leq 3D_\mu(x,y)$.

First we handle the unconditional case. We claim that

$$\Pr[\bar{\mathcal{E}}] \leq D_\mu(x,y). \tag{9}$$

Note that Equation (9) immediately implies that $\mathcal{P}$ is L-correlated w.r.t. (X, D_μ). This claim is not very difficult to prove, essentially since the condition in Equation (7) is specifically engineered to satisfy the unconditional L-correlation property. We provide the proof in detail.

Let $w \in \operatorname{argmin}_{u \in P_{xy}} \mu(u)$, where P_{xy} is the $x \to y$ path. Let $(z = x_0, x_1, \ldots, x_n = x)$ be the $z \to x$ path. For each $i \geq 1$ by Equation (7) the probability of having a mutation at x_i is at most $D(x_i, x_{i-1})/\mu(w)$, so the probability of having a mutation on the $z \to x$ path is at most $D(x,z)/\mu(w)$. Likewise for the $z \to y$ path. So $\Pr[\bar{\mathcal{E}}] \leq D(x,y)/\mu(w) \leq D(x,y)/\alpha$.

It remains to prove that

$$\Pr[\bar{\mathcal{E}}] \leq D(x,y) \frac{3}{\mu(x)+\mu(y)}. \tag{10}$$

Indeed, by L-continuity it holds that

$$\mu(w) \geq \mu(x) - D(x,w),$$
$$\mu(w) \geq \mu(y) - D(y,w).$$

Since $D(x,y) = D(x,w) + D(y,w)$, it follows that

$$\mu(w) \geq \tfrac{\mu(x)+\mu(y)-D(x,y)}{2}. \tag{11}$$

Now, either the right-hand side of Equation (11) is at least $\frac{\mu(x)+\mu(y)}{3}$, or the right-hand side of Equation (10) is at least 1. In both cases Equation (10) holds. This completes the proof of the claim (9).

The conditional case is much more difficult. We handle it by showing that

$$\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq 3 \Pr[\bar{\mathcal{E}}]. \tag{12}$$

In fact, Equation (12) holds even if Equation (7) is replaced with a much weaker bound: $\max(q_0(u), q_1(u)) \leq \frac{1}{2}$ for each u.

The mathematically subtle proof of Equation (12) can be found in Section B. The crux in this proof is that event Z_S is more likely if document z is not relevant to the user:

$$\Pr[Z_S \,|\, z = 0] \geq \Pr[Z_S \,|\, z = 1].$$

■

4.2 Arbitrary Metric Spaces

We can extend Theorem 3.1 to arbitrary metric spaces using prior work on *metric embeddings*. Fix an N-point metric space (X,D) and a function $\mu : X \to [\alpha, \frac{1}{2}]$ that is L-continuous on (X,D). It is known (Bartal, 1996; Fakcharoenphol et al., 2004) that there exists a distribution $\mathcal{P}_{\text{tree}}$ over tree metric spaces $(X,\mathcal{T})$ such that $D(x,y) \leq \mathcal{T}(x,y)$ and

$$\mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}[\mathcal{T}(x,y)] \leq c\, D(x,y) \quad \forall x,y \in X,$$

where $c = O(\log N)$.[6]

Our construction (Algorithm 2) is simple: first sample a tree metric space $(X,\mathcal{T})$ from $\mathcal{P}_{\text{tree}}$, then independently generate a user distribution $\mathcal{P}_{\mathcal{T}}$ for $(X,\mathcal{T})$ as per Algorithm 1.

Theorem 6 *The user distribution $\mathcal{P}$ produced by Algorithm 2 has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3c\,D_\mu)$, where D_μ is given by*

$$D_\mu(x,y) = D(x,y) \min\left(\tfrac{1}{\alpha}, \tfrac{3}{\mu(x)+\mu(y)}\right).$$

6. This is the main result in Fakcharoenphol et al. (2004), which improves on an earlier result in Bartal (1996) with $c = O(\log^2 N)$. For point sets in a d-dimensional Euclidean space one could take $c = O(d \log \frac{1}{\varepsilon})$, where ε is the minimal distance. In fact, this result extends to a much more general family of metric spaces—those of doubling dimension d (Gupta et al., 2003). Doubling dimension, the smallest d such that any ball can be covered by 2^d balls of half the radius, has been introduced to the theoretical computer science literature in Gupta et al. (2003), and has been a well-studied concept since then.

Algorithm 2 User distribution for arbitrary metric spaces

Input: metric space (X,D); function $\mu : X \to [\alpha, \frac{1}{2}]$ that is L-continuous on (X,D).
Output: relevance vector $\pi : X \to \{0,1\}$

1. Sample a tree metric space $(X,\mathcal{T})$ from $\mathcal{P}_{\text{tree}}$,
2. Run Algorithm 1 for $(X,\mathcal{T})$, output the resulting π.

Proof The function μ is L-continuous w.r.t. each tree metric space $(X,\mathcal{T})$, so by Theorem 3.1 user distribution $\mathcal{P}_{\mathcal{T}}$ has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3\mathcal{T}_\mu)$. It follows that the aggregate user distribution $\mathcal{P}$ has pointwise mean μ, and moreover for any $x,y \in X$ and $S \subset X$ we have

$$\begin{aligned}
&\Pr_{\pi\sim\mathcal{P}}[\pi(x) \neq \pi(y) \,|Z_S] \\
&\quad \leq \mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}\left[\Pr_{\pi\sim\mathcal{P}_{\mathcal{T}}}[\pi(x) \neq \pi(y) \,|Z_S]\right] \\
&\quad \leq \mathbb{E}_{\mathcal{T}\sim\mathcal{P}_{\text{tree}}}\left[3\,\mathcal{T}_\mu(x,y)\right] \\
&\quad \leq 3c\,D_\mu(x,y).
\end{aligned}$$

■

4.3 Distributions over User Distributions

Let us verify that conditional L-continuity is *robust*, in the sense that any distribution over conditionally L-continuous user distributions is itself conditionally L-continuous. This result considerably extends the family of user distributions for which we have conditional L-continuity guarantees.

Lemma 7 *Let $\mathcal{P}$ be a distribution over countably many user distributions $\mathcal{P}_i$ that are conditionally L-continuous w.r.t. a metric space (X,D). Then $\mathcal{P}$ is conditionally L-continuous w.r.t. (X,D).*

Proof Let μ and μ_i be the (conditional) pointwise means of $\mathcal{P}$ and $\mathcal{P}_i$, respectively. Formally, let us treat each $\mathcal{P}_i$ as a measure, so that $\mathcal{P}_i(E)$ is the probability of event E under $\mathcal{P}_i$. Let $\mathcal{P} = \sum_i q_i \mathcal{P}_i$, where $\{q_i\}$ are positive coefficients that sum up to 1. Fix documents $x,y \in X$ and a subset $S \subset X$. Then

$$\begin{aligned}
\mu(x|S) = \mathcal{P}(x=1\,|Z_S) &= \frac{\mathcal{P}(x=1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\
&= \frac{\sum_i q_i\, \mathcal{P}_i(x=1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\
&= \frac{\sum_i q_i\, \mathcal{P}_i(Z_S)\, \mu_i(x|Z_S)}{\mathcal{P}(Z_S)}.
\end{aligned}$$

It follows that

$$\begin{aligned}
&|\mu(x|S) - \mu(y|S)| \\
&\quad = \frac{\sum_i q_i \, \mathcal{P}_i(Z_S) \, (\mu_i(x|Z_S) - \mu_i(y|Z_S))}{\mathcal{P}(Z_S)} \\
&\quad \leq \frac{\sum_i q_i \, \mathcal{P}_i(Z_S) \, D(x,y)}{\mathcal{P}(Z_S)} \\
&\quad \leq D(x,y).
\end{aligned}$$

■

5. Algorithms from Prior Work

Let us discuss some algorithmic ideas from prior work that can be adapted to our setting. Interestingly, one can combine these algorithms in a *modular* way, which we make particularly transparent by putting forward a suitable naming scheme. Throughout this section, we let `Bandit` be some algorithm for the MAB problem.

5.1 Ranked Bandits

Given some bandit algorithm `Bandit`, the "ranked" algorithm `RankBandit` for the multi-slot MAB problem is defined as follows (Radlinski et al., 2008). We have k slots (i.e., ranks) for which we wish to find the best documents to present. In each slot i, a separate instance $\mathcal{A}_i$ of `Bandit` is created. In each round these instances select the documents to show independently of one another. If a user clicks on slot i, then this slot receives a reward of 1, and all higher (i.e., skipped) slots $j < i$ receive a reward of 0. For slots $j > i$, the state is rolled back as if this round had never happened (as if the user never considered these documents). If no slot is clicked, then all slots receive a reward of 0.

Let us emphasize that the above approach can be applied to *any* algorithm `Bandit`. In Radlinski et al. (2008), this approach gives rise to algorithms `RankUCB1` and `RankEXP3`, based on MAB algorithms `UCB1` and `EXP3`(Auer et al., 2002a,b). `EXP3` is designed for the *adversarial* setting with no assumptions on how the clicks are generated, which translates into concrete provable guarantees for `RankEXP3`. `UCB1` is geared towards the *stochastic* setting with i.i.d. rewards on each arm, although the per-slot i.i.d. assumption breaks for slots $i > 1$ because of the influence of the higher slots. Nevertheless, in small-scale experiments `RankUCB1` performs much better than `RankEXP3`(Radlinski et al., 2008).

Provable guarantees. Letting T be the number of rounds and `OPT` be the probability of clicking on the optimal ranking, algorithm `RankBandit` achieves

$$\mathbb{E}[\texttt{\#clicks}] \geq (1 - \tfrac{1}{e})\, T \times \texttt{OPT} - k R(T), \tag{13}$$

where $R(T)$ is any upper bound on regret for `Bandit` in each slot (Radlinski et al., 2008; Streeter and Golovin, 2008).

In the multi-slot setting, *performance* of an algorithm up to time T is defined as the time-averaged expected total number of clicks. We will consider performance as a function of T. Assuming $R(T) = o(T)$ in Equation (13), performance of `RankBandit` converges to or exceeds $(1 - \frac{1}{e})$`OPT`.

Convergence to $(1-\frac{1}{e})\texttt{OPT}$ is proved to be worst-case optimal. Thus, as long as $R(T)$ scales well with time, for the document collection sizes that are typical for the application at hand, Radlinski et al. (2008) interpret Equation (13) as a proof of an algorithm's scalability in the multi-slot MAB setting.

`RankBandit` is presented in Radlinski et al. (2008) as the online version of the *greedy algorithm*: an offline fully informed algorithm that selects documents greedily slot by slot from top to bottom. The performance of this algorithm is called the *greedy optimum*,[7] which is equal to $(1-\frac{1}{e})\,\texttt{OPT}$ in the worst case, but for "benign" problem instances it can be as good as `OPT`. The greedy optimum is a more natural benchmark for `RankBandit` than $(1-\frac{1}{e})\texttt{OPT}$. However, results w.r.t. this benchmark are absent in the literature.[8]

5.2 Lipschitz Bandits

Both `UCB1` and `EXP3` are impractical when there are too many documents to explore them all. To alleviate this issue, one can use the similarity information provided by the metric space and the Lipschitz assumption; this setting is called *Lipschitz MAB*.

Below we describe two "metric-aware" algorithms from Kleinberg (2004) and Kleinberg et al. (2008b). Both are well-defined for arbitrary metric spaces, but for simplicity we present them for a special case in which documents are leaves in a *document tree* (denoted τ_{d}) with an ε-exponential tree metric. In both algorithms, a *subtree* is chosen in each round, then a document in this subtree is sampled at random, choosing uniformly at each branch.

Given some bandit algorithm `Bandit`, Kleinberg (2004) define algorithm `GridBandit` for the Lipschitz MAB setting. This algorithm proceeds in phases: in phase i, the depth-i subtrees are treated as "arms", and a fresh copy of `Bandit` is run on these arms.[9] Phase i lasts for $k\varepsilon^{-2i}$ rounds, where k is the number of depth-i subtrees. This meta-algorithm, coupled with an adversarial MAB algorithm such as `EXP3`, is the only algorithm in the literature that takes advantage of the metric space in the adversarial setting. Following Radlinski et al. (2008), we expect `GridEXP3` to be overly pessimistic for our problem, trumped by the corresponding stochastic MAB approaches such as `GridUCB1`.

The "zooming algorithm" (Kleinberg et al., 2008b, Algorithm 3) is a more efficient version of `GridUCB1`: instead of iteratively reducing the grid size in the entire metric space, it *adaptively* refines the grid in promising areas. It maintains a set $\mathcal{A}$ of *active subtrees* which collectively partition the leaf set. In each round the active subtree with the maximal *index* is chosen. The index of a subtree is (assuming stochastic rewards) the best available upper confidence bound on the click probabilities in this subtree. It is defined via the *confidence radius*[10] given (letting T be the time horizon) by

$$\texttt{rad}(\cdot) \triangleq \sqrt{4\log(T)/(1+\#\texttt{samples}(\cdot))}. \tag{14}$$

The algorithm "zooms in" on a given active subtree u (de-activates u and activates all its children) when $\texttt{rad}(u)$ becomes smaller than its *width* $\mathtt{W}(u) \triangleq \varepsilon^{\mathrm{depth}(u)} = \max_{x,x'\in u} D(x,x')$.

7. If due to ties there are multiple "greedy rankings", define the greedy optimum via the *worst* of them.
8. Following the conference publication of this paper, Streeter and Golovin claimed that the techniques in Streeter and Golovin (2008) can be used to extend Equation (13) to the greedy optimum benchmark. If so, then it may be possible to use the same approach to improve our guarantees.
9. As an empirical optimization, previous events can also be replayed to better initialize later phases.
10. The meaning of $\texttt{rad}(\cdot)$ is that w.h.p. the sample average is within $\pm\texttt{rad}(\cdot)$ from the true mean.

Algorithm 3 "Zooming algorithm" in trees

initialize (document tree τ_d)**:**
$\mathcal{A} \leftarrow \emptyset$; activate($\texttt{root}(\tau_d)$)

activate($u \in \texttt{nodes}(\tau_d)$)**:**
$\mathcal{A} \leftarrow \mathcal{A} \cup \{u\}$; $n(u) \leftarrow 0$; $r(u) \leftarrow 0$

Main loop:
$u \leftarrow \text{argmax}_{u \in \mathcal{A}}\, \texttt{index}(u)$,
where $\texttt{index}(u) = \frac{r(u)}{n(u)} + 2\,\texttt{rad}(u)$
"Play" a random document from $\texttt{subtree}(u)$
$r(u) \leftarrow r(u) + \{\text{reward}\}$; $n(u) \leftarrow n(u) + 1$
if $\texttt{rad}(u) < \texttt{W}(u)$ **then**
deactivate u: remove u from $\mathcal{A}$
activate all children of u

Provable guarantees. Regret guarantees for the two algorithms above are independent of the number of arms (which, in particular, can be infinite). Instead, they depend on the covering properties of the metric space (X,D). A crucial notion here is the *covering number* $N_r(X)$, defined as the minimal number of balls of radius r sufficient to cover X. It is often useful to summarize the covering numbers $N_r(X)$, $r > 0$ with a single number called the *covering dimension*:

$$\texttt{CovDim}(X,D) \triangleq \inf\{d \geq 0 : N_r(X) \leq \alpha r^{-d} \quad \forall r > 0\}. \tag{15}$$

(Here $\alpha > 0$ is a constant which we will keep implicit in the notation.) In particular, for an arbitrary point set in $\mathbb{R}^d$ under the standard (ℓ_2) distance, the covering dimension is d, for some $\alpha = O(1)$. For an ε-exponential tree metric with maximal branching factor b, the covering dimension is $d = \log_{1/\varepsilon}(b)$, with $\alpha = 1$.

Against an oblivious adversary, GridEXP3 has regret

$$R(T) = \tilde{O}(\alpha T^{(d+1)/(d+2)}), \tag{16}$$

where d is the covering dimension of (X,D).

For the stochastic setting, GridUCB1 and the zooming algorithm enjoy strong instance-dependent regret guarantees. These guarantees reduce to Equation (16) in the worst case, but are much better for "nice" problem instances. Informally, regret guarantees improve for problem instances in which the set of near-optimal arms has smaller covering numbers than the set of all arms. Regret guarantees for the zooming algorithm are (typically) much stronger than for GridUCB1. In particular, one can derive a version of Equation (16) with a different d called the *zooming dimension*, which is equal to the covering dimension in the worst case but can be much smaller, even $d = 0$. These issues are further discussed in Appendix C.

5.3 Anytime Guarantees and the Doubling Trick

While the zooming algorithm, and also the contextual zooming algorithm from Section 5.5, are defined for a fixed time horizon, one can obtain the corresponding *anytime* versions using a simple *doubling trick*: in each phase $i \in \mathbb{N}$, run a fresh instance of the algorithm for 2^i rounds. These

versions are run indefinitely and enjoy the same asymptotic upper bounds on regret as the original algorithms (but now these bounds hold for each round).

5.4 Ranked Bandits in Metric Spaces

Using and combining the algorithms in the previous two subsections, we obtain the following battery of algorithms for k-slot Lipschitz MAB problem:

- metric-oblivious algorithms: `RankUCB1` and `RankEXP3`.
- simple metric-aware algorithms: `RankGridUCB1` and `RankGridEXP3` (ranked versions of `GridUCB1` and `GridEXP3`, respectively).
- `RankZoom`: the ranked version of the zooming algorithm.

In theory, `RankGridEXP3` scales to large document collections, in the sense that it achieves Equation (13) with $R(T)$ that does not degenerate with #documents:

Theorem 8 *Consider the k-slot Lipschitz MAB problem on a metric space with covering dimension d (as defined in Equation (15), with constant α). Then after T rounds* `RankGridEXP3` *achieves*

$$\frac{\mathbb{E}[\texttt{\#clicks}]}{T} \geq (1-\tfrac{1}{e})\,\texttt{OPT} - \tilde{O}\left(\frac{\alpha k}{T^{1/(d+2)}}\right).$$

The theorem follows from the respective regret bounds for `GridEXP3` (Equation (16)) and `RankBandit` (Equation (13)). We do not have any provable guarantees for other algorithms because the corresponding regret bounds for the single-slot setting do not directly plug into Equation (13). However, the strong instance-dependent guarantees for `GridUCB1` and especially for the zooming algorithm (even though they do not directly apply to the ranked bandit setting) suggest that `RankGridUCB1` and `RankZoom` are promising. We shall see that these two algorithms perform much better than `RankGridEXP3` in the experiments.

5.5 Contextual Lipschitz Bandits

We also leverage prior work on contextual bandits. The relevant contextual MAB setting, called contextual Lipschitz MAB, is as follows. In each round nature reveals a *context* h, an algorithm chooses a document x, and the resulting reward is an independent $\{0,1\}$ sample with expectation $\mu(x|h)$. Further, one is given *similarity information*: metrics D and D_{c} on documents and contexts, respectively, such that for any two documents x,x' and any two contexts h,h' we have

$$|\mu(x|h)-\mu(x'|h')| \leq D(x,x')+D_{\mathrm{c}}(h,h').$$

Let X_{c} be the set of contexts, and $X_{\mathrm{dc}} = X \times X_{\mathrm{c}}$ be the set of all (document, context) pairs. Abstractly, one considers the metric space $(X_{\mathrm{dc}}, D_{\mathrm{dc}})$, henceforth the *DC-space*, where the metric is

$$D_{\mathrm{dc}}((x,h),(x',h')) = D(x,x')+D_{\mathrm{c}}(h,h').$$

We will use the "contextual zooming algorithm" (`ContextZoom`) from Slivkins (2009). This algorithm is well-defined for arbitrary D_{dc}, but for simplicity we will state it for the case when D and D_{c} are ε-exponential tree metrics.

Let us assume that documents and contexts are leaves in a document tree τ_{d} and context tree τ_{c}, respectively. The algorithm (see Algorithm 4 for pseudocode) maintains a set $\mathcal{A}$ of *active strategies*

Algorithm 4 `ContextZoom` in trees

initialize (document tree τ_d, context tree τ_c)**:**
$\mathcal{A} \leftarrow \emptyset$; activate($\texttt{root}(\tau_d)$, $\texttt{root}(\tau_c)$)

activate ($u \in \texttt{nodes}(\tau_d)$, $u_c \in \texttt{nodes}(\tau_c)$)**:**
$\mathcal{A} \leftarrow \mathcal{A} \cup \{(u,u_c)\}$; $n(u,u_c) \leftarrow 0$; $r(u,u_c) \leftarrow 0$

Main loop:
Input a context $h \in \texttt{nodes}(\tau_c)$
$(u,u_c) \leftarrow \operatorname{argmax}_{(u,u_c)\in\mathcal{A}:\, h\in u_c} \texttt{index}(u,u_c)$,
where $\texttt{index}(u,u_c) = \texttt{W}(u\times u_c) + \frac{r(u,u_c)}{n(u,u_c)} + \texttt{rad}(u,u_c)$
"Play" a random document from $\texttt{subtree}(u)$
$r(u,u_c) \leftarrow r(u,u_c) + \{\text{reward}\}$; $n(u,u_c) \leftarrow n(u,u_c)+1$
if $\texttt{rad}(u,u_c) < \texttt{W}(u,u_c)$ **then**
deactivate (u,u_c): remove (u,u_c) from $\mathcal{A}$
activate all pairs (child(u), child(u_c))

of the form (u,u_c), where u is a subtree in τ_d and u_c is a subtree in τ_c. At any given time the active strategies partition X_{dc}. In each round, a context h arrives, and one of the active strategies (u,u_c) with $h \in u_c$ is chosen: namely the one with the maximal *index*, and then a document $x \in u$ is picked uniformly at random. The index of (u,u_c) is, essentially, the best available upper confidence bound on expected rewards from choosing a document $x \in u$ given a context $h \in u_c$. The index is defined via sample average, confidence radius (14), and "width" $\texttt{W}(u\times u_c)$. The latter can be any upper bound on the diameter of the product set $u \times u_c$ in the DC-space:

$$\texttt{W}(u,u_c) \geq \max_{x,x'\in u,\ h,h'\in u_c} D(x,x') + D_c(h,h'). \tag{17}$$

The (de)activation rule ensures that the active strategies form a finer partition in the regions of the DC-space that correspond to higher rewards and more frequently occurring contexts.

Provable guarantees. The provable guarantees for the contextual MAB problem are in terms of *contextual regret*, which is regret is with respect to a much stronger benchmark: the best arm in hindsight *for every given context*.

Regret guarantees for `ContextZoom` focus on the *DC-space* (X_{dc}, D_{dc}). A very pessimistic regret bound is Equation (16) with $d = \texttt{CovDim}(X_{dc}, D_{dc})$. However, as for the zooming algorithm, much better instance-dependent bounds are possible. See Appendix C for further discussion.

6. New Approach: Ranked Contextual Bandits

We now present a new approach in which the upper slot selections are taken into account as a *context* in the contextual MAB setting.

The slot algorithms in the `RankBandit` setting can make their selections sequentially. Then without loss of generality each slot algorithm $\mathcal{A}_i$ knows the set S of documents in the upper slots. We propose to treat S as a "context" to $\mathcal{A}_i$. Specifically, $\mathcal{A}_i$ will assume that none of the documents in S is clicked, that is, event Z_S happens (else the i-th slot is ignored by the user). For each such round, the click probabilities for $\mathcal{A}_i$ are given by $\mu(\cdot\,|Z_S)$, which is an L-continuous function on (X,D).

6.1 `RankCorrZoom`: "Light-weight" Ranked Contextual Algorithm

We first propose a simple modification to `RankZoom`, called `RankCorrZoom`, which uses the contexts as discussed above.

Recall that in the zooming algorithm, the index of an active subtree u is defined so that, assuming stochastic rewards, it is an upper confidence bound on the click probability of any document x in this subtree:

$$\texttt{w.h.p. } \texttt{index}(u) \geq \max_{x \in u} \mu(x). \tag{18}$$

Moreover, it follows from the analysis in (Kleinberg et al., 2008b) that performance of the algorithm improves if the index is decreased as long as Equation (18) holds.

Now consider `RankZoom`, and let $\mathcal{A}_i$ be the instance of the zooming algorithm in slot $i \geq 2$. While for $\mathcal{A}_i$ the rewards are no longer stochastic, our intuition for why `RankZoom` may be a good algorithm is still based on Equation (18). In other words, we *wish* that for each context $S \subset X$ we have

$$\texttt{w.h.p. } \texttt{index}(u) \geq \max_{x \in u} \mu(x|Z_S), \tag{19}$$

and our intuition is that it is desirable to decrease the index as long as Equation (19) holds.

We will derive an upper bound on $\max_{x \in u} \mu(x|Z_S)$ using correlation between u and S, and we will cap the index of u at this quantity. Since $\mu(y|Z_S) = 0$ for any $y \in S$, we have

$$\begin{aligned} &\mu(x|Z_S) = |\mu(x|Z_S) - \mu(y|Z_S)| \leq D(x,y), \quad \forall y \in S \\ &\mu(x|Z_S) \leq D(x,S) \triangleq \min_{y \in S} D(x,y). \end{aligned} \tag{20}$$

In other words, if document x is close to some document in S, the event Z_S limits the conditional probability $\mu(x|Z_S)$. Therefore we can cap the index of u at $\max_{x \in u} D(x,S)$:

$$\texttt{index}(u) \leftarrow \min\left(\texttt{index}(u),\ \max_{x \in u} D(x,S)\right).$$

The version of `RankZoom` with the above "correlation rule" will be called `RankCorrZoom`.

To simplify the computation of $\max_{x \in u} D(x,S)$ in an ε-exponential tree metric, we note that it is equal to $D(\texttt{root}(u), S)$ if u is disjoint with S, and in general it is equal to $D(\texttt{root}(v), S)$, where v is the largest subtree of u that is disjoint with S.

6.2 Contextual *Lipschitz* MAB Interpretation

Let us cast each slot algorithm $\mathcal{A}_i$ as a contextual algorithm in the contextual *Lipschitz* MAB setting (as defined in Section 5.5). We need to specify a metric D_c on contexts $S \subset X$ which can be computed by the algorithm and satisfies the Lipschitz condition:

$$|\mu(x|Z_S) - \mu(x|Z_{S'})| \leq D_c(S,S') \quad \text{for all } x \in X \text{ and } S, S' \subset X. \tag{21}$$

Lemma 9 *Consider the k-slot Lipschitz MAB problem. For any $S, S' \subset X$, define*

$$D_c(S,S') \triangleq 4 \inf \textstyle\sum_{j=1}^{n} D(x_j, x'_j), \tag{22}$$

where the infimum is taken over all $n \in \mathbb{N}$ and over all n-element sequences $\{x_j\}$ and $\{x'_j\}$ that enumerate, possibly with repetitions, all documents in S and S'. Then D_c satisfies Equation (21).

Proof For shorthand, let us write

$$\sigma(x|S) \triangleq 1-\mu(x|Z_S),$$
$$\sigma(x|S,y) \triangleq \sigma(x|S\cup\{y\}).$$

First, we claim that for any $y \in X$ and $y' \in S$

$$|\sigma(x|S,y)-\sigma(x|S,y')| \leq 4D(y,y'). \tag{23}$$

Indeed, noting that $\sigma(x|S,y) = \sigma(y|S,x)\ \frac{\sigma(x|S)}{\sigma(y|S)}$, we can re-write the left-hand side of Equation (23) as

$$\begin{aligned}\text{LHS}(23) &= \sigma(x,S)\left|\frac{\sigma(y|S,x)}{\sigma(y|S)}-\frac{\sigma(y'|S,x)}{\sigma(y'|S)}\right| \\ &\leq \sigma(x,S)\ D(y,y')\ \frac{\sigma(y|S)+\sigma(y|S,x)}{\sigma(y|S)\,\sigma(y'|S)} \\ &= D(y,y')\ \frac{\sigma(x|S)+\sigma(x|S,y)}{\sigma(y'|S)} \leq 2D(y,y').\end{aligned} \tag{24}$$

In Equation (24), we have used the L-continuity of $\sigma(\cdot|S)$ and $\sigma(\cdot|S,x)$. To achieve the constant of 2, it was crucial that $y' \in S$, so that $\sigma(y'|S) = 1$. This completes the proof of Equation (23).

Fix some $n \in \mathbb{N}$ and some n-element sequences $\{x_i\}$ and $\{x'_i\}$ that enumerate, possibly with repetitions, all values in S and S', respectively. Consider sets

$$S_i = \{x'_1, \ldots, x'_i\} \cup \{x_{i+1}, \ldots, x_n\},\ \ 1 \leq i \leq n-1,$$

and let $S_0 = S$ and $S_{n+1} = S'$. To prove the lemma, it suffices to show that

$$|\sigma(x|S_i)-\sigma(x|S_{i+1})| \leq 4D(x_{i+1}, x'_{i+1}) \tag{25}$$

for each $i \leq n$. To prove Equation (25), fix i and let $y = x_{i+1}$ and $y' = x'_{i+1}$. Note that $S_i \cup \{y'\} = S_{i+1} \cup \{y\}$, call this set S^*. Then using Equation (23) (note, $y \in S_i$ and $y' \in S'_i$) we obtain

$$\begin{aligned}|\sigma(x|S_i)-\sigma(x|S^*)| &= |\sigma(x|S_i,y)-\sigma(x|S_i,y')| \\ &\leq 2D(y,y'), \\ |\sigma(x|S_{i+1})-\sigma(x|S^*)| &= |\sigma(x|S_{i+1},y')-\sigma(x|S_{i+1},y)| \\ &\leq 2D(y,y'),\end{aligned}$$

which implies Equation (25). ■

6.3 `RankContextZoom`: "Full-blown" Ranked Contextual Algorithm

Now we can take any algorithm for the contextual Lipschitz MAB problem (with metric D_c on contexts given by Equation (22)), and use it as a slot algorithm. We will use `ContextZoom`, augmented by the "correlation rule" similar to the one in Section 6.1. The resulting "ranked" algorithm will be called `RankContextZoom`.

The implementation details are not difficult. Suppose the metric space on documents is the ε-exponential tree metric, and let τ_d be the document tree. Consider slot $(i+1)$-th slot, $i \geq 1$.[11] Then the contexts are unordered i-tuples of documents. Let us define *context tree* τ_c as follows. Depth-ℓ nodes of τ_c are unordered i-tuples of depth-ℓ nodes from τ_d, and leaves are contexts. The root of τ_c is $(r \dots r)$, where $r = \texttt{root}(\tau_d)$. For each internal node $u_c = (u_1 \dots u_i)$ of τ_c, its children are all unordered tuples $(v_1 \dots v_i)$ such that each v_j is a child of u_j in τ_d. This completes the definition of τ_c. Letting u and u_c be level-ℓ subtrees of τ_d and τ_c, respectively, it follows from the definition of D_c in Equation (22) that $D_c(S,S') \leq 4i\varepsilon^\ell$ for any contexts $S, S' \in u_c$. Thus setting $\mathbb{W}(u \times u_c) \triangleq \varepsilon^\ell(4i+1)$ satisfies Equation (17).

We define the "correlation rule" as follows. Let (u, u_c) be an active strategy in the execution of `ContextZoom`, where u is a subtree of the document tree τ_d, and u_c is a subtree of the context tree τ_c. It follows from the analysis in (Slivkins, 2009) that decreasing the index of (u, u_c) improves performance, as long it holds that

$$\texttt{index}(u, u_c) \geq \mu(x|Z_S), \quad \forall x \in u, S \in u_c.$$

Recall that $\mu(x|Z_S) \leq D(x,S)$ by Equation (20), so we can cap $\texttt{index}(u, u_c)$ at $\max_{x \in u} D(x|S)$:

$$\texttt{index}(u,S) \leftarrow \min\left(\texttt{index}(u,S),\ \max_{x \in u} D(x|S)\right).$$

This completes the description of `RankContextZoom`.

7. Provable Scalability Guarantees and Discussion

Noting that for each slot $i \geq k$ the covering dimension of the DC-space is at most k times the covering dimension of (X,D), it follows that a (very pessimistic) upper bound on contextual regret of `RankContextZoom` is $R(T) = \tilde{O}(\alpha T^{1-1/(kd+2)})$. Plugging this into Equation (13), we obtain:

Theorem 10 *Consider the k-slot Lipschitz MAB problem on a metric space with covering dimension d (as defined in Equation (15), with constant α). Then after T rounds algorithm* `RankContextZoom` *achieves*

$$\frac{\mathbb{E}[\texttt{\#clicks}]}{T} \geq (1 - \tfrac{1}{e})\,\texttt{OPT} - \tilde{O}\left(\frac{\alpha k}{T^{1/(kd+2)}}\right).$$

This is just a basic scalability guarantee which does not degenerate with the number of documents. (Note that it is *worse* than the one for `RankGridEXP3`.) We believe that this guarantee is very pessimistic, as it builds on a very pessimistic version of the result for `ContextZoom`. In particular, we ignore the intuition that for a given slot, contexts $S \subset X$ may gradually converge over time to the greedy optimum, which effectively results in a much smaller set of possible contexts.[12] We believe this effect is very important to the performance `RankContextZoom`. In particular, it causes `RankContextZoom` to perform much better than `RankGridEXP3` in simulations.

11. For slot 1, contexts are empty, so `ContextZoom` reduces to Algorithm 3.
12. It is also wasteful (but perhaps less so) that we use a slot-k bound for each slot $i < k$.

7.1 A Better Benchmark

Recall that while the bound in Equation (13) uses $(1-\frac{1}{e})\,\mathrm{OPT}$ as a benchmark, a more natural benchmark would be the greedy optimum. We provide a preliminary convergence result for RankContextZoom, without any specific regret bounds.

Such result is more elegantly formulated in terms of a version of RankContextZoom, henceforth called anytime-RankContextZoom, which uses the anytime version of ContextZoom (see Section 5.3).

Theorem 11 *Fix an instance of the k-slot MAB problem. The performance of anytime-*RankContextZoom *up to any given time t is equal to the greedy optimum minus $f(t)$ such that $f(t) \to 0$.*

Proof Sketch It suffices to prove that with high probability, anytime-RankContextZoom outputs a greedy ranking in all but $f_k(t)$ rounds among the first t rounds, where $f_k(t) \to 0$.

We prove this claim by induction on k, the number of slots. Suppose it holds for some $k-1$ slots, and focus on the k-th slot. Consider all rounds in which a greedy ranking is chosen for the upper slots but not for the k-th slot. In each such round, the k-th slot replica of anytime-ContextZoom incurs contextual regret at least δ_k, for some instance-specific constant $\delta_k > 0$. Thus, with high probability there can be at most $R_k(t)/\delta_k$ such rounds, where $R_k(t) = o(t)$ is an upper bound on contextual regret for slot k. Thus, one can take $f_k(t) = f_{k-1}(t) + R_k(t)/\delta_k$. ■

Theorem 11 is about the "metric-less" setting from Radlinski et al. (2008). It easily extends to the "ranked" version of any bandit algorithm whose contextual regret is sublinear with high probability.

It is an open question whether (and under which assumptions) Theorem 11 can be extended to the "ranked" versions of non-contextual bandit algorithms such as RankUCB1. One assumption that appears essential is the uniqueness of the greedy ranking. To see that multiple greedy rankings may cause problems for ranked non-contextual algorithms, consider a simple example:

- There are two slots and three documents x_1, x_2, x_3 such that $\mu = (\frac{1}{2}, \frac{1}{2}, \frac{1}{3})$ and the relevance of each arm is independent of that of the other arms.[13]

An optimal ranking for this example is a greedy ranking that puts x_1 and x_2 in the two slots, achieving aggregate click probability $\frac{3}{4}$. According to our intuition, a "reasonable" ranked non-contextual algorithm will behave as follows. The slot 1 algorithm will alternate between x_1 and x_2, each with frequency $\to \frac{1}{2}$. Since the slot-2 algorithm is oblivious to the slot 1 selection, it will observe averages that converge over time to $(\frac{1}{4}, \frac{1}{4}, \frac{1}{3})$,[14] so it will select document x_3 with frequency $\to 1$. Therefore frequency $\to 1$ the ranked algorithm will alternate between (x,z) or (y,z), each of which has aggregate click probability $\frac{2}{3}$.

13. Here documents x_1, x_2, x_3 can stand for disjoint *subsets* of documents with highly correlated payoffs. Documents within a given subset can lie far from one another in the metric space.

14. Suppose x_j, $j \in \{1,2\}$ is chosen in slot 1. Then, letting $S = \{x_j\}$, $\mu(x_1|Z_S)$ equals 0 if $j = 1$ and $\frac{1}{2}$ otherwise (which averages to $\frac{1}{4}$), whereas $\mu(x_3|Z_S) = \frac{1}{3}$.

RankUCB1 RankEXP3	metric-oblivious algorithms: ranked versions of UCB1 and EXP3	Section 5.1
RankGridUCB1 RankGridEXP3	simple metric-aware algorithms: ranked versions of GridUCB1 and GridEXP3	Section 5.4
RankZoom	the ranked version of the zooming algorithm	Section 5.4
	contextual algorithms:	
RankCorrZoom	"light-weight"(based on the zooming algorithm)	Section 6.1
RankContextZoom	"full-blown" (based on ContextZoom).	Section 6.3

Table 1: Algorithms for the k-slot Lipschitz MAB problem.

7.2 Desiderata

We believe that the above guarantees do not reflect the full power of our algorithms, and more generally the full power of conditional L-continuity. The "ideal" performance guarantee for RankBandit in our setting would use the greedy optimum as a benchmark, and would have a bound on regret that is free from the inefficiencies outlined in the discussion after Theorem 10. Furthermore, this guarantee would only rely on some general property of Bandit such as a bound on regret or contextual regret. We conjecture that such guarantee is possible for RankContextZoom, and, perhaps under some assumptions, also for RankCorrZoom and RankZoom.

Further, one would like to study the relative benefits of the new "contextual" algorithms (RankContextZoom and RankCorrZoom) and the prior work such as RankZoom. The discussion Section 7.1 suggests that the difference can be particularly pronounced when the pointwise mean has multiple peaks of similar value. In fact, we confirm this experimentally in Section 8.4.

8. Evaluation

Let us evaluate the performance of the algorithms presented in Section 5 and Section 6. We summarize these algorithms in Table 8.

In all UCB1-based algorithms in Table 8, including all extensions of the zooming algorithm, one can damp exploration by replacing the $4\log(T)$ factor in Equation (14) with 1. Such change effectively makes the algorithm more *optimistic*; it was found beneficial for RankUCB1 by Radlinski et al. (2008). We find (see Section 8.3) that this change greatly improves the average performance in our experiments. So, by a slight abuse of notation, we will assume this change from now on.

8.1 Experimental Setup

Using the generative model from Section 4 (Algorithm 1 with Equation (8)), we created a document collection with $|X| = 2^{15} \approx 32,000$ documents[15] in a binary ε-exponential tree metric space with $\varepsilon = 0.837$ (and constant $c = 1$, see Section 3.1). The value for ε was chosen so that the most dissimilar documents in the collection still have a non-trivial similarity, as may be expected for web documents. Each document's expected relevance $\mu(x)$ was set by first identifying a small number

15. This is a realistic number of documents that may be considered in detail for a typical web search query after pruning very unlikely documents.

of "peaks" $y_i \in X$, choosing $\mu(\cdot)$ for these documents, and then defining the relevance of other documents as the minimum allowed while obeying L-continuity and a background relevance rate μ_0:

$$\mu(x) \triangleq \max(\mu_0,\ \tfrac{1}{2} - \min_i D(x, y_i)). \tag{26}$$

For internal nodes in the tree, μ is defined bottom-up (from leaves to the root) as the mean value of all children nodes. As a result, we obtain a set of documents X where each document $x \in X$ has an expected click probability $\mu(x)$ that obeys L-continuity.

Our simulation was run over a 5-slot ranked bandit setting, learning the best 5 documents. We evaluated over 300,000 user visits sampled from $\mathcal{P}$ per Algorithm 1. Performance within 50,000 impressions, typical for the number of times relatively frequent queries are seen by commercial search engines in a month, is essential for any practical applicability of this approach. However, we also measure performance for a longer time period to obtain a deeper understanding of the convergence properties of the algorithms.

We consider two models for $\mu(\cdot)$ in Equation (26). In the first model, two "peaks" $\{y_1, y_2\}$ are selected at random with $\mu(\cdot) = \frac{1}{2}$, and μ_0 set to 0.05. The second model is less "rigid" (and thus more realistic): the relevant documents y_i and their expected relevance rates $\mu(\cdot)$ are selected according to a Chinese Restaurant Process (Aldous, 1985) with parameters $n = 20$ and $\theta = 2$, and setting $\mu_0 = 0.01$. The Chinese Restaurant Process is inspired by customers coming in to a restaurant with an infinite number of tables, each with infinite capacity. At time t, a customer arrives and can choose to sit at a new table with probability $\theta/(t-1+\theta)$, and otherwise sits at an already occupied table with probability proportional to the number of customers already sitting at that table. By considering each table as equivalent to a peak in the distrubtion, this leads to a set of peaks with expected relevance rates distributed accoring to a power law. Following Radlinski et al. (2008), we assign users to one of the peaks, then select relevant documents so as to obey the expected relevance rate $\mu(x)$ for each document x.

As baselines we use an algorithm ranking the documents at random, and the (offline) greedy algorithm discussed in Section 5.1.

8.2 Main Experimental Results

Our experimental results are summarized in Figure 1 and Figure 2.

`RankEXP3` and `RankUCB1` perform as poorly as picking documents randomly: the three curves are indistinguishable. This is due to the large number of available documents and slow convergence rates of these algorithms. Other algorithms that explore all strategies (such as REC Radlinski et al., 2008) would perform just as poorly. This result is consistent with results reported by Radlinski et al. (2008) on just 50 documents. On the other hand, algorithms that progressively refine the space of strategies explored perform much better.

`RankCorrZoom` achieves the best empirical performance, converging rapidly to near-optimal rankings. `RankZoom` is a close second. The theoretically preferred `RankContextZoom` comes third, with a significant gap. This appears to be due to the much larger branching factor in the strategies activated by `RankContextZoom` slowing down the convergence. (However, as we investigate in Section 8.4, `RankContextZoom` may significantly outperform the other algorithms if μ has multiple peaks with similar values.)

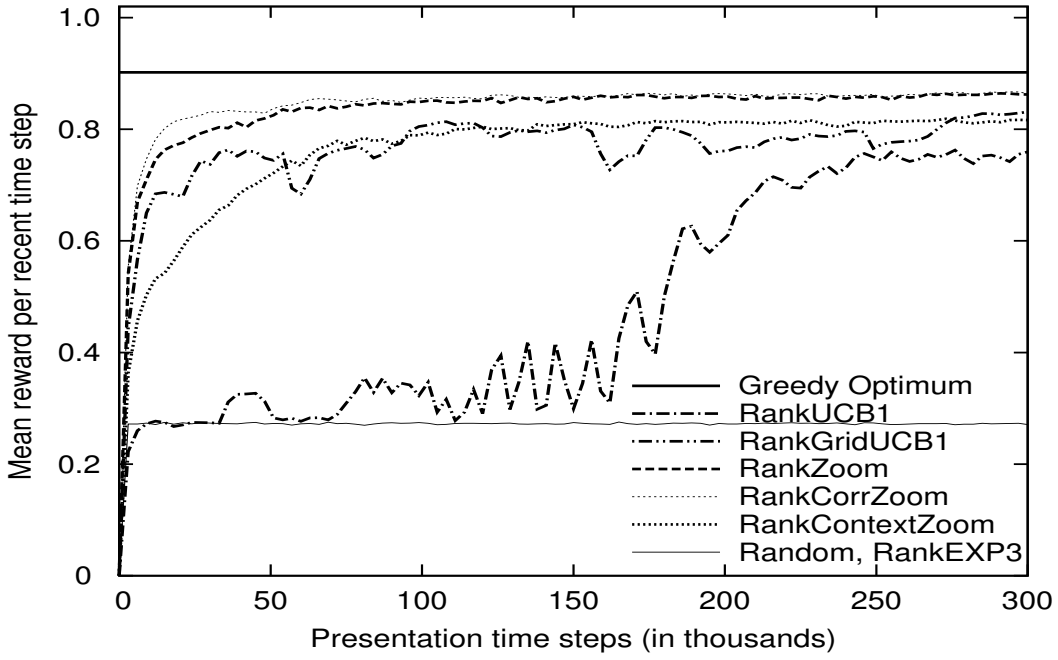


Figure 1: The learning algorithms on 5-slot problem instances with two relevance peaks.

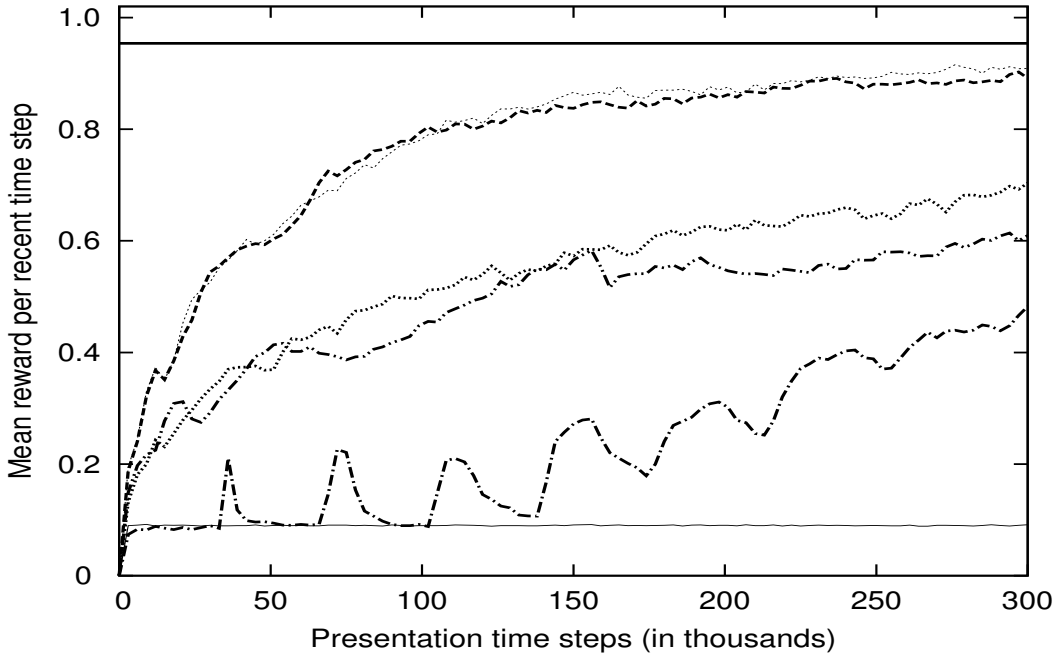


Figure 2: The learning algorithms on 5-slot problem instances with random relevance rates $\mu(\cdot)$ selected according to the Chinese Restaurant Process.

8.3 "Optimistic" vs. "Pessimistic" UCB1-style Algorithms

We find that the "optimistic" UCB1-style algorithms (obtained by replacing the $4\log(T)$ factor in Equation (14) with 1) perform dramatically better than their "pessimistic" counterparts. In Figure 3 and Figure 4 we compare RankUCB1 and RankZoom with their respective "pessimistic" versions (which are marked with a "- -" after the algorithm name). We saw a similar increase in performance for other UCB1-style algorithms, too.

8.4 Secondary Experiment

As discussed in Section 7.1, some RankBandit-style algorithms may converge to a suboptimal ranking if μ has multiple peaks with similar values. To investigate this, we designed a small-scale experiment presented in Figure 5. We generated a small collection of 128 documents using the same setup with two "peaks", and assumed 2 slots. Each peak corresponds to a half of the user population, with peak value $\mu = \frac{1}{2}$ and background value $\mu_0 = 0.05$.

We see that RankContextZoom converges more slowly than the other zooming variants, but eventually outperforms them. This confirms our intuition, and suggests that RankContextZoom may eventually outperform the other algorithms on a larger collection, such as that used for Figures 1 and 2.

9. Further Directions

This paper initiates the study of bandit learning-to-rank with side information on similarity between documents, focusing on an idealized model of document similarity based on the new notion of "conditional Lipschitz-continuity". As discussed in Section 7, we conjecture that provable performance guarantees can be improved significantly. On the experimental side, future work will include evaluating the model on web search data, and designing sufficiently memory- and time-efficient implementations to allow experiments on real users. An interesting challenge in such an endeavor would be to come up with effective similarity measures. A natural next step would be to also exploit the similarity between search queries.

Appendix A. Proof of Lemma 4 (Extending μ from Leaves to Tree Nodes)

Recall that Lemma 4 is needed to define the generative model in Section 4. We will prove a slightly more general statement:

Lemma 12 *Let D be the shortest-paths metric of an edge-weighted rooted tree with node set V and leaf set X. Let $\mu : X \to [a,b]$ be an L-continuous function on (X,D). Then μ can be extended to V so that $\mu : V \to [a,b]$ is L-continuous w.r.t. (V,D).*

Proof For each $x \in V$, let $\mathcal{L}(x)$ be the set of all leaves in the subtree rooted at x. For each $z \in \mathcal{L}(y)$ the assignment $\mu(x)$ should satisfy

$$\mu(z) - D(x,z) \leq \mu(x) \leq \mu(z) + D(x,z)$$

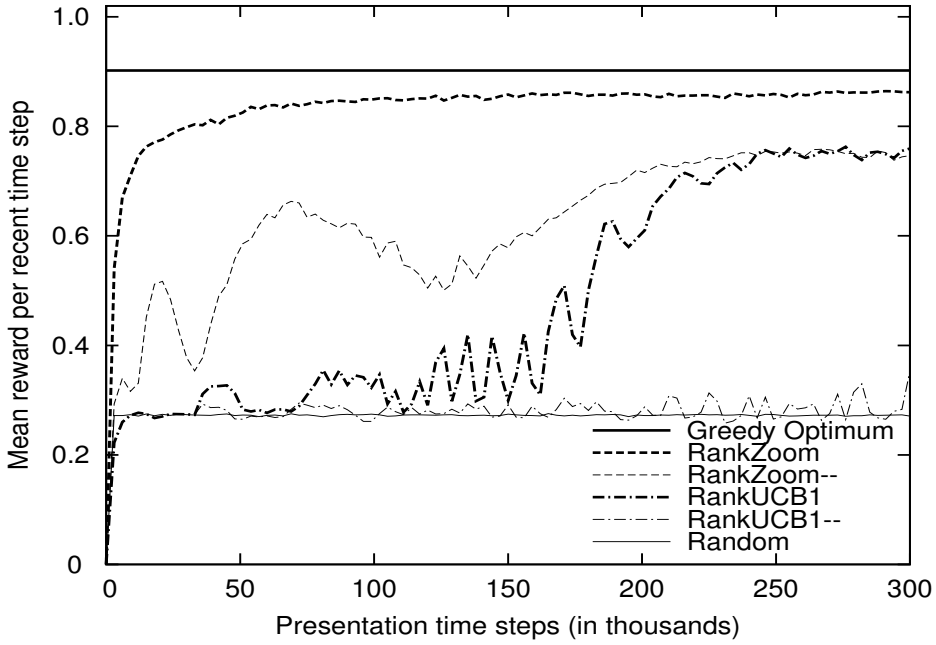


Figure 3: "Optimistic" vs. "pessimistic" UCB1-style algorithms:
The learning algorithms on 5-slot problem instances with two relevance peaks.

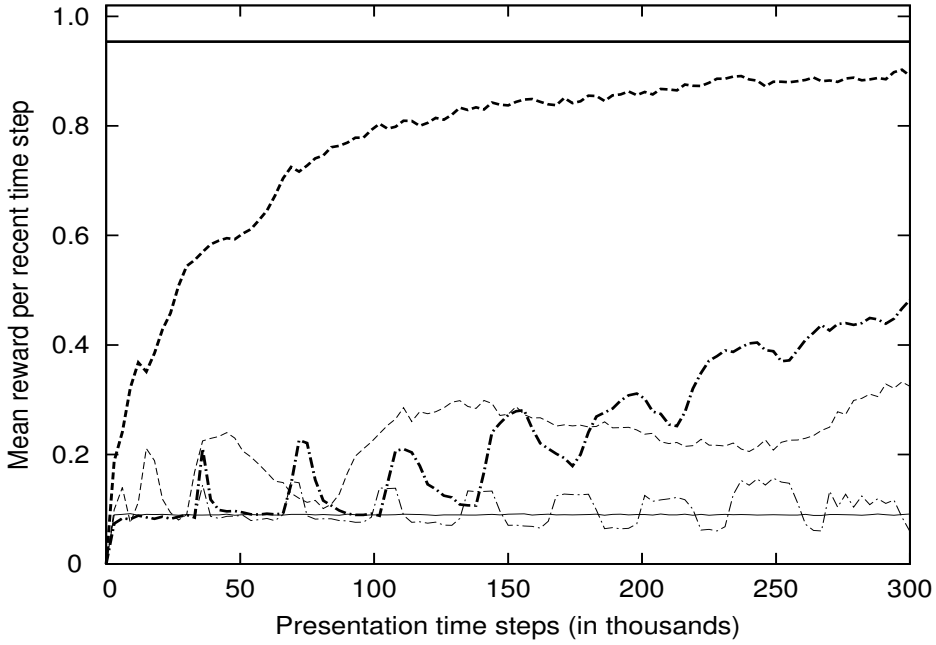


Figure 4: "Optimistic" vs. "pessimistic" UCB1-style algorithms:
The learning algorithms on 5-slot problem instances with random relevance rates $\mu(\cdot)$ selected according to the Chinese Restaurant Process.

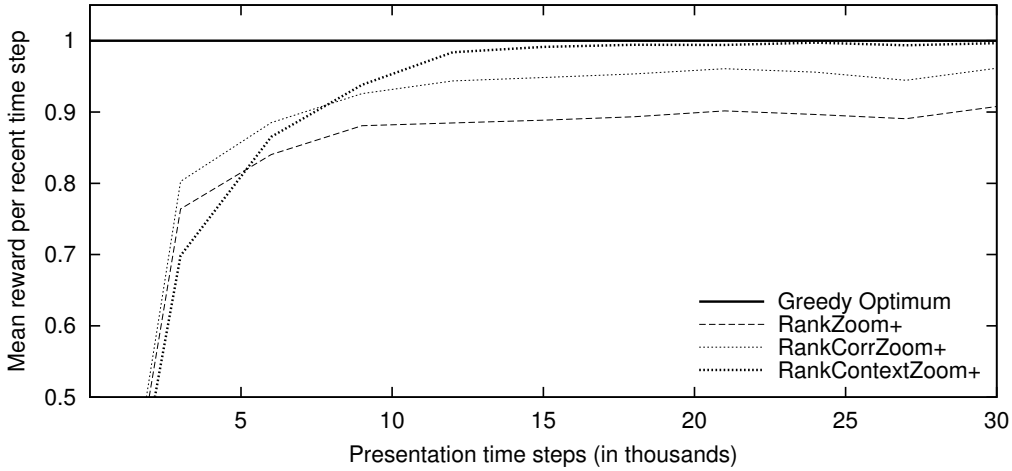


Figure 5: Zooming-style algorithms in a two-slot setting over a small document collection.

Thus $\mu(x)$ should lie in the interval $I(x) \triangleq [\mu^-(x), \mu^+(x)]$, where

$$\mu^-(x) \triangleq \sup_{z \in \mathcal{L}(x)} \mu(z) - D(x,z),$$
$$\mu^+(x) \triangleq \inf_{z \in \mathcal{L}(x)} \mu(z) + D(x,z).$$

This interval is always well-defined, that is, $\mu^-(x) \leq \mu^+(x)$. Indeed, if not then for some $z, z' \in \mathcal{L}(x)$

$$\mu(z) - D(x,z) > \mu(z') + D(x,z')$$
$$\mu(z) - \mu(z') > D(x,z) + D(x,z') \geq D(z,z'),$$

contradiction, claim proved. Note that $\mu^+(x) \geq a$ and $\mu^-(x) \leq b$, so the intervals $I(x)$ and $[a,b]$ overlap.

Using induction on the tree, we will construct values $\mu(x)$, $x \in V$ such that the Lipschitz condition

$$|\mu(x) - \mu(y)| \leq D(x,y) \qquad \text{for all } x, y \in X$$

holds whenever x is a parent of y. For the root x_0, let $\mu(x_0)$ be an arbitrary value in the interval $I(x_0) \cap [a,b]$. For the induction step, suppose for some x we have chosen $\mu(x) \in I(x) \cap [a,b]$ and y is a child of x. We need to choose $\mu(y) \in I(y) \cap [a,b]$ so that $|\mu(x) - \mu(y)| \leq D(x,y)$. Note that

$$\begin{aligned}
\mu(x) \geq \mu^-(x) &\geq \sup_{z \in \mathcal{L}(y)} \left[\mu(z) - D(x,y) - D(y,z)\right] \\
&= \mu^-(y) - D(x,y), \\
\mu(x) \leq \mu^+(x) &\leq \inf_{z \in \mathcal{L}(y)} \left[\mu(z) + D(x,y) + D(y,z)\right] \\
&= \mu^+(y) + D(x,y).
\end{aligned}$$

It follows that $I(y)$ and $[\mu(x) - D(x,y), \mu(x) + D(x,y)]$ have a non-empty intersection. Therefore, both intervals have a non-empty intersection with $[a,b]$. So we can choose $\mu(y)$ as required. This completes the construction of $\mu()$ on V.

To check that μ is Lipschitz-continuous on V, fix $x,y \in V$, let P be the $x \to y$ path in the tree, and note that

$$\begin{aligned}|\mu(x)-\mu(y)| &\leq \textstyle\sum_{(u,v)\in P} |\mu(u)-\mu(v)| \\ &\leq \textstyle\sum_{(u,v)\in P} D(u,v) = D(x,y).\end{aligned}$$

■

Appendix B. Proof of Theorem 5 (Expressiveness of the Model)

Recall that a proof sketch for Theorem 5 was given in Section 4. In this section we complete this proof sketch by proving Equation (12).

Notation. Let us introduce the notation (some of it is from the proof sketch).

For a tree node u, let $\mathcal{T}_u$ be the node set of the subtree rooted at u. For convenience (and by a slight abuse of notation) we will write $u = b$, $b \in \{0,1\}$ to mean $\pi(u) = b$.

Fix documents $x,y \in X$. We focus on the key event, denoted $\mathcal{E}$, that no mutation happened on the $x \to y$ path. Recall that in Algorithm 1, for each tree node u with parent v we assign $\pi(u) \leftarrow M_u(\pi(v))$, where $M_u : \{0,1\} \to \{0,1\}$ is a random mutation which flips the input bit b with probability $q_b(u)$. If M_u is the identity function, then we say that no mutation happened at u. We say that no mutation happened on the $x \to y$ path if no mutation happened at each node in N_{xy}, the set of all nodes on the $x \to y$ path except z. This event is denoted $\mathcal{E}$; note that it implies $\pi(x) = \pi(y) = \pi(z)$. Its complement $\bar{\mathcal{E}}$ is, intuitively, a low-probability "failure event".

Fix a subset of documents $S \subset X$. Recall that Z_S denotes the event that all documents in S are irrelevant, that is, $\pi(x) = 0$ for all $x \in S$.

What we need to prove. We need to prove Equation (12), which states that

$$\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq 3 \Pr[\bar{\mathcal{E}}].$$

It suffices to prove the following lemma:

Lemma 13 $\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq \Pr[\bar{\mathcal{E}}] \times (2/\Pr[\mathcal{E}])$.

(Indeed, letting $p = \Pr[\bar{\mathcal{E}}]$ it holds that $\Pr[\bar{\mathcal{E}} \,|\, Z_S] \leq \min\left(1, \frac{2p}{1-p}\right) \leq 3p$.)

Remark. Lemma 13 inherits assumptions (6-7) on the mutation probabilities. Specifically for this Lemma, the upper bound (6) on mutation probabilities can be replaced with a much weaker upper bound:

$$\max(q_0(u), q_1(u)) \leq \tfrac{1}{2} \quad \text{for each tree node } u. \tag{27}$$

Our goal is to prove Lemma 13. In a sequence on claims, we will establish that

$$\Pr[Z_S \,|\, z = 0] \geq \Pr[Z_S \,|\, z = 1]. \tag{28}$$

Intuitively, (28) means that the low-probability mutations are more likely to zero out a given subset of the leaves if the value at some fixed internal node is zero (rather than one).

B.1 Using Equation (28) to Prove Lemma 13

Let us extend the notion of mutation from a single node to the $x \to y$ path. Recall that N_{xy} denotes the set of all nodes on this path except z. Then the individual node mutations $\{M_u : u \in N_{xy}\}$ collectively provide a mutation on N_{xy}, which we define simply as a function $M : N_{xy} \times \{0,1\} \to \{0,1\}$ such that $\pi(\cdot) = M(\cdot, \pi(z))$. Crucially, M is chosen independently of $\pi(z)$ (and of all other mutations). Let $\mathcal{M}$ be the set of all possible mutations of N_{xy}. By a slight abuse of notation, we treat the event $\mathcal{E}$ as the identity mutation.

Claim 14 *Fix $M \in \mathcal{M}$ and $b \in \{0,1\}$. Then*

$$\Pr[Z_S \,|\, M, \pi(z) = b] \leq \Pr[Z_S \,|\, \mathcal{E}, \pi(z) = 0].$$

Proof For each tree node u, let $S_u = S \cap \mathcal{T}_u$ be the subset of S that lies in the subtree $\mathcal{T}_u$. Then by (28)

$$\begin{aligned}\Pr[Z_S \,|\, M,\, \pi(z) = b] &= \textstyle\prod_u \Pr[Z_{S_u} \,|\, \pi(u) = M(u,b)] \\ &\leq \textstyle\prod_u \Pr[Z_{S_u} \,|\, \pi(u) = 0] \\ &= \Pr[Z_S \,|\, \mathcal{E},\, \pi(z) = 0],\end{aligned}$$

where the product is over all tree nodes $u \in N_{xy}$ such that the intersection S_u is non-empty. ■

Proof [Proof of Lemma 13] On one hand, by Claim 14

$$\begin{aligned}\Pr[Z_S \cap \bar{\mathcal{E}}] &= \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, M, z = b] \\ &\leq \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, \mathcal{E}, z = 0] \\ &= \Pr[\bar{\mathcal{E}}] \times \Pr[Z_S \,|\, \mathcal{E}, z = 0],\end{aligned}$$

where the sums are over bits $b \in \{0,1\}$ and all mutations $M \in \mathcal{M} \setminus \{\mathcal{E}\}$. On the other hand,

$$\Pr[Z_S] = \textstyle\sum_{b,M} \Pr[M]\ \Pr[z = b]\ \Pr[Z_S \,|\, M, z = b]$$

(where the sum is over $b \in \{0,1\}$ and $M \in \mathcal{M}$)

$$\geq \Pr[\mathcal{E}]\ \Pr[z = 0]\ \Pr[Z_S \,|\, \mathcal{E}, z = 0].$$

Since $\Pr[z = 0] \geq \frac{1}{2}$, it follows that

$$\begin{aligned}\Pr[\bar{\mathcal{E}} \,|\, Z_S] &= \Pr[Z_S \cap \bar{\mathcal{E}}] \,/\, \Pr[Z_S] \\ &\leq 2\,\Pr[\bar{\mathcal{E}}] / \Pr[\mathcal{E}].\end{aligned}$$

■

B.2 Proof of Equation (28)

First we prove (28) for the case $S \subset \mathcal{T}_z$, then we build on it to prove the (similar, but considerably more technical) case $S \cap T_z = \emptyset$. The general case follows since the events $Z_{S \cap \mathcal{T}_z}$ and $Z_{S \setminus \mathcal{T}_z}$ are conditionally independent given $\pi(z)$.

Claim 15 *If $S \subset \mathcal{T}_z$ then (28) holds.*

Proof Let us use induction the depth of z. For the base case, the case $x = y = z$. Then $S = \{z\}$ is the only possibility, and the claim is trivial.

For the induction step, consider children u_i of z such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let $u_1, \ldots, u_k$ be all such children. For brevity, denote $Z_i \triangleq Z_{S_i}$, and

$$\nu_i(a|b) \triangleq \Pr[u_i = a \,|\, z = b], \quad a, b \in \{0,1\}.$$

Note that $v_i(1,0) = q_0(x_i)$ and $v_i(0,1) = q_1(x_i)$.

Then for each $b \in \{0,1\}$ we have

$$\Pr[Z_S \,|\, z = b] = \textstyle\prod_{i=1}^{k} \Pr[Z_i \,|\, z = b] \tag{29}$$

$$\Pr[Z_i \,|\, z = b] = \textstyle\sum_{a \in \{0,1\}} \nu_i(a|b) \, \Pr[Z_i \,|\, u_i = a]. \tag{30}$$

By (29), to prove the claim it suffices to show that

$$\Pr[Z_i \,|\, z = 0] \geq \Pr[Z_i \,|\, z = 1]$$

holds for each i. By the induction hypothesis we have

$$\Pr[Z_i \,|\, u_i = 0] \geq \Pr[Z_i \,|\, u_i = 1]. \tag{31}$$

Combining (31) and (27), and noting that by (30) we have $\nu_i(0|0) \geq \nu_i(0|1)$, it follows that

$$\begin{aligned}
&\Pr[Z_i \,|\, z = 0] - \Pr[Z_i \,|\, z = 1] \\
&\quad = \textstyle\sum_{a \in \{0,1\}} \Pr[Z_i \,|\, u_i = a] \; (\nu_i(a|0) - \nu_i(a|1)) \\
&\quad \geq \Pr[Z_i \,|\, u_i = 1] \quad \textstyle\sum_{a \in \{0,1\}} (\nu_i(a|0) - \nu_i(a|1)) \\
&\quad = 0
\end{aligned}$$

because $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$. ■

Corollary 16 *Consider tree nodes r, v, w such that r is an ancestor of v which in turn is an ancestor of w. Then for any $c \in \{0,1\}$*

$$\Pr[u = 0 \,|\, w = 0, r = c] \geq \Pr[u = 0 \,|\, w = 1, r = c].$$

Proof We claim that for each $b \in \{0,1\}$

$$\Pr[w = b \,|\, u = b] \geq \Pr[w = b \,|\, u = 1 - b]. \tag{32}$$

Indeed, truncating the subtree $\mathcal{T}_w$ to a single node w and specializing Lemma 15 to a singleton set $S = \{w\}$ (with $z = u$) we obtain (32) for $b = 0$. The case $b = 1$ is symmetric.

Now, for brevity we will omit conditioning on $\{r = c\}$ in the remainder of the proof. (Formally, we will work on in the probability space obtained by conditioning on this event.) Then for each $b \in \{0,1\}$

$$\begin{aligned}
&\Pr[u = 0 \,|\, w = b] \\
&\quad = \frac{\Pr[u = 0 \wedge w = b]}{\Pr[u = 0 \wedge w = b] \cup \Pr[u = 1 \wedge w = b]} \\
&\quad = \frac{1}{1 + \Phi(b)},
\end{aligned}$$

where

$$\begin{aligned}
\Phi(b) &\triangleq \frac{\Pr[u = 1 \wedge w = b]}{\Pr[u = 0 \wedge w = b]} \\
&= \frac{\Pr[w = b \,|\, u = 1] \Pr[u = 1]}{\Pr[w = b \,|\, u = 0] \Pr[u = 0]}
\end{aligned}$$

is decreasing in b by (32). ■

We will also need a stronger, *conditional*, version of Lemma 15 whose proof is essentially identical (and omitted).

Claim 17 *Suppose $S \subset \mathcal{T}_z$ and $u \neq z$ is a tree node such that $\mathcal{T}_u$ is disjoint with S. Then*

$$\Pr[Z_S \,|\, z = 0, u = 1] \geq \Pr[Z_S \,|\, z = 1, u = 1].$$

We will use Corollary 16 and Lemma 17 to prove (28) for the case $S \cap T_z = \emptyset$.

Claim 18 *If S is disjoint with $\mathcal{T}_z$ then (28) holds.*

Proof Suppose S is disjoint with $\mathcal{T}_z$, and let r be the root of the tree. We will use induction on the tree to prove the following: for each $c \in \{0,1\}$,

$$\Pr[Z_S \,|\, r = c, z = 0] \geq \Pr[Z_S \,|\, r = c, z = 1] \tag{33}$$

For the induction base, consider a tree of depth 2, consisting of the root r and the leaves. Then $z \notin S$ is a leaf, so Z_S is independent of $\pi(z)$ given $\pi(r)$, so (33) holds with equality.

For the induction step, fix $c \in \{0,1\}$. Let us set up the notation similarly to the proof of Claim 15. Consider children u_i of r such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let $u_1, \ldots, u_k$ be all such children. Assume $z \in \mathcal{T}_{u_i}$ for some i (else, Z_S is independent from $\pi(z)$ given $\pi(r)$, so (33) holds with equality); without loss of generality, assume this happens for $i = 1$. For brevity, for $a, b \in \{0,1\}$ denote

$$\begin{aligned}
f_i(a,b) &\triangleq \Pr[Z_{S_i} \quad | \, u_i = a, z = b] \\
\nu_i(a|b) &\triangleq \Pr[u_i = a \,|\, r = c, \;\; z = b].
\end{aligned}$$

Note that $f_i(a,b)$ and $\nu_i(a|b)$ do not depend on b for $i > 1$.

Then for each $b \in \{0,1\}$

$$\begin{aligned}
&\Pr[Z_S \,|\, r = c, z = b] \\
&\quad = \sum_{a_i \in \{0,1\},\, i \geq 1} \prod_{i \geq 1} f_i(a_i,b)\, \nu_i(a_i|b) \\
&\quad = \Phi \times \textstyle\sum_{a \in \{0,1\}} f_1(a,b)\, \nu_1(a|b),
\end{aligned}$$

where

$$\Phi \triangleq \sum_{a_i \in \{0,1\},\, i \geq 2} \prod_{i \geq 2} f_i(a_i,b)\, \nu_i(a_i|b)$$

does not depend on of b. Therefore:

$$\begin{aligned}
&\Pr[Z_S \,|\, r = c, z = 1] - \Pr[Z_S \,|\, r = c, z = 1] \\
&\quad = \Phi \times \textstyle\sum_{a \in \{0,1\}} \\
&\qquad\qquad [\, f_1(a,0)\, \nu_1(a|0) - f_1(a,1)\, \nu_1(a|1)\,] && (34) \\
&\quad \geq \Phi \times \textstyle\sum_{a \in \{0,1\}} f_1(a,1)\, [\, \nu_1(a|0) - \nu_1(a|1)\,] && (35) \\
&\quad \geq \Phi \times f_1(1,1) \textstyle\sum_{a \in \{0,1\}} [\, \nu_1(a|0) - \nu_1(a|1)\,] && (36) \\
&\quad = 0. && (37)
\end{aligned}$$

The above transitions hold for the following reasons:

(34 $\to$ 35) By Induction Hypothesis, $f_1(a,0) \geq f_1(a,1)$

(35 $\to$ 36) By Lemma 17 $f_1(0,1) \geq f_1(1,1)$, and moreover we have $\nu_1(0|0) \geq \nu_1(0|1)$ by Corollary 16.

(36 $\to$ 37) Since $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$

This completes the proof of the inductive step. ■

Appendix C. Instance-Dependent Regret Bounds from Prior Work

In this section we discuss instance-dependent regret bounds from prior work on UCB1-style algorithms for the single-slot setting. The purpose is to put forward a concrete mathematical evidence which suggests that RankGridUCB1, RankZoom and RankCorrZoom are likely to satisfy strong upper bounds on regret in the k-slot setting (perhaps under some additional assumptions), even if such bounds are beyond the reach of our current techniques. Similarly, we believe that the regret bound for RankContextZoom that we have been able to prove (Theorem 10) is overly pessimistic. A secondary purpose is to provide more intuition for when these algorithms are likely to excel.

Our story begins with the comparison between the guarantees for EXP3 and UCB1 in the standard (single-slot, metric-free) bandit setting, and then progresses to Lipschtz MAB and contextual Lispchitz MAB.

In what follows, we let μ denote the vector of expected rewards in the stochastic reward setting, so that $\mu(x)$ is the expected reward of arm x. Let $\Delta(x) \triangleq \max \mu(\cdot) - \mu(x)$ denote the "badness" of arm x compared to the optimum.

C.1 Standard Bandits: UCB1 vs. EXP3

Algorithm EXP3(Auer et al., 2002b) achieves regret $R(T) = \tilde{O}(\sqrt{nT})$ against an oblivious adversary. In the stochastic setting, UCB1(Auer et al., 2002a) performs much better, with *logarithmic* regret for every fixed μ. More specifically, each arm $x \in X$ contributes only $O(\log T)/\Delta(x)$ to regret. Noting that the total regret from playing arms with $\Delta(\cdot) \leq \delta$ can be a priori upper-bounded by δT, we bound regret of UCB1 as:

$$R(T) = \min_{\delta>0} \left(\delta T + \sum_{x \in X: \Delta(x) > \delta} \frac{O(\log T)}{\Delta(x)} \right). \tag{38}$$

Note that Equation (38) depends on μ. In particular, if $\Delta(\cdot) \geq \delta$ then $R(T) = O(\frac{n}{\delta} \log T)$.

However, for any given T there exists a "worst-case" pointwise mean μ_T such that $R(T) = \tilde{\Theta}(\sqrt{nT})$ in Equation (38), matching EXP3. The above regret guarantees for EXP3 and UCB1 are optimal up to constant factors (Auer et al., 2002b; Kleinberg et al., 2008a).

C.2 Bandits in Metric Spaces

Let (X, D) denote the metric space. Recall that the *covering number* $N_r(X)$ is the minimal number of balls of radius r sufficient to cover X, and the *covering dimension* is defined as

$$\texttt{CovDim}(X, D) \triangleq \inf\{d \geq 0 : N_r(X) \leq \alpha r^{-d} \quad \forall r > 0\}.$$

(Here $\alpha > 0$ is a constant which we will keep implicit in the notation.)

Against an oblivious adversary, GridEXP3 has regret

$$R(T) = \tilde{O}(\alpha T^{(d+1)/(d+2)}), \tag{39}$$

where d is the covering dimension of (X, D).

For the stochastic setting, GridUCB1 and the zooming algorithm have better μ-specific regret guarantees in terms of the covering numbers. These guarantees are similar to Equation (38) for UCB1. In fact, it is possible, and instructive, to state the guarantees for all three algorithms in a common form.

Consider reward scales $\mathcal{S} = \{2^i : i \in \mathbb{N}\}$, and for each scale $r \in \mathcal{S}$ define

$$X_r = \{x \in X : r < \Delta(x) \leq 2r\}.$$

Then regret (38) of UCB1 can be restated as

$$R(T) = \min_{\delta>0} \left(\delta T + \sum_{r \in \mathcal{S}: r \geq \delta} N_{(\delta,r)} \frac{O(\log T)}{r} \right), \tag{40}$$

where $N_{(\delta,r)} = |X_r|$. Further, it follows from the analysis in (Kleinberg, 2004; Kleinberg et al., 2008b) that regret of GridUCB1 is Equation (40) with $N_{(\delta,r)} = N_\delta(X_r)$. For the zooming algorithm, the μ-specific bound can be improved to Equation (40) with $N_{(\delta,r)} = N_r(X_r)$. These results are summarized in Table C.2.

For the worst-case μ one could have $N_\delta(X_r) = N_\delta(X)$, in which case the μ-specific bound for GridUCB1 essentially reduces to Equation (39).

algorithm	regret is (40) with ...
`UCB1`	$N_{(\delta,r)} = \lvert X_r \rvert$
`GridUCB1`	$N_{(\delta,r)} = N_\delta(X_r)$
zooming algorithm	$N_{(\delta,r)} = N_r(X_r)$
`ContextZoom`	$N_{(\delta,r)} = N_r(X_{\text{dc},r})$.

Table 2: Regret bounds in terms of covering numbers

For the zooming algorithm, the μ-specific bound above implies an improved version of Equation (39) with a different, smaller d called the *zooming dimension*:

$$\texttt{ZoomDim}(X,D,\mu) \triangleq \inf\{d \geq 0 : N_r(X_r) \leq c\, r^{-d} \quad \forall r > 0\}.$$

Note that the zooming dimension depends on the triple (X,D,μ) rather than on the metric space alone. It can be as high as the covering dimension for the worst-case μ, but can be much smaller (e.g., $d = 0$) for "nice" problem instances, see (Kleinberg et al., 2008b) for further discussion. For a simple example, suppose an ε-exponential tree metric has a "high-reward" branch and a "low-reward" branch with respective branching factors $b \ll b'$. Then the zooming dimension is $\log_{1/\varepsilon}(b)$, whereas the covering dimension is $\log_{1/\varepsilon}(b')$.

C.3 Contextual Bandits in Metric Spaces

Let $\mu(x|h)$ denote the expected reward from arm x given context h. Recall that the algorithm is given metrics D and D_{c} on documents and contexts, respectively, such that for any two documents x, x' and any two contexts h, h' we have

$$|\mu(x|h) - \mu(x'|h')| \leq D(x,x') + D_{\text{c}}(h,h').$$

Let X_{c} be the set of contexts, and $X_{\text{dc}} = X \times X_{\text{c}}$ be the set of all (document, context) pairs. More abstractly, one considers the metric space $(X_{\text{dc}}, D_{\text{dc}})$, henceforth the *DC-space*, where the metric is

$$D_{\text{dc}}((x,h),(x',h')) = D(x,x') + D_{\text{c}}(h,h').$$

We partition X_{dc} according to reward scales $r \in \mathcal{S}$:

$$\Delta(x|h) \triangleq \max \mu(\cdot|h) - \mu(x|h), \quad x \in X, h \in X_{\text{c}}.$$
$$X_{\text{dc},r} \triangleq \{(x,h) \in X_{\text{dc}} : r < \Delta(x|h) \leq 2r\}.$$

Then contextual regret of `ContextZoom` can be bounded by Equation (40) with $N_{(\delta,r)} = N_r(X_{\text{dc},r})$, where $N_r(\cdot)$ now refers to the covering numbers in the DC-space (see Table C.2).

Further, one can define the *contextual* zooming dimension as

$$d_{\text{dc}}(X,D,\mu) \triangleq \inf\{d \geq 0 : N_r(X_r) \leq c\, r^{-d} \quad \forall r > 0\}.$$

Then one obtains Equation (39) with $d = d_{\text{dc}}$. In the worst case, we could have μ such that $N_r(X_{\text{dc},r}) = N_r(X_{\text{dc}})$, in which case $d_{\text{dc}} \leq \texttt{CovDim}(X_{\text{dc}}, D_{\text{dc}})$.

The regret bounds for `ContextZoom` can be improved by taking into account "benign" context arrivals: effectively, one can prune the regions of X_{c} that correspond to infrequent context arrivals, see (Slivkins, 2009) for details. This improvement can be especially significant if $\texttt{CovDim}(X_{\text{c}}, D_{\text{c}}) > \texttt{CovDim}(X,D)$.

References

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21th Conf. on Learning Theory (COLT)*, pages 263–274, 2008.

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6): 1926–1951, 1995.

David J. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII*, pages 1–198, 1985.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. of Machine Learning Research (JMLR)*, 3:397–422, 2002. Preliminary version in *41st IEEE FOCS*, 2000.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a. Preliminary version in *15th ICML*, 1998.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. Preliminary version in *36th IEEE FOCS*, 1995.

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *20th Conf. on Learning Theory (COLT)*, pages 454–468, 2007.

Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *J. of Computer and System Sciences*, 74(1):97–114, February 2008. Preliminary version in *36th ACM STOC*, 2004.

Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 1996.

Dirk Bergemann and Juuso Välimäki. Bandit problems. In Steven Durlauf and Larry Blume, editors, *The New Palgrave Dictionary of Economics, 2nd ed.* Macmillan Press, 2006.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning (Draft under submission)*, 2012. Available at `www.princeton.edu/∼sbubeck/pub.html`.

Sébastien Bubeck and Rémi Munos. Open loop optimistic planning. In *23rd Conf. on Learning Theory (COLT)*, pages 477–489, 2010.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online optimization in x-armed bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011. Preliminary version in *NIPS 2008*.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Intl. Conf. on Machine Learning (ICML)*, pages 89–96, 2005.

Jaime G. Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Intl. Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 335–336, 1998.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge Univ. Press, 2006.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *J. of Machine Learning Research*, 6:1019–1041, 2005.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *14thIntl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Varsha Dani, Thomas P. Hayes, and Sham Kakade. The price of bandit information for online optimization. In *20th Advances in Neural Information Processing Systems (NIPS)*, 2007.

Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. of Computer and System Sciences*, 69(3):485–497, 2004.

Abraham Flaxman, Adam Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 385–394, 2005.

Daniel Golovin, Andreas Krause, and Matthew Streeter. Online learning of assignments. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2003.

Elad Hazan and Satyen Kale. Better algorithms for benign bandits. In *20th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 38–47, 2009.

Elad Hazan and Nimrod Megiddo. Online learning with prior information. In *20th Conf. on Learning Theory (COLT)*, pages 499–513, 2007.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

Satyen Kale, Lev Reyzin, and Robert E. Schapire. Non-stochastic bandit slate problems. In *24th Advances in Neural Information Processing Systems (NIPS)*, pages 1054–1062, 2010.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.

Robert Kleinberg and Aleksandrs Slivkins. Sharp dichotomies for regret minimization in metric spaces. In *21st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. In *21st Conf. on Learning Theory (COLT)*, pages 425–436, 2008a.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008b.

Levente Kocsis and Csaba Szepesvari. Bandit based Monte-Carlo planning. In *17th European Conf. on Machine Learning (ECML)*, pages 282–293, 2006.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *21st Advances in Neural Information Processing Systems (NIPS)*, 2007.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *19th Intl. World Wide Web Conf. (WWW)*, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *4th ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, 2011.

Tyler Lu, Dávid Pál, and Martin Pál. Showing relevant ads via Lipschitz context multi-armed bandits. In *14thIntl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Odalric-Ambrym Maillard and Rémi Munos. Online learning in adversarial lipschitz environments. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 305–320, 2010.

Rémi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2007.

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: A model-based approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *25th Intl. Conf. on Machine Learning (ICML)*, pages 784–791, 2008.

Philippe Rigollet and Assaf Zeevi. Nonparametric bandits with covariates. In *23rd Conf. on Learning Theory (COLT)*, pages 54–66, 2010.

Aleksandrs Slivkins. Contextual bandits with similarity information. http://arxiv.org/abs/0907.3986, 2009. Has been published in *24th COLT 2011*.

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *27th Intl. Conf. on Machine Learning (ICML)*, pages 1015–1022, 2010.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1577–1584, 2008.

Rangarajan K. Sundaram. Generalized bandit problems. In David Austen-Smith and John Duggan, editors, *Social Choice and Strategic Decisions: Essays in Honor of Jeffrey S. Banks (Studies in Choice and Welfare)*, pages 131–162. Springer, 2005. First appeared as *Working Paper, Stern School of Business*, 2003.

Michael J. Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimizing non-smooth rank metrics. In *ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, pages 77–86, 2008.

Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. Algorithms for adversarial bandit problems with multiple plays. In *21st Intl. Conf. on Algorithmic Learning Theory (ALT)*, pages 375–389, 2010.

Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Bandit problems with side observations. *IEEE Trans. on Automatic Control*, 50(3):338355, 2005.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1729–1736, 2008.

Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *J. Amer. Statist. Assoc.*, 74(368), 1979.

Journal of Machine Learning Research 14 (2013) 437-497 Submitted 8/11; Revised 1/13; Published 2/13

A Theory of Multiclass Boosting

Indraneel Mukherjee IMUKHERJ@CS.PRINCETON.EDU
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043, USA

Robert E. Schapire SCHAPIRE@CS.PRINCETON.EDU
Princeton University
Department of Computer Science
Princeton, NJ 08540 USA

Editor: Manfred Warmuth

Abstract

Boosting combines weak classifiers to form highly accurate predictors. Although the case of binary classification is well understood, in the multiclass setting, the "correct" requirements on the weak classifier, or the notion of the most efficient boosting algorithms are missing. In this paper, we create a broad and general framework, within which we make precise and identify the optimal requirements on the weak-classifier, as well as design the most effective, in a certain sense, boosting algorithms that assume such requirements.

Keywords: multiclass, boosting, weak learning condition, drifting games

1. Introduction

Boosting (Schapire and Freund, 2012) refers to a general technique of combining rules of thumb, or weak classifiers, to form highly accurate combined classifiers. Minimal demands are placed on the weak classifiers, so that a variety of learning algorithms, also called weak-learners, can be employed to discover these simple rules, making the algorithm widely applicable. The theory of boosting is well-developed for the case of binary classification. In particular, the exact requirements on the weak classifiers in this setting are known: any algorithm that predicts better than random on any distribution over the training set is said to satisfy the weak learning assumption. Further, boosting algorithms that minimize loss as efficiently as possible have been designed. Specifically, it is known that the Boost-by-majority (Freund, 1995) algorithm is optimal in a certain sense, and that AdaBoost (Freund and Schapire, 1997) is a practical approximation.

Such an understanding would be desirable in the multiclass setting as well, since many natural classification problems involve more than two labels, for example, recognizing a digit from its image, natural language processing tasks such as part-of-speech tagging, and object recognition in vision. However, for such multiclass problems, a complete theoretical understanding of boosting is lacking. In particular, we do not know the "correct" way to define the requirements on the weak classifiers, nor has the notion of optimal boosting been explored in the multiclass setting.

Straightforward extensions of the binary weak-learning condition to multiclass do not work. Requiring less error than random guessing on every distribution, as in the binary case, turns out to be too weak for boosting to be possible when there are more than two labels. On the other hand,

requiring more than 50% accuracy even when the number of labels is much larger than two is too stringent, and simple weak classifiers like decision stumps fail to meet this criterion, even though they often can be combined to produce highly accurate classifiers (Freund and Schapire, 1996a). The most common approaches so far have relied on reductions to binary classification (Allwein et al., 2000), but it is hardly clear that the weak-learning conditions implicitly assumed by such reductions are the most appropriate.

The purpose of a weak-learning condition is to clarify the goal of the weak-learner, thus aiding in its design, while providing a specific minimal guarantee on performance that can be exploited by a boosting algorithm. These considerations may significantly impact learning and generalization because knowing the correct weak-learning conditions might allow the use of simpler weak classifiers, which in turn can help prevent overfitting. Furthermore, boosting algorithms that more efficiently and effectively minimize training error may prevent underfitting, which can also be important.

In this paper, we create a broad and general framework for studying multiclass boosting that formalizes the interaction between the boosting algorithm and the weak-learner. Unlike much, but not all, of the previous work on multiclass boosting, we focus specifically on the most natural, and perhaps weakest, case in which the weak classifiers are genuine classifiers in the sense of predicting a single multiclass label for each instance. Our new framework allows us to express a range of weak-learning conditions, both new ones and most of the ones that had previously been assumed (often only implicitly). Within this formalism, we can also now finally make precise what is meant by *correct* weak-learning conditions that are neither too weak nor too strong.

We focus particularly on a family of novel weak-learning conditions that have an especially appealing form: like the binary conditions, they require performance that is only slightly better than random guessing, though with respect to performance measures that are more general than ordinary classification error. We introduce a whole family of such conditions since there are many ways of randomly guessing on more than two labels, a key difference between the binary and multiclass settings. Although these conditions impose seemingly mild demands on the weak-learner, we show that each one of them is powerful enough to guarantee boostability, meaning that some combination of the weak classifiers has high accuracy. And while no individual member of the family is necessary for boostability, we also show that the entire family taken together is necessary in the sense that for every boostable learning problem, there exists one member of the family that is satisfied. Thus, we have identified a family of conditions which, as a whole, is necessary and sufficient for multiclass boosting. Moreover, we can combine the entire family into a single weak-learning condition that is necessary and sufficient by taking a kind of union, or logical OR, of all the members. This combined condition can also be expressed in our framework.

With this understanding, we are able to characterize previously studied weak-learning conditions. In particular, the condition implicitly used by AdaBoost.MH (Schapire and Singer, 1999), which is based on a one-against-all reduction to binary, turns out to be strictly stronger than necessary for boostability. This also applies to AdaBoost.M1 (Freund and Schapire, 1996a), the most direct generalization of AdaBoost to multiclass, whose conditions can be shown to be equivalent to those of AdaBoost.MH in our setting. On the other hand, the condition implicit to the SAMME algorithm by Zhu et al. (2009) is too weak in the sense that even when the condition is satisfied, no boosting algorithm can guarantee to drive down the training error. Finally, the condition implicit to AdaBoost.MR (Schapire and Singer, 1999; Freund and Schapire, 1996a) (also called AdaBoost.M2) turns out to be exactly necessary and sufficient for boostability.

Employing proper weak-learning conditions is important, but we also need boosting algorithms that can exploit these conditions to effectively drive down error. For a given weak-learning condition, the boosting algorithm that drives down training error most efficiently in our framework can be understood as the optimal strategy for playing a certain two-player game. These games are non-trivial to analyze. However, using the powerful machinery of drifting games (Freund and Opper, 2002; Schapire, 2001), we are able to compute the optimal strategy for the games arising out of each weak-learning condition in the family described above. Compared to earlier work, our optimality results hold more generally and also achieve tighter bounds. These optimal strategies have a natural interpretation in terms of random walks, a phenomenon that has been observed in other settings (Abernethy et al., 2008; Freund, 1995).

We also analyze the optimal boosting strategy when using the minimal weak learning condition, and this poses additional challenges. Firstly, the minimal weak learning condition has multiple natural formulations—for example, as the union of all the conditions in the family described above, or the formulation used in AdaBoost.MR—and each formulation leading to a different game specification. A priori, it is not clear which game would lead to the best strategy. We resolve this dilemma by proving that the optimal strategies arising out of different formulations of the same weak learning condition lead to algorithms that are essentially equally good, and therefore we are free to choose whichever formulation leads to an easier analysis without fear of suffering in performance. We choose the union of conditions formulation, since it leads to strategies that share the same interpretation in terms of random walks as before. However, even with this choice, the resulting games are hard to analyze, and although we can explicitly compute the optimum strategies in general, the computational complexity is usually exponential in the number of classes. Nevertheless, we identify key situations under which efficient computation is possible.

The game-theoretic strategies are non-adaptive in that they presume prior knowledge about the *edge*, that is, how much better than random are the weak classifiers. Algorithms that are adaptive, such as AdaBoost, are much more practical because they do not require such prior information. We show therefore how to derive an adaptive boosting algorithm by modifying the game-theoretic strategy based on the minimal condition. This algorithm enjoys a number of theoretical guarantees. Unlike some of the non-adaptive strategies, it is efficiently computable, and since it is based on the minimal weak learning condition, it makes minimal assumptions. In fact, whenever presented with a boostable learning problem, this algorithm can approach zero training error at an exponential rate. More importantly, the algorithm is effective even beyond the boostability framework. In particular, we show empirical consistency, that is, the algorithm always converges to the minimum of a certain exponential loss over the training data, whether or not the data set is boostable. Furthermore, using the results in Mukherjee et al. (2011) we can show that this convergence occurs rapidly.

Our focus in this paper is only on minimizing training error, which, for the algorithms we derive, provably decreases exponentially fast with the number of rounds of boosting under boostability assumptions. Such results can be used in turn to derive bounds on the generalization error using standard techniques that have been applied to other boosting algorithms (Schapire et al., 1998; Freund and Schapire, 1997; Koltchinskii and Panchenko, 2002). Consistency in the multiclass classification setting has been studied by Tewari and Bartlett (2007) and has been shown to be trickier than binary classification consistency. Nonetheless, by following the approach in Bartlett and Traskin (2007) for showing consistency in the binary setting, we are able to extend the empirical consistency guarantees to general consistency guarantees in the multiclass setting: we show that

under certain conditions and with sufficient data, our adaptive algorithm approaches the Bayes-optimum error on the *test* data set.

We present experiments aimed at testing the efficacy of the adaptive algorithm when working with a very weak weak-learner to check that the conditions we have identified are indeed weaker than others that had previously been used. We find that our new adaptive strategy achieves low test error compared to other multiclass boosting algorithms which usually heavily underfit. This validates the potential practical benefit of a better theoretical understanding of multiclass boosting.

1.1 Previous Work

The first boosting algorithms were given by Schapire (1990) and Freund (1995), followed by their AdaBoost algorithm (Freund and Schapire, 1997). Multiclass boosting techniques include AdaBoost.M1 and AdaBoost.M2 (Freund and Schapire, 1997), as well as AdaBoost.MH and AdaBoost.MR (Schapire and Singer, 1999). Other approaches include the work by Eibl and Pfeiffer (2005) and Zhu et al. (2009). There are also more general approaches that can be applied to boosting including Allwein et al. (2000), Beygelzimer et al. (2009), Dietterich and Bakiri (1995), Hastie and Tibshirani (1998) and Li (2010). Two game-theoretic perspectives have been applied to boosting. The first one (Freund and Schapire, 1996b; Rätsch and Warmuth, 2005) views the weak-learning condition as a minimax game, while drifting games (Schapire, 2001; Freund, 1995) were designed to analyze the most efficient boosting algorithms. These games have been further analyzed in the multiclass and continuous time setting in Freund and Opper (2002).

2. Framework

We introduce some notation. Unless otherwise stated, matrices will be denoted by bold capital letters like $\mathbf{M}$, and vectors by bold small letters like $\mathbf{v}$. Entries of a matrix and vector will be denoted as $M(i,j)$ or $v(i)$, while $\mathbf{M}(i)$ will denote the ith row of a matrix. The inner product of two vectors $\mathbf{u},\mathbf{v}$ is denoted by $\langle \mathbf{u},\mathbf{v}\rangle$. The Frobenius inner product $\mathrm{Tr}(\mathbf{A}\mathbf{B}')$ of two matrices $\mathbf{A},\mathbf{B}$ will be denoted by $\mathbf{A}\bullet\mathbf{B}'$, where $\mathbf{B}'$ is the transpose of $\mathbf{B}$. The indicator function is denoted by $\mathbf{1}[\cdot]$. The set of all distributions over the set $\{1,\ldots,k\}$ will be denoted by $\Delta\{1,\ldots,k\}$, and in general, the set of all distributions over any set S will be denoted by $\Delta(S)$.

In multiclass classification, we want to predict the labels of examples lying in some set X. We are provided a training set of labeled examples $\{(x_1,y_1),\ldots,(x_m,y_m)\}$, where each example $x_i \in X$ has a label y_i in the set $\{1,\ldots,k\}$.

Boosting combines several mildly powerful predictors, called *weak classifiers*, to form a highly accurate combined classifier, and has been previously applied for multiclass classification. In this paper, we only allow weak classifiers that predict a single class for each example. This is appealing, since the combined classifier has the same form, although it differs from what has been used in much previous work. Later we will expand our framework to include *multilabel* weak classifiers, that may predict multiple labels per example.

We adopt a game-theoretic view of boosting. A game is played between two players, Booster and Weak-Learner, for a fixed number of rounds T. With binary labels, Booster outputs a distribution in each round, and Weak-Learner returns a weak classifier achieving more than 50% accuracy on that distribution. The multiclass game is an extension of the binary game. In particular, in each round t:

- Booster creates a cost-matrix $\mathbf{C}_t \in \mathbb{R}^{m\times k}$, specifying to Weak-Learner that the cost of classifying example x_i as l is $C_t(i,l)$. The cost-matrix may not be arbitrary, but should conform to certain restrictions as discussed below.

- Weak-Learner returns some weak classifier $h_t : X \to \{1,\ldots,k\}$ from a fixed space $h_t \in \mathcal{H}$ so that the cost incurred is
$$\mathbf{C}_t \bullet \mathbf{1}_{h_t} = \sum_{i=1}^{m} C_t(i, h_t(x_i)),$$
is "small enough", according to some conditions discussed below. Here by $\mathbf{1}_h$ we mean the $m \times k$ matrix whose (i,j)-th entry is $\mathbf{1}[h(i) = j]$.

- Booster computes a weight α_t for the current weak classifier based on how much cost was incurred in this round.

At the end, Booster predicts according to the weighted plurality vote of the classifiers returned in each round:

$$H(x) \triangleq \operatorname*{argmax}_{l\in\{1,\ldots,k\}} f_T(x,l), \text{ where } f_T(x,l) \triangleq \sum_{t=1}^{T} \mathbf{1}[h_t(x) = l]\,\alpha_t. \tag{1}$$

By carefully choosing the cost matrices in each round, Booster aims to minimize the training error of the final classifier H, even when Weak-Learner is adversarial. The restrictions on cost-matrices created by Booster, and the maximum cost Weak-Learner can suffer in each round, together define the *weak-learning condition* being used. For binary labels, the traditional weak-learning condition states: for any non-negative weights $w(1),\ldots,w(m)$ on the training set, the error of the weak classifier returned is at most $(1/2-\gamma/2)\sum_i w_i$. Here γ parametrizes the condition. There are many ways to translate this condition into our language. The one with fewest restrictions on the cost-matrices requires labeling correctly should be less costly than labeling incorrectly:

$$\forall i : C(i,y_i) \le C(i,\bar{y}_i) \text{ (here } \bar{y}_i \neq y_i \text{ is the other binary label)},$$

while the restriction on the returned weak classifier h requires less cost than predicting randomly:

$$\sum_i C(i,h(x_i)) \le \sum_i \left\{ \left(\frac{1}{2}-\frac{\gamma}{2}\right) C(i,\bar{y}_i) + \left(\frac{1}{2}+\frac{\gamma}{2}\right) C(i,y_i) \right\}.$$

By the correspondence $w(i) = C(i,\bar{y}_i) - C(i,y_i)$, we may verify the two conditions are the same.

We will rewrite this condition after making some simplifying assumptions. Henceforth, without loss of generality, we assume that the true label is always 1. Let $\mathcal{C}^{\text{bin}} \subseteq \mathbb{R}^{m\times 2}$ consist of matrices $\mathbf{C}$ which satisfy $C(i,1) \le C(i,2)$. Further, let $\mathbf{U}^{\text{bin}}_\gamma \in \mathbb{R}^{m\times 2}$ be the matrix whose each row is $(1/2+\gamma/2, 1/2-\gamma/2)$. Then, Weak-Learner searching space $\mathcal{H}$ satisfies the binary weak-learning condition if: $\forall \mathbf{C} \in \mathcal{C}^{\text{bin}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{U}^{\text{bin}}_\gamma\right) \le \mathbf{0}$. There are two main benefits to this reformulation. With linear homogeneous constraints, the mathematics is simplified, as will be apparent later. More importantly, by varying the restrictions $\mathcal{C}^{\text{bin}}$ on the cost vectors and the matrix $\mathbf{U}^{\text{bin}}$, we can generate a vast variety of weak-learning conditions for the multiclass setting $k \ge 2$ as we now show.

Let $\mathcal{C} \subseteq \mathbb{R}^{m\times k}$ and let $\mathbf{B} \in \mathbb{R}^{m\times k}$ be a matrix which we call the *baseline*. We say a weak classifier space $\mathcal{H}$ satisfies the condition $(\mathcal{C}, \mathbf{B})$ if

$$\forall \mathbf{C} \in \mathcal{C}, \exists h \in \mathcal{H} : \quad \mathbf{C} \bullet (\mathbf{1}_h - \mathbf{B}) \leq \mathbf{0}, \quad \text{i.e., } \sum_{i=1}^{m} C(i,h(i)) \leq \sum_{i=1}^{m} \langle \mathbf{C}(i), \mathbf{B}(i) \rangle . \tag{2}$$

In (2), the variable matrix $\mathbf{C}$ specifies how costly each misclassification is, while the baseline $\mathbf{B}$ specifies a weight for each misclassification. The condition therefore states that a weak classifier should not exceed the average cost when weighted according to baseline $\mathbf{B}$. This large class of weak-learning conditions captures many previously used conditions, such as the ones used by AdaBoost.M1 (Freund and Schapire, 1996a), AdaBoost.MH (Schapire and Singer, 1999) and AdaBoost.MR (Freund and Schapire, 1996a; Schapire and Singer, 1999) (see below), as well as novel conditions introduced in the next section.

By studying this vast class of weak-learning conditions, we hope to find the one that will serve the main purpose of the boosting game: finding a convex combination of weak classifiers that has zero training error. For this to be possible, at the minimum the weak classifiers should be sufficiently rich for such a perfect combination to exist. Formally, a collection $\mathcal{H}$ of weak classifiers is *boostable* if it is eligible for boosting in the sense that there exists a weighting $\boldsymbol{\lambda}$ on the votes forming a distribution that linearly separates the data: $\forall i : \operatorname{argmax}_{l\in\{1,\ldots,k\}} \sum_{h\in\mathcal{H}} \boldsymbol{\lambda}(h)\mathbf{1}[h(x_i) = l] = y_i$. The weak-learning condition plays two roles. It rejects spaces that are not boostable, and provides an algorithmic means of searching for the right combination. Ideally, the second factor will not cause the weak-learning condition to impose additional restrictions on the weak classifiers; in that case, the weak-learning condition is merely a reformulation of being boostable that is more appropriate for deriving an algorithm. In general, it could be *too strong*, that is, certain boostable spaces will fail to satisfy the conditions. Or it could be *too weak*, that is, non-boostable spaces might satisfy such a condition. Booster strategies relying on either of these conditions will fail to drive down error, the former due to underfitting, and the latter due to overfitting. Later we will describe conditions captured by our framework that avoid being too weak or too strong. But before that, we show in the next section how our flexible framework captures weak learning conditions that have appeared previously in the literature.

3. Old Conditions

In this section, we rewrite, in the language of our framework, the weak learning conditions explicitly or implicitly employed in the multiclass boosting algorithms SAMME (Zhu et al., 2009), AdaBoost.M1 (Freund and Schapire, 1996a), and AdaBoost.MH and AdaBoost.MR (Schapire and Singer, 1999). This will be useful later on for comparing the strengths and weaknesses of the various conditions. We will end this section with a curious equivalence between the conditions of AdaBoost.MH and AdaBoost.M1.

Recall that we have assumed the correct label is 1 for every example. Nevertheless, we continue to use y_i to denote the correct label in this section.

3.1 Old Conditions in the New Framework

Here we restate, in the language of our new framework, the weak learning conditions of four algorithms that have earlier appeared in the literature.

3.1.1 SAMME

The SAMME algorithm (Zhu et al., 2009) requires less error than random guessing on any distribution on the examples. Formally, a space $\mathcal{H}$ satisfies the condition if there is a $\gamma' > 0$ such that,

$$\forall d(1),\ldots,d(m) \geq 0, \exists h \in \mathcal{H} : \sum_{i=1}^{m} d(i)\mathbf{1}\left[h(x_i) \neq y_i\right] \leq (1 - 1/k - \gamma')\sum_{i=1}^{m} d(i). \tag{3}$$

Define a cost matrix $\mathbf{C}$ whose entries are given by

$$C(i,j) = \begin{cases} d(i) & \text{if } j \neq y_i, \\ 0 & \text{if } j = y_i. \end{cases}$$

Then the left hand side of (3) can be written as

$$\sum_{i=1}^{m} C(i,h(x_i)) = \mathbf{C} \bullet \mathbf{1}_h.$$

Next let $\gamma = (k/(k-1))\gamma'$ and define baseline $\mathbf{U}_\gamma$ to be the multiclass extension of $\mathbf{U}^{\text{bin}}$,

$$U_\gamma(i,l) = \begin{cases} \frac{(1-\gamma)}{k} + \gamma & \text{if } l = y_i, \\ \frac{(1-\gamma)}{k} & \text{if } l \neq y_i. \end{cases}$$

Then notice that for each i, we have

$$\begin{aligned} \left\langle \mathbf{C}(i), \mathbf{U}_\gamma(i) \right\rangle &= \sum_{l \neq y_i} C(i,l) U_\gamma(i,l) \\ &= (k-1)\frac{(1-\gamma)}{k} d(i) \\ &= \left(1 - \frac{1}{k} - \left(1 - \frac{1}{k}\right)\gamma\right) d(i) \\ &= \left(1 - \frac{1}{k} - \gamma'\right) d(i). \end{aligned}$$

Therefore, the right hand side of (3) can be written as

$$\sum_{i=1}^{m} \sum_{l \neq y_i} C(i,l) U_\gamma(i,l) = \mathbf{C} \bullet \mathbf{U}_\gamma,$$

since $C(i,y_i) = 0$ for every example i. Define $\mathcal{C}^{\text{SAM}}$ to be the following collection of cost matrices:

$$\mathcal{C}^{\text{SAM}} \triangleq \left\{ \mathbf{C} : C(i,l) = \begin{cases} 0 & \text{if } l = y_i, \\ t_i & \text{if } l \neq y_i, \end{cases} \text{ for non-negative } t_1,\ldots,t_m. \right\}$$

Using the last two equations, (3) is equivalent to

$$\forall \mathbf{C} \in \mathcal{C}^{\text{SAM}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{U}_\gamma\right) \leq 0.$$

Therefore, the weak-learning condition of SAMME is given by $(\mathcal{C}^{\text{SAM}}, \mathbf{U}_\gamma)$.

3.1.2 ADABOOST.M1

AdaBoost.M1 (Freund and Schapire, 1997) measures the performance of weak classifiers using ordinary error. It requires $1/2+\gamma/2$ accuracy with respect to any non-negative weights $d(1),\ldots,d(m)$ on the training set:

$$\begin{aligned} \sum_{i=1}^{m} d(i)\mathbf{1}\left[h(x_i)\neq y_i\right] &\leq (1/2-\gamma/2)\sum_{i=1}^{m} d(i), \\ \text{i.e., } \sum_{i=1}^{m} d(i)[\![h(x_i)\neq y_i]\!] &\leq -\gamma\sum_{i=1}^{m} d(i). \end{aligned} \tag{4}$$

where $[\![\cdot]\!]$ is the ± 1 indicator function, taking value $+1$ when its argument is true, and -1 when false. Using the transformation

$$C(i,l) = [\![l\neq y_i]\!]d(i)$$

we may rewrite (4) as

$$\begin{gathered} \forall C\in\mathbb{R}^{m\times k} \text{ satisfying } 0\leq -C(i,y_i)=C(i,l) \text{ for } l\neq y_i, \\ \exists h\in\mathcal{H}: \sum_{i=1}^{m} C(i,h(x_i)) \leq \gamma\sum_{i=1}^{m} C(i,y_i) \\ \text{i.e.,}\quad \forall \mathbf{C}\in\mathcal{C}^{\mathrm{M1}}, \exists h\in\mathcal{H}: \mathbf{C}\bullet\left(\mathbf{1}_h-\mathbf{B}_{\gamma}^{\mathrm{M1}}\right)\leq 0, \end{gathered} \tag{5}$$

where $\mathbf{B}_{\gamma}^{\mathrm{M1}}(i,l)=\gamma\mathbf{1}\left[l=y_i\right]$, and $\mathcal{C}^{\mathrm{M1}}\subseteq\mathbb{R}^{m\times k}$ consists of matrices satisfying the constraints in (5).

3.1.3 ADABOOST.MH

AdaBoost.MH (Schapire and Singer, 1999) is a popular multiclass boosting algorithm that is based on the one-against-all reduction, and was originally designed to use weak-hypotheses that return a prediction for every example and every label. The implicit weak learning condition requires that for any matrix with non-negative entries $d(i,l)$, the weak-hypothesis should achieve $1/2+\gamma$ accuracy

$$\sum_{i=1}^{m}\left\{\mathbf{1}\left[h(x_i)\neq y_i\right]d(i,y_i)+\sum_{l\neq y_i}\mathbf{1}\left[h(x_i)=l\right]d(i,l)\right\} \leq \left(\frac{1}{2}-\frac{\gamma}{2}\right)\sum_{i=1}^{m}\sum_{l=1}^{k} d(i,l). \tag{6}$$

This can be rewritten as

$$\begin{aligned} &\sum_{i=1}^{m}\left\{-\mathbf{1}\left[h(x_i)=y_i\right]d(i,y_i)+\sum_{l\neq y_i}\mathbf{1}\left[h(x_i)=l\right]d(i,l)\right\} \\ &\leq \sum_{i=1}^{m}\left\{\left(\frac{1}{2}-\frac{\gamma}{2}\right)\sum_{l\neq y_i} d(i,l)-\left(\frac{1}{2}+\frac{\gamma}{2}\right)d(i,y_i)\right\}. \end{aligned}$$

Using the mapping

$$C(i,l)=\begin{cases} d(i,l) & \text{if } l\neq y_i \\ -d(i,l) & \text{if } l=y_i, \end{cases}$$

their weak-learning condition may be rewritten as follows

$$\begin{aligned}&\forall \mathbf{C} \in \mathbb{R}^{m\times k} \text{ satisfying } C(i,y_i) \le 0, C(i,l) \ge 0 \text{ for } l \ne y_i, \\ &\quad \exists h \in \mathcal{H}: \\ &\quad sum_{i=1}^{m} C(i,h(x_i)) \le \sum_{i=1}^{m} \left\{ \left(\frac{1}{2}+\frac{\gamma}{2}\right) C(i,y_i) + \left(\frac{1}{2}-\frac{\gamma}{2}\right) \sum_{l\ne y_i} C(i,l) \right\}.\end{aligned} \tag{7}$$

Defining $\mathcal{C}^{\mathrm{MH}}$ to be the space of all cost matrices satisfying the constraints in (7), the above condition is the same as

$$\forall \mathbf{C} \in \mathcal{C}^{\mathrm{MH}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}_\gamma^{\mathrm{MH}}\right) \le 0,$$

where $\mathbf{B}_\gamma^{\mathrm{MH}}(i,l) = (1/2 + \gamma[\![l = y_i]\!]/2)$.

3.1.4 ADABOOST.MR

AdaBoost.MR (Schapire and Singer, 1999) is based on the all-pairs multiclass to binary reduction. Like AdaBoost.MH, it was originally designed to use weak-hypotheses that return a prediction for every example and every label. The weak learning condition for AdaBoost.MR requires that for any non-negative cost-vectors $\{d(i,l)\}_{l\ne y_i}$, the weak-hypothesis returned should satisfy the following:

$$\begin{aligned}\sum_{i=1}^{m} \sum_{l\ne y_i} \left(\mathbf{1}\left[h(x_i)=l\right] - \mathbf{1}\left[h(x_i)=y_i\right]\right) d(i,l) &\le -\gamma \sum_{i=1}^{m} \sum_{l\ne y_i} d(i,l) \\ \text{i.e., } \sum_{i=1}^{m} \left\{ -\mathbf{1}\left[h(x_i)=y_i\right] \sum_{l\ne y_i} d(i,l) + \sum_{l\ne y_i} \mathbf{1}\left[h(x_i)=l\right] d(i,l) \right\} &\le -\gamma \sum_{i=1}^{m} \sum_{l\ne y_i} d(i,l).\end{aligned}$$

Substituting

$$C(i,l) = \begin{cases} d(i,l) & l \ne y_i \\ -\sum_{l\ne y_i} d(i,l) & l = y_i, \end{cases}$$

we may rewrite AdaBoost.MR's weak-learning condition as

$$\begin{aligned}&\forall \mathbf{C} \in \mathbb{R}^{m\times k} \text{ satisfying } C(i,l) \ge 0 \text{ for } l \ne y_i, C(i,y_i) = -\sum_{l\ne y_i} C(i,l), \\ &\exists h \in \mathcal{H} : \sum_{i=1}^{m} C(i,h(x_i)) \le -\frac{\gamma}{2} \sum_{i=1}^{m} \left\{ -C(i,y_i) + \sum_{l\ne y_i} C(i,l) \right\}.\end{aligned} \tag{8}$$

Defining $\mathcal{C}^{\mathrm{MR}}$ to be the collection of cost matrices satisfying the constraints in (8), the above condition is the same as

$$\forall \mathbf{C} \in \mathcal{C}^{\mathrm{MR}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}_\gamma^{\mathrm{MR}}\right) \le 0,$$

where $\mathbf{B}_\gamma^{\mathrm{MR}}(i,l) = [\![l = y_i]\!]\gamma/2$.

3.2 A Curious Equivalence

We show that the weak learning conditions of AdaBoost.MH and AdaBoost.M1 are identical in our framework. This is surprising because the original motivations behind these algorithms were completely different. AdaBoost.M1 is a direct extension of binary AdaBoost to the multiclass setting,

whereas AdaBoost.MH is based on the one-against-all multiclass to binary reduction. This equivalence is a sort of degeneracy, and arises because the weak classifiers being used predict single labels per example. With multilabel weak classifiers, for which AdaBoost.MH was originally designed, the equivalence no longer holds.

The proofs in this and later sections will make use of the following minimax result, that is a weaker version of Corollary 37.3.2 of Rockafellar (1970).

Theorem 1 *(Minimax Theorem) Let C, D be non-empty closed convex subsets of $\mathbb{R}^m, \mathbb{R}^n$ respectively, and let K be a linear function on $C \times D$. If either C or D is bounded, then*

$$\min_{v \in D} \max_{u \in C} K(u,v) = \max_{u \in C} \min_{v \in D} K(u,v).$$

Lemma 2 *A weak classifier space $\mathcal{H}$ satisfies $(\mathcal{C}^{M1}, \mathbf{B}_\gamma^{M1})$ if and only if it satisfies $(\mathcal{C}^{MH}, \mathbf{B}_\gamma^{MH})$.*

Note that $\mathcal{C}^{\mathrm{M1}}$ and $\mathcal{C}^{\mathrm{MH}}$ depend implicitly on the training set. This lemma is valid for all training sets.

Proof We will refer to $(\mathcal{C}^{\mathrm{M1}}, \mathbf{B}_\gamma^{\mathrm{M1}})$ by M1 and $(\mathcal{C}^{\mathrm{MH}}, \mathbf{B}_\gamma^{\mathrm{MH}})$ by MH for brevity. The proof is in three steps.

Step (i): If $\mathcal{H}$ satisfies MH, then it also satisfies M1. This follows since any constraint (4) imposed by M1 on $\mathcal{H}$ can be reproduced by MH by plugging the following values of $d(i,l)$ in (6)

$$d(i,l) = \begin{cases} d(i) & \text{if } l = y_i \\ 0 & \text{if } l \neq y_i. \end{cases}$$

Step (ii): If $\mathcal{H}$ satisfies M1, then there is a convex combination $\mathbf{H}_{\boldsymbol{\lambda}^*}$ of the matrices $\mathbf{1}_h \in \mathcal{H}$, defined as

$$\mathbf{H}_{\boldsymbol{\lambda}^*} \stackrel{\triangle}{=} \sum_{h \in \mathcal{H}} \lambda^*(h) \mathbf{1}_h,$$

such that

$$\forall i : \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}_\gamma^{\mathrm{M1}}\right)(i,l) \begin{cases} \geq 0 & \text{if } l = y_i \\ \leq 0 & \text{if } l \neq y_i. \end{cases} \tag{9}$$

Indeed, Theorem 1 yields

$$\min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{M1}}} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}_\gamma^{\mathrm{M1}}\right) = \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{M1}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}_\gamma^{\mathrm{M1}}\right) \leq 0, \tag{10}$$

where the inequality is a restatement of our assumption that $\mathcal{H}$ satisfies M1. If $\boldsymbol{\lambda}^*$ is a minimizer of the minimax expression, then $\mathbf{H}_{\boldsymbol{\lambda}^*}$ must satisfy

$$\forall i : \mathbf{H}_{\boldsymbol{\lambda}^*}(i,l) \begin{cases} \geq \frac{1}{2} + \frac{\gamma}{2} & \text{if } l = y_i \\ \leq \frac{1}{2} - \frac{\gamma}{2} & \text{if } l \neq y_i, \end{cases} \tag{11}$$

or else some choice of $\mathbf{C} \in \mathcal{C}^{\mathrm{M1}}$ can cause $\mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}_\gamma^{\mathrm{M1}}\right)$ to exceed 0. In particular, if $\mathbf{H}_{\boldsymbol{\lambda}^*}(i_0, l) < 1/2 + \gamma/2$, then

$$\left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}_\gamma^{\mathrm{M1}}\right)(i_0, y_{i_0}) < \sum_{l \neq y_{i_0}} \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}_\gamma^{\mathrm{M1}}\right)(i_0, l).$$

Now, if we choose $\mathbf{C} \in \mathcal{C}^{\mathrm{M1}}$ as

$$C(i,l) = \begin{cases} 0 & \text{if } i \neq i_0 \\ 1 & \text{if } i = i_0, l \neq y_{i_0} \\ -1 & \text{if } i = i_0, l = y_{i_0}, \end{cases}$$

then,

$$\mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}^{\mathrm{M1}}_{\gamma}\right) = -\left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}^{\mathrm{M1}}_{\gamma}\right)(i_0, y_{i_0}) + \sum_{l \neq y_{i_0}} \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}^{\mathrm{M1}}_{\gamma}\right)(i_0, l) > 0,$$

contradicting the inequality in (10). Therefore (11) holds. Equation (9), and thus Step (ii), now follows by observing that $\mathbf{B}^{\mathrm{MH}}_{\gamma}$, by definition, satisfies

$$\forall i : \mathbf{B}^{\mathrm{MH}}_{\gamma}(i,l) = \begin{cases} \frac{1}{2} + \frac{\gamma}{2} & \text{if } l = y_i \\ \frac{1}{2} - \frac{\gamma}{2} & \text{if } l \neq y_i. \end{cases}$$

Step (iii) If there is some convex combination $\mathcal{H}_{\boldsymbol{\lambda}^*}$ satisfying (9), then $\mathcal{H}$ satisfies MH. Recall that $\mathbf{B}^{\mathrm{MH}}$ consists of entries that are non-positive on the correct labels and non-negative for incorrect labels. Therefore, (9) implies

$$0 \geq \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}^{\mathrm{MH}}_{\gamma}\right) \geq \min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}^{\mathrm{MH}}_{\gamma}\right).$$

On the other hand, using Theorem 1 we have

$$\min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}^{\mathrm{MH}}_{\gamma}\right) = \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}^{\mathrm{MH}}_{\gamma}\right).$$

Combining the two, we get

$$0 \geq \max_{\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}^{\mathrm{MH}}_{\gamma}\right),$$

which is the same as saying that $\mathcal{H}$ satisfies MH's condition.

Steps (ii) and (iii) together imply that if $\mathcal{H}$ satisfies M1, then it also satisfies MH. Along with Step (i), this concludes the proof. ■

4. Necessary and Sufficient Weak-learning Conditions

The binary weak-learning condition has an appealing form: for any distribution over the examples, the weak classifier needs to achieve error not greater than that of a random player who guesses the correct answer with probability $1/2+\gamma/2$. Further, this is the weakest condition under which boosting is possible as follows from a game-theoretic perspective (Freund and Schapire, 1996b; Rätsch and Warmuth, 2005) . Multiclass weak-learning conditions with similar properties are missing in the literature. In this section we show how our framework captures such conditions.

4.1 Edge-over-random Conditions

In the multiclass setting, we model a random player as a baseline predictor $\mathbf{B} \in \mathbb{R}^{m \times k}$ whose rows are distributions over the labels, $\mathbf{B}(i) \in \Delta\{1,\ldots,k\}$. The prediction on example i is a sample from $\mathbf{B}(i)$. We only consider the space of *edge-over-random* baselines $\mathcal{B}_\gamma^{\text{eor}} \subseteq \mathbb{R}^{m \times k}$ who have a faint clue about the correct answer. More precisely, any baseline $\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}$ in this space is γ more likely to predict the correct label than an incorrect one on every example i: $\forall l \neq 1, B(i,1) \geq B(i,l) + \gamma$, with equality holding for some l, that is:

$$B(i,1) = \max\{B(i,l) + \gamma : l \neq 1\}.$$

Notice that the edge-over-random baselines are different from the baselines used by earlier weak learning conditions discussed in the previous section.

When $k = 2$, the space $\mathcal{B}_\gamma^{\text{eor}}$ consists of the unique player $\mathbf{U}_\gamma^{\text{bin}}$, and the binary weak-learning condition is given by $(\mathcal{C}^{\text{bin}}, \mathbf{U}_\gamma^{\text{bin}})$. The new conditions generalize this to $k > 2$. In particular, define $\mathcal{C}^{\text{eor}}$ to be the multiclass extension of $\mathcal{C}^{\text{bin}}$: any cost-matrix in $\mathcal{C}^{\text{eor}}$ should put the least cost on the correct label, that is, the rows of the cost-matrices should come from the set $\left\{\mathbf{c} \in \mathbb{R}^k : \forall l, c(1) \leq c(l)\right\}$. Then, for every baseline $\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}$, we introduce the condition $(\mathcal{C}^{\text{eor}}, \mathbf{B})$, which we call an *edge-over-random* weak-learning condition. Since $\mathbf{C} \bullet \mathbf{B}$ is the expected cost of the edge-over-random baseline $\mathbf{B}$ on matrix $\mathbf{C}$, the constraints (2) imposed by the new condition essentially require better than random performance.

Also recall that we have assumed that the true label y_i of example i in our training set is always 1. Nevertheless, we may occasionally continue to refer to the true labels as y_i.

We now present the central results of this section. The seemingly mild edge-over-random conditions guarantee boostability, meaning weak classifiers that satisfy any one such condition can be combined to form a highly accurate combined classifier.

Theorem 3 (Sufficiency) *If a weak classifier space $\mathcal{H}$ satisfies a weak-learning condition $(\mathcal{C}^{eor}, \mathbf{B})$, for some $\mathbf{B} \in \mathcal{B}_\gamma^{eor}$, then $\mathcal{H}$ is boostable.*

Proof The proof is in the spirit of the ones in Freund and Schapire (1996b). Applying Theorem 1 yields

$$0 \geq \max_{\mathbf{C} \in \mathcal{C}^{\text{eor}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet (\mathbf{1}_h - \mathbf{B}) = \min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\text{eor}}} \mathbf{C} \bullet (\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}),$$

where the first inequality follows from the definition (2) of the weak-learning condition. Let $\boldsymbol{\lambda}^*$ be a minimizer of the min-max expression. Unless the first entry of each row of $(\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B})$ is the largest, the right hand side of the min-max expression can be made arbitrarily large by choosing $\mathbf{C} \in \mathcal{C}^{\text{eor}}$ appropriately. For example, if in some row i, the j_0^{th} element is strictly larger than the first element, by choosing

$$C(i,j) = \begin{cases} -1 & \text{if } j = 1 \\ 1 & \text{if } j = j_0 \\ 0 & \text{otherwise,} \end{cases}$$

we get a matrix in $\mathcal{C}^{\text{eor}}$ which causes $\mathbf{C} \bullet (\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B})$ to be equal to $C(i,j_0) - C(i,1) > 0$, an impossibility by the first inequality.

Therefore, the convex combination of the weak classifiers, obtained by choosing each weak classifier with weight given by $\boldsymbol{\lambda}^*$, perfectly classifies the training data, in fact with a margin γ. ■

On the other hand, the family of such conditions, taken as a whole, is necessary for boostability in the sense that every eligible space of weak classifiers satisfies some edge-over-random condition.

Theorem 4 (Relaxed necessity) *For every boostable weak classifier space $\mathcal{H}$, there exists a $\gamma > 0$ and $\mathbf{B} \in \mathcal{B}_\gamma^{eor}$ such that $\mathcal{H}$ satisfies the weak-learning condition $(\mathcal{C}^{eor}, \mathbf{B})$.*

Proof The proof shows existence through non-constructive averaging arguments. We will reuse notation from the proof of Theorem 3 above. $\mathcal{H}$ is boostable implies there exists some distribution $\boldsymbol{\lambda}^* \in \Delta(\mathcal{H})$ such that

$$\forall j \neq 1, i : \mathbf{H}_{\boldsymbol{\lambda}^*}(i,1) - \mathbf{H}_{\boldsymbol{\lambda}^*}(i,j) > 0.$$

Let $\gamma > 0$ be the minimum of the above expression over all possible (i, j), and let $\mathbf{B} = \mathbf{H}_{\boldsymbol{\lambda}^*}$. Then $\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}$, and

$$\max_{\mathbf{C} \in \mathcal{C}^{\text{eor}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet (\mathbf{1}_h - \mathbf{B}) \leq \min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\text{eor}}} \mathbf{C} \bullet (\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}) \leq \max_{\mathbf{C} \in \mathcal{C}^{\text{eor}}} \mathbf{C} \bullet (\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B}) = 0,$$

where the equality follows since by definition $\mathbf{H}_{\boldsymbol{\lambda}^*} - \mathbf{B} = \mathbf{0}$. The max-min expression is at most zero is another way of saying that $\mathcal{H}$ satisfies the weak-learning condition $(\mathcal{C}^{\text{eor}}, \mathbf{B})$ as in (2). ■

Theorem 4 states that any boostable weak classifier space will satisfy some condition in our family, but it does not help us choose the right condition. Experiments in Section 10 suggest $\left(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma\right)$ is effective with very simple weak-learners compared to popular boosting algorithms. (Recall $\mathbf{U}_\gamma \in \mathcal{B}_\gamma^{\text{eor}}$ is the edge-over-random baseline closest to uniform; it has weight $(1-\gamma)/k$ on incorrect labels and $(1-\gamma)/k + \gamma$ on the correct label.) However, there are theoretical examples showing each condition in our family is too strong.

Theorem 5 *For any $\mathbf{B} \in \mathcal{B}_\gamma^{eor}$, there exists a boostable space $\mathcal{H}$ that fails to satisfy the condition $(\mathcal{C}^{eor}, \mathbf{B})$.*

Proof We provide, for any $\gamma > 0$ and edge-over-random baseline $\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}$, a data set and weak classifier space that is boostable but fails to satisfy the condition $(\mathcal{C}^{\text{eor}}, \mathbf{B})$.

Pick $\gamma' = \gamma/k$ and set $m > 1/\gamma'$ so that $\lfloor m(1/2 + \gamma') \rfloor > m/2$. Our data set will have m labeled examples $\{(0, y_0), \ldots, (m-1, y_{m-1})\}$, and m weak classifiers. We want the following symmetries in our weak classifiers:

- Each weak classifier correctly classifies $\lfloor m(1/2 + \gamma') \rfloor$ examples and misclassifies the rest.
- On each example, $\lfloor m(1/2 + \gamma') \rfloor$ weak classifiers predict correctly.

Note the second property implies boostability, since the uniform convex combination of all the weak classifiers is a perfect predictor.

The two properties can be satisfied by the following design. A window is a contiguous sequence of examples that may wrap around; for example

$$\{i, (i+1) \mod m, \ldots, (i+k) \mod m\}$$

is a window containing k elements, which may wrap around if $i + k \geq m$. For each window of length $\lfloor m(1/2 + \gamma') \rfloor$ create a hypothesis that correctly classifies within the window, and misclassifies outside. This weak-hypothesis space has size m, and has the required properties.

We still have flexibility as to how the misclassifications occur, and which cost-matrix to use, which brings us to the next two choices:

- Whenever a hypothesis misclassifies on example i, it predicts label

$$\hat{y}_i \stackrel{\triangle}{=} \operatorname{argmin}\{B(i,l) : l \neq y_i\}. \tag{12}$$

- A cost-matrix is chosen so that the cost of predicting $\hat{y}_i$ on example i is 1, but for any other prediction the cost is zero. Observe this cost-matrix belongs to $\mathcal{C}^{\text{eor}}$.

Therefore, every time a weak classifier predicts incorrectly, it also suffers cost 1. Since each weak classifier predicts correctly only within a window of length $\lfloor m(1/2+\gamma')\rfloor$, it suffers cost $\lceil m(1/2-\gamma')\rceil$. On the other hand, by the choice of $\hat{y}_i$ in (12), and by our assumption that $y_i = 1$, we have

$$\begin{aligned} B(i,\hat{y}_i) &= \min\{B(i,1)-\gamma, B(i,2), \ldots, B(i,k)\} \\ &\leq \frac{1}{k}\left(B(i,1)-\gamma+B(i,2)+B(i,3)+\ldots+B(i,k)\right) \\ &= 1/k-\gamma/k. \end{aligned}$$

So the cost of $\mathbf{B}$ on the chosen cost-matrix is at most $m(1/k-\gamma/k)$, which is less than the cost $\lceil m(1/2-\gamma')\rceil \geq m(1/2-\gamma/k)$ of any weak classifier whenever the number of labels k is more than two. Hence our boostable space of weak classifiers fails to satisfy $(\mathcal{C}^{\text{eor}}, \mathbf{B})$. ■

Theorems 4 and 5 can be interpreted as follows. While a boostable space will satisfy *some* edge-over-random condition, without further information about the data set it is not possible to know *which* particular condition will be satisfied. The kind of prior knowledge required to make this guess correctly is provided by Theorem 3: the appropriate weak learning condition is determined by the distribution of votes on the labels for each example that a target weak classifier combination might be able to get. Even with domain expertise, such knowledge may or may not be obtainable in practice before running boosting. We therefore need conditions that assume less.

4.2 The Minimal Weak Learning Condition

A perhaps extreme way of weakening the condition is by requiring the performance on a cost matrix to be competitive not with a *fixed* baseline $\mathbf{B} \in \mathcal{B}^{\text{eor}}_{\gamma}$, but with the *worst* of them:

$$\forall \mathbf{C} \in \mathcal{C}^{\text{eor}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \mathbf{1}_h \leq \max_{\mathbf{B} \in \mathcal{B}^{\text{eor}}_{\gamma}} \mathbf{C} \bullet \mathbf{B}. \tag{13}$$

Condition (13) states that during the course of the same boosting game, Weak-Learner may choose to beat *any* edge-over-random baseline $\mathbf{B} \in \mathcal{B}^{\text{eor}}_{\gamma}$, possibly a different one for every round and every cost-matrix. This may superficially seem much too weak. On the contrary, this condition turns out to be equivalent to boostability. In other words, according to our criterion, it is neither too weak nor too strong as a weak-learning condition. However, unlike the edge-over-random conditions, it also turns out to be more difficult to work with algorithmically.

Furthermore, this condition can be shown to be equivalent to the one used by AdaBoost.MR (Schapire and Singer, 1999; Freund and Schapire, 1996a). This is perhaps remarkable since the latter is based on the apparently completely unrelated all-pairs multiclass to binary reduction. In Section 3 we saw that the MR condition is given by $(\mathcal{C}^{\text{MR}}, \mathbf{B}^{\text{MR}}_{\gamma})$, where $\mathcal{C}^{\text{MR}}$ consists of cost-matrices that put non-negative costs on incorrect labels and whose rows sum up to zero, while $\mathbf{B}^{\text{MR}}_{\gamma} \in \mathbb{R}^{m \times k}$ is the matrix that has γ on the first column and $-\gamma$ on all other columns. Further, the MR condition, and hence (13), can be shown to be neither too weak nor too strong.

Theorem 6 (MR) *A weak classifier space $\mathcal{H}$ satisfies AdaBoost.MR's weak-learning condition $(\mathcal{C}^{MR}, \mathbf{B}_\gamma^{MR})$ if and only if it satisfies* (13)*. Moreover, this condition is equivalent to being boostable.*

Proof We will show the following three conditions are equivalent:

(A) $\mathcal{H}$ is boostable

(B) $\exists \gamma > 0$ such that $\forall \mathbf{C} \in \mathcal{C}^{\text{eor}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \mathbf{1}_h \leq \max_{\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}} \mathbf{C} \bullet \mathbf{B}$

(C) $\exists \gamma > 0$ such that $\forall \mathbf{C} \in \mathcal{C}^{\text{MR}}, \exists h \in \mathcal{H} : \mathbf{C} \bullet \mathbf{1}_h \leq \mathbf{C} \bullet \mathbf{B}^{\text{MR}}$.

We will show (A) implies (B), (B) implies (C), and (C) implies (A) to achieve the above.

(A) implies (B): Immediate from Theorem 4.

(B) implies (C): Suppose (B) is satisfied with 2γ. We will show that this implies $\mathcal{H}$ satisfies $(\mathcal{C}^{\text{MR}}, \mathbf{B}_\gamma^{\text{MR}})$. Notice $\mathcal{C}^{\text{MR}} \subset \mathcal{C}^{\text{eor}}$. Therefore it suffices to show that

$$\forall \mathbf{C} \in \mathcal{C}^{\text{MR}}, \mathbf{B} \in \mathcal{B}_{2\gamma}^{\text{eor}} : \mathbf{C} \bullet \left(\mathbf{B} - \mathbf{B}_\gamma^{\text{MR}}\right) \leq 0.$$

Notice that $\mathbf{B} \in \mathcal{B}_{2\gamma}^{\text{eor}}$ implies $\mathbf{B}' = \mathbf{B} - \mathbf{B}_\gamma^{\text{MR}}$ is a matrix whose largest entry in each row is in the first column of that row. Then, for any $\mathbf{C} \in \mathcal{C}^{\text{MR}}$, $\mathbf{C} \bullet \mathbf{B}'$ can be written as

$$\mathbf{C} \bullet \mathbf{B}' = \sum_{i=1}^{m} \sum_{j=2}^{k} C(i,j) \left(B'(i,j) - B'(i,1)\right).$$

Since $C(i,j) \geq 0$ for $j > 1$, and $B'(i,j) - B'(i,1) \leq 0$, we have our result.

(C) implies (A): Applying Theorem 1

$$\begin{aligned} 0 &\geq \max_{\mathbf{C} \in \mathcal{C}^{\text{MR}}} \min_{h \in \mathcal{H}} \mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}_\gamma^{\text{MR}}\right) \\ &\geq \max_{\mathbf{C} \in \mathcal{C}^{\text{MR}}} \min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}_\gamma^{\text{MR}}\right) \\ &= \min_{\boldsymbol{\lambda} \in \Delta(\mathcal{H})} \max_{\mathbf{C} \in \mathcal{C}^{\text{MR}}} \mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}_\gamma^{\text{MR}}\right). \end{aligned}$$

For any i_0 and $l_0 \neq 1$, the following cost-matrix $\mathbf{C}$ satisfies $\mathbf{C} \in \mathcal{C}^{\text{MR}}$,

$$C(i,l) = \begin{cases} 0 & \text{if } i \neq i_0 \text{ or } l \notin \{1, l_0\} \\ 1 & \text{if } i = i_0, l = l_0 \\ -1 & \text{if } i = i_0, l = 1. \end{cases}$$

Let $\boldsymbol{\lambda}$ belong to the argmin of the min max expression. Then $\mathbf{C} \bullet \left(\mathbf{H}_{\boldsymbol{\lambda}} - \mathbf{B}_\gamma^{\text{MR}}\right) \leq 0$ implies $\mathbf{H}_{\boldsymbol{\lambda}}(i_0, 1) - \mathbf{H}_{\boldsymbol{\lambda}}(i_0, l_0) \geq 2\gamma$. Since this is true for all i_0 and $l_0 \neq 1$, we conclude that the $(\mathcal{C}^{\text{MR}}, \mathbf{B}_\gamma^{\text{MR}})$ condition implies boostability.

This concludes the proof of equivalence. ■

Next, we illustrate the strengths of our minimal weak-learning condition through concrete comparisons with previous algorithms.

	h_1	h_2
a	1	2
b	1	2

Figure 1: A weak classifier space which satisfies SAMME's weak learning condition but is not boostable.

4.2.1 COMPARISON WITH SAMME

The SAMME algorithm of Zhu et al. (2009) requires the weak classifiers to achieve less error than uniform random guessing for multiple labels; in our language, their weak-learning condition is $(\mathcal{C}^{\text{SAM}}, \mathbf{U}_\gamma)$, as shown in Section 3, where $\mathcal{C}^{\text{SAM}}$ consists of cost matrices whose rows are of the form $(0,t,t,\ldots)$ for some non-negative t. As is well-known, this condition is not sufficient for boosting to be possible. In particular, consider the data set $\{(a,1),(b,2)\}$ with $k=3, m=2$, and a weak classifier space consisting of h_1, h_2 which always predict $1,2$, respectively (Figure 1). Since neither classifier distinguishes between a,b we cannot achieve perfect accuracy by combining them in any way. Yet, due to the constraints on the cost-matrix, one of h_1, h_2 will always manage non-positive cost while random always suffers positive cost. On the other hand our weak-learning condition allows the Booster to choose far richer cost matrices. In particular, when the cost matrix $\mathbf{C} \in \mathcal{C}^{\text{eor}}$ is given by

	1	2	3
a	-1	$+1$	0
b	$+1$	-1	0,

both classifiers in the above example suffer more loss than the random player $\mathbf{U}_\gamma$, and fail to satisfy our condition.

4.2.2 COMPARISON WITH ADABOOST.MH

AdaBoost.MH (Schapire and Singer, 1999) was designed for use with weak hypotheses that on each example return a prediction for every label. When used in our framework, where the weak classifiers return only a single multiclass prediction per example, the implicit demands made by AdaBoost.MH on the weak classifier space turn out to be too strong. We cannot use Theorem 5 to demonstrate this, since it applies to only fixed edge-over-random conditions. Instead, we construct a classifier space that satisfies the condition $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$ in our family, but cannot satisfy AdaBoost.MH's weak-learning condition. Note that this does not imply that the conditions are too strong when used with more powerful weak classifiers that return multilabel multiclass predictions.

Consider a space $\mathcal{H}$ that has, for every $(1/k+\gamma)m$ element subset of the examples, a classifier that predicts correctly on exactly those elements. The expected loss of a randomly chosen classifier from this space is the same as that of the random player $\mathbf{U}_\gamma$. Hence $\mathcal{H}$ satisfies this weak-learning condition. On the other hand, it was shown in Section 3 that AdaBoost.MH's weak-learning condition is the pair $(\mathcal{C}^{\text{MH}}, \mathbf{B}_\gamma^{\text{MH}})$, where $\mathcal{C}^{\text{MH}}$ consists of cost matrices with non-negative entries on incorrect labels and non-positive entries on real labels, and where each row of the matrix $\mathbf{B}_\gamma^{\text{MH}}$ is the vector $(1/2+\gamma/2, 1/2-\gamma/2, \ldots, 1/2-\gamma/2)$. A quick calculation shows that for any $h \in \mathcal{H}$, and

$\mathbf{C} \in \mathcal{C}^{\mathrm{MH}}$ with -1 in the first column and zeroes elsewhere, $\mathbf{C} \bullet \left(\mathbf{1}_h - \mathbf{B}_\gamma^{\mathrm{MH}}\right) = 1/2 - 1/k$. This is positive when $k > 2$, so that $\mathcal{H}$ fails to satisfy AdaBoost.MH's condition.

We have seen how our framework allows us to capture the strengths and weaknesses of old conditions, describe a whole new family of conditions and also identify the condition making minimal assumptions. In the next few sections, we show how to design boosting algorithms that employ these new conditions and enjoy strong theoretical guarantees.

5. Algorithms

In this section we devise algorithms by analyzing the boosting games that employ weak-learning conditions in our framework. We compute the optimum Booster strategy against a completely adversarial Weak-Learner, which here is permitted to choose weak classifiers without restriction, that is, the entire space $\mathcal{H}^{\mathrm{all}}$ of all possible functions mapping examples to labels. By modeling Weak-Learner adversarially, we make absolutely no assumptions on the algorithm it might use. Hence, error guarantees enjoyed in this situation will be universally applicable. Our algorithms are derived from the very general drifting games framework (Schapire, 2001) for solving boosting games, which in turn was inspired by Freund's Boost-by-majority algorithm (Freund, 1995), which we review next.

5.1 The OS Algorithm

Fix the number of rounds T and a weak-learning condition $(\mathcal{C}, \mathbf{B})$. We will only consider conditions that are not *vacuous*, that is, at least some classifier space satisfies the condition, or equivalently, the space $\mathcal{H}^{\mathrm{all}}$ satisfies $(\mathcal{C}, \mathbf{B})$. Additionally, we assume the constraints placed by $\mathcal{C}$ are on individual rows. In other words, there is some subset $\mathcal{C}_0 \subseteq \mathbb{R}^k$ of all possible rows, such that a cost matrix $\mathbf{C}$ belongs to the collection $\mathcal{C}$ if and only if each of its rows belongs to this subset:

$$\mathbf{C} \in \mathcal{C} \iff \forall i : \mathbf{C}(i) \in \mathcal{C}_0. \tag{14}$$

Further, we assume $\mathcal{C}_0$ forms a convex cone, that is, $\mathbf{c}, \mathbf{c}' \in \mathcal{C}_0$ implies $t\mathbf{c} + t'\mathbf{c}' \in \mathcal{C}_0$ for any non-negative t, t'. This also implies that $\mathcal{C}$ is a convex cone. This is a very natural restriction, and is satisfied by the space $\mathbf{C}$ used by the weak learning conditions of AdaBoost.MH, AdaBoost.M1, AdaBoost.MR, SAMME as well as every edge-over-random condition.[1] For simplicity of presentation we fix the weights $\alpha_t = 1$ in each round. With $\mathbf{f}_T$ defined as in (1), whether the final hypotheses output by Booster makes a prediction error on an example i is decided by whether an incorrect label received the maximum number of votes, $f_T(i,1) \leq \max_{l=2}^{k} f_T(i,l)$. Therefore, the optimum Booster payoff can be written as

$$\min_{\mathbf{C}_1 \in \mathcal{C}} \max_{\substack{h_1 \in \mathcal{H}^{\mathrm{all}}: \\ \mathbf{C}_1 \bullet \left(\mathbf{1}_{h_1} - \mathbf{B}\right) \leq \mathbf{0}}} \cdots \min_{\mathbf{C}_T \in \mathcal{C}} \max_{\substack{h_T \in \mathcal{H}^{\mathrm{all}}: \\ \mathbf{C}_T \bullet \left(\mathbf{1}_{h_T} - \mathbf{B}\right) \leq \mathbf{0}}} \frac{1}{m} \sum_{i=1}^{m} L^{\mathrm{err}}(f_T(x_i,1), \ldots, f_T(x_i,k)). \tag{15}$$

where the function $L^{\mathrm{err}} : \mathbb{R}^k \to \mathbb{R}$ encodes 0-1 error

$$L^{\mathrm{err}}(\mathbf{s}) = \mathbf{1}\left[s(1) \leq \max_{l>1} s(l)\right]. \tag{16}$$

1. All our results hold under the weaker restriction on the space $\mathcal{C}$, where the set of possible cost vectors $\mathcal{C}_0$ for a row i could depend on i. For simplicity of exposition, we stick to the more restrictive assumption that $\mathcal{C}_0$ is common across all rows.

In general, we will also consider other loss functions $L : \mathbb{R}^k \to \mathbb{R}$ such as exponential loss, hinge loss, etc. that upper-bound error and are *proper*: that is, $L(\mathbf{s})$ is increasing in the weight of the correct label $s(1)$, and decreasing in the weights of the incorrect labels $s(l), l \neq 1$.

Directly analyzing the optimal payoff is hard. However, Schapire (2001) observed that the payoffs can be very well approximated by certain potential functions. Indeed, for any $\mathbf{b} \in \mathbb{R}^k$ define the *potential function* $\phi_t^{\mathbf{b}} : \mathbb{R}^k \to \mathbb{R}$ by the following recurrence:

$$\begin{array}{rcl} \phi_0^{\mathbf{b}}(\mathbf{s}) & = & L(\mathbf{s}) \\ \phi_t^{\mathbf{b}}(\mathbf{s}) & = & \min\limits_{\mathbf{c} \in \mathcal{C}_0} \quad \max\limits_{\mathbf{p} \in \Delta\{1,\ldots,k\}} \quad \mathbb{E}_{l \sim \mathbf{p}} \left[\phi_{t-1}^{\mathbf{b}} \left(\mathbf{s} + \mathbf{e}_l \right) \right] \\ & & \qquad\qquad \text{s.t.} \qquad \mathbb{E}_{l \sim \mathbf{p}} \left[c(l) \right] \leq \langle \mathbf{b}, \mathbf{c} \rangle \,, \end{array} \tag{17}$$

where $l \sim \mathbf{p}$ denotes that label l is sampled from the distribution $\mathbf{p}$, and $\mathbf{e}_l \in \mathbb{R}^k$ is the unit-vector whose lth coordinate is 1 and the remaining coordinates zero. Notice the recurrence uses the collection of rows $\mathcal{C}_0$ instead of the collection of cost matrices $\mathcal{C}$. When there are $T-t$ rounds remaining (that is, after t rounds of boosting), these potential functions compute an estimate $\phi_{T-t}^{\mathbf{b}}(\mathbf{s}_t)$ of whether an example x will be misclassified, based on its current state $\mathbf{s}_t$ consisting of counts of votes received so far on various classes:

$$s_t(l) = \sum_{t'=1}^{t-1} \mathbf{1} \left[h_{t'}(x) = l \right]. \tag{18}$$

Notice this definition of state assumes that $\alpha_t = 1$ in each round. Sometimes, we will choose the weights differently. In such cases, a more appropriate definition is the weighted state $\mathbf{f}_t \in \mathbb{R}^k$, tracking the weighted counts of votes received so far:

$$f_t(l) = \sum_{t'=1}^{t-1} \alpha_{t'} \mathbf{1} \left[h_{t'}(x) = l \right]. \tag{19}$$

However, unless otherwise noted, we will assume $\alpha_t = 1$, and so the definition in (18) will suffice.

The recurrence in (17) requires the max player's response $\mathbf{p}$ to satisfy the constraint that the expected cost under the distribution $\mathbf{p}$ is at most the inner-product $\langle \mathbf{c}, \mathbf{b} \rangle$. If there is no distribution satisfying this requirement, then the value of the max expression is $-\infty$. The existence of a valid distribution depends on both $\mathbf{b}$ and $\mathbf{c}$ and is captured by the following:

$$\exists \mathbf{p} \in \Delta\{1, \ldots, k\} : \mathbb{E}_{l \sim \mathbf{p}} \left[c(l) \right] \leq \langle \mathbf{c}, \mathbf{b} \rangle \iff \min_l c(l) \leq \langle \mathbf{b}, \mathbf{c} \rangle \,. \tag{20}$$

In this paper, the vector $\mathbf{b}$ will always correspond to some row $\mathbf{B}(i)$ of the baseline used in the weak learning condition. In such a situation, the next lemma shows that a distribution satisfying the required constraints will always exist.

Lemma 7 *If $\mathcal{C}_0$ is a cone and* (14) *holds, then for any row $\mathbf{b} = \mathbf{B}(i)$ of the baseline and any cost vector $\mathbf{c} \in \mathcal{C}_0$,* (20) *holds unless the condition $(\mathcal{C}, \mathbf{B})$ is vacuous.*

Proof We show that if (20) does not hold, then the condition is vacuous. Assume that for row $\mathbf{b} = \mathbf{B}(i_0)$ of the baseline, and some choice of cost vector $\mathbf{c} \in \mathcal{C}_0$, (20) does not hold. We pick a cost-matrix $\mathbf{C} \in \mathcal{C}$, such that no weak classifier h can satisfy the requirement (2), implying the condition

must be vacuous. The i_0^{th} row of the cost matrix is $\mathbf{c}$, and the remaining rows are $\mathbf{0}$. Since $\mathcal{C}_0$ is a cone, $\mathbf{0} \in \mathcal{C}_0$ and hence the cost matrix lies in $\mathcal{C}$. With this choice for $\mathbf{C}$, the condition (2) becomes

$$c(h(x_i)) = C(i, h(x_i)) \leq \langle \mathbf{C}(i), \mathbf{B}(i) \rangle = \langle \mathbf{c}, \mathbf{b} \rangle < \min_l c(l),$$

where the last inequality holds since, by assumption, (20) is not true for this choice of $\mathbf{c}, \mathbf{b}$. The previous equation is an impossibility, and hence no such weak classifier h exists, showing the condition is vacuous. ■

Lemma 7 shows that the expression in (17) is well defined, and takes on finite values. We next record an alternate dual form for the same recurrence which will be useful later.

Lemma 8 *The recurrence in* (17) *is equivalent to*

$$\phi_t^{\mathbf{b}}(\mathbf{s}) = \min_{\mathbf{c} \in \mathcal{C}_0} \max_{l=1}^{k} \left\{ \phi_{t-1}^{\mathbf{b}}(\mathbf{s} + \mathbf{e}_l) - (c(l) - \langle \mathbf{c}, \mathbf{b} \rangle) \right\}. \tag{21}$$

Proof Using Lagrangean multipliers, we may convert (17) to an unconstrained expression as follows:

$$\phi_t^{\mathbf{b}}(\mathbf{s}) = \min_{\mathbf{c} \in \mathcal{C}_0} \max_{\mathbf{p} \in \Delta\{1,\ldots,k\}} \min_{\lambda \geq 0} \left\{ \mathbb{E}_{l \sim \mathbf{p}} \left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s} + \mathbf{e}_l) \right] - \lambda \left(\mathbb{E}_{l \sim \mathbf{p}} [c(l)] - \langle \mathbf{c}, \mathbf{b} \rangle \right) \right\}.$$

Applying Theorem 1 to the inner min-max expression we get

$$\phi_t^{\mathbf{b}}(\mathbf{s}) = \min_{\mathbf{c} \in \mathcal{C}_0} \min_{\lambda \geq 0} \max_{\mathbf{p} \in \Delta\{1,\ldots,k\}} \left\{ \mathbb{E}_{l \sim \mathbf{p}} \left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s} + \mathbf{e}_l) \right] - \left(\mathbb{E}_{l \sim \mathbf{p}} [\lambda c(l)] - \langle \lambda \mathbf{c}, \mathbf{b} \rangle \right) \right\}.$$

Since $\mathcal{C}_0$ is a cone, $\mathbf{c} \in \mathcal{C}_0$ implies $\lambda \mathbf{c} \in \mathcal{C}_0$. Therefore we may absorb the Lagrange multiplier into the cost vector:

$$\phi_t^{\mathbf{b}}(\mathbf{s}) = \min_{\mathbf{c} \in \mathcal{C}_0} \max_{\mathbf{p} \in \Delta\{1,\ldots,k\}} \mathbb{E}_{l \sim \mathbf{p}} \left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s} + \mathbf{e}_l) - (c(l) - \langle \mathbf{c}, \mathbf{b} \rangle) \right].$$

For a fixed choice of $\mathbf{c}$, the expectation is maximized when the distribution $\mathbf{p}$ is concentrated on a single label that maximizes the inner expression, which completes our proof. ■

The dual form of the recurrence is useful for optimally choosing the cost matrix in each round. When the weak learning condition being used is $(\mathcal{C}, \mathbf{B})$, Schapire (2001) proposed a Booster strategy, called the OS strategy, which always chooses the weight $\alpha_t = 1$, and uses the potential functions to construct a cost matrix $\mathbf{C}_t$ as follows. Each row $\mathbf{C}_t(i)$ of the matrix achieves the minimum of the right hand side of (21) with $\mathbf{b}$ replaced by $\mathbf{B}(i)$, t replaced by $T-t$, and $\mathbf{s}$ replaced by current state $\mathbf{s}_t(i)$:

$$\mathbf{C}_t(i) = \operatorname*{argmin}_{\mathbf{c} \in \mathcal{C}_0} \max_{l=1}^{k} \left\{ \phi_{T-t-1}^{\mathbf{B}(i)}(\mathbf{s} + \mathbf{e}_l) - (c(l) - \langle \mathbf{c}, \mathbf{B}(i) \rangle) \right\}. \tag{22}$$

The following theorem, proved in the appendix, provides a guarantee for the loss suffered by the OS algorithm, and also shows that it is the game-theoretically optimum strategy when the number of examples is large. Similar results have been proved by Schapire (2001), but our theorem holds much more generally, and also achieves tighter lower bounds.

Theorem 9 (Extension of results in Schapire (2001)) *Suppose the weak-learning condition is not vacuous and is given by* $(\mathcal{C},\mathbf{B})$*, where* $\mathcal{C}$ *is such that, for some convex cone* $\mathcal{C}_0 \subseteq \mathbb{R}^k$*, the condition* (14) *holds. Let the potential functions* $\phi_t^{\mathbf{b}}$ *be defined as in* (17)*, and assume the Booster employs the OS algorithm, choosing* $\alpha_t = 1$ *and* $\mathbf{C}_t$ *as in* (22) *in each round* t*. Then the average potential of the states,*

$$\frac{1}{m}\sum_{i=1}^{m} \phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}_t(i)),$$

never increases in any round. In particular, the loss suffered after T *rounds of play is at most*

$$\frac{1}{m}\sum_{i=1}^{m} \phi_{T}^{\mathbf{B}(i)}(\mathbf{0}). \tag{23}$$

Further, under certain conditions, this bound is nearly tight. In particular, assume the loss function does not vary too much but satisfies

$$\sup_{\mathbf{s},\mathbf{s}' \in \mathcal{S}_T} |L(\mathbf{s}) - L(\mathbf{s}')| \leq \varnothing(L,T), \tag{24}$$

where $\mathcal{S}_T$*, a subset of* $\left\{\mathbf{s} \in \mathbb{R}^k : \|\mathbf{s}\|_\infty \leq T\right\}$*, is the set of all states reachable in* T *iterations, and* $\varnothing(L,T)$ *is an upper bound on the discrepancy of losses between any two reachable states when the loss function is* L *and the total number of iterations is* T*. Then, for any* $\varepsilon > 0$*, when the number of examples* m *is sufficiently large,*

$$m \geq \frac{T\varnothing(L,T)}{\varepsilon}, \tag{25}$$

no Booster strategy can guarantee to achieve in T *rounds a loss that is* ε *less than the bound* (23)*.*

In order to implement the nearly optimal OS strategy, we need to solve (22). This is computationally only as hard as evaluating the potentials, which in turn reduces to computing the recurrences in (17). In the next few sections, we study how to do this when using various losses and weak learning conditions.

6. Solving for Any Fixed Edge-over-random Condition

In this section we show how to implement the OS strategy when the weak learning condition is any fixed edge-over-random condition: $(\mathcal{C},\mathbf{B})$ for some $\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}$. By our previous discussions, this is equivalent to computing the potential $\phi_t^{\mathbf{b}}$ by solving the recurrence in (17), where the vector $\mathbf{b}$ corresponds to some row of the baseline $\mathbf{B}$. Let $\Delta_\gamma^k \subseteq \Delta\{1,\ldots,k\}$ denote the set of all edge-over-random distributions on $\{1,\ldots,k\}$ with γ more weight on the first coordinate:

$$\Delta_\gamma^k = \{\mathbf{b} \in \Delta\{1,\ldots,k\} : b(1) - \gamma = \max\{b(2),\ldots,b(k)\}\}. \tag{26}$$

Note, that $\mathcal{B}_\gamma^{\text{eor}}$ consists of all matrices whose rows belong to the set Δ_γ^k. Therefore we are interested in computing $\phi^{\mathbf{b}}$, where $\mathbf{b}$ is an arbitrary edge-over-random distribution: $\mathbf{b} \in \Delta_\gamma^k$. We begin by simplifying the recurrence (17) satisfied by such potentials, and show how to compute the optimal cost matrix in terms of the potentials.

Lemma 10 *Assume L is proper, and $\mathbf{b} \in \Delta_\gamma^k$ is an edge-over-random distribution. Then the recurrence* (17) *may be simplified as*

$$\phi_t^{\mathbf{b}}(\mathbf{s}) \quad = \quad \mathbb{E}_{l\sim\mathbf{b}}\left[\phi_{t-1}(\mathbf{s}+\mathbf{e}_l)\right]. \tag{27}$$

Further, if the cost matrix $\mathbf{C}_t$ is chosen as follows

$$C_t(i,l) = \phi_{T-t-1}^{\mathbf{b}}(\mathbf{s}_t(i)+\mathbf{e}_l), \tag{28}$$

then $\mathbf{C}_t$ satisfies the condition in (22)*, and hence is the optimal choice.*

Proof Let $\mathcal{C}_0^{\text{eor}} \subseteq \mathbb{R}^k$ denote all vectors $\mathbf{c}$ satisfying $\forall l : c(1) \le c(l)$. Then, we have

$$\begin{aligned}
\phi_t^{\mathbf{b}}(\mathbf{s}) &= \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} \max_{\mathbf{p}\in\Delta\{1,\ldots,k\}} \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}(\mathbf{s}+\mathbf{e}_l)\right] \\
&\qquad\qquad \text{s.t.} \quad \mathbb{E}_{l\sim\mathbf{p}}[c(l)] \le \mathbb{E}_{l\sim\mathbf{b}}[c(l)], \qquad (\text{ by (17) }) \\
&= \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} \max_{\mathbf{p}\in\Delta} \min_{\lambda\ge 0} \left\{\mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s}+\mathbf{e}_l)\right] + \lambda\left(\mathbb{E}_{l\sim\mathbf{b}}[c(l)] - \mathbb{E}_{l\sim\mathbf{p}}[c(l)]\right)\right\} \text{(Lagrangean)} \\
&= \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} \min_{\lambda\ge 0} \max_{\mathbf{p}\in\Delta} \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s}+\mathbf{e}_l)\right] + \lambda\langle\mathbf{b}-\mathbf{p},\mathbf{c}\rangle \text{ (Theorem 1)} \\
&= \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} \max_{\mathbf{p}\in\Delta} \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s}+\mathbf{e}_l)\right] + \langle\mathbf{b}-\mathbf{p},\mathbf{c}\rangle \text{ (absorb } \lambda \text{ into } \mathbf{c}) \\
&= \max_{\mathbf{p}\in\Delta} \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s}+\mathbf{e}_l)\right] + \langle\mathbf{b}-\mathbf{p},\mathbf{c}\rangle \text{ (Theorem 1)}.
\end{aligned}$$

Unless $b(1)-p(1) \le 0$ and $b(l)-p(l) \ge 0$ for each $l > 1$, the quantity $\langle\mathbf{b}-\mathbf{p},\mathbf{c}\rangle$ can be made arbitrarily small for appropriate choices of $\mathbf{c} \in \mathcal{C}_0^{\text{eor}}$. The max-player is therefore forced to constrain its choices of $\mathbf{p}$, and the above expression becomes

$$\begin{aligned}
&\max_{\mathbf{p}\in\Delta} \quad \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{b}}(\mathbf{s}+\mathbf{e}_l)\right] \\
&\text{s.t.} \quad b(l)-q(l)\begin{cases}\ge 0 & \text{if } l=1,\\ \le 0 & \text{if } l>1.\end{cases}
\end{aligned}$$

Lemma 6 of Schapire (2001) states that if L is *proper* (as defined here), so is $\phi_t^{\mathbf{b}}$; the same result can be extended to our drifting games. This implies the optimal choice of $\mathbf{p}$ in the above expression is in fact the distribution that puts as small weight as possible in the first coordinate, namely $\mathbf{b}$. Therefore the optimum choice of $\mathbf{p}$ is $\mathbf{b}$, and the potential is the same as in (27).

We end the proof by showing that the choice of cost matrix in (28) is optimum. Theorem 9 states that a cost matrix $\mathbf{C}_t$ is the optimum choice if it satisfies (22), that is, if the expression

$$\max_{l=1}^{k}\left\{\phi_{T-t-1}^{\mathbf{B}(i)}(\mathbf{s}+\mathbf{e}_l) - \left(C_t(i,l) - \langle\mathbf{C}_t(i),\mathbf{B}(i)\rangle\right)\right\} \tag{29}$$

is equal to

$$\min_{\mathbf{c}\in C_0}\max_{l=1}^{k}\left\{\phi_{T-t-1}^{\mathbf{B}(i)}(\mathbf{s}+\mathbf{e}_l) - \left(c(l) - \langle\mathbf{c},\mathbf{B}(i)\rangle\right)\right\} = \phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}), \tag{30}$$

where the equality in (30) follows from (21). If $\mathbf{C}_t(i)$ is chosen as in (28), then, for any label l, the expression within max in (29) evaluates to

$$\begin{aligned}
\phi^{\mathbf{B}(i)}_{T-t-1}(\mathbf{s}+\mathbf{e}_l) &- \left(\phi^{\mathbf{B}(i)}_{T-t-1}(\mathbf{s}+\mathbf{e}_l) - \langle \mathbf{C}_t(i), \mathbf{B}(i)\rangle\right) \\
&= \langle \mathbf{B}(i), \mathbf{C}_t(i)\rangle \\
&= \mathbb{E}_{l\sim\mathbf{B}(i)}\left[C_t(i,l)\right] \\
&= \mathbb{E}_{l\sim\mathbf{B}(i)}\left[\phi^{\mathbf{B}(i)}_{T-t-1}(\mathbf{s}+\mathbf{e}_l)\right] \\
&= \phi^{\mathbf{B}(i)}_{T-t}(\mathbf{s}),
\end{aligned}$$

where the last equality follows from (27). Therefore the max expression in (29) is also equal to $\phi^{\mathbf{B}(i)}_{T-t}(\mathbf{s})$, which is what we needed to show. ■

Equation (28) in Lemma 10 implies the cost matrix chosen by the OS strategy can be expressed in terms of the potentials, which is the only thing left to calculate. Fortunately, the simplification (27) of the drifting games recurrence, allows the potentials to be solved completely in terms of a random-walk $\mathcal{R}^t_{\mathbf{b}}(\mathbf{x})$. This random variable denotes the position of a particle after t time steps, that starts at location $\mathbf{x}\in\mathbb{R}^k$, and in each step moves in direction $\mathbf{e}_l$ with probability $b(l)$.

Corollary 11 *The recurrence in* (27) *can be solved as follows:*

$$\phi^{\mathbf{b}}_t(\mathbf{s}) = \mathbb{E}\left[L\left(\mathcal{R}^t_{\mathbf{b}}(\mathbf{s})\right)\right]. \tag{31}$$

Proof Inductively assuming $\phi^{\mathbf{b}}_{t-1}(\mathbf{x}) = \mathbb{E}\left[L(\mathcal{R}^{t-1}_{\mathbf{b}}(\mathbf{x}))\right]$,

$$\phi_t(\mathbf{s}) = \mathbb{E}_{l\sim\mathbf{b}}\left[L(\mathcal{R}^{t-1}_{\mathbf{b}}(\mathbf{s})+\mathbf{e}_l)\right] = \mathbb{E}\left[L(\mathcal{R}^t_{\mathbf{b}}(\mathbf{s}))\right].$$

The last equality follows by observing that the random position $\mathcal{R}^{t-1}_{\mathbf{b}}(\mathbf{s})+\mathbf{e}_l$ is distributed as $\mathcal{R}^t_{\mathbf{b}}(\mathbf{s})$ when l is sampled from $\mathbf{b}$. ■

Lemma 10 and Corollary 11 together imply:

Theorem 12 *Assume L is proper and $\mathbf{b}\in\Delta^k_\gamma$ is an edge-over-random distribution. Then the potential $\phi^{\mathbf{b}}_t$, defined by the recurrence in* (17)*, has the solution given in* (31) *in terms of random walks.*

Before we can compute (31), we need to choose a loss function L. We next consider two options for the loss—the non-convex 0-1 error, and exponential loss.

6.1 Exponential Loss

The exponential loss serves as a smooth convex proxy for discontinuous non-convex 0-1 error (16) that we would ultimately like to bound, and is given by

$$L^{\exp}_\eta(\mathbf{s}) = \sum_{l=2}^{k} e^{\eta(s_l - s_1)}. \tag{32}$$

The parameter η can be thought of as the weight in each round, that is, $\alpha_t = \eta$ in each round. Then notice that the weighted state $\mathbf{f}_t$ of the examples, defined in (19), is related to the unweighted states

$\mathbf{s}_t$ as $f_t(l) = \eta s_t(l)$. Therefore the exponential loss function in (32) directly measures the loss of the weighted state as

$$L^{\exp}(\mathbf{f}_t) = \sum_{l=2}^{k} e^{f_t(l) - f_t(1)}. \tag{33}$$

Because of this correspondence, the optimal strategy with the loss function $L^{\exp}$ and $\alpha_t = \eta$ is the same as that using loss $L_\eta^{\exp}$ and $\alpha_t = 1$. We study the latter setting so that we may use the results derived earlier. With the choice of the exponential loss $L_\eta^{\exp}$, the potentials are easily computed, and in fact have a closed form solution.

Theorem 13 *If $L_\eta^{\exp}$ is as in* (32)*, where η is non-negative, then the solution in Theorem 12 evaluates to $\phi_t^{\mathbf{b}}(\mathbf{s}) = \sum_{l=2}^{k} (a_l)^t e^{\eta_l (s_l - s_1)}$, where $a_l = 1 - (b_1 + b_l) + e^{\eta} b_l + e^{-\eta} b_1$.*

The proof by induction is straightforward. By tuning the weight η, each a_l can be always made less than 1. This ensures the exponential loss decays exponentially with rounds. In particular, when $\mathbf{B} = \mathbf{U}_\gamma$ (so that the condition is $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$), the relevant potential $\phi_t(\mathbf{s})$ or $\phi_t(\mathbf{f})$ is given by

$$\phi_t(\mathbf{s}) = \phi_t(f) = \kappa(\gamma, \eta)^t \sum_{l=2}^{k} e^{\eta(s_l - s_1)} = \kappa(\gamma, \eta)^t \sum_{l=2}^{k} e^{f_l - f_1} \tag{34}$$

where

$$\kappa(\gamma, \eta) = 1 + \frac{(1-\gamma)}{k} \left(e^{\eta} + e^{-\eta} - 2\right) - \left(1 - e^{-\eta}\right) \gamma. \tag{35}$$

The cost-matrix output by the OS algorithm can be simplified by rescaling, or adding the same number to each coordinate of a cost vector, without affecting the constraints it imposes on a weak classifier, to the following form

$$C(i,l) = \begin{cases} (e^{\eta} - 1)\, e^{\eta(s_l - s_1)} & \text{if } l > 1, \\ (e^{-\eta} - 1) \sum_{l=2}^{k} e^{\eta(s_l - s_1)} & \text{if } l = 1. \end{cases}$$

Using the correspondence between unweighted and weighted states, the above may also be rewritten as:

$$C(i,l) = \begin{cases} (e^{\eta} - 1)\, e^{f_l - f_1} & \text{if } l > 1, \\ (e^{-\eta} - 1) \sum_{l=2}^{k} e^{f_l - f_1} & \text{if } l = 1. \end{cases} \tag{36}$$

With such a choice, Theorem 9 and the form of the potential guarantee that the average loss

$$\frac{1}{m} \sum_{i=1}^{m} L_\eta^{\exp}(\mathbf{s}_t(i)) = \frac{1}{m} \sum_{i=1}^{m} L^{\exp}(\mathbf{f}_t(i)) \tag{37}$$

of the states changes by a factor of at most $\kappa(\gamma, \eta)$ every round. Therefore the final loss, which upper bounds the error, that is, the fraction of misclassified training examples, is at most $(k-1)\kappa(\gamma, \eta)^T$. Since this upper bound holds for any value of η, we may tune it to optimize the bound. Setting $\eta = \ln(1+\gamma)$, the error can be upper bounded by $(k-1)e^{-T\gamma^2/2}$.

6.2 Zero-one Loss

There is no simple closed form solution for the potential when using the zero-one loss L^{err} (16). However, we may compute the potentials efficiently as follows. To compute $\phi_t^{\mathbf{b}}(\mathbf{s})$, we need to find the probability that a random walk (making steps according to $\mathbf{b}$) of length t in $\mathbb{Z}^k$, starting at $\mathbf{s}$ will end up in a region where the loss function is 1. Any such random walk will consist of x_l steps in direction $\mathbf{e}_l$ where the non-negative $\sum_l x_l = t$. The probability of each such path is $\prod_l b_l^{x_l}$. Further, there are exactly $\binom{t}{x_1,\ldots,x_k}$ such paths. Starting at state $\mathbf{s}$, such a path will lead to a correct answer only if $s_1 + x_1 > s_l + x_l$ for each $l > 1$. Hence we may write the potential $\phi_t^{\mathbf{b}}(\mathbf{s})$ as

$$\begin{aligned}
\phi_t^{\mathbf{b}}(\mathbf{s}) = 1 - \sum_{x_1,\ldots,x_k}^{t} & \binom{t}{x_1,\ldots,x_k} \prod_{l=1}^{k} b_l^{x_l} \\
\text{s.t.} \quad & x_1 + \ldots + x_k \quad = t \\
\forall l: \quad & x_l \quad \geq 0 \\
\forall l: \quad & x_l + s_l \quad \leq x_1 + s_1.
\end{aligned}$$

Since the x_l's are restricted to be integers, this problem is presumably hard. In particular, the only algorithms known to the authors that take time logarithmic in t is also exponential in k. However, by using dynamic programming, we can compute the summation in time polynomial in $|s_l|$, t and k. In fact, the run time is always $O(t^3 k)$, and at least $\Omega(tk)$.

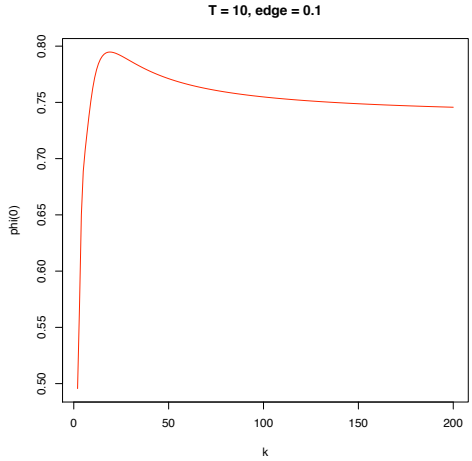
The bounds on error we achieve, although not in closed form, are much tighter than those obtainable using exponential loss. The exponential loss analysis yields an error upper bound of $(k-1)e^{-T\gamma^2/2}$. Using a different initial distribution, Schapire and Singer (1999) achieve the slightly better bound $\sqrt{(k-1)}e^{-T\gamma^2/2}$. However, when the edge γ is small and the number of rounds are few, each bound is greater than 1 and hence trivial. On the other hand, the bounds computed by the above dynamic program are sensible for all values of k, γ and T. When $\mathbf{b}$ is the γ-biased uniform distribution $\mathbf{b} = (\frac{1-\gamma}{k} + \gamma, \frac{1-\gamma}{k}, \frac{1-\gamma}{k}, \ldots, \frac{1-\gamma}{k})$ a table containing the error upper bound $\phi_T^{\mathbf{b}}(0)$ for $k = 6$, $\gamma = 0$ and small values for the number of rounds T is shown in Figure 2(a); note that with the exponential loss, the bound is always 1 if the edge γ is 0. Further, the bounds due to the exponential loss analyses seem to imply that the dependence of the error on the number of labels is monotonic. However, a plot of the tighter bounds with edge $\gamma = 0.1$, number of rounds $T = 10$ against various values of k, shown in Figure 2(b), indicates that the true dependence is more complicated. Therefore the tighter analysis also provides qualitative insights not obtainable via the exponential loss bound.

7. Solving for the Minimal Weak Learning Condition

In the previous section we saw how to find the optimal boosting strategy when using any fixed edge-over-random condition. However as we have seen before, these conditions can be stronger than necessary, and therefore lead to boosting algorithms that require additional assumptions. Here we show how to compute the optimal algorithm while using the weakest weak learning condition, provided by (13), or equivalently the condition used by AdaBoost.MR, $(\mathcal{C}^{\text{MR}}, \mathbf{B}_\gamma^{\text{MR}})$. Since there are two possible formulations for the minimal condition, it is not immediately clear which to use to compute the optimal boosting strategy. To resolve this, we first show that the optimal boosting strategy based on any formulation of a necessary and sufficient weak learning condition is the same. Having resolved this ambiguity, we show how to compute this strategy for the exponential loss and 0-1 error using the first formulation.

T	$\phi^{\mathbf{b}}_T(\mathbf{0})$	T	$\phi^{\mathbf{b}}_T(\mathbf{0})$
0	1.00	6	0.90
1	0.83	7	0.91
2	0.97	8	0.90
3	0.93	9	0.89
4	0.89	10	0.89
5	0.89		

(a)



(b)

Figure 2: Plot of potential value $\phi^{\mathbf{b}}_T(\mathbf{0})$ where $\mathbf{b}$ is the γ-biased uniform distribution: $\mathbf{b} = (\frac{1-\gamma}{k} + \gamma, \frac{1-\gamma}{k}, \frac{1-\gamma}{k}, \ldots, \frac{1-\gamma}{k})$. **(a):** Potential values (rounded to two decimal places) for different number of rounds T using $\gamma = 0$ and $k = 6$. These are bounds on the error, and less than 1 even when the edge and number of rounds are small. **(b):** Potential values for different number of classes k, with $\gamma = 0.1$, and $T = 10$. These are tight estimates for the optimal error, and yet not monotonic in the number of classes.

7.1 Game-theoretic Equivalence of Necessary and Sufficient Weak-learning Conditions

In this section we study the effect of the weak learning condition on the game-theoretically optimal boosting strategy. We introduce the notion of *game-theoretic equivalence* between two weak learning conditions, that determines if the payoffs (15) of the optimal boosting strategies based on the two conditions are identical. This should hold whenever both games last for the same number of iterations T, for any value of T. This is different from the usual notion of equivalence between two conditions, which holds if any weak classifier space satisfies both conditions or neither condition. In fact we prove that game-theoretic equivalence is a broader notion; in other words, equivalence implies game-theoretic equivalence. A special case of this general result is that any two weak learning conditions that are necessary and sufficient, and hence equivalent to boostability, are also game-theoretically equivalent. In particular, so are the conditions of AdaBoost.MR and (13), and the resulting optimal Booster strategies enjoy equally good payoffs. We conclude that in order to derive the optimal boosting strategy that uses the minimal weak-learning condition, it is sound to use either of these two formulations.

The purpose of a weak learning condition $(\mathcal{C}, \mathbf{B})$ is to impose restrictions on the Weak-Learner's responses in each round. These restrictions are captured by subsets of the weak classifier space as follows. If Booster chooses cost-matrix $\mathbf{C} \in \mathcal{C}$ in a round, the Weak-Learner's response h is restricted to the subset $S_{\mathbf{C}} \subseteq \mathcal{H}^{\text{all}}$ defined as

$$S_{\mathbf{C}} = \left\{ h \in \mathcal{H}^{\text{all}} : \mathbf{C} \bullet \mathbf{1}_h \leq \mathbf{C} \bullet \mathbf{B} \right\}.$$

Thus, a weak learning condition is essentially a family of subsets of the weak classifier space. Further, smaller subsets mean fewer options for Weak-Learner, and hence better payoffs for the optimal boosting strategy. Based on this idea, we may define when a weak learning condition $(\mathcal{C}_1, \mathbf{B}_1)$ is *game-theoretically stronger* than another condition $(\mathcal{C}_2, \mathbf{B}_2)$ if the following holds: For every subset $S_{\mathbf{C}_2}$ in the second condition (that is $\mathbf{C}_2 \in \mathcal{C}_2$), there is a subset $S_{\mathbf{C}_1}$ in the first condition (that is $\mathbf{C}_1 \in \mathcal{C}_1$), such that $S_{\mathbf{C}_1} \subseteq S_{\mathbf{C}_2}$. Mathematically, this may be written as follows:

$$\forall \mathbf{C}_1 \in \mathcal{C}_1, \exists \mathbf{C}_2 \in \mathcal{C}_2 : S_{\mathbf{C}_1} \subseteq S_{\mathbf{C}_2}.$$

Intuitively, a game theoretically stronger condition will allow Booster to place similar or stricter restrictions on Weak-Learner in each round. Therefore, the optimal Booster payoff using a game-theoretically stronger condition is at least equally good, if not better. It therefore follows that if two conditions are both game-theoretically stronger than each other, the corresponding Booster payoffs must be equal, that is they must be *game-theoretically equivalent*.

Note that game-theoretic equivalence of two conditions does not mean that they are identical as families of subsets, for we may arbitrarily add large and "useless" subsets to the two conditions without affecting the Booster payoffs, since these subsets will never be used by an optimal Booster strategy. In fact we next show that game-theoretic equivalence is a broader notion than just equivalence.

Theorem 14 *Suppose* $(\mathcal{C}_1, \mathbf{B}_1)$ *and* $(\mathcal{C}_2, \mathbf{B}_2)$ *are two equivalent weak learning conditions, that is, every space* $\mathcal{H}$ *satisfies both or neither condition. Then each condition is game-theoretically stronger than the other, and hence game-theoretically equivalent.*

Proof We argue by contradiction. Assume that despite equivalence, the first condition (without loss of generality) includes a particularly hard subset $S_{\mathbf{C}_1} \subseteq \mathcal{H}^{\text{all}}, \mathbf{C}_1 \in \mathcal{C}_1$ which is not smaller than any subset in the second condition. In particular, for every subset $S_{\mathbf{C}_2}, \mathbf{C}_2 \in \mathcal{C}_2$ in the second condition is satisfied by some weak classifier $h_{\mathbf{C}_2}$ not satisfying the hard subset in the first condition: $h_{\mathbf{C}_2} \in S_{\mathbf{C}_2} \setminus S_{\mathbf{C}_1}$. Therefore, the space

$$\mathcal{H} = \{h_{\mathbf{C}_2} : \mathbf{C}_2 \in \mathcal{C}_2\},$$

formed by just these classifiers satisfies the second condition, but has an empty intersection with $S_{\mathbf{C}_1}$. In other words, $\mathcal{H}$ satisfies the second but not the first condition, a contradiction to their equivalence. ■

An immediate corollary is the game theoretic equivalence of necessary and equivalent conditions.

Corollary 15 *Any two necessary and sufficient weak learning conditions are game-theoretically equivalent. In particular the optimum Booster strategies based on AdaBoost.MR's condition* $(\mathcal{C}^{MR}, \mathbf{B}^{MR}_{\gamma})$ *and* (13) *have equal payoffs.*

Therefore, in deriving the optimal Booster strategy, it is sound to work with either AdaBoost.MR's condition or (13). In the next section, we actually compute the optimal strategy using the latter formulation.

7.2 Optimal Strategy with the Minimal Conditions

In this section we compute the optimal Booster strategy that uses the minimum weak learning condition provided in (13). We choose this instead of AdaBoost.MR's condition because this description is more closely related to the edge-over-random conditions, and the resulting algorithm has a close relationship to the ones derived for fixed edge-over-random conditions, and therefore more insightful. However, this formulation does not state the condition as a single pair $(\mathbf{C},\mathbf{B})$, and therefore we cannot directly use the result of Theorem 9. Instead, we define new potentials and a modified OS strategy that is still nearly optimal, and this constitutes the first part of this section. In the second part, we show how to compute these new potentials and the resulting OS strategy.

7.2.1 MODIFIED POTENTIALS AND OS STRATEGY

The condition in (13) is not stated as a single pair $(\mathcal{C}^{\text{eor}},\mathbf{B})$, but uses all possible edge-over-random baselines $\mathbf{B}\in\mathcal{B}_\gamma^{\text{eor}}$. Therefore, we modify the definitions (17) of the potentials accordingly to extract an optimal Booster strategy. Recall that Δ_γ^k is defined in (26) as the set of all edge-over-random distributions which constitute the rows of edge-over-random baselines $\mathbf{B}\in\mathcal{B}_\gamma^{\text{eor}}$. Using these, define new potentials $\phi_t(\mathbf{s})$ as follows:

$$\phi_t(\mathbf{s}) = \begin{array}{lll} \min\limits_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}} & \max\limits_{\mathbf{b}\in\Delta_\gamma^k}\max\limits_{\mathbf{p}\in\Delta\{1,\ldots,k\}} & \mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right] \\ & \text{s.t.} & \mathbb{E}_{l\sim\mathbf{p}}[c(l)]\le\langle\mathbf{b},\mathbf{c}\rangle\,. \end{array} \tag{38}$$

The main difference between (38) and (17) is that while the older potentials were defined using a fixed vector $\mathbf{b}$ corresponding to some row in the fixed baseline $\mathbf{B}$, the new definition takes the maximum over all possible rows $\mathbf{b}\in\Delta_\gamma^k$ that an edge-over-random baseline $\mathbf{B}\in\mathcal{B}_\gamma^{\text{eor}}$ may have. As before, we may write the recurrence in (38) in its dual form

$$\phi_t(\mathbf{s}) = \min_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}}\max_{\mathbf{b}\in\Delta_\gamma^k}\max_{l=1}^{k}\left\{\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)-\left(c(l)-\langle\mathbf{c},\mathbf{b}\rangle\right)\right\}.$$

The proof is very similar to that of Lemma 8 and is omitted. We may now define a new OS strategy that chooses a cost-matrix in round t analogously:

$$\mathbf{C}_t(i)\in\operatorname*{argmin}_{\mathbf{c}\in\mathcal{C}_0^{\text{eor}}}\max_{\mathbf{b}\in\Delta_\gamma^k}\max_{l=1}^{k}\left\{\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)-\left(c(l)-\langle\mathbf{c},\mathbf{b}\rangle\right)\right\}. \tag{39}$$

where recall that $\mathbf{s}_t(i)$ denotes the state vector (defined in (18)) of example i. With this strategy, we can show an optimality result very similar to Theorem 9.

Theorem 16 *Suppose the weak-learning condition is given by* (13)*. Let the potential functions* $\phi_t^{\mathbf{b}}$ *be defined as in* (38)*, and assume the Booster employs the modified OS strategy, choosing* $\alpha_t=1$ *and* $\mathbf{C}_t$ *as in* (39) *in each round* t*. Then the average potential of the states,*

$$\frac{1}{m}\sum_{i=1}^{m}\phi_{T-t}\left(\mathbf{s}_t(i)\right),$$

never increases in any round. In particular, the loss suffered after T *rounds of play is at most* $\phi_T(\mathbf{0})$*.*

Further, for any $\varepsilon>0$*, when the loss function satisfies* (24) *and the number of examples* m *is as large as in* (25)*, no Booster strategy can guarantee to achieve less than* $\phi_T(\mathbf{0})-\varepsilon$ *loss in* T *rounds.*

The proof is very similar to that of Theorem 9 and is omitted.

7.2.2 COMPUTING THE NEW POTENTIALS

Here we show how to compute the new potentials. The resulting algorithms will require exponential time, and we provide some empirical evidence showing that this might be necessary. Finally, we show how to carry out the computations efficiently in certain special situations.

An Exponential Time Algorithm. Here we show how the potentials may be computed as the expected loss of some random walk, just as we did for the potentials arising with fixed edge-over-random conditions. The main difference is there will be several random walks to choose from.

We first begin by simplifying the recurrence (38), and expressing the optimal cost matrix in (39) in terms of the potentials, just as we did in Lemma 10 for the case of fixed edge-over-random conditions.

Lemma 17 *Assume L is proper. Then the recurrence* (38) *may be simplified as*

$$\phi_t(\mathbf{s}) \quad = \quad \max_{\mathbf{b}\in\Delta_\gamma^k} \mathbb{E}_{l\sim\mathbf{b}}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right]. \tag{40}$$

Further, if the cost matrix $\mathbf{C}_t$ is chosen as follows:

$$C_t(i,l) = \phi_{T-t-1}(\mathbf{s}_t(i)+\mathbf{e}_l), \tag{41}$$

then $\mathbf{C}_t$ satisfies the condition in (39)*.*

The proof is very similar to that of Lemma 10 and is omitted. Equation (41) implies that, as before, computing the optimal Booster strategy reduces to computing the new potentials. One computational difficulty created by the new definitions (38) or (40) is that they require infinitely many possible distributions $\mathbf{b}\in\Delta_\gamma^k$ to be considered. We show that we may in fact restrict our attention to only finitely many of such distributions described next.

At any state $\mathbf{s}$ and number of remaining iterations t, let π be a permutation of the coordinates $\{2,\ldots,k\}$ that sorts the potential values:

$$\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_{\pi(k)}\right) \geq \phi_{t-1}\left(\mathbf{s}+\mathbf{e}_{\pi(k-1)}\right) \geq \ldots \geq \phi_{t-1}\left(\mathbf{s}+\mathbf{e}_{\pi(2)}\right). \tag{42}$$

For any permutation π of the coordinates $\{2,\ldots,k\}$, let $\mathbf{b}_a^\pi$ denote the γ-biased uniform distribution on the a coordinates $\{1,\pi_k,\pi_{k-1},\ldots,\pi_{k-a+2}\}$:

$$b_a^\pi(l) = \begin{cases} \frac{1-\gamma}{a}+\gamma & \text{if } l=1 \\ \frac{1-\gamma}{a} & \text{if } l\in\{\pi_k,\ldots,\pi_{k-a+2}\} \\ 0 & \text{otherwise.} \end{cases} \tag{43}$$

Then, the next lemma shows that we may restrict our attention to only the distributions $\left\{\mathbf{b}_2^\pi,\ldots,\mathbf{b}_k^\pi\right\}$ when evaluating the recurrence in (40).

Lemma 18 *Fix a state $\mathbf{s}$ and remaining rounds of boosting t. Let π be a permutation of the coordinates $\{2,\ldots,k\}$ satisfying* (42)*, and define $\mathbf{b}_a^\pi$ as in* (43)*. Then the recurrence* (40) *may be simplified as follows:*

$$\phi_t(\mathbf{s}) = \max_{\mathbf{b}\in\Delta_\gamma^k} \mathbb{E}_{l\sim\mathbf{b}}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right] = \max_{2\leq a\leq k} \mathbb{E}_{l\sim\mathbf{b}_a^\pi}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right]. \tag{44}$$

Proof Assume (by relabeling the coordinates if necessary) that π is the identity permutation, that is, $\pi(2) = 2, \ldots, \pi(k) = k$. Observe that the right hand side of (40) is at least as much the right hand side of (44) since the former considers more distributions. We complete the proof by showing that the former is also at most the latter.

By (40), we may assume that some optimal $\mathbf{b}$ satisfies

$$\begin{aligned} b(k) = \cdots = b(k-a+2) &= b(1) - \gamma, \\ b(k-a+1) &\leq b(1) - \gamma, \\ b(k-a) = \cdots = b(2) &= 0. \end{aligned}$$

Therefore, $\mathbf{b}$ is a distribution supported on $a+1$ elements, with the minimum weight placed on element $k-a+1$. This implies $b(k-a+1) \in [0, 1/(a+1)]$.

Now, $\mathbb{E}_{l\sim\mathbf{b}}\left[\phi_{t-1}(\mathbf{s}+\mathbf{e}_l)\right]$ may be written as

$$\begin{aligned} & \gamma \cdot \phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + b(k-a+1)\phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+1}) \\ + \quad & (1-\gamma-b(k-a+1))\frac{\phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + \phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+2}) + \ldots \phi_{t-1}(\mathbf{s}+\mathbf{e}_k)}{a} \\ = \quad & \gamma \cdot \phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + \frac{b(k-a+1)}{1-\gamma}\phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+1}) \\ + \quad & (1-\gamma)\left\{\left(1 - \frac{b(k-a+1)}{1-\gamma}\right)\frac{\phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + \phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+2}) + \ldots \phi_{t-1}(\mathbf{s}+\mathbf{e}_k)}{a}\right\} \end{aligned}$$

Replacing $b(k-a+1)$ by x in the above expression, we get a linear function of x. When restricted to $[0, 1/(a+1)]$ the maximum value is attained at a boundary point. For $x = 0$, the expression becomes

$$\gamma \cdot \phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + (1-\gamma)\frac{\phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + \phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+2}) + \ldots \phi_{t-1}(\mathbf{s}+\mathbf{e}_k)}{a}.$$

For $x = 1/(a+1)$, the expression becomes

$$\gamma \cdot \phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + (1-\gamma)\frac{\phi_{t-1}(\mathbf{s}+\mathbf{e}_1) + \phi_{t-1}(\mathbf{s}+\mathbf{e}_{k-a+1}) + \ldots \phi_{t-1}(\mathbf{s}+\mathbf{e}_k)}{a+1}.$$

Since $b(k-a+1)$ lies in $[0, 1/(a+1)]$, the optimal value is at most the maximum of the two. However each of these last two expressions is at most the right hand side of (44), completing the proof. ■

Unraveling (44), we find that $\phi_t(\mathbf{s})$ is the expected loss of the final state reached by some random walk of t steps starting at state $\mathbf{s}$. However, the number of possibilities for the random-walk is huge; indeed, the distribution at each step can be any of the $k-1$ possibilities $\mathbf{b}_a^\pi$ for $a \in \{2, \ldots, k\}$, where the parameter a denotes the size of the support of the γ-biased uniform distribution chosen at each step. In other words, at a given state $\mathbf{s}$ with t rounds of boosting remaining, the parameter a determines the number of directions the optimal random walk will consider taking; we will therefore refer to a as the *degree* of the random walk given $(\mathbf{s}, t)$. Now, the total number of states reachable in T steps is $O\left(T^{k-1}\right)$. If the degree assignment every such state, for every value of $t \leq T$ is fixed in advance, $\mathbf{a} = \{a(\mathbf{s}, t) : t \leq T, \mathbf{s} \text{ reachable}\}$, we may identify a unique random walk $\mathcal{R}^{\mathbf{a},t}(\mathbf{s})$ of length t starting at step $\mathbf{s}$. Therefore the potential may be computed as

$$\phi_t(\mathbf{s}) = \max_{\mathbf{a}} \mathbb{E}\left[\mathcal{R}^{\mathbf{a},t}(\mathbf{s})\right]. \tag{45}$$

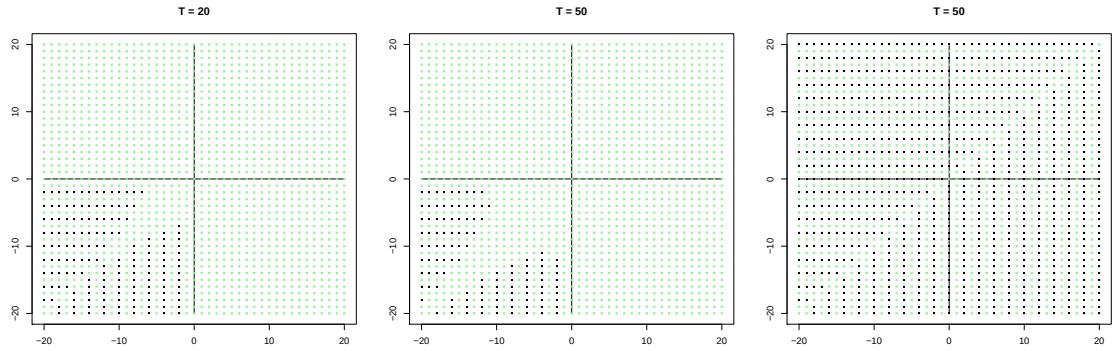


Figure 3: Green pixels (crosses) have degree 3, black pixels (solid squares) have degree 2. Each step is diagonally down (left), and up (if $x < y$) and right (if $x > y$) and both when degree is 3. The rightmost figure uses $\gamma = 0.4$, and the other two $\gamma = 0$. The loss function is 0-1.

A dynamic programming approach for computing (45) requires time and memory linear in the number of different states reachable by a random walk that takes T coordinate steps: $O(T^{k-1})$. This is exponential in the data set size, and hence impractical. In the next two sections we show that perhaps there may not be any way of computing these efficiently in general, but provide efficient algorithms in certain special cases.

Hardness of Evaluating the Potentials. Here we provide empirical evidence for the hardness of computing the new potentials. We first identify a computationally easier problem, and show that even that is probably hard to compute. Equation (44) implies that if the potentials were efficiently computable, the correct value of the degree a could be determined efficiently. The problem of determining the degree a given the state $\mathbf{s}$ and remaining rounds t is therefore easier than evaluating the potentials. However, a plot of the degrees against states and remaining rounds, henceforth called a *degree map*, reveals very little structure that might be captured by a computationally efficient function.

We include three such degree maps in Figure 3. Only three classes $k = 3$ are used, and the loss function is 0-1 error. We also fix the number T of remaining rounds of boosting and the value of the edge γ for each plot. For ease of presentation, the 3-dimensional states $\mathbf{s} = (s_1, s_2, s_3)$ are compressed into 2-dimensional pixel coordinates $(u = s_2 - s_1, v = s_3 - s_2)$. It can be shown that this does not take away information required to evaluate the potentials or the degree at any pixel (u, v). Further, only those states are considered whose compressed coordinates u, v lie in the range $[-T, T]$; in T rounds, these account for all the reachable states. The degrees are indicated on the plot by colors. Our discussion in the previous sections implies that the possible values of the degree is 2 or 3. When the degree at a pixel (u, v) is 3, the pixel is colored green, and when the degree is 2, it is colored black.

Note that a random walk over the space $\mathbf{s} \in \mathbb{R}^3$ consisting of distributions over coordinate steps $\{(1,0,0),(0,1,0),(0,0,1)\}$ translates to a random walk over $(u, v) \in \mathbb{R}^2$ where each step lies in the set $\{(-1,-1),(1,0),(0,1)\}$. In Figure 3, these correspond to the directions diagonally down, up or right. Therefore at a black pixel, the random walk either chooses between diagonally down and up, or between diagonally down and right, with probabilities $\{1/2+\gamma/2, 1/2-\gamma/2\}$. On the

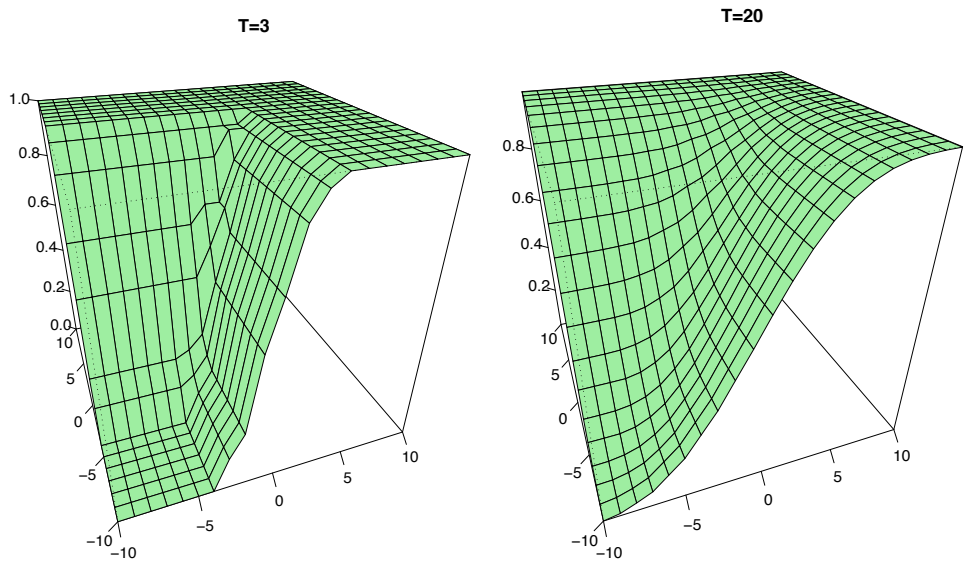


Figure 4: Optimum recurrence value. We set $\gamma = 0$. Surface is irregular for smaller values of T, but smoother for larger values, admitting hope for approximation.

other hand, at a green pixel, the random walk chooses among diagonally down, up and right with probabilities $(\gamma + (1-\gamma)/3, (1-\gamma)/3, (1-\gamma)/3)$. The degree maps are shown for varying values of T and the edge γ. While some patterns emerge for the degrees, such as black or green depending on the parity of u or v, the authors found the region near the line $u = v$ still too complex to admit any solution apart from a brute-force computation.

We also plot the potential values themselves in Figure 4 against different states. In each plot, the number of iterations remaining, T, is held constant, the number of classes is chosen to be 3, and the edge $\gamma = 0$. The states are compressed into pixels as before, and the potential is plotted against each pixel, resulting in a 3-dimensional surface. We include two plots, with different values for T. The surface is irregular for $T = 3$ rounds, but smoother for 20 rounds, admitting some hope for approximation.

An alternative approach would be to approximate the potential ϕ_t by the potential $\phi_t^{\mathbf{b}}$ for some fixed $\mathbf{b} \in \Delta_\gamma^k$ corresponding to some particular edge-over-random condition. Since $\phi_t \geq \phi_t^{\mathbf{b}}$ for all edge-over-random distributions $\mathbf{b}$, it is natural to approximate by choosing $\mathbf{b}$ that maximizes the fixed edge-over-random potential. (It can be shown that this $\mathbf{b}$ corresponds to the γ-biased uniform distribution.) Two plots of comparing the potential values at $\mathbf{0}$, $\phi_T(\mathbf{0})$ and $\max_{\mathbf{b}} \phi_T^{\mathbf{b}}(\mathbf{0})$, which correspond to the respective error upper bounds, is shown in Figure 5. In the first plot, the number of classes k is held fixed at 6, and the values are plotted for different values of iterations T. In the second plot, the number of classes vary, and the number of iterations is held at 10. Both plots show that the difference in the values is significant, and hence $\max_{\mathbf{b}} \phi_T^{\mathbf{b}}(\mathbf{0})$ would be a rather optimistic upper bound on the error when using the minimal weak learning condition.

If we use exponential loss (32), the situation is not much better. The degree maps for varying values of the weight parameter η against fixed values of edge $\gamma = 0.1$, rounds remaining $T = 20$ and

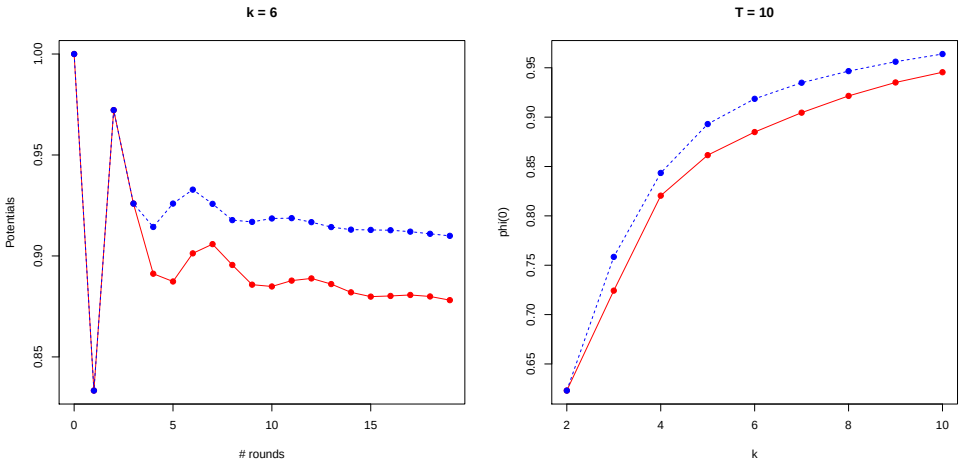


Figure 5: Comparison of $\phi_t(\mathbf{0})$ (blue, dashed) with $\max_{\mathbf{q}} \phi_t^{\mathbf{q}}(\mathbf{0})$ (red, solid) over different rounds t and different number of classes k. We set $\gamma = 0$ in both.

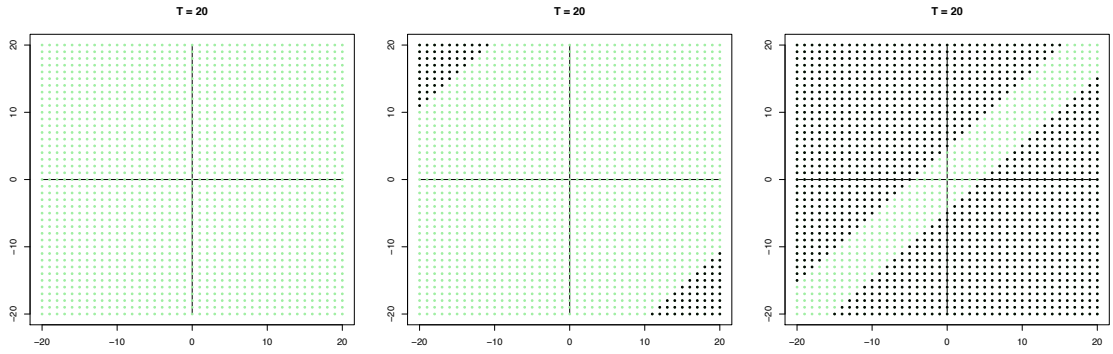


Figure 6: Green pixels (crosses) have degree 3, black pixels (squares) have degree 2. Each step is diagonally down (left), and up (if $x < y$) and right (if $x > y$) and both when degree is 3. Each plot uses $T = 20, \gamma = 0.1$. The values of η are 0.08, 0.1 and 0.3, respectively. With smaller values of η, more pixels have degree 3.

number of classes $k = 3$ are plotted in Figure 6. Although the patterns are simple, with the degree 3 pixels forming a diagonal band, we found it hard to prove this fact formally, or compute the exact boundary of the band. However the plots suggest that when η is small, all pixels have degree 3. We next find conditions under which this opportunity for tractable computation exists.

Efficient Computation in Special Cases. Here we show that when using the exponential loss, if the edge γ is very small, then the potentials can be computed efficiently. We first show an intermediate result. We already observed empirically that when the weight parameter η is small, the degrees all become equal to k. Indeed, we can prove this fact.

Lemma 19 *If the loss function being used is exponential loss* (32) *and the weight parameter* η *is small compared to the number of rounds*

$$\eta \leq \frac{1}{4}\min\left\{\frac{1}{k-1}, \frac{1}{T}\right\}, \tag{46}$$

then the optimal value of the degree a in (44) *is always k. Therefore, in this situation, the potential* ϕ_t *using the minimal weak learning condition is the same as the potential* $\phi_t^{\mathbf{u}}$ *using the* γ*-biased uniform distribution* $\mathbf{u}$*,*

$$\mathbf{u} = \left(\frac{1-\gamma}{k}+\gamma, \frac{1-\gamma}{k}, \ldots, \frac{1-\gamma}{k}\right), \tag{47}$$

and hence can be efficiently computed.

Proof We show $\phi_t = \phi_t^{\mathbf{u}}$ by induction on the remaining number t of boosting iterations. The base case holds since, by definition, $\phi_0 = \phi_0^{\mathbf{u}} = L_\eta^{\exp}$. Assume, inductively that

$$\phi_{t-1}(\mathbf{s}) = \phi_{t-1}^{\mathbf{u}}(\mathbf{s}) = \kappa(\gamma,\eta)^{t-1}\sum_{l=2}^{k} e^{\eta(s_l - s_1)}, \tag{48}$$

where the second equality follows from (34). We show that

$$\phi_t(\mathbf{s}) = \mathbb{E}_{l\sim\mathbf{u}}\left[\phi_{t-1}(\mathbf{s}+\mathbf{e}_l)\right]. \tag{49}$$

By the inductive hypothesis and (27), the right hand side of (49) is in fact equal to $\phi_t^{\mathbf{u}}$, and we will have shown $\phi_t = \phi_t^{\mathbf{u}}$. The proof will then follow by induction.

In order to show (49), by Lemma 18, it suffices to show that the optimal degree a maximizing the right hand side of (44) is always k:

$$\mathbb{E}_{l\sim\mathbf{b}_a^\pi}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right] \leq \mathbb{E}_{l\sim\mathbf{b}_k^\pi}\left[\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_l\right)\right]. \tag{50}$$

By (48), $\phi_{t-1}\left(\mathbf{s}+\mathbf{e}_{l_0}\right)$ may be written as $\phi_{t-1}(\mathbf{s}) + \kappa(\gamma,\eta)^{t-1}\cdot\xi_{l_0}$, where the term ξ_{l_0} is:

$$\xi_{l_0} = \begin{cases} (e^{\eta}-1)e^{\eta(s_{l_0}-s_1)} & \text{if } l_0 \neq 1, \\ (e^{-\eta}-1)\sum_{l=2}^{k} e^{\eta(s_l-s_1)} & \text{if } l_0 = 1. \end{cases}$$

Therefore (50) is the same as: $\mathbb{E}_{l\sim\mathbf{b}_a^\pi}[\xi_l] \leq \mathbb{E}_{l\sim\mathbf{b}_k^\pi}[\xi_l]$. Assume (by relabeling if necessary) that π is the identity permutation on coordinates $\{2,\ldots,k\}$. Then the expression $\mathbb{E}_{l\sim\mathbf{b}_a^\pi}[\xi_l]$ can be written as

$$\begin{aligned} \mathbb{E}_{l\sim\mathbf{b}_a^\pi}[\xi_l] &= \left(\frac{1-\gamma}{a}+\gamma\right)\xi_1 + \sum_{l=k-a+2}^{k}\left(\frac{1-\gamma}{a}\right)\xi_l \\ &= \gamma\xi_1 + (1-\gamma)\left\{\frac{\xi_1 + \sum_{l=k-a+2}^{k}\xi_l}{a}\right\}. \end{aligned}$$

It suffices to show that the term in curly brackets is maximized when $a = k$. We will in fact show that the numerator of the term is negative if $a < k$, and non-negative for $a = k$, which will complete

our proof. Notice that the numerator can be written as

$$\begin{aligned}
& (e^{\eta}-1)\left\{\sum_{l=k-a+2}^{k} e^{\eta(s_l-s_1)}\right\} - (1-e^{-\eta})\sum_{l=2}^{k} e^{\eta(s_l-s_1)} \\
= \;& (e^{\eta}-1)\left\{\sum_{l=k-a+2}^{k} e^{\eta(s_l-s_1)} - \sum_{l=2}^{k} e^{\eta(s_l-s_1)}\right\} + \left\{(e^{\eta}-1)-(1-e^{-\eta})\right\}\sum_{l=2}^{k} e^{\eta(s_l-s_1)} \\
= \;& \left\{e^{\eta}+e^{-\eta}-2\right\}\sum_{l=2}^{k} e^{\eta(s_l-s_1)} - (e^{\eta}-1)\left\{\sum_{l=2}^{k-a+1} e^{\eta(s_l-s_1)}\right\}.
\end{aligned}$$

When $a = k$, the second summation disappears, and we are left with a non-negative expression. However when $a < k$, the second summation is at least $e^{\eta(s_2-s_1)}$. Since $t \le T$, and in t iterations the absolute value of any state coordinate $|s_t(l)|$ is at most T, the first summation is at most $(k-1)e^{2\eta T}$ and the second summation is at least $e^{-2\eta T}$. Therefore the previous expression is at most

$$\begin{aligned}
& (k-1)\left(e^{\eta}+e^{-\eta}-2\right)e^{2\eta T} - (e^{\eta}-1)e^{-2\eta T} \\
= \;& (e^{\eta}-1)e^{-2\eta T}\left\{(k-1)(1-e^{-\eta})e^{4\eta T}-1\right\}.
\end{aligned}$$

We show that the term in curly brackets is negative. Firstly, using $e^x \ge 1+x$, we have $1-e^{-\eta} \le \eta \le 1/(4(k-1))$ by choice of η. Therefore it suffices to show that $e^{4\eta T} < 4$. By choice of η again, $e^{4\eta T} \le e^1 < 4$. This completes our proof. ■

The above lemma seems to suggest that under certain conditions, a sort of degeneracy occurs, and the optimal Booster payoff (15) is nearly unaffected by whether we use the minimal weak learning condition, or the condition $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$. Indeed, we next prove this fact.

Theorem 20 *Suppose the loss function is as in Lemma 19, and for some parameter $\varepsilon > 0$, the number of examples m is large enough*

$$m \ge \frac{Te^{1/4}}{\varepsilon}. \tag{51}$$

Consider the minimal weak learning condition (13)*, and the fixed edge-over-random condition $(\mathcal{C}^{eor}, \mathbf{U}_\gamma)$ corresponding to the γ-biased uniform baseline $\mathbf{U}_\gamma$. Then the optimal booster payoffs using either condition is within ε of each other.*

Proof We show that the OS strategies arising out of either condition is the same. In other words, at any iteration t and state $\mathbf{s}_t$, both strategies play the same cost matrix and enforce the same constraints on the response of Weak-Learner. The theorem will then follow if we can invoke Theorems 9 and 16. For that, the number of examples needs to be as large as in (25). The required largeness would follow from (51) if the loss function satisfied (24) with $\varnothing(L,T)$ at most $\exp(1/4)$. Since the largest discrepancy in losses between two states reachable in T iterations is at most $e^{\eta T} - 0$, the bound follows from the choice of η in (46). Therefore, it suffices to show the equivalence of the OS strategies corresponding to the two weak learning conditions.

We first show both strategies play the same cost-matrix. Lemma 19 states that the potential function using the minimal weak learning condition is the same as when using the fixed condition $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$: $\phi_t = \phi_t^{\mathbf{u}}$, where $\mathbf{u}$ is as in (47). Since, according to (28) and (41), given a state $\mathbf{s}_t$ and iteration t, the two strategies choose cost matrices that are identical functions of the respective

potentials, by the equivalence of the potential functions, the resulting cost matrices must be the same.

Even with the same cost matrix, the two different conditions could be imposing different constraints on Weak-Learner, which might affect the final payoff. For instance, with the baseline $\mathbf{U}_\gamma$, Weak-Learner has to return a weak classifier h satisfying

$$\mathbf{C}_t \bullet \mathbf{1}_h \leq \mathbf{C}_t \bullet \mathbf{U}_\gamma,$$

whereas with the minimal condition, the constraint on h is

$$\mathbf{C}_t \bullet \mathbf{1}_h \leq \max_{\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}} \mathbf{C}_t \bullet \mathbf{B}.$$

In order to show that the constraints are the same we therefore need to show that for the common cost matrix $\mathbf{C}_t$ chosen, the right hand side of the two previous expressions are the same:

$$\mathbf{C}_t \bullet \mathbf{U}_\gamma = \max_{\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}} \mathbf{C}_t \bullet \mathcal{B}_\gamma^{\text{eor}}. \tag{52}$$

We will in fact show the stronger fact that the equality holds for every row separately:

$$\forall i : \langle \mathbf{C}_t(i), \mathbf{u} \rangle = \max_{\mathbf{b} \in \Delta_\gamma^k} \langle \mathbf{C}_t(i), \mathbf{b} \rangle . \tag{53}$$

To see this, first observe that the choice of the optimal cost matrix $\mathbf{C}_t$ in (41) implies the following identity

$$\langle \mathbf{C}_t(i), \mathbf{b} \rangle = \mathbb{E}_{l \sim \mathbf{b}} \left[\phi_{T-t-1}(\mathbf{s}_t(i) + \mathbf{e}_l) \right].$$

On the other hand, (44) and Lemma 19 together imply that the distribution $\mathbf{b}$ maximizing the right hand side of the above is the γ-biased uniform distribution, from which (53) follows. Therefore, the constraints placed on Weak-Learner by the cost-matrix $\mathbf{C}_t$ is the same whether we use minimum weak learning condition or the fixed condition $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$. ■

One may wonder why η would be chosen so small, especially since the previous theorem indicates that such choices for η lead to degeneracies. To understand this, recall that η represents the size of the weights α_t chosen in every round, and was introduced as a tunable parameter to help achieve the best possible upper bound on zero-one error. More precisely, recall that the exponential loss $L_\eta^{\exp}(\mathbf{s})$ of the unweighted state, defined in (32), is equal to the exponential loss $L^{\exp}(\mathbf{f})$ on the weighted state, defined in (33), which in turn is an upper bound on the error $L^{\text{err}}(\mathbf{f}_T)$ of the final weighted state $\mathbf{f}_T$. Therefore the potential value $\phi_T(\mathbf{0})$ based on the exponential loss $L_\eta^{\exp}$ is an upper bound on the minimum error attainable after T rounds of boosting. At the same time, $\phi_T(\mathbf{0})$ is a function of η. Therefore, we may tune this parameter to attain the best bound possible. Even with this motivation, it may seem that a properly tuned η will not be as small as in Lemma 19, especially since it can be shown that the resulting loss bound $\phi_T(\mathbf{0})$ will always be larger than a fixed constant (depending on γ, k), no matter how many rounds T of boosting is used. However, the next result identifies conditions under which the tuned value of η is indeed as small as in Lemma 19. This happens when the edge γ is very small, as is described in the next theorem. Intuitively, a weak classifier achieving small edge has low accuracy, and a low weight reflects Booster's lack of confidence in this classifier.

Theorem 21 *When using the exponential loss function* (32)*, and the minimal weak learning condition* (13)*, the loss upper bound $\phi_T(\mathbf{0})$ provided by Theorem 16 is more than* 1 *and hence trivial unless the parameter η is chosen sufficiently small compared to the edge γ:*

$$\eta \leq \frac{k\gamma}{1-\gamma}. \tag{54}$$

In particular, when the edge is very small:

$$\gamma \leq \min\left\{\frac{1}{2}, \frac{1}{8k}\min\left\{\frac{1}{k}, \frac{1}{T}\right\}\right\}, \tag{55}$$

the value of η needs to be as small as in (46)*.*

Proof Comparing solutions (45) and (31) to the potentials corresponding to the minimal weak learning condition and a fixed edge-over-random condition, we may conclude that the loss bound $\phi_T(\mathbf{0})$ is in the former case is larger than $\phi_T^{\mathbf{b}}(\mathbf{0})$, for any edge-over-random distribution $\mathbf{b} \in \Delta_\gamma^k$. In particular, when $\mathbf{b}$ is set to be the γ-biased uniform distribution $\mathbf{u}$, as defined in (47), we get $\phi_T(\mathbf{0}) \geq \phi_T^{\mathbf{u}}(\mathbf{0})$. Now the latter bound, according to (34), is $\kappa(\gamma,\eta)^T$, where κ is defined as in (35). Therefore, to get non-trivial loss bounds which are at most 1, we need to choose η such that $\kappa(\gamma,\eta) \leq 1$. By (35), this happens when

$$\begin{aligned} \left(1-e^{-\eta}\right)\gamma &\geq \left(e^{\eta}+e^{-\eta}-2\right)\left(\frac{1-\gamma}{k}\right) \\ \text{i.e., } \frac{k\gamma}{1-\gamma} &\geq \frac{e^{\eta}+e^{-\eta}-2}{1-e^{-\eta}} = e^{\eta}-1 \geq \eta. \end{aligned}$$

Therefore (54) holds. When γ is as small as in (55), then $1-\gamma \leq \frac{1}{2}$, and therefore, by (54), the bound on η becomes identical to that in (55). ■

The condition in the previous theorem, that of the existence of only a very small edge, is the most we can assume for most practical data sets. Therefore, in such situations, we can compute the optimal Booster strategy that uses the minimal weak learning conditions. More importantly, using this result, we derive, in the next section, a highly efficient and practical *adaptive* algorithm, that is, one that does not require any prior knowledge about the edge γ, and will therefore work with any data set.

8. Variable Edges

So far we have required Weak-Learner to beat random by at least a fixed amount $\gamma > 0$ in each round of the boosting game. In reality, the edge over random is larger initially, and gets smaller as the OS algorithm creates harder cost matrices. Therefore requiring a fixed edge is either unduly pessimistic or overly optimistic. If the fixed edge is too small, not enough progress is made in the initial rounds, and if the edge is too large, Weak-Learner fails to meet the weak-learning condition in latter rounds. We fix this by not making any assumption about the edges, but instead *adaptively* responding to the edges returned by Weak-Learner. In the rest of the section we describe the adaptive procedure, and the resulting loss bounds guaranteed by it.

The philosophy behind the adaptive algorithm is a boosting game where Booster and Weak Learner no longer have opposite goals, but cooperate to reduce error as fast as possible. However, in

order to create a clean abstraction and separate implementations of the boosting algorithms and the weak learning procedures as much as possible, we assume neither of the players has any knowledge of the details of the algorithm employed by the other player. In particular Booster may only assume that Weak Learner's strategy is barely strong enough to guarantee boosting. Therefore, Booster's demands on the weak classifiers returned by Weak Learner should be minimal, and it should send the weak learning algorithm the "easiest" cost matrices that will ensure boostability. In turn, Weak Learner may only assume a very weak Booster strategy, and therefore return a weak classifier that performs as well as possible with respect to the cost matrix sent by Booster.

At a high level, the adaptive strategy proceeds as follows. At any iteration, based on the states of the examples and number of remaining rounds of boosting, Booster chooses the game-theoretically optimal cost matrix assuming only infinitesimal edges in the remaining rounds. Intuitively, Booster has no high expectations of Weak Learner, and supplies it the easiest cost matrices with which it may be able to boost. However, in the adaptive setting, Weak-Learner is no longer adversarial. Therefore, although only infinitesimal edges are anticipated by Booster, Weak Learner cooperates in returning weak classifiers that achieve as large edges as possible, which will be more than just inifinitesimal. Based on the exact edge received in each round, Booster chooses the weight α_t adaptively to reach the most favourable state possible. Therefore, Booster plays game theoretically assuming an adversarial Weak Learner and expecting only the smallest edges in the future rounds, although Weak Learner actually cooperates, and Booster adaptively exploits this favorable behavior as much as possible. This way the boosting algorithm remains robust to a poorly performing Weak Learner, and yet can make use of a powerful weak learning algorithm whenever possible.

We next describe the details of the adaptive procedure. With variable weights we need to work with the weighted state $\mathbf{f}_t(i)$ of each example i, defined in (19). To keep the computations tractable, we will only be working with the exponential loss $L^{\exp}(\mathbf{f})$ on the weighted states. We first describe how Booster chooses the cost-matrix in each round. Following that we describe how it adaptively computes the weights in each round based on the edge of the weak classifier received.

8.1 Choosing the Cost-matrix

As discussed before, at any iteration t and state $\mathbf{f}_t$ Booster assumes that it will receive an infinitesimal edge γ in each of the remaining rounds. Since the step size is a function of the edge, which in turn is expected to be the same tiny value in each round, we may assume that the step size in each round will also be some fixed value η. We are therefore in the setting of Theorem 21, which states that the parameter η in the exponential loss function (32) should also be tiny to get any non-trivial bound. But then the loss function satisfies the conditions in Lemma 19, and by Theorem 20, the game theoretically optimal strategy remains the same whether we use the minimal condition or $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$. When using the latter condition, the optimal choice of the cost-matrix at iteration t and state $\mathbf{f}_t$, according to (36), is

$$C_t(i,l) = \begin{cases} (e^{\eta} - 1)\, e^{f_{t-1}(i,j) - f_{t-1}(i,1)} & \text{if } l > 1, \\ (e^{-\eta} - 1) \sum_{j=2}^{k} e^{f_{t-1}(i,j) - f_{t-1}(i,1)} & \text{if } l = 1. \end{cases}$$

Further, when using the condition $(\mathcal{C}^{\text{eor}}, \mathbf{U}_\gamma)$, the average potential of the states $\mathbf{f}_t(i)$, according to (34), is given by the average loss (37) of the state times $\kappa(\gamma,\eta)^{T-t}$, where the function κ is defined in (35). Our goal is to choose η as a function of γ so that $\kappa(\gamma,\eta)$ is as small as possible. Now, there is no lower bound on how small the edge γ may get, and, anticipating the worst, it makes sense to

choose an infinitesimal γ, in the spirit of Freund (2001). Equation (35) then implies that the choice of η should also be infinitesimal. Then the above choice of the cost matrix becomes the following (after some rescaling):

$$\begin{aligned} C_t(i,l) &= \lim_{\eta\to 0} C_\eta(i,l) \triangleq \frac{1}{\eta}\begin{cases} (e^{\eta}-1)\, e^{f_{t-1}(i,j)-f_{t-1}(i,1)} & \text{if } l>1, \\ (e^{-\eta}-1)\sum_{j=2}^{k} e^{f_{t-1}(i,j)-f_{t-1}(i,1)} & \text{if } l=1. \end{cases} \\ &= \begin{cases} e^{f_{t-1}(i,j)-f_{t-1}(i,1)} & \text{if } l>1, \\ -\sum_{j=2}^{k} e^{f_{t-1}(i,j)-f_{t-1}(i,1)} & \text{if } l=1. \end{cases} \end{aligned} \tag{56}$$

We have therefore derived the optimal cost matrix played by the adaptive boosting strategy, and we record this fact.

Lemma 22 *Consider the boosting game using the minimal weak learning condition* (13)*. Then, in iteration t at state $\mathbf{f}_t$, the game-theoretically optimal Booster strategy chooses the cost matrix $\mathbf{C}_t$ given in* (56)*.*

We next show how to adaptively choose the weights α_t.

8.2 Adaptively Choosing Weights

Once Weak Learner returns a weak classifier h_t, Booster chooses the optimum weight α_t so that the resulting states $\mathbf{f}_t = \mathbf{f}_{t-1} + \alpha_t \mathbf{1}_{h_t}$ are as favorable as possible, that is, minimizes the total potential of its states. By our previous discussions, these are proportional to the total loss given by $Z_t = \sum_{i=1}^{m}\sum_{l=2}^{k} e^{f_t(i,l)-f_t(i,1)}$. For any choice of α_t, the difference $Z_t - Z_{t-1}$ between the total loss in rounds $t-1$ and t is given by

$$\begin{aligned} & (e^{\alpha_t}-1)\sum_{i\in S_-} e^{f_{t-1}(i,h_t(i))-f_{t-1}(i,1)} - \left(1-e^{-\alpha_t}\right)\sum_{i\in S_+} L^{\exp}(\mathbf{f}_{t-1}(i)) \\ = \;& (e^{\alpha_t}-1)A_-^t - \left(1-e^{-\alpha_t}\right)A_+^t \\ = \;& \left(A_+^t e^{-\alpha_t} + A_-^t e^{\alpha_t}\right) - \left(A_+^t + A_-^t\right), \end{aligned}$$

where S_+ denotes the set of examples that h_t classifies correctly, S_- the incorrectly classified examples, and A_-^t, A_+^t denote the first and second summations, respectively. Therefore, the task of choosing α_t can be cast as a simple optimization problem minimizing the previous expression. In fact, the optimal value of α_t is given by the following closed form expression

$$\alpha_t = \frac{1}{2}\ln\left(\frac{A_+^t}{A_-^t}\right). \tag{57}$$

With this choice of weight, one can show (with some straightforward algebra) that the total loss of the state falls by a factor less than 1. In fact the factor is exactly

$$(1-c_t) - \sqrt{c_t^2 - \delta_t^2}, \tag{58}$$

where

$$c_t = (A_+^t + A_-^t)/Z_{t-1},$$

and δ_t is the edge of the returned classifier h_t on the supplied cost-matrix $\mathbf{C}_t$. Notice that the quantity c_t is at most 1, and hence the factor (58) can be upper bounded by $\sqrt{1-\delta_t^2}$. We next show how to compute the edge δ_t. The definition of the edge depends on the weak learning condition being used, and in this case we are using the minimal condition (13). Therefore the edge δ_t is the largest γ such that the following still holds

$$\mathbf{C}_t \bullet \mathbf{1}_h \leq \max_{\mathbf{B} \in \mathcal{B}_\gamma^{\text{eor}}} \mathbf{C}_t \bullet \mathbf{B}.$$

However, since $\mathbf{C}_t$ is the optimal cost matrix when using exponential loss with a tiny value of η, we can use arguments in the proof of Theorem 20 to simplify the computation. In particular, eq. (52) implies that the edge δ_t may be computed as the largest γ satisfying the following simpler inequality

$$\begin{aligned}
\delta_t &= \sup\left\{\gamma : \mathbf{C}_t \bullet \mathbf{1}_{h_t} \leq \mathbf{C}_t \bullet \mathbf{U}_\gamma\right\} \\
&= \sup\left\{\gamma : \mathbf{C}_t \bullet \mathbf{1}_{h_t} \leq -\gamma \sum_{i=1}^{m}\sum_{l=2}^{k} e^{f_{t-1}(i,l)-f_{t-1}(i,1)}\right\} \\
\implies \delta_t &= \gamma :\ \mathbf{C}_t \bullet \mathbf{1}_{h_t} = -\gamma \sum_{i=1}^{m}\sum_{l=2}^{k} e^{f_{t-1}(i,l)-f_{t-1}(i,1)} \\
\implies \delta_t &= \frac{-\mathbf{C}_t \bullet \mathbf{1}_{h_t}}{\sum_{i=1}^{m}\sum_{l=2}^{k} e^{f_{t-1}(i,l)-f_{t-1}(i,1)}} = \frac{-\mathbf{C}_t \bullet \mathbf{1}_{h_t}}{Z_t},
\end{aligned} \tag{59}$$

where the first step follows by expanding $\mathbf{C}_t \bullet \mathbf{U}_\gamma$. We have therefore an adaptive strategy which efficiently reduces error. We record our results.

Lemma 23 *If the weight α_t in each round is chosen as in* (57)*, and the edge δ_t is given by* (59)*, then the total loss Z_t falls by the factor given in* (58)*, which is at most $\sqrt{1-\delta_t^2}$.*

The choice of α_t in (57) is optimal, but depends on quantities other than just the edge δ_t. We next show a way of choosing α_t based only on δ_t that still causes the total loss to drop by a factor of $\sqrt{1-\delta_t^2}$.

Lemma 24 *Suppose cost matrix $\mathbf{C}_t$ is chosen as in* (56)*, and the returned weak classifier h_t has edge δ_t, that is, $\mathbf{C}_t \bullet \mathbf{1}_{h_t} \leq \mathbf{C}_t \bullet \mathbf{U}_{\delta_t}$. Then choosing any weight $\alpha_t > 0$ for h_t makes the loss Z_t at most a factor*

$$1 - \frac{1}{2}(e^{\alpha_t} - e^{-\alpha_t})\delta_t + \frac{1}{2}(e^{\alpha_t} + e^{-\alpha_t} - 2)$$

of the previous loss Z_{t-1}. In particular by choosing

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1+\delta_t}{1-\delta_t}\right), \tag{60}$$

the drop factor is at most $\sqrt{1-\delta_t^2}$.

Proof We borrow notation from earlier discussions. The edge-condition implies

$$A_-^t - A_+^t = \mathbf{C}_t \bullet \mathbf{1}_{h_t} \leq \mathbf{C}_t \bullet \mathbf{U}_{\delta_t} = -\delta_t Z_{t-1} \implies A_+^t - A_-^t \geq \delta_t Z_{t-1}.$$

On the other hand, the drop in loss after choosing h_t with weight α_t is

$$\begin{aligned} & \left(1-e^{-\alpha_t}\right)A_+^t - (e^{\alpha_t}-1)A_-^t \\ = & \left(\frac{e^{\alpha_t}-e^{-\alpha_t}}{2}\right)\left(A_+^t - A_-^t\right) - \left(\frac{e^{\alpha_t}+e^{-\alpha_t}-2}{2}\right)\left(A_+^t + A_-^t\right). \end{aligned}$$

We have already shown that $A_+^t - A_-^t \geq \delta_t Z_{t-1}$. Further, $A_+^t + A_-^t$ is at most Z_{t-1}. Therefore the loss drops by a factor of at least

$$1 - \frac{1}{2}(e^{\alpha_t}-e^{-\alpha_t})\delta_t + \frac{1}{2}(e^{\alpha_t}+e^{-\alpha_t}-2) = \frac{1}{2}\left\{(1-\delta_t)e^{\alpha_t} + (1+\delta_t)e^{-\alpha_t}\right\}.$$

Tuning α_t as in (60) causes the drop factor to be at least $\sqrt{1-\delta_t^2}$. ■

Algorithm 1 contains pseudocode for the adaptive algorithm, and includes both ways of choosing α_t. We call both versions of this algorithm AdaBoost.MM. With the approximate way of choosing the step length in (61), AdaBoost.MM turns out to be identical to AdaBoost.M2 (Freund and Schapire, 1997) or AdaBoost.MR (Schapire and Singer, 1999), provided the weak classifier space is transformed in an appropriate way to be acceptable by AdaBoost.M2 or AdaBoost.MR. We emphasize that AdaBoost.MM and AdaBoost.M2 are products of very different theoretical considerations, and this similarity should be viewed as a coincidence arising because of the particular choice of loss function, infinitesimal edge and approximate step size. For instance, when the step sizes are chosen instead as in (62), the training error falls more rapidly, and the resulting algorithm is different.

As a summary of all the discussions in the section, we record the following theorem.

Theorem 25 *The boosting algorithm AdaBoost.MM, shown in Algorithm 1, is the optimal strategy for playing the adaptive boosting game, and is based on the minimal weak learning condition. Further if the edges returned in each round are $\delta_1, \ldots, \delta_T$, then the error after T rounds is $(k-1)\prod_{t=1}^T \sqrt{1-\delta_t^2} \leq (k-1)\exp\left\{-(1/2)\sum_{t=1}^T \delta_t^2\right\}$.*

In particular, if a weak hypothesis space is used that satisfies the optimal weak learning condition (13)*, for some γ, then the edge in each round is large, $\delta_t \geq \gamma$, and therefore the error after T rounds is exponentially small, $(k-1)e^{-T\gamma^2/2}$.*

The theorem above states that as long as the minimal weak learning condition is satisfied, the error will decrease exponentially fast. Even if the condition is not satisfied, the error rate will keep falling rapidly provided the edges achieved by the weak classifiers are relatively high. However, our theory so far can provide no guarantees on these edges, and therefore it is not clear what is the best error rate achievable in this case, and how quickly it is achieved. The assumptions of boostability, and hence our minimal weak learning condition does not hold for the vast majority of practical data sets, and as such it is important to know what happens in such settings. In particular, an important requirement is *empirical consistency*, where we want that for any given weak classifier space, the algorithm converge, if allowed to run forever, to the weighted combination of classifiers that minimizes error on the training set. Another important criterion is *universal consistency*, which requires that the algorithm converge, when provided sufficient training data, to the classifier combination that minimizes error on the test data set. In the next section, we show that AdaBoost.MM satisfies such consistency requirements. Both the choice of the minimal weak learning condition as well as the setup of the adaptive game framework will play crucial roles in ensuring consistency. These results therefore provide evidence that game theoretic considerations can have strong statistical implications.

Algorithm 1 AdaBoost.MM

Require: Number of classes k, number of examples m.

Require: Training set $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ with $y_i \in \{1,\ldots,k\}$ and $x_i \in X$.

- Initialize $m \times k$ matrix $f_0(i,l) = 0$ for $i = 1,\ldots,m$, and $l = 1,\ldots,k$.

for $t = 1$ to T **do**

- Choose cost matrix $\mathbf{C}_t$ as follows:

$$C_t(i,l) \quad = \quad \begin{cases} e^{f_{t-1}(i,l)-f_{t-1}(i,y_i)} & \text{if } l \neq y_i, \\ -\sum_{l \neq y_i} e^{f_{t-1}(i,j)-f_{t-1}(i,y_i)} & \text{if } l = 1. \end{cases}$$

- Receive weak classifier $h_t : X \to \{1,\ldots,k\}$ from weak learning algorithm
- Compute edge δ_t as follows:

$$\delta_t = \frac{-\sum_{i=1}^m C_t(i,h_t(x_i))}{\sum_{i=1}^m \sum_{l \neq y_i} e^{f_{t-1}(i,l)-f_{t-1}(i,y_i)}}$$

- Choose α_t either as

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1+\delta_t}{1-\delta_t} \right), \tag{61}$$

or, for a slightly bigger drop in the loss, as

$$\alpha_t = \frac{1}{2} \ln \left(\frac{\sum_{i:h_t(x_i)=y_i} \sum_{l \neq y_i} e^{f_{t-1}(i,l)-f_{t-1}(i,y_i)}}{\sum_{i:h_t(x_i)\neq y_i} e^{f_{t-1}(i,h_t(x_i))-f_{t-1}(i,y_i)}} \right) \tag{62}$$

- Compute $\mathbf{f}_t$ as:

$$f_t(i,l) = f_{t-1}(i,l) + \alpha_t \mathbf{1}\left[h_t(x_i) = l\right].$$

end for

- Output weighted combination of weak classifiers $F_T : X \times \{1,\ldots,k\} \to \mathbb{R}$ defined as:

$$F_T(x,l) = \sum_{t=1}^T \alpha_t \mathbf{1}\left[h_t(x) = l\right]. \tag{63}$$

- Based on F_T, output a classifier $H_T : X \to \{1,\ldots,k\}$ that predicts as

$$H_T(x) = \operatorname*{argmax}_{l=1}^{k} F_T(x,l).$$

9. Consistency of the Adaptive Algorithm

The goal in a classification task is to design a classifier that predicts with high accuracy on unobserved or test data. This is usually carried out by ensuring the classifier fits training data well without being overly complex. Assuming the training and test data are reasonably similar, one can show that the above procedure achieves high test accuracy, or is consistent. Here we work in a

probabilistic setting that connects training and test data by assuming both consist of examples and labels drawn from a common, unknown distribution.

Consistency for multiclass classification in the probabilistic setting has been studied by Tewari and Bartlett (2007), who show that, unlike in the binary setting, many natural approaches fail to achieve consistency. In this section, we show that AdaBoost.MM described in the previous section avoids such pitfalls and enjoys various consistency results. We begin by laying down some standard assumptions and setting up some notation. Then we prove our first result showing that our algorithm minimizes a certain exponential loss function on the training data at a fast rate. Next, we build upon this result and improve along two fronts: firstly we change our metric from exponential loss to the more relevant classification error metric, and secondly we show fast convergence on not just training data, but also the test set. For the proofs, we heavily reuse existing machinery in the literature.

Throughout the rest of this section we consider the version of AdaBoost.MM that picks weights according to the approximate rule in (61). All our results most probably hold with the other rule for picking weights in (62) as well, but we did not verify that. These results hold without any boostability requirements on the space $\mathcal{H}$ of weak classifiers, and are therefore widely applicable in practice. While we do not assume any weak learning condition, we will require a fully cooperating Weak Learner. In particular, we will require that in each round Weak Learner picks the weak classifier suffering minimum cost with respect to the cost matrix provided by the boosting algorithm, or equivalently achieves the highest edge as defined in (59). Such assumptions are both necessary and standard in the literature, and are frequently met in practice.

In order to state our results, we will need to setup some notation. The space of examples will be denoted by $\mathcal{X}$, and the set of labels by $\mathcal{Y} = \{1, \ldots, k\}$. We also fix a finite weak classifier space $\mathcal{H}$ consisting of classifiers $h : \mathcal{X} \to \mathcal{Y}$. We will be interested in functions $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that assign a score to every example and label pair. Important examples of such functions are the weighted majority combinations (63) output by the adaptive algorithm. In general, any such combination of the weak classifiers in space $\mathcal{H}$ is specified by some weight function $\alpha : \mathcal{H} \to \mathbb{R}$; the resulting function is denoted by $F_\alpha : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and satisfies:

$$F_\alpha(x,l) = \sum_{h \in \mathcal{H}} \alpha(h) \mathbf{1}\left[h(x) = l\right].$$

We will be interested in measuring the average exponential loss of such functions. To measure this, we introduce the $\widehat{\text{risk}}$ operator:

$$\widehat{\text{risk}}(F) \stackrel{\triangle}{=} \frac{1}{m} \sum_{i=1}^{m} \sum_{l \neq y_i} e^{F(x_i,l) - F(x_i,y_i)}.$$

With this setup, we can now state our simplest consistency result, which ensures that the algorithm converges to a weighted combination of classifiers in the space $\mathcal{H}$ that achieves the minimum exponential loss over the training set at an efficient rate.

Lemma 26 *The* $\widehat{\text{risk}}$ *of the predictions* F_T*, as defined in* (63)*, converges to that of the optimal predictions of any combination of the weak classifiers in* $\mathcal{H}$ *at the rate* $O(1/T)$*:*

$$\widehat{\text{risk}}(F_T) - \inf_{\alpha : \mathcal{H} \to \mathbb{R}} \widehat{\text{risk}}(F_\alpha) \leq \frac{C}{T},$$

where C is a constant depending only on the data set.

A slightly stronger result would state that the average exponential loss when measured with respect to the *test set*, and not just the empirical set, also converges. The test set is generated by some target distribution D over example label pairs, and we introduce the risk_D operator to measure the exponential loss for any function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ with respect to D:

$$\text{risk}_D(F) = \mathbb{E}_{(x,y)\sim D}\left[\sum_{l\neq y} e^{F(x,l)-F(x,y)}\right].$$

We show this stronger result holds if the function F_T is modified to the function $\bar{F}_T : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that takes values in the range $[0, -C]$, for some large constant C:

$$\bar{F}_T(x,l) \stackrel{\triangle}{=} \max\left\{-C, F_T(x,l) - \max_{l'} F_T(x,l')\right\}. \tag{64}$$

Lemma 27 *If $\bar{F}_T$ is as in* (64)*, and the number of rounds T is set to $T_m = \sqrt{m}$, then its* risk_D *converges to the optimal value as $m \to \infty$ with high probability:*

$$\Pr\left[\text{risk}_D\left(\bar{F}_{T_m}\right) \leq \inf_{F:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \text{risk}_D(F) + O\left(m^{-c}\right)\right] \geq 1 - \frac{1}{m^2},$$

where $c > 0$ is some absolute constant, and the probability is over the draw of training examples.

We prove Lemmas 26 and 27 by demonstrating a strong correspondence between AdaBoost.MM and binary AdaBoost, and then leveraging almost identical known consistency results for AdaBoost (Bartlett and Traskin, 2007). Our proofs will closely follow the exposition in Chapter 12 of Schapire and Freund (2012) on the consistency of AdaBoost, and are deferred to the appendix.

So far we have focused on risk_D, but a more desirable consistency result would state that the test *error* of the final classifier output by AdaBoost.MM converges to the Bayes optimal error. The test error is measured by the err_D operator, and is given by

$$\text{err}_D(H) = \Pr_{(x,y)\sim D}\left[H(x) \neq y\right].$$

The Bayes optimal classifier H_{opt} is a classifier achieving the minimum error among all possible classifying functions

$$\text{err}_D(H_{\text{opt}}) = \inf_{H:\mathcal{X}\to\mathcal{Y}} \text{err}_D(H),$$

and we want our algorithm to output a classifier whose err_D approaches $\text{err}_D(H_{\text{opt}})$. In designing the algorithm, our main focus was on reducing the exponential loss, captured by risk_D and $\widehat{\text{risk}}$. Unless these loss functions are aligned properly with classification error, we cannot hope to achieve optimal error. The next result shows that our loss functions are correctly aligned, or more technically *Bayes consistent*. In other words, if a scoring function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is close to achieving optimal risk_D, then the classifier $H : \mathcal{X} \to \mathcal{Y}$ derived from it as follows:

$$H(x) \in \operatorname*{argmax}_{l\in\mathcal{Y}} F(x,y), \tag{65}$$

also approaches the Bayes optimal error.

Lemma 28 *Suppose F is a scoring function achieving close to optimal risk*

$$\mathrm{risk}_D(F) \leq \inf_{F':\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathrm{risk}_D(F') + \varepsilon, \tag{66}$$

for some $\varepsilon \geq 0$. If H is the classifier derived from it as in (65)*, then it achieves close to the Bayes optimal error*

$$\mathrm{err}_D(H) \leq \mathrm{err}_D(H_{\mathrm{opt}}) + \sqrt{2\varepsilon}.$$

Proof The proof is similar to that of Theorem 12.1 in Schapire and Freund (2012), which in turn is based on the work by Zhang (2004) and Bartlett et al. (2006). Let $p(x) = \Pr_{(x',y')\sim D}(x' = x)$ denote the the marginalized probability of drawing example x from D, and let $p^x_y = \Pr_{(x',y')\sim D}[y' = y | x' = x]$ denote the conditional probability of drawing label y given we have drawn example x. We first rewrite the difference in errors between H and H_{opt} using these probabilities. Firstly note that the accuracy of any classifier H' is given by

$$\sum_{x\in\mathcal{X}} D(x, H'(x)) = \sum_{x\in\mathcal{X}} p(x) p^x_{H'(x)}.$$

If $\mathcal{X}'$ is the set of examples where the predictions of H and H_{opt} differ, $\mathcal{X}' = \left\{x \in \mathcal{X} : H(x) \neq H_{\mathrm{opt}}(x)\right\}$, then we may bound the error differences as

$$\mathrm{err}_D(H) - \mathrm{err}_D(H_{\mathrm{opt}}) = \sum_{x\in\mathcal{X}'} p(x)\left(p^x_{H_{\mathrm{opt}}(x)} - p^x_{H(x)}\right). \tag{67}$$

We next relate this expression to the difference of the losses.

Notice that for any scoring function F', the risk_D can be rewritten as follows :

$$\mathrm{risk}_D(F') = \sum_{x\in\mathcal{X}} p(x) \sum_{l<l'} \left\{ p^x_l e^{F'(x,l')-F'(x,l)} + p^x_{l'} e^{F'(x,l)-F'(x,l')} \right\}.$$

Denote the inner summation in curly brackets by $L^{l,l'}_{F'}(x)$, and notice this quantity is minimized if

$$e^{F'(x,l)-F'(x,l')} = \sqrt{p^x_l / p^x_{l'}}, \quad \text{i.e., if } F'(x,l) - F'(x,l') = \tfrac{1}{2}\ln p^x_l - \tfrac{1}{2}\ln p^x_{l'}.$$

Therefore, defining $F^*(x,l) = \frac{1}{2}\ln p^x_l$ leads to a risk_D minimizing function F^*. Furthermore, for any example and pair of labels l, l', the quantity $L^{l,l'}_{F^*}(x)$ is at most $L^{l,l'}_F(x)$, and therefore the difference in losses of F^* and F may be lower bounded as follows:

$$\begin{aligned} \varepsilon \geq \mathrm{risk}_D(F) - \mathrm{risk}_D(F^*) &= \sum_{x\in\mathcal{X}} p(x) \sum_{l\neq l'} \left(L^{l,l'}_F - L^{l,l'}_{F^*}\right) \\ &\geq \sum_{x\in\mathcal{X}'} p(x) \left\{ L^{H(x),H_{\mathrm{opt}}(x)}_F - L^{H(x),H_{\mathrm{opt}}(x)}_{F^*} \right\}. \end{aligned} \tag{68}$$

We next study the term in the curly brackets for a fixed x. Let A and B denote $H(x)$ and $H_{\mathrm{opt}}(x)$, respectively. We have already seen that $L^{A,B}_{F^*} = 2\sqrt{p^x_A p^x_B}$. Further, by definition of Bayes optimality,

$p_A^x \geq p_B^x$. On the other hand, since $x \in \mathcal{X}'$, we know that $B \neq A$, and hence, $F(x,A) \geq F(x,B)$. Let $e^{F(x,B)-F(x,A)} = 1+\eta$, for some $\eta \geq 0$. The quantity $L_F^{A,B}$ may be lower bounded as:

$$\begin{aligned} L_F^{A,B} &= p_A^x e^{F(x,B)-F(x,A)} + p_B^x e^{F(x,A)-F(x,B)} \\ &= (1+\eta)p_A^x + (1+\eta)^{-1} p_B^x \\ &\geq (1+\eta)p_A^x + (1-\eta)p_B^x \\ &= p_A^x + p_B^x + \eta(p_A^x - p_B^x) \geq p_A^x + p_B^x. \end{aligned}$$

Combining we get

$$L_F^{A,B} - L_{F^*}^{A,B} \geq p_A^x + p_B^x - 2\sqrt{p_A^x p_B^x} = \left(\sqrt{p_A^x} - \sqrt{p_B^x}\right)^2.$$

Plugging back into (68) we get

$$\sum_{x \in \mathcal{X}'} p(x) \left(\sqrt{p_{H(x)}^x} - \sqrt{p_{H_{\text{opt}}(x)}^x}\right)^2 \leq \varepsilon. \tag{69}$$

Now we connect (67) to the previous expression as follows

$$\begin{aligned} &\left\{\text{err}_D(H) - \text{err}_D(H_{\text{opt}})\right\}^2 \\ &= \left\{\sum_{x \in \mathcal{X}'} p(x)\left(p_{H_{\text{opt}}(x)}^x - p_{H(x)}^x\right)\right\}^2 \\ &\leq \left(\sum_{x \in \mathcal{X}'} p(x)\right)\left(\sum_{x \in \mathcal{X}'} p(x)\left(p_{H_{\text{opt}}(x)}^x - p_{H(x)}^x\right)^2\right) \text{ (Cauchy-Schwartz)} \\ &\leq \sum_{x \in \mathcal{X}'} p(x)\left(\sqrt{p_{H_{\text{opt}}(x)}^x} - \sqrt{p_{H(x)}^x}\right)^2 \left(\sqrt{p_{H_{\text{opt}}(x)}^x} + \sqrt{p_{H(x)}^x}\right)^2 \qquad (70) \\ &\leq 2\sum_{x \in \mathcal{X}'} p(x)\left(\sqrt{p_{H_{\text{opt}}(x)}^x} - \sqrt{p_{H(x)}^x}\right)^2 \qquad (71) \\ &\leq 2\varepsilon, \text{ (by (69))} \end{aligned}$$

where (70) holds since

$$\sum_{x \in \mathcal{X}'} p(x) = \Pr_{(x',y') \sim D}\left[x' \in \mathcal{X}'\right] \leq 1,$$

and (71) holds since

$$\begin{aligned} p_{H(x)}^x + p_{H_{\text{opt}}(x)}^x &= \Pr_{(x',y') \sim D}\left[y' \in \left\{H(x), H_{\text{opt}}(x)\right\} | x\right] \leq 1 \\ \implies \sqrt{p_{H(x)}^x} + \sqrt{p_{H_{\text{opt}}(x)}^x} &\leq \sqrt{2}. \end{aligned}$$

Therefore, $\text{err}_D(H) - \text{err}_D(H_{\text{opt}}) \leq \sqrt{2\varepsilon}$. ■

Note that the classifier $\bar{H}_T$, derived from the truncated scoring function $\bar{F}_T$ in the manner provided in (65), makes identical predictions to, and hence has the same err_D as, the classifier H_T output by

the adaptive algorithm. Further, Lemma 27 seems to suggest that $\bar{F}_T$ satisfies the condition in (66), which, combined with our previous observation $\mathrm{err}_D(H) = \mathrm{err}_D(\bar{H}_T)$, would imply H_T approaches the optimal error. However, the condition (66) requires achieving optimal risk over all scoring functions, and not just ones achievable as a combination of weak classifiers in $\mathcal{H}$. Therefore, in order to use Lemma 28, we require the weak classifier space to be sufficiently rich, so that some combination of the weak classifiers in $\mathcal{H}$ attains risk_D arbitrarily close to the minimum attainable by any function:

$$\inf_{\alpha:\mathcal{H}\to\mathbb{R}} \mathrm{risk}_D(F_\alpha) = \inf_{F:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathrm{risk}_D(F). \tag{72}$$

The richness condition, along with our previous arguments and Lemma 27, immediately imply the following result.

Theorem 29 *If the weak classifier space $\mathcal{H}$ satisfies the richness condition* (72)*, and the number of rounds T is set to $\sqrt{m}$, then the error of the final classifier H_T approaches the Bayes optimal error:*

$$\Pr\left[\mathrm{err}_D\left(H_{\sqrt{m}}\right) \le \mathrm{err}_D(H_{\mathrm{opt}}) + O\left(m^{-c}\right)\right] \ge 1 - \frac{1}{m^2},$$

where $c > 0$ is some positive constant, and the probability is over the draw of training examples.

A consequence of the theorem is our strongest consistency result:

Corollary 30 *Let H_{opt} be the Bayes optimal classifier, and let the weak classifier space $\mathcal{H}$ satisfy the richness condition* (72)*. Suppose m example and label pairs $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ are sampled from the distribution D, the number of rounds T is set to be $\sqrt{m}$, and these are supplied to AdaBoost.MM. Then, in the limit $m\to\infty$, the final classifier $H_{\sqrt{m}}$ output by AdaBoost.MM achieves the Bayes optimal error almost surely:*

$$\Pr\left[\left\{\lim_{m\to\infty} \mathrm{err}_D(H_{\sqrt{m}})\right\} = \mathrm{err}_D(H_{\mathrm{opt}})\right] = 1,$$

where the probability is over the randomness due to the draw of training examples.

The proof of Corollary 30, based on the Borel-Cantelli Lemma, is very similar to that of Corollary 12.3 in Schapire and Freund (2012), and so we omit it. When $k = 2$, AdaBoost.MM is identical to AdaBoost. For Theorem 29 to hold for AdaBoost, the richness assumption (72) is necessary, since there are examples due to Long and Servedio (2010) showing that the theorem may not hold when that assumption is violated.

Although we have seen that technically AdaBoost.MM is consistent under broad assumptions, intuitively perhaps it is not clear what properties were responsible for this desirable behavior. We next briefly study the high level ingredients necessary for consistency in boosting algorithms.

9.1 Key Ingredients for Consistency

We show here how both the choice of the loss function as well as the weak learning condition play crucial roles in ensuring consistency. If the loss function were not Bayes consistent as in Lemma 28, driving it down arbitrarily could still lead to high test error. For example, the loss employed by SAMME (Zhu et al., 2009) does not upper bound the error, and therefore although it can manage to

drive down its loss arbitrarily when supplied by the data set discussed in Figure 1, although its error remains high.

Equally important is the weak learning condition. Even if the loss function is chosen to be error, so that it is trivially Bayes consistent, choosing the wrong weak learning condition could lead to inconsistency. In particular, if the weak learning condition is stronger than necessary, then, even on a boostable data set where the error can be driven to zero, the boosting algorithm may get stuck prematurely because its stronger than necessary demands cannot be met by the weak classifier space. We have already seen theoretical examples of such data sets, and we will see some practical instances of this phenomenon in the next section.

On the other hand, if the weak learning condition is too weak, then a lazy Weak Learner may satisfy the Booster's demands by returning weak classifiers belonging only to a non-boostable subset of the available weak classifier space. For instance, consider again the data set in Figure 1, and assume that this time the weak classifier space is much richer, and consists of all possible classifying functions. However, in any round, Weak Learner searches through the space, first trying hypotheses h_1 and h_2 shown in the figure, and only if neither satisfy the Booster, search for additional weak classifiers. In that case, any algorithm using SAMME's weak learning condition, which is known to be too weak and satisfiable by just the two hypotheses $\{h_1, h_2\}$, would only receive h_1 or h_2 in each round, and therefore be unable to reach the optimum accuracy. Of course, if the Weak Learner is extremely generous and helpful, then it may return the right collection of weak classifiers even with a null weak learning condition that places no demands on it. However, in practice, many Weak Learners used are similar to the lazy weak learner described since these are computationally efficient.

To see the effect of inconsistency arising from too weak learning conditions in practice, we need to test boosting algorithms relying on such data sets on significantly hard data sets, where only the strictest Booster strategy can extract the necessary service from Weak Learner for creating an optimal classifier. We did not include such experiments, and it will be an interesting empirical conjecture to be tested in the future. However, we did include experiments that illustrate the consequence of using too strong conditions, and we discuss those in the next section.

10. Experiments

In the final section of this paper, we report preliminary experimental results on 13 UCI data sets: letter, nursery, pendigits, satimage, segmentation, vowel, car, chess, connect4, forest, magic04, poker, abalone. These data sets are all multiclass except for magic04, have a wide range of sizes, contain all combinations of real and categorical features, have different number of examples to number of features per example ratios, and are drawn from a variety of real-life situations. A summary of each data set is provided in Figure 7. Most sets come with prespecified train and test splits which we use; if not, we picked a random $4:1$ split. Sometimes the prespecified test set was too large compared to the training set, and we restricted ourselves to the first ten thousand examples of the specified test set. Throughout this section by `MM` we refer to the version of AdaBoost.MM studied in the consistency section, which uses the approximate step size (61).

There were two kinds of experiments. In the first, we took a standard implementation `M1` of AdaBoost.M1 with C4.5 as weak learner, and the Boostexter implementation `MH` of AdaBoost.MH using stumps (Schapire and Singer, 2000), and compared it against our method `MM` with a naive greedy tree-searching weak-learner `Greedy`. We will refer to the number of leaves as the *size* of

data set	classes	test	train	discrete	real
abalone*	28	1044	3133	1	7
car	4	345	1383	6	0
chess	2	799	2397	36	0
connect4	3	13511	54046	42	0
forest*	7	10000*	15120	44	10
letter	26	4000	16000	0	16
magic04	2	3804	15216	0	10
nursery	5	2591	10369	8	0
pendigits*	10	3498	7494	0	16
poker*	10	10000*	25010	5	5
satimage*	6	2000	4435	0	36
segmentation*	7	2100	210	0	19
vowel*	11	462	528	0	10

Figure 7: Summaries of the data sets used in the experiments. Each row contains the name of data set, number of labels, number of test examples, and training examples, and number of discrete and real features, in that order. The data sets that come with prespecified training-test splits are marked with an asterisk. When the prespecified test set was too large compared to the training set, only the first ten thousand examples were used. These test set sizes are marked with an asterisk.

the tree. Note that since the trees are full binary trees, with each internal node having exactly two children, the total number of nodes in the tree is one less than twice the number of leaves. When C4.5 is run as the weak learner, it grows the tree till a desired accuracy is reached in each round, and thereby automatically picks the tree sizes. Those sizes were used to pick the maximum size of the trees that `Greedy` was allowed to return when run as a weak learner by `MM`, so that both `M1` and `MM` output ensembles of trees of roughly similar sizes. The test-errors after 500 rounds of boosting for each algorithm and data set are shown in Figure 8. The performance is comparable with `M1` and far better than `MH` (understandably since stumps are far weaker than trees), even though our weak-learner is very naive. The convergence rates of error with rounds of `M1` and `MM` are also comparable, as shown in Figure 9 (we omitted the curve for `MH` since it lay far above both `M1` and `MM`).

We next investigated how each algorithm performs with less powerful weak-learners. We modified `MH` so that it uses a tree returning a single multiclass prediction on each example. For `MH` and `MM` we used the `Greedy` weak learner, while for `M1` we used a more powerful-variant `Greedy-Info` whose greedy criterion was information gain rather than error or cost (we also ran `M1` on top of `Greedy` but `Greedy-Info` consistently gave better results so we only report the latter). The reason for using weak learners that optimize different cost functions with the different boosting algorithms is as follows. `M1` is based on the error-metric, where every example incurs a penalty of 0 when classified correctly and 1 when classified incorrectly. Information gain, measuring how "impure" the nodes resulting from a particular split are, is well aligned with the error metric. However, `MH` and `MM` use more general cost functions, and we could not come up with appropriate generalizations of information gain for these setting. So we just used the cost itself in deciding how to grow the de-

data set	MH	M1	MM
abalone	0.732	0.751	0.750
car	0.264	0.336	0.159
chess	0.025	0.003	0.005
connect4	0.321	0.306	0.282
forest	0.326	0.238	0.239
letter	0.146	0.031	0.027
magic04	0.148	0.117	0.118
nursery	0.081	0.164	0.196
pendigits	0.046	0.026	0.095
poker	0.497	0.341	0.228
satimage	0.121	0.088	0.093
segmentation	0.050	0.053	0.149
vowel	0.569	0.485	0.567

Figure 8: This is a table of the final test-errors of standard implementations of MH, M1 and MM after 500 rounds of boosting on different data sets. Both M1 and MM achieve comparable error, which is often smaller than that achieved by MH. This is because M1 and MM used trees of comparable sizes which were often much larger and powerful than the decision stumps that MH boosted.

cision tree. We tried all tree-sizes in the set $\{10, 20, 50, 100, 200, 500, 1000, 2000, 4000\}$ up to the tree-size used by M1 on C4.5 for each data-set. We plotted the error of each algorithm against tree size for each data-set in Figure 10. As predicted by our theory, our algorithm succeeds in boosting the accuracy even when the tree size is too small to meet the stronger weak learning assumptions of the other algorithms. More insight is provided by plots in Figure 11 of the rate of convergence of error with rounds when the tree size allowed is very small (5). Both M1 and MH drive down the error for a few rounds. But since boosting keeps creating harder distributions, very soon the small-tree learning algorithms Greedy and Greedy-Info are no longer able to meet the excessive requirements of M1 and MH respectively. However, our algorithm makes more reasonable demands that are easily met by Greedy.

11. Conclusion and Discussion

In summary, we create a new framework for studying multiclass boosting. This framework is very general and captures the weak learning conditions implicitly used by many earlier multiclass boosting algorithms as well as novel conditions, including the minimal condition under which boosting is possible. We also show how to design boosting algorithms relying on these weak learning conditions that drive down training error rapidly. These algorithms are the optimal strategies for playing certain two player games. Based on this game-theoretic approach, we also design a multiclass boosting algorithm that is consistent, that is, approaches the minimum empirical risk, and under some basic assumptions, the Bayes optimal test error. Preliminary experiments show that this algorithm can achieve much lower error compared to existing algorithms when used with very weak classifiers.

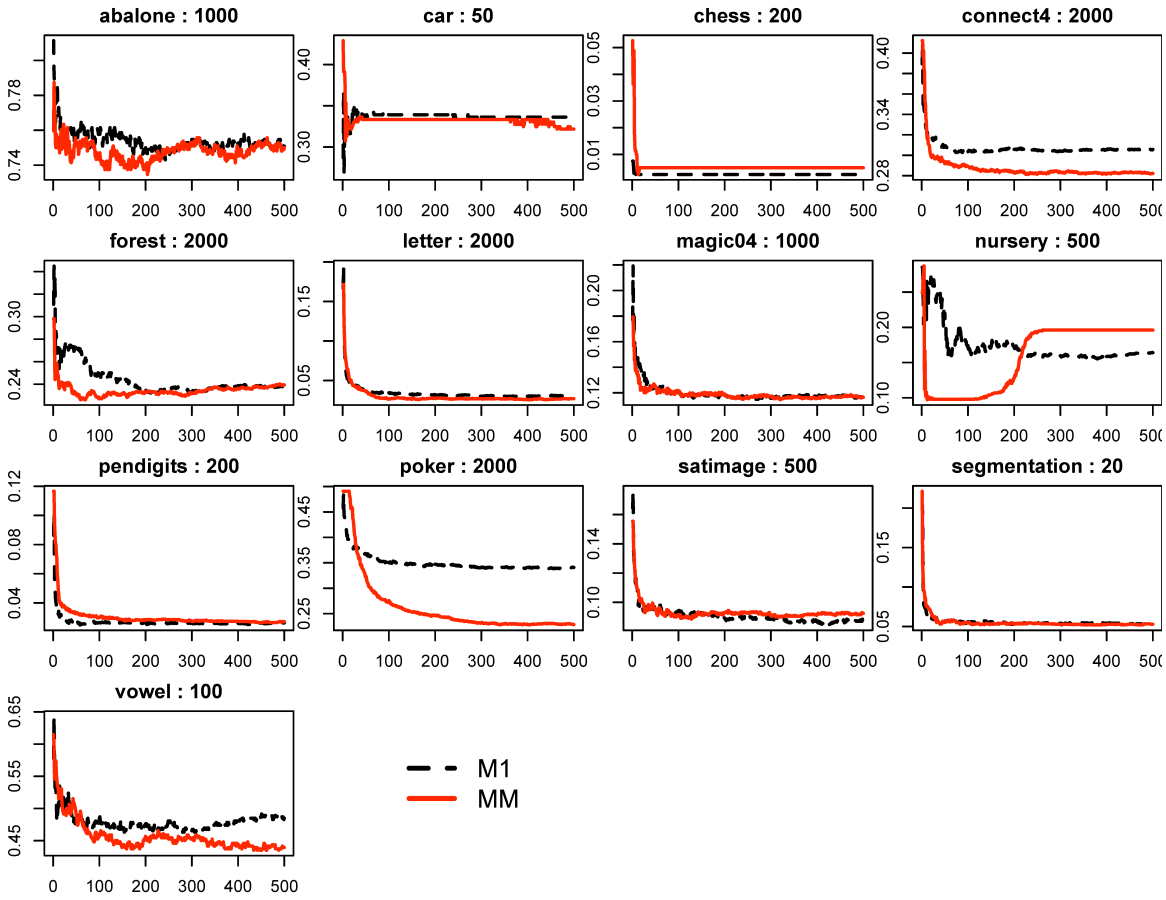


Figure 9: Plots of the rates at which `M1`(black,dashed) and `MM`(red,solid) drive down test-error on different data-sets when using trees of comparable sizes as weak classifiers. `M1` called C4.5, and `MM` called `Greedy`, respectively, as weak-learner. The tree sizes returned by C4.5 were used as a bound on the size of the trees that `Greedy` was allowed to return. This bound on the tree-size depended on the data set, and are shown next to the data set labels.

The notion of game-theoretic equivalence in Section 7.1 is based upon a weak learner that may return any weak hypothesis, which is absurd from a practical viewpoint. However, designing optimal boosting algorithms separately for different kinds of weak learners, which we leave as an open problem, will lead to a much more complex theory. Further, it is not clear what the additional gain (in terms of improvement in loss bounds) may be. Our philosophy here was to take the ultra-conservative approach, so that the resulting boosting algorithm enjoys bounds that hold under all settings. Theorem 14 then says, that in that ultra-conservative framework, the best algorithm remains the same if you change the weak-learning condition to another ”equivalent” condition.

Although we can efficiently compute the game-theoretically optimal strategies under most conditions, when using the minimal weak learning condition, and non-convex 0-1 error as loss function,

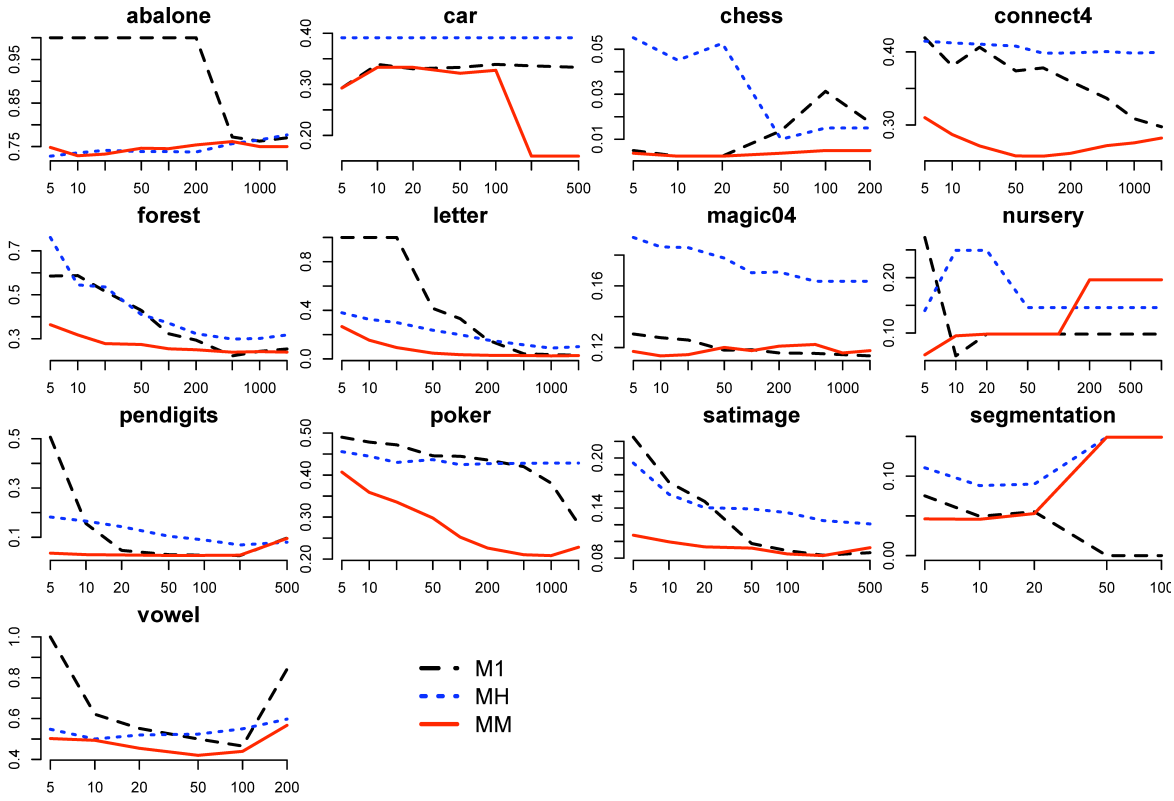


Figure 10: For this figure, `M1`(black, dashed), `MH`(blue, dotted) and `MM`(red,solid) were designed to boost decision trees of restricted sizes. The final test-errors of the three algorithms after 500 rounds of boosting are plotted against the maximum tree-sizes allowed for the weak classifiers. `MM` achieves much lower error when the weak classifiers are very weak, that is, with smaller trees.

we require exponential computational time to solve the corresponding boosting games. Boosting algorithms based on error are potentially far more noise tolerant than those based on convex loss functions, and finding efficiently computable near-optimal strategies in this situation is an important problem left for future work. Further, we primarily work with weak classifiers that output a single multiclass prediction per example, whereas weak hypotheses that make multilabel multiclass predictions are typically more powerful. We believe that multilabel predictions do not increase the power of the weak learner in our framework, and our theory can be extended without much work to include such hypotheses, but we do not address this here. Finally, it will be interesting to see if the notion of minimal weak learning condition can be extended to boosting settings beyond classification, such as ranking.

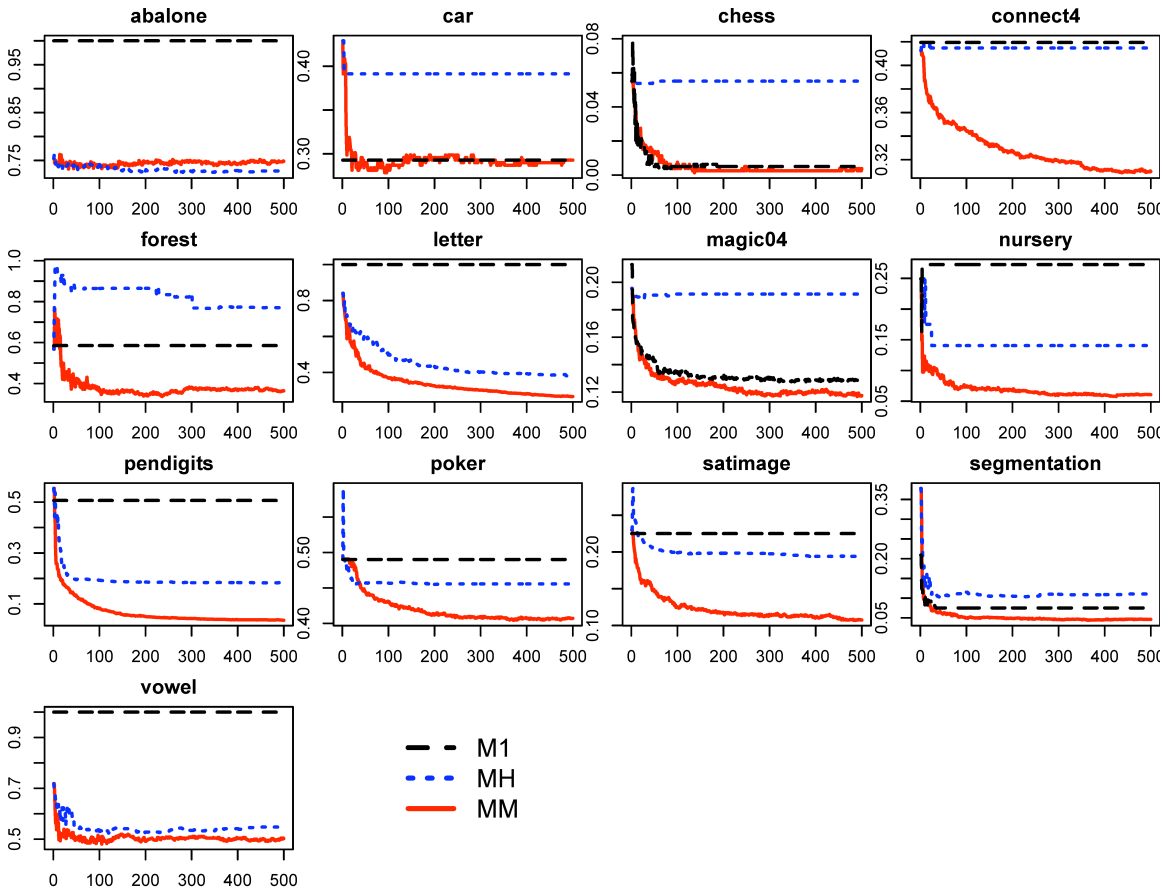


Figure 11: A plot of how fast the test-errors of the three algorithms drop with rounds when the weak classifiers are trees with a size of at most 5. Algorithms `M1` and `MH` make strong demands which cannot be met by the extremely weak classifiers after a few rounds, whereas `MM` makes gentler demands, and is hence able to drive down error through all the rounds of boosting.

Acknowledgments

This research was funded by the National Science Foundation under grants IIS-0325500 and IIS-1016029.

Appendix A. Omitted Proofs

We include proofs and details that were omitted earlier in the paper.

A.1 Optimality of the OS Strategy

Here we prove Theorem 9. The proof of the upper bound on the loss is very similar to the proof of Theorem 2 in Schapire (2001). For the lower bound, a similar result is proved in Theorem 3 in Schapire (2001). However, the proof relies on certain assumptions that may not hold in our setting, and we instead follow the more direct lower bounding techniques in Section 5 of Mukherjee and Schapire (2010).

We first show that the average potential of states does not increase in any round. The dual form of the recurrence (21) and the choice of the cost matrix $\mathbf{C}_t$ in (22) together ensure that for each example i,

$$\begin{aligned}\phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}_t(i)) &= \max_{l=1}^{k}\left\{\phi_{T-t-1}^{\mathbf{B}(i)}(\mathbf{s}_t(i)+\mathbf{e}_l)-(\mathbf{C}_t(i)(l)-\langle\mathbf{C}_t(i),\mathbf{B}(i)\rangle)\right\}\\ &\geq \phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i)+\mathbf{e}_{h_t(x_i)}\right)-(C_t(i,h_t(x_i))-\langle\mathbf{C}_t(i),\mathbf{B}(i)\rangle)\,.\end{aligned}$$

Summing up the inequalities over all examples, we get

$$\sum_{i=1}^{m}\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i)+\mathbf{e}_{h_t(x_i)}\right)\leq\sum_{i=1}^{m}\phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}_t(i))+\sum_{i=1}^{m}\{C_t(i,h_t(x_i))-\langle\mathbf{C}_t(i),\mathbf{B}(i)\rangle\}$$

The first two summations are the total potentials in round $t+1$ and t, respectively, and the third summation is the difference in the costs incurred by the weak-classifier h_t returned in iteration t and the baseline $\mathbf{B}$. By the weak learning condition, this difference is non-positive, implying that the average potential does not increase.

Next we show that the bound is tight. In particular choose any accuracy parameter $\varepsilon>0$, and total number of iterations T, and let m be as large as in (25). We show that in any iteration $t\leq T$, based on Booster's choice of cost-matrix $\mathbf{C}$, an adversary can choose a weak classifier $h_t\in\mathcal{H}^{\text{all}}$ such that the weak learning condition is satisfied, and the average potential does not fall by more than an amount ε/T. In fact, we show how to choose labels $l_1,\ldots,l_m$ such that the following hold simultaneously:

$$\sum_{i=1}^{m}C(i,l_i)\leq\sum_{i=1}^{m}\langle\mathbf{C}(i),\mathbf{B}(i)\rangle \tag{73}$$

$$\sum_{i=1}^{m}\phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}_t(i))\leq\frac{m\varepsilon}{T}+\sum_{i=1}^{m}\phi_{T-t-1}^{\mathbf{B}(i)}(\mathbf{s}_t(i)+\mathbf{e}_{l_i}) \tag{74}$$

This will imply that the final potential or loss is at least ε less than the bound in (23).

We first construct, for each example i, a distribution $\mathbf{p}_i \in \Delta\{1,\ldots,k\}$ such that the size of the support of $\mathbf{p}_i$ is either 1 or 2, and

$$\phi_{T-t}^{\mathbf{B}(i)}(\mathbf{s}_t(i)) = \mathbb{E}_{l\sim\mathbf{p}_i}\left[\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i)+\mathbf{e}_l\right)\right]. \tag{75}$$

To satisfy (75), by (17), we may choose $\mathbf{p}_i$ as any optimal response of the max player in the minimax recurrence when the min player chooses $\mathbf{C}(i)$:

$$\mathbf{p}_i \in \operatorname*{argmax}_{\mathbf{p}\in\mathcal{P}_i}\left\{\mathbb{E}_{l\sim\mathbf{p}}\left[\phi_{t-1}^{\mathbf{B}(i)}\left(\mathbf{s}+\mathbf{e}_l\right)\right]\right\} \tag{76}$$

$$\text{where } \mathcal{P}_i = \left\{\mathbf{p}\in\Delta\{1,\ldots,k\} : \mathbb{E}_{l\sim\mathbf{p}}\left[C(i,l)\right] \le \langle \mathbf{C}(i),\mathbf{B}(i)\rangle\right\}. \tag{77}$$

The existence of $\mathbf{p}_i$ is guaranteed, since, by Lemma 7, the polytope $\mathcal{P}_i$ is non-empty for each i. The next result shows that we may choose $\mathbf{p}_i$ to have a support of size 1 or 2.

Lemma 31 *There is a $\mathbf{p}_i$ satisfying* (76) *with either 1 or 2 non-zero coordinates.*

Proof Let $\mathbf{p}^*$ satisfy (76), and let its support set be S. Let μ_i denote the mean cost under this distribution:

$$\mu_i = \mathbb{E}_{l\sim\mathbf{p}^*}\left[C(i,l)\right] \le \langle \mathbf{C}(i),\mathbf{B}(i)\rangle.$$

If the support has size at most 2, then we are done. Further, if each non-zero coordinate $l\in S$ of $\mathbf{p}^*$ satisfies $C(i,l)=\mu_i$, then the distribution $\mathbf{p}_i$ that concentrates all its weight on the label $l^{\min}\in S$ minimizing $\phi_{t-1}^{\mathbf{B}(i)}\left(\mathbf{s}+\mathbf{e}_{l^{\min}}\right)$ is an optimum solution with support of size 1. Otherwise, we can pick labels $l_1^{\min}, l_2^{\min}\in S$ such that

$$C(i,l_1^{\min}) < \mu_i < C(i,l_2^{\min}).$$

Then we may choose a distribution $\mathbf{q}$ supported on these two labels with mean μ_i:

$$\mathbb{E}_{l\sim\mathbf{q}}\left[C(i,l)\right] = q(l_1^{\min})C(i,l_1^{\min}) + q(l_2^{\min})C(i,l_2^{\min}) = \mu_i.$$

Choose λ as follows:

$$\lambda = \min\left\{\frac{p^*(l_1^{\min})}{q(l_1^{\min})}, \frac{p^*(l_2^{\min})}{q(l_2^{\min})}\right\},$$

and write $\mathbf{p}^* = \lambda\mathbf{q} + (1-\lambda)\mathbf{p}$. Then both $\mathbf{p},\mathbf{q}$ belong to the polytope $\mathcal{P}_i$, and have strictly fewer non-zero coordinates than $\mathbf{p}^*$. Further, by linearity, one of $\mathbf{q},\mathbf{p}$ is also optimal. We repeat the process on the new optimal distribution till we find one which has only 1 or 2 non-zero entries. ■

We next show how to choose the labels $l_1,\ldots,l_m$ using the distributions $\mathbf{p}_i$. For each i, let $\left\{l_i^+, l_i^-\right\}$ be the support of $\mathbf{p}_i$ so that

$$C\left(i,l_i^+\right) \le \mathbb{E}_{l\sim\mathbf{p}_i}\left[C(i,l)\right] \le C\left(i,l_i^-\right).$$

(When $\mathbf{p}_i$ has only one non-zero element, then $l_i^+ = l_i^-$.) For brevity, we use p_i^+ and p_i^- to denote $p_i\left(l_i^+\right)$ and $p_i\left(l_i^-\right)$, respectively. If the costs of both labels are equal, we assume without loss of generality that $\mathbf{p}_i$ is concentrated on label l_i^-:

$$C\left(i,l_i^-\right) - C\left(i,l_i^-\right) = 0 \implies p_i^+ = 0, p_i^- = 1. \tag{78}$$

We will choose each label l_i from the set $\{l_i^-, l_i^+\}$. In fact, we will choose a partition S_+, S_- of the examples $1,\ldots,m$ and choose the label depending on which side S_ξ, for $\xi \in \{-,+\}$, of the partition element i belongs to:

$$l_i = l_i^\xi \text{ if } i \in S_\xi.$$

In order to guide our choice for the partition, we introduce parameters a_i, b_i as follows:

$$\begin{aligned} a_i &= C(i,l_i^-) - C(i,l_i^+), \\ b_i &= \phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_{l_i^-}\right) - \phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_{l_i^+}\right). \end{aligned}$$

Notice that for each example i and each sign-bit $\xi \in \{-1,+1\}$, we have the following relations:

$$C(i,l_i^\xi) = \mathbb{E}_{l\sim\mathbf{p}_i}[C(i,l)] - \xi(1-p_i^\xi)a_i \tag{79}$$

$$\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_{l_i^\xi}\right) = \mathbb{E}_{l\sim\mathbf{p}_i}\left[\phi_{T-t}^{\mathbf{B}(i)}(i,l)\right] - \xi(1-p_i^\xi)b_i. \tag{80}$$

Then the cost incurred by the choice of labels can be expressed in terms of the parameters a_i, b_i as follows:

$$\begin{aligned} \sum_{i\in S_+} C(i,l_i^+) + \sum_{i\in S_-} C(i,l_i^-) &= \sum_{i\in S_+}\left\{\mathbb{E}_{l\sim\mathbf{p}_i}[C(i,l)] - a_i + p_i^+ a_i\right\} \\ &\quad + \sum_{i\in S_-}\left\{\mathbb{E}_{l\sim\mathbf{p}_i}[C(i,l)] + p_i^+ a_i\right\} \\ &= \sum_{i=1}^m \mathbb{E}_{l\sim\mathbf{p}_i}[C(i,l)] + \left(\sum_{i=1}^m p_i^+ a_i - \sum_{i\in S_+} a_i\right) \\ &\le \sum_{i=1}^m \langle \mathbf{C}(i), \mathbf{B}(i)\rangle + \left(\sum_{i=1}^m p_i^+ a_i - \sum_{i\in S_+} a_i\right), \end{aligned} \tag{81}$$

where the first equality follows from (79), and the inequality follows from the constraint on $\mathbf{p}_i$ in (77). Similarly, the potential of the new states is given by

$$\begin{aligned} &\sum_{i\in S_+} \phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_{l_i^+}\right) + \sum_{i\in S_-} \phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_{l_i^-}\right) \\ &= \sum_{i\in S_+}\left\{\mathbb{E}_{l\sim\mathbf{p}_i}\left[\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_l\right)\right] - b_i + p_i^+ b_i\right\} \\ &\quad + \sum_{i\in S_-}\left\{\mathbb{E}_{l\sim\mathbf{p}_i}\left[\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_l\right)\right] + p_i^+ b_i\right\} \\ &= \sum_{i=1}^m \mathbb{E}_{l\sim\mathbf{p}_i}\left[\phi_{T-t-1}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i) + \mathbf{e}_l\right)\right] + \left(\sum_{i=1}^m p_i^+ b_i - \sum_{i\in S_+} b_i\right) \\ &= \sum_{i=1}^m \phi_{T-t}^{\mathbf{B}(i)}\left(\mathbf{s}_t(i)\right) + \left(\sum_{i=1}^m p_i^+ b_i - \sum_{i\in S_+} b_i\right), \end{aligned} \tag{82}$$

where the first equality follows from (80), and the last equality from an optimal choice of $\mathbf{p}_i$ satisfying (75). Now, (81) and (82) imply that in order to satisfy (73) and (74), it suffices to choose a

subset S_+ satisfying

$$\sum_{i \in S_+} a_i \geq \sum_{i=1}^{m} p_i^+ a_i, \qquad \sum_{i \in S_+} b_i \leq \frac{m\varepsilon}{T} + \sum_{i=1}^{m} p_i^+ b_i. \tag{83}$$

We simplify the required conditions. Notice the first constraint tries to ensure that S_+ is big, while the second constraint forces it to be small, provided the b_i are non-negative. However, if $b_i < 0$ for any example i, then adding this example to S_+ only helps both inequalities. In other words, if we can always construct a set S_+ satisfying (83) in the case where the b_i are non-negative, then we may handle the more general situation by just adding the examples i with negative b_i to the set S_+ that would be constructed by considering only the examples $\{i : b_i \geq 0\}$. Therefore we may assume without loss of generality that the b_i are non-negative. Further, assume (by relabeling if necessary) that $a_1, \ldots, a_{m'}$ are positive and $a_{m'+1}, \ldots a_m = 0$, for some $m' \leq m$. By (78), we have $p_i^+ = 0$ for $i > m'$. Therefore, by assigning the examples $m'+1, \ldots, m$ to the opposite partition S_-, we can ensure that (83) holds if the following is true:

$$\sum_{i \in S_+} a_i \quad \geq \quad \sum_{i=1}^{m'} p_i^+ a_i, \tag{84}$$

$$\sum_{i \in S_+} b_i \quad \leq \quad \max_{i=1}^{m'} |b_i| + \sum_{i=1}^{m'} p_i^+ b_i, \tag{85}$$

where, for (85), we additionally used that, by the choice of m (25) and the bound on loss variation (24), we have

$$\frac{m\varepsilon}{T} \geq \varnothing(L,T) \geq b_i \text{ for } i = 1, \ldots, m.$$

The next lemma shows how to construct such a subset S_+, and concludes our lower bound proof.

Lemma 32 *Suppose $a_1, \ldots, a_{m'}$ are positive and $b_1, \ldots, b_{m'}$ are non-negative reals, and $p_1^+, \ldots, p_{m'}^+ \in [0,1]$ are probabilities. Then there exists a subset $S_+ \subseteq \{1, \ldots, m'\}$ such that* (84) *and* (85) *hold.*

Proof Assume, by relabeling if necessary, that the following ordering holds:

$$\frac{a(1) - b(1)}{a(1)} \geq \cdots \geq \frac{a(m') - b(m')}{a(m')}. \tag{86}$$

Let $I \leq m'$ be the largest integer such that

$$a_1 + a_2 + \cdots + a_I < \sum_{i=1}^{m'} p_i^+ a_i. \tag{87}$$

Since the p_i^+ are at most 1, I is in fact at most $m' - 1$. We will choose S_+ to be the first $I+1$ examples $S_+ = \{1, \ldots, I+1\}$. Observe that (84) follows immediately from the definition of I. Further, (85) will hold if the following is true

$$b_1 + b_2 + \cdots + b_I \leq \sum_{i=1}^{m'} p_i^+ b_i, \tag{88}$$

since the addition of one more example $I+1$ can exceed this bound by at most $b_{I+1} \leq \max_{i=1}^{m'} |b_i|$. We prove (88) by showing that the left hand side of this equation is not much more than the left hand side of (87). We first rewrite the latter summation differently. The inequality in (87) implies we can pick $\tilde{p}_1^+, \ldots, \tilde{p}_{m'}^+ \in [0,1]$ (e.g., by simply scaling the p_i^+'s appropriately) such that

$$\begin{aligned} a_1 + \ldots + a_I &= \sum_{i=1}^{m'} \tilde{p}_i^+ a_i & (89) \\ \text{for } i = 1, \ldots, m' \text{: } \tilde{p}_i^+ &\leq p_i. & (90) \end{aligned}$$

By subtracting off the first I terms in the right hand side of (89) from both sides we get

$$(1-\tilde{p}_1^+)a_1 + \cdots + (1-\tilde{p}_I^+)a_I = \tilde{p}_{I+1}^+ a_{I+1} + \cdots + \tilde{p}_{m'}^+ a_{m'}.$$

Since the terms in the summations are non-negative, we may combine the above with the ordering property in (86) to get

$$\begin{aligned} &(1-\tilde{p}_1^+)a_1 \left(\frac{a_1 - b_1}{a_1}\right) + \cdots + (1-\tilde{p}_I^+)a_I \left(\frac{a_I - b_I}{a_I}\right) \\ &\geq \tilde{p}_{I+1}^+ a_{I+1} \left(\frac{a_{I+1} - b_{I+1}}{a_{I+1}}\right) + \cdots + \tilde{p}_{m'}^+ a_{m'} \left(\frac{a_{m'} - b_{m'}}{a_{m'}}\right). \end{aligned} \quad (91)$$

Adding the expression

$$\tilde{p}_1^+ a_1 \left(\frac{a_1 - b_1}{a_1}\right) + \cdots + \tilde{p}_I^+ a_I \left(\frac{a_I - b_I}{a_I}\right)$$

to both sides of (91) yields

$$\begin{aligned} \sum_{i=1}^{I} a_i \left(\frac{a_i - b_i}{a_i}\right) &\geq \sum_{i=1}^{m'} \tilde{p}_i^+ a_i \left(\frac{a_i - b_i}{a_i}\right) \\ \text{i.e., } \sum_{i=1}^{I} a_i - \sum_{i=1}^{I} b_i &\geq \sum_{i=1}^{m'} \tilde{p}_i^+ a_i - \sum_{i=1}^{m'} \tilde{p}_i^+ b_i \\ \text{i.e., } \sum_{i=1}^{I} b_i &\leq \sum_{i=1}^{m'} \tilde{p}_i^+ b_i, & (92) \end{aligned}$$

where the last inequality follows from (89). Now (88) follows from (92) using (90) and the fact that the b_i's are non-negative. ■

This completes the proof of the lower bound.

A.2 Consistency Proofs

Here we sketch the proofs of Lemmas 26 and 27. Our approach will be to relate our algorithm to AdaBoost and then use relevant known results on the consistency of AdaBoost. We first describe the correspondence between the two algorithms, and then state and connect the relevant results on AdaBoost to the ones in this section.

For any given multiclass data set and weak classifier space, we will obtain a transformed binary data set and weak classifier space, such that the run of AdaBoost.MM on the original data set will be

in perfect correspondence with the run of AdaBoost on the transformed data set. In particular, the loss and error on both the training and test set of the combined classifiers produced by our algorithm will be exactly equal to those produced by AdaBoost, while the space of functions and classifiers on the two data sets will be in correspondence.

Intuitively, we transform our multiclass classification problem into a single binary classification problem in a way similar to the all-pairs multiclass to binary reduction. A very similar reduction was carried out by Freund and Schapire (1997). Borrowing their terminology, the transformed data set roughly consists of *mislabel* triples (x,y,l) where y is the true label of the example and l is an incorrect example. The new binary label of a mislabel triple is always -1, signifying that l is not the true label. A multiclass classifier becomes a binary classifier that predict ± 1 on the mislabel triple (x,y,l) depending on whether the prediction on x matches label l; therefore error on the transformed binary data set is low whenever the multiclass accuracy is high. The details of the transformation are provided in Figure 12.

Some of the properties between the functions and their transformed counterparts are described in the next lemma, showing that we are essentially dealing with similar objects.

Lemma 33 *The following are identities for any scoring function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and weight function $\alpha : \mathcal{H} \to \mathbb{R}$:*

$$\widehat{\mathrm{risk}}(F_\alpha) = \widetilde{\widehat{\mathrm{risk}}}\left(\widetilde{F}_{\widetilde{\alpha}}\right) \tag{93}$$

$$\mathrm{risk}_D(\bar{F}) = \widetilde{\mathrm{risk}_D}\left(\bar{\widetilde{F}}\right). \tag{94}$$

The proofs involve doing straightforward algebraic manipulations to verify the identities and are omitted.

The next lemma connects the two algorithms. We show that the scoring function output by AdaBoost when run on the transformed data set is the transformation of the function output by our algorithm. The proof again involves tedious but straightforward checking of details and is omitted.

Lemma 34 *If AdaBoost.MM produces scoring function F_α when run for T rounds with the training set S and weak classifier space $\mathcal{H}$, then AdaBoost produces the scoring function $\widetilde{F}_{\widetilde{\alpha}}$ when run for T rounds with the training set $\widetilde{S}$ and space $\widetilde{\mathcal{H}}$. We assume that for both the algorithms, Weak Learner returns the weak classifier in each round that achieves the maximum edge. Further we consider the version of AdaBoost.MM that chooses weights according to the approximate rule* (61).

We next state the result for AdaBoost corresponding to Lemma 26 , which appears in Mukherjee et al. (2011).

Lemma 35 (Theorem 8 in Mukherjee et al. (2011)) *Suppose AdaBoost produces the scoring function $\widetilde{F}_{\widetilde{\alpha}}$ when run for T rounds with the training set $\widetilde{S}$ and space $\widetilde{\mathcal{H}}$. Then*

$$\widetilde{\widehat{\mathrm{risk}}}\left(\widetilde{F}_{\widetilde{\alpha}}\right) \leq \inf_{\widetilde{\beta}:\widetilde{\mathcal{H}}\to\mathbb{R}} \widetilde{\widehat{\mathrm{risk}}}\left(\widetilde{F}_{\widetilde{\beta}}\right) + C/T,$$

where the constant C depends only on the data set.

The previous lemma, along with (93) immediately proves Lemma 26. The result for AdaBoost corresponding to Lemma 27 appears in Schapire and Freund (2012).

	AdaBoost.MM	AdaBoost
Labels	$\mathcal{Y} = \{1, \ldots, k\}$	$\widetilde{\mathcal{Y}} = \{-1, +1\}$
Examples	$\mathcal{X}$	$\widetilde{\mathcal{X}} = \mathcal{X} \times ((\mathcal{Y} \times \mathcal{Y}) \setminus \{(y,y) : y \in \mathcal{Y}\})$
Weak classifiers	$h : \mathcal{X} \to \mathcal{Y}$	$\widetilde{h} : \widetilde{X} \to \{-1, 0, +1\}$, where $\widetilde{h}(x,y,l) = \mathbf{1}[h(x) = l] - \mathbf{1}[h(x) = y]$
Classifier space	$\mathcal{H}$	$\widetilde{\mathcal{H}} = \left\{\widetilde{h} : h \in \mathcal{H}\right\}$
Scoring function	$F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$	$\widetilde{F} : \widetilde{\mathcal{X}} \to \mathbb{R}$ where $\widetilde{F}(x,y,l) = F(x,l) - F(x,y)$
Clamped function	$\bar{F}(x,y) =$ $\max\{-C, F(x,l) - \max_{l'} F_T(x,l')\}$	$\bar{\widetilde{F}}(x,y,l) = \widetilde{F}(x,y,l)$, if $\lvert\widetilde{F}(x,y,l)\rvert \le C$ $\bar{\widetilde{F}}(x,y,l) = C$, if $\lvert\widetilde{F}(x,y,l)\rvert > C$
Classifier weights	$\alpha : \mathcal{H} \to \mathbb{R}$	$\widetilde{\alpha} : \widetilde{\mathcal{H}} \to \mathbb{R}$ where $\widetilde{\alpha}\left(\widetilde{h}\right) = \alpha(h)$
Combined hypothesis	F_α where $F_\alpha(x,l) = \sum_{h \in \mathcal{H}} \alpha(h) \mathbf{1}[h(x) = l]$	$\widetilde{F}_{\widetilde{\alpha}}$ where $\widetilde{F}_{\widetilde{\alpha}}(x,y,l) = \sum_{\widetilde{h} \in \widetilde{\mathcal{H}}} \widetilde{\alpha}\left(\widetilde{h}\right) \widetilde{h}(x,y,l)$
Training set	$S = \{(x_i, y_i) : 1 \le i \le m\}$	$\widetilde{S} =$ $\{((x_i, y_i, l), \xi) : \xi = -1, l \ne y_i, 1 \le i \le m\}$
Test distribution	D over $\mathcal{X} \times \mathcal{Y}$	$\widetilde{D}$ over $\widetilde{X} \times \widetilde{Y}$ where $\widetilde{D}((x,y,l), -1) = D(x,y)/(k-1)$ $\widetilde{D}((x,y,l), +1) = 0$
Empirical risk	$\widehat{\text{risk}}(F) =$ $\frac{1}{m}\sum_{i=1}^m \sum_{l \ne y_i} e^{F(x_i,l) - F(x_i,y_i)}$	$\widehat{\widetilde{\text{risk}}}\left(\widetilde{F}\right)$ $\frac{1}{m(k-1)}\sum_{i=1}^m \sum_{l \ne y_i} e^{-\xi \widetilde{F}(x_i,y_i,l)}$
Test risk	$\text{risk}_D(F) =$ $\mathbb{E}_{(x,y) \sim D}\left[\sum_{l \ne y} e^{F(x,l) - F(x,y)}\right]$	$\widehat{\widetilde{\text{risk}}_D}\left(\widetilde{F}\right) =$ $\mathbb{E}_{((x,y,l),\xi) \sim \widetilde{D}}\left[e^{-\xi \widetilde{F}(x,y,l)}\right]$

Figure 12: Details of transformation between AdaBoost.MM and AdaBoost.

Lemma 36 (Theorem 12.2 in Schapire and Freund (2012)) *Suppose AdaBoost produces the scoring function $\widetilde{F}$ when run for $T = \sqrt{m}$ rounds with the training set $\widetilde{S}$ and space $\widetilde{\mathcal{H}}$. Then*

$$\Pr\left[\text{risk}_D\left(\bar{\widetilde{F}}\right) \le \inf_{\widetilde{F}' : \widetilde{\mathcal{X}} \to \mathbb{R}} \text{risk}_D(\widetilde{F}') + O\left(m^{-c}\right)\right] \ge 1 - \frac{1}{m^2},$$

where the constant C depends only on the data set.

The proof of Lemma 27 follows immediately from the above lemma and (94).

References

Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal stragies and minimax lower bounds for online convex games. In *COLT*, pages 415–424, 2008.

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

Peter L. Bartlett and Mikhail Traskin. AdaBoost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006.

Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error-correcting tournaments. In *Algorithmic Learning Theory: 20th International Conference*, pages 247–262, 2009.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, January 1995.

Günther Eibl and Karl-Peter Pfeiffer. Multiclass boosting for weak classifiers. *Journal of Machine Learning Research*, 6:189–210, 2005.

Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121 (2):256–285, 1995.

Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3): 293–318, June 2001.

Yoav Freund and Manfred Opper. Continuous drifting games. *Journal of Computer and System Sciences*, pages 113–132, 2002.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996a.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996b.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, 26 (2):451–471, 1998.

Vladimir Koltchinskii and Dmitriy Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1), February 2002.

Ping Li. Robust logitboost and adaptive base class (abc) logitboost. In *UAI*, pages 302–311, 2010.

Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78:287–304, 2010.

Indraneel Mukherjee and Robert E. Schapire. Learning with continuous experts using drifting games. *Theoretical Computer Science*, 411(29-30):2670–2683, June 2010.

Indraneel Mukherjee, Cynthia Rudin, and Robert E. Schapire. The rate of convergence of AdaBoost. In *The 24th Annual Conference on Learning Theory*, 2011.

Gunnar Rätsch and Manfred K. Warmuth. Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, 6:2131–2152, 2005.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Robert E. Schapire. Drifting games. *Machine Learning*, 43(3):265–291, June 2001.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999.

Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, May/June 2000.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, October 1998.

Ambuj Tewari and Peter L. Bartlett. On the Consistency of Multiclass Classification Methods. *Journal of Machine Learning Research*, 8:1007–1025, May 2007.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004.

Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class AdaBoost. *Statistics and Its Interface*, 2:349360, 2009.

Journal of Machine Learning Research 14 (2013) 499-566 Submitted 11/11; Revised 7/12; Published 1/13

Algorithms for Discovery of Multiple Markov Boundaries

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Nikita I. Lytkin NIKITA.LYTKIN@GMAIL.COM
Center for Health Informatics and Bioinformatics, Department of Medicine
New York University School of Medicine
New York, NY 10016, USA

Jan Lemeire[*] JAN.LEMEIRE@VUB.AC.BE
Department of Electronics and Informatics, Faculty of Applied Sciences
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels, Belgium

Constantin F. Aliferis CONSTANTIN.ALIFERIS@NYUMC.ORG
Center for Health Informatics and Bioinformatics, Department of Pathology
New York University School of Medicine
New York, NY 10016, USA

Editor: Peter Spirtes

Abstract

Algorithms for Markov boundary discovery from data constitute an important recent development in machine learning, primarily because they offer a principled solution to the variable/feature selection problem and give insight on local causal structure. Over the last decade many sound algorithms have been proposed to identify a single Markov boundary of the response variable. Even though faithful distributions and, more broadly, distributions that satisfy the intersection property always have a single Markov boundary, other distributions/data sets may have multiple Markov boundaries of the response variable. The latter distributions/data sets are common in practical data-analytic applications, and there are several reasons why it is important to induce multiple Markov boundaries from such data. However, there are currently no sound and efficient algorithms that can accomplish this task. This paper describes a family of algorithms TIE* that can discover all Markov boundaries in a distribution. The broad applicability as well as efficiency of the new algorithmic family is demonstrated in an extensive benchmarking study that involved comparison with 26 state-of-the-art algorithms/variants in 15 data sets from a diversity of application domains.

Keywords: Markov boundary discovery, variable/feature selection, information equivalence, violations of faithfulness

1. Introduction

The problem of variable/feature selection is of fundamental importance in machine learning, especially when it comes to analysis, modeling, and discovery from high-dimensional data sets (Guyon and Elisseeff, 2003; Kohavi and John, 1997). In addition to the promise of cost effectiveness (as a result of reducing the number of observed variables), two major goals of variable selection are to improve the predictive performance of classification/regression models and to provide a better un-

∗. Also at Interdisciplinary Institute for Broadband Technology (IBBT), FMI Dept., Gaston Crommenlaan 8 (box 102), B-9050 Ghent, Belgium.

derstanding of the data-generative process (Guyon and Elisseeff, 2003). An emerging class of filter algorithms proposes solution of the variable selection problem by identification of a Markov boundary of the response variable of interest (Aliferis et al., 2010a, 2003a; Mani and Cooper, 2004; Peña et al., 2007; Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a,b). The Markov boundary $\boldsymbol{M}$ is a minimal set of variables conditioned on which all the remaining variables in the data set, excluding the response variable T, are rendered statistically independent of the response variable T. Under certain assumptions about the learner and the loss function, Markov boundary is the solution of the variable selection problem (Tsamardinos and Aliferis, 2003), that is, it is the minimal set of variables with optimal predictive performance for the current distribution and response variable. Furthermore, in faithful distributions, Markov boundary corresponds to a local causal neighborhood of the response variable and consists of all its direct causes, effects, and causes of the direct effects (Neapolitan, 2004; Tsamardinos and Aliferis, 2003).

An important theoretical result states that if the distribution satisfies the intersection property (which is defined in Section 2.2), then it is guaranteed to have a unique Markov boundary of the response variable (Pearl, 1988). Faithful distributions, which constitute a subclass of distributions that satisfy the intersection property, also have a unique Markov boundary (Neapolitan, 2004; Tsamardinos and Aliferis, 2003). However, some real-life distributions contain multiple Markov boundaries and thus violate the intersection property and faithfulness condition. For example, a phenomenon ubiquitous in analysis of high-throughput molecular data, known as the "multiplicity" of molecular signatures (i.e., different gene/biomarker sets perform equally well in terms of predictive accuracy of phenotypes) suggests existence of multiple Markov boundaries in these distributions (Dougherty and Brun, 2006; Somorjai et al., 2003; Aliferis et al., 2010a). Likewise, many engineering systems such as digital circuits and engines typically contain deterministic components and thus can lead to multiple Markov boundaries (Gopnik and Schulz, 2007; Lemeire, 2007).

Related to the above, a distinguished statistician, the late Professor Leo Breiman, in his seminal work (Breiman, 2001) coined the term "Rashomon effect" that describes the phenomenon of multiple different predictive models that fit the data equally well. Breiman emphasized that "*multiplicity problem and its effect on conclusions drawn from models needs serious attention*" (Breiman, 2001).

There are at least three practical benefits of algorithms that could systematically discover from data multiple Markov boundaries of the response variable of interest:

First, such algorithms would improve discovery of the underlying mechanisms by not missing causative variables. For example, if a causal Bayesian network with the graph $X \leftarrow Y \rightarrow T \rightarrow Z$ is parameterized such that variables X and Y contain equivalent information about T (see section 2.3 and the work by Lemeire, 2007), then there are two Markov boundaries of T: $\{X, Z\}$ and $\{Y, Z\}$. If an algorithm discovers only a single Markov boundary $\{X, Z\}$, then it would miss the directly causative variable Y.

Second, such algorithms can be useful in exploring alternative cost-effective but equally predictive solutions in cases where different variables may have different costs associated with their acquisition. For example, some variables may correspond to cheaper and safer medical tests, while other equally predictive variables may correspond to more expensive and/or potentially unsafe tests. The American College of Radiology maintains Appropriateness Criteria for Diagnostic Imaging (http://www.acr.org/Quality-Safety/Appropriateness-Criteria/) that list diagnostic protocols (sets of radiographic procedures/variables) with the same sensitivity and specificity (i.e., these protocols can be thought of Markov boundaries of the diagnostic response variable) but different cost and

radiation exposure level. Algorithms for induction of multiple Markov boundaries can be helpful for de-novo identification of such protocols from patient data.

Third, such algorithms would shed light on the predictor multiplicity phenomenon and how it affects the reproducibility of predictors. For example, in the domain of high-throughput molecular analytics, induction of multiple Markov boundaries with subsequent validation in independent data would allow testing whether multiple and equally predictive molecular signatures are due to intrinsic information redundancy in the biological networks, small sample statistical indistinguishability of signatures, correlated measurement noise, normalization/data preprocessing steps, or other factors (Aliferis et al., 2010a).

Even though there are several well-developed algorithms for learning a single Markov boundary (Aliferis et al., 2010a, 2003a; Mani and Cooper, 2004; Peña et al., 2007; Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a,b), little research has been done in development of algorithms for identification of multiple Markov boundaries. The most notable advances in the field are stochastic Markov boundary algorithms that involve running multiple times either a standard or approximate Markov boundary induction algorithm initialized with a random seed, for example, KIAMB (Peña et al., 2007), EGS-NCMIGS and EGS-CMIM (Liu et al., 2010b). Another approach exemplified in the EGSG algorithm (Liu et al., 2010) involves first grouping variables into multiple clusters such that each cluster (i) has variables that are similar to each other and (ii) contributes "unique" information about the response variable, and then randomly sampling a representative from each cluster for the output Markov boundaries. In genomics data analysis, researchers try to induce multiple variable sets (that sometimes approximate Markov boundaries) via application of a standard variable selection algorithm to resampled data, for example, bootstrap samples (Ein-Dor et al., 2005; Michiels et al., 2005; Roepman et al., 2006). Finally, other bioinformatics researchers proposed a multiple variable set selection algorithm that iteratively applies a standard variable selection algorithm after removing from the data all variables that participate in the previously discovered variable sets with optimal classification performance (Natsoulis et al., 2005). As we will see in Sections 3 and 5 of this paper, the above early approaches are either highly heuristic and/or cannot be practically used to induce multiple Markov boundaries in high-dimensional data sets with relatively small sample size.

To address the limitations of prior methods, this work presents an algorithmic family TIE* (which is an acronym for "Target Information Equivalence") for multiple Markov boundary induction. TIE* is presented in the form of a generative algorithm and can be instantiated differently for different distributions. TIE* is sound and can be practically applied in typical data-analytic tasks. We have previously introduced in the bioinformatics domain a specific instantiation of TIE* for development of multiple molecular signatures of the phenotype using microarray gene expression data (Statnikov and Aliferis, 2010a). The current paper significantly extends the earlier work for general machine learning use. This includes a detailed description of the generative algorithm, expanded theoretical and complexity analyses, various instantiations of the generative algorithm and its implementation details, and an extensive benchmarking study in 15 data sets from a diversity of application domains.

The remainder of this paper is organized as follows. Section 2 provides general theory and background. Section 3 lists prior algorithms for induction of multiple Markov boundaries and variable sets. Section 4 describes the TIE* generative algorithm, traces its execution, presents specific instantiations, proves algorithm correctness, and analyzes its computational complexity. This section also introduces a simpler and faster algorithm iTIE* for special distributions. Section 5 describes empir-

ical experiments with the TIE* algorithm and comparison with prior methods in simulated and real data. The paper concludes with Section 6 that summarizes main findings, reiterates key principles of TIE* efficiency, demonstrates how the generative algorithm TIE* can be configured for optimal results, presents limitations of this study, and outlines directions for future research. The paper includes several appendices with additional details about our work: Appendix A proves theorems and lemmas; Appendix B presents parameterizations of example structures; Appendix C describes and performs theoretical analysis of prior algorithms for induction of multiple Markov boundaries and variable sets; Appendix D provides details about the TIE* algorithm and its implementations; Appendix E provides additional details about experiments with simulated and real data.

2. Background and Theory

This section provides general theory and background.

2.1 Notation and Key Definitions

In this paper upper-case letters in italics denote random variables (e.g., A, B, C) and lower-case letters in italics denote their values (e.g., a, b, c). Upper-case bold letters in italics denote random variable sets (e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$) and lower-case bold letters in italics denote their values (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$). The terms "variables" and "vertices" are used interchangeably. If a graph contains an edge $X \rightarrow Y$, then X is a *parent* of Y and Y is a *child* of X. A vertex X is a *spouse* of Y if they share a common child vertex. An undirected edge $X - Y$ denotes an *adjacency relation* between X and Y (i.e., presence of an edge directly connecting X and Y). A *path* p is a set of consecutive edges (independent of the direction) without visiting a vertex more than once. A *directed path* p from X to Y is a set of consecutive edges with direction "$\rightarrow$" connecting X with Y, that is, $X \rightarrow \ldots \rightarrow Y$. X is an *ancestor* of Y (and Y is a *descendant* of X) if there exists a directed path p from X to Y. A *directed cycle* is a nonempty directed path that starts and ends on the same vertex X. Three classes of graphs are considered in this work: (i) *directed graphs*: graphs where vertices are connected only with edges "$\rightarrow$"; (ii) *directed acyclic graphs* (DAGs): graphs without directed cycles and where vertices are connected only with edges "$\rightarrow$"; and (iii) *ancestral graphs*: graphs without directed cycles and where vertices are connected with edges "$\rightarrow$" or "$\leftrightarrow$" (an edge $X \leftrightarrow Y$ implies that X is not an ancestor of Y and Y is not an ancestor of X).

When the two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given a set of variables $\boldsymbol{Z}$ in the joint probability distribution $\mathbb{P}$, we denote this as $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z}$. For notational convenience, conditional dependence is defined as absence of conditional independence and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y} \mid \boldsymbol{Z}$. Two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are considered independent and denoted as $\boldsymbol{X} \perp \boldsymbol{Y}$, when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given an empty set of variables. Similarly, the dependence of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y}$.

We further refer the readers to the work by Glymour and Copper (1991), Neapolitan (2004), Pearl (2009) and Spirtes et al. (2000) to review the standard definitions of collider, blocked path, d-separation, m-separation, Bayesian network, causation, direct/indirect causation, and causal Bayesian network that are used in this work. Below we state several essential definitions:

Definition 1 ***Local Markov condition:*** *The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the local Markov condition for a directed acyclic graph (DAG) $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$ if and only if for*

each W *in* $\boldsymbol{V}$, W *is conditionally independent of all variables in* $\boldsymbol{V}$ *excluding descendants of* W *given parents of* W *(Richardson and Spirtes, 1999).*

Definition 2 ***Global Markov condition:*** *The joint probability distribution* $\mathbb{P}$ *over variables* $\boldsymbol{V}$ *satisfies the global Markov condition for a directed graph (ancestral graph)* $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$ *if and only if for any three disjoint subsets of variables* $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ *from* $\boldsymbol{V}$, *if* $\boldsymbol{X}$ *is d-separated (m-separated) from* $\boldsymbol{Y}$ *given* $\boldsymbol{Z}$ *in* $\mathbb{G}$ *then* $\boldsymbol{X}$ *is independent of* $\boldsymbol{Y}$ *given* $\boldsymbol{Z}$ *in* $\mathbb{P}$ *(Richardson and Spirtes, 1999, 2002).*

It follows that if the underlying graph $\mathbb{G}$ is a DAG, then the global Markov condition is equivalent to the local Markov condition (Richardson and Spirtes, 1999).

Finally, we provide several definitions of the faithfulness condition. This condition is fundamental for causal discovery and Markov boundary induction algorithms.

Definition 3 ***DAG-faithfulness:*** *If all and only the conditional independence relations that are true in* $\mathbb{P}$ *defined over variables* $\boldsymbol{V}$ *are entailed by the local Markov condition applied to a DAG* $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$, *then* $\mathbb{P}$ *and* $\mathbb{G}$ *are DAG-faithful to one another (Spirtes et al., 2000).*

The following definition extends DAG-faithfulness to any directed or ancestral graphs:

Definition 4 ***Graph-faithfulness:*** *If all and only the conditional independence relations that are true in* $\mathbb{P}$ *defined over variables* $\boldsymbol{V}$ *are entailed by the global Markov condition applied to a directed or ancestral graph* $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$, *then* $\mathbb{P}$ *and* $\mathbb{G}$ *are graph-faithful to one another.*

A relaxed version of the standard faithfulness assumption is given in the following definition:

Definition 5 ***Adjacency faithfulness:*** *Given a directed or ancestral graph* $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$ *and a joint probability distribution* $\mathbb{P}$ *defined over variables* $\boldsymbol{V}$, $\mathbb{P}$ *and* $\mathbb{G}$ *are adjacency faithful to one another if every adjacency relation between* X *and* Y *in* $\mathbb{G}$ *implies that* X *and* Y *are conditionally dependent given any subset of* $\boldsymbol{V} \setminus \{\mathrm{X}, \mathrm{Y}\}$ *in* $\mathbb{P}$ *(Ramsey et al., 2006).*

The adjacency faithfulness assumption can be relaxed to focus on the specific response variable of interest:

Definition 6 ***Local adjacency faithfulness:*** *Given a directed or ancestral graph* $\mathbb{G} = <\boldsymbol{V}, \mathbb{E}>$ *and a joint probability distribution* $\mathbb{P}$ *defined over variables* $\boldsymbol{V}$, $\mathbb{P}$ *and* $\mathbb{G}$ *are locally adjacency faithful with respect to* T *if every adjacency relation between* T *and* X *in* $\mathbb{G}$ *implies that* T *and* X *are conditionally dependent given any subset of* $\boldsymbol{V} \setminus \{\mathrm{T}, \mathrm{X}\}$ *in* $\mathbb{P}$.

2.2 Basic Properties of Probability Distributions

The following theorem provides a set of useful tools for theoretical analysis of probability distributions and proofs of correctness of Markov boundary algorithms. It is stated similarly to the work by Peña et al. (2007) and its proof is given in the book by Pearl (1988).

Theorem 1 *Let* $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, *and* $\boldsymbol{W}$ *be any four subsets of variables from* $\boldsymbol{V}$.[1] *The following five properties hold in any joint probability distribution* $\mathbb{P}$ *over variables* $\boldsymbol{V}$:

1. Pearl originally provided this theorem for disjoint sets of variables (Pearl, 1988). However, he stated that the disjoint requirement is made for the sake of clarity, and that the theorem can be extended to include overlapping subsets as well.

- Symmetry: $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z} \Leftrightarrow \boldsymbol{Y} \perp \boldsymbol{X} \mid \boldsymbol{Z}$,
- Decomposition: $\boldsymbol{X} \perp (\boldsymbol{Y} \cup \boldsymbol{W}) \mid \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z}$ *and* $\boldsymbol{X} \perp \boldsymbol{W} \mid \boldsymbol{Z}$,
- Weak union: $\boldsymbol{X} \perp (\boldsymbol{Y} \cup \boldsymbol{W}) \mid \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp \boldsymbol{Y} \mid (\boldsymbol{Z} \cup \boldsymbol{W})$,
- Contraction: $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z}$ *and* $\boldsymbol{X} \perp \boldsymbol{W} \mid (\boldsymbol{Z} \cup \boldsymbol{Y}) \Rightarrow \boldsymbol{X} \perp (\boldsymbol{Y} \cup \boldsymbol{W}) \mid \boldsymbol{Z}$,
- Self-conditioning: $\boldsymbol{X} \perp \boldsymbol{Z} \mid \boldsymbol{Z}$.

If $\mathbb{P}$ is strictly positive, then in addition to the above five properties a sixth property holds:

- Intersection: $\boldsymbol{X} \perp \boldsymbol{Y} \mid (\boldsymbol{Z} \cup \boldsymbol{W})$ *and* $\boldsymbol{X} \perp \boldsymbol{W} \mid (\boldsymbol{Z} \cup \boldsymbol{Y}) \Rightarrow \boldsymbol{X} \perp (\boldsymbol{Y} \cup \boldsymbol{W}) \mid \boldsymbol{Z}$.

If $\mathbb{P}$ is faithful to $\mathbb{G}$, then $\mathbb{P}$ satisfies the above six properties and:

- Composition: $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z}$ *and* $\boldsymbol{X} \perp \boldsymbol{W} \mid \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp (\boldsymbol{Y} \cup \boldsymbol{W}) \mid \boldsymbol{Z}$.

The definition given below provides a relaxed version of the composition property that will be used later in the theoretical analysis of Markov boundary induction algorithms.

Definition 7 ***Local composition property:*** *Let $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ be any three subsets of variables from $\boldsymbol{V}$. The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the local composition property with respect to* T *if* $\mathrm{T} \perp \boldsymbol{X} | \boldsymbol{Z}$ *and* $\mathrm{T} \perp \boldsymbol{Y} | \boldsymbol{Z} \Rightarrow \mathrm{T} \perp (\boldsymbol{X} \cup \boldsymbol{Y}) | \boldsymbol{Z}$.

2.3 Information Equivalence

In this subsection we review relevant information equivalence theory (Lemeire, 2007). We first formally define information equivalence that leads to violations of the intersection property and eliminates uniqueness of the Markov boundary (see next subsection). We then describe distributions that have information equivalence relations and point to a theoretical result that characterizes violations of the intersection property.

Definition 8 ***Equivalent information:*** *Two subsets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ from $\boldsymbol{V}$ contain equivalent information about a variable* T *iff the following conditions hold:* $\mathrm{T} \not\perp \boldsymbol{X}$, $\mathrm{T} \not\perp \boldsymbol{Y}$, $\mathrm{T} \perp \boldsymbol{X} \mid \boldsymbol{Y}$ *and* $\mathrm{T} \perp \boldsymbol{Y} \mid \boldsymbol{X}$.

It follows from the definition of equivalent information and the definition of mutual information (Cover and Thomas, 1991) that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ contain the same information about T, that is, mutual information $I(\boldsymbol{X}, T) = I(\boldsymbol{Y}, T)$ (Lemeire, 2007).

Information equivalences can result from deterministic relations. For example, if we consider a Bayesian network with the graph $\begin{matrix} A & \searrow \\ B & \nearrow \end{matrix}$ $X \rightarrow T$ that is parameterized such that X = AND(A, B) and $T \not\perp X$, then $\{X\}$ and $\{A, B\}$ contain equivalent information with respect to T according to the above definition. However, information equivalences follow from a broader class of relations than just deterministic ones (see Example 2 and Figure 1 in the next subsection). We thus define the notion of equivalent partition that was originally introduced in the work by Lemeire (2007). To do so we first provide the definition of T-partition:

Definition 9 ***T-partition:*** *The domain of* X, *denoted by* X_{dom}, *can be partitioned into disjoint subsets* X_{dom}^{k} *for which* $P(\mathrm{T} \mid \mathrm{x})$ *is the same for all* $\mathrm{x} \in \mathrm{X}_{k}^{dom}$. *We call this the* T*-partition of* X_{dom}. *We define* $\kappa_{\mathrm{T}}(\mathrm{X})$ *as the index of the subset of the partition.*

Accordingly, the conditional distribution can be rewritten solely based on the index of T-partition, that is, $\mathrm{P}(T \mid X) = P(T \mid \kappa_T(X))$.

Definition 10 ***Equivalent partition:*** *A relation* $\mathfrak{R} \subset \mathrm{X} \times \mathrm{Y}$ *(where the "*$\times$*" operator denotes the Cartesian product) defines an equivalent partition in* Y_{dom} *to a partition of* X_{dom} *if:*

• *for any* x_1 *and* $\mathrm{x}_2 \in \mathrm{X}_{dom}$ *that do not belong to the same partition and for any* $\mathrm{y}_1 \in \mathrm{Y}_{dom}$ *with* $\mathrm{x}_1\ \mathfrak{R}\ \mathrm{y}_1$*, it must be that* $\neg(\mathrm{x}_2 \mathfrak{R} \mathrm{y}_1)$.

• *for all subsets* X^k_{dom} *of the partition,* $\exists \mathrm{x}_1 \in \mathrm{X}^k_{dom}$ *and* $\exists \mathrm{y}_1 \in \mathrm{Y}_{dom}$ *such that* $\mathrm{x}_1 \mathfrak{R} \mathrm{y}_1$.

In other words, for an equivalent partition, every partition X^k_{dom} corresponds to a partition Y^l_{dom}. If an element of Y_{dom} is related to an element of partition X_{dom}, then it is not related to an element of another partition, and each partition of X_{dom} has at least one element that is related to a partition of Y_{dom}. An example of an equivalent partition is provided in Figure 1 in the next subsection.

In the following theorem the concept of equivalent partition is used to characterize violations of the intersection property; the proof of this theorem is given in the work by Lemeire (2007).

Theorem 2 *If* $\mathrm{T} \not\perp \mathrm{X}$ *and* $\mathrm{T} \perp \mathrm{Y} \mid \mathrm{X}$ *then* $\mathrm{T} \perp \mathrm{X} \mid \mathrm{Y}$ *if and only if the relation* $\mathrm{x}\ \mathfrak{R}\ \mathrm{y}$ *defined by* $P(\mathrm{x}, \mathrm{y}) > 0$ *with* $\mathrm{x} \in \mathrm{X}_{dom}$ *and* $y \in \mathrm{Y}_{dom}$ *defines an equivalent partition in* Y_{dom} *to the* T*-partition of* X_{dom}.

It is worthwhile to mention that the above definitions of T-partition, equivalent partition, and Theorem 2 can be trivially extended to sets of variables instead of individual variables X and Y.

Next we provide two more definitions of equivalent information that take into consideration values of other variables and also lead to violations of the intersection property.

Definition 11 ***Conditional equivalent information:*** *Two subsets of variables* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *from* $\boldsymbol{V}$ *contain equivalent information about a variable* T *conditioned on a non-empty subset of variables* $\boldsymbol{W}$ *iff the following conditions hold* $\mathrm{T} \not\perp \boldsymbol{X} \mid \boldsymbol{W}$, $\mathrm{T} \not\perp \boldsymbol{Y} \mid \boldsymbol{W}$, $\mathrm{T} \perp \boldsymbol{X} \mid (\boldsymbol{Y} \cup \boldsymbol{W})$, *and* $\mathrm{T} \perp \boldsymbol{Y} \mid (\boldsymbol{X} \cup \boldsymbol{W})$.

Definition 12 ***Context-independent equivalent information:*** *Two subsets of variables* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *from* $\boldsymbol{V}$ *contain context-independent equivalent information about a variable* T *iff* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *contain equivalent information about* T *conditioned on any subset of variables* $\boldsymbol{V} \setminus (\boldsymbol{X} \cup \boldsymbol{Y} \cup \{\mathrm{T}\})$.

Finally, we point out that, in general, equivalent information does not always imply context-independent equivalent information. However, equivalent information due to deterministic relations always implies context-independent equivalent information.

2.4 Markov Boundary Theory

In this subsection we first define the concepts of Markov blanket and Markov boundary and theoretically characterize distributions with multiple Markov boundaries of the same response variable. Then we provide examples of such distributions and demonstrate that the number of Markov boundaries can even be exponential in the number of variables in the underlying network. We also state and prove theoretical results that connect the concepts of Markov blanket and Markov boundary with the data-generative graph. Finally, we define optimal predictor and prove a theorem that links the concept of Markov blanket with optimal predictor.

Definition 13 ***Markov blanket:*** *A* Markov blanket $\boldsymbol{M}$ *of the response variable* $\mathrm{T} \in \boldsymbol{V}$ *in the joint probability distribution* $\mathbb{P}$ *over variables* $\boldsymbol{V}$ *is a set of variables conditioned on which all other variables are independent of* T*, that is, for every* $\mathrm{X} \in (\boldsymbol{V} \setminus \boldsymbol{M} \setminus \{\mathrm{T}\})$, $\mathrm{T} \perp \mathrm{X} \mid \boldsymbol{M}$.

Trivially, the set of all variables $\boldsymbol{V}$ excluding T is a Markov blanket of T. Also one can take a small Markov blanket and produce a larger one by adding arbitrary (predictively redundant or irrelevant) variables. Hence, only minimal Markov blankets are of interest.

Definition 14 ***Markov boundary:*** *If no proper subset of $\boldsymbol{M}$ satisfies the definition of Markov blanket of* T*, then $\boldsymbol{M}$ is called a Markov boundary of* T.

The following theorem states a sufficient assumption for the uniqueness of Markov boundaries and its proof is given in the book by Pearl (1988).

Theorem 3 *If a joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the intersection property, then for each $\boldsymbol{X} \in \boldsymbol{V}$, there exists a unique Markov boundary of* X.

Since every joint probability distribution $\mathbb{P}$ that is faithful to $\mathbb{G}$ satisfies the intersection property (Theorem 1), then there is a unique Markov boundary in such distributions according to Theorem 3. However, Theorem 3 does not guarantee that Markov boundaries will be unique in distributions that do not satisfy the intersection property. In fact, as we will see below, Markov boundaries may not be unique in such distributions.

The following two lemmas allow us to explicitly construct and verify multiple Markov blankets and Markov boundaries when the distribution violates the intersection property (proofs are given in Appendix A).

Lemma 1 *If $\boldsymbol{M}$ is a Markov blanket of* T *that contains a set $\boldsymbol{Y}$, and there is a subset of variables $\boldsymbol{Z}$ such that $\boldsymbol{Z}$ and $\boldsymbol{Y}$ contain context-independent equivalent information about* T*, then $\boldsymbol{M}_{new} = (\boldsymbol{M} \setminus \boldsymbol{Y}) \cup \boldsymbol{Z}$ is also a Markov blanket of* T.

Lemma 2 *If $\boldsymbol{M}$ is a Markov blanket of* T *and there exists a subset of variables $\boldsymbol{M}_{new} \subseteq \boldsymbol{V} \setminus \{\mathrm{T}\}$ such that $\mathrm{T} \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$, then $\boldsymbol{M}_{new}$ is also a Markov blanket of* T.

The above lemmas also hold when $\boldsymbol{M}$ is a Markov boundary and immediately imply that $\boldsymbol{M}_{new}$ is a Markov boundary assuming minimality of this subset.

The following three examples provide graphical structures and related probability distributions where multiple Markov boundaries exist. Notably, these examples also demonstrate that multiple Markov boundaries can exist even in large samples. Thus it is not an exclusively small-sample phenomenon, as it was postulated by earlier research (Ein-Dor et al., 2005, 2006).

Example 1 *Consider a joint probability distribution $\mathbb{P}$ described by a Bayesian network with graph* $\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{T}$ *where* A*,* B*, and* T *are binary random variables that take values* $\{0,1\}$*. Given the local Markov condition, the joint probability distribution can be defined as follows:* $P(\mathrm{A}=0) = 0.3$, $P(\mathrm{B}=0 \mid \mathrm{A}=1) = 1.0$, $P(\mathrm{B}=1 \mid \mathrm{A}=0) = 1.0$, $P(\mathrm{T}=0 \mid \mathrm{B}=1) = 0.2$, $P(\mathrm{T}=0 \mid \mathrm{B}=0) = 0.4$. *Two Markov boundaries of* T *exist in this distribution:* $\{\mathrm{A}\}$ *and* $\{\mathrm{B}\}$.

Example 2 *Figure 1 shows a graph of a causal Bayesian network and constraints on its parameterization.*[2] *As can be seen, there is an equivalent partition in the domain of* A *to the* T*-partition*

2. This example has been previously published in the work by Statnikov and Aliferis (2010a) and is presented here with the intent to illustrate the definition of equivalent partition.

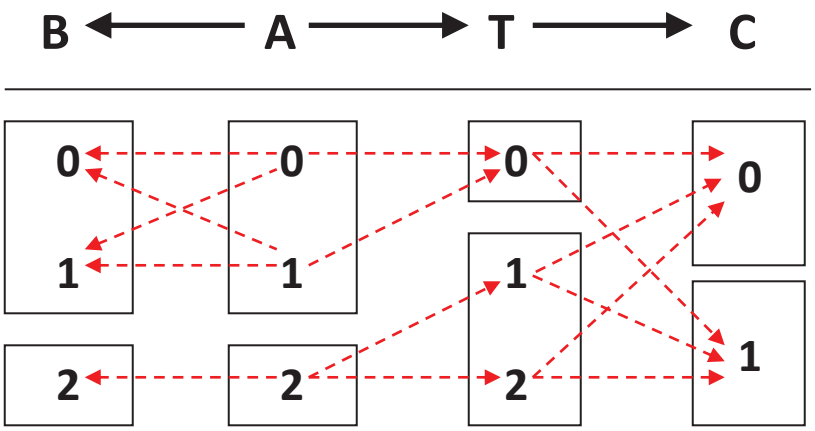


Figure 1: Graph of a causal Bayesian network with four variables (top) and constraints on its parameterization $(bottom)$. Variables A, B, T take three values $\{0,1,2\}$, and variable C takes two values $\{0,1\}$. Red dashed arrows denote non-zero conditional probabilities of each variable given its parents. For example, $P(T=0 \mid A=1) \neq 0$, while $\mathrm{P}(T=0 \mid A=2) = 0$.

of the domain of B. *The following hold in any joint probability distribution of a causal Bayesian network that satisfies the constraints in the figure:*

- A *and* B *are not deterministically related, yet they contain equivalent information about* T*;*
- *There are two Markov boundaries of* T $(\{\mathrm{A},\mathrm{C}\}$ *and* $\{\mathrm{B},\mathrm{C}\})$*;*
- *If an algorithm selects only one Markov boundary of* T *(e.g.,* $\{\mathrm{B},\mathrm{C}\}$*), then there is danger to miss causative variables (i.e., direct cause* A*) and focus instead on confounded ones (i.e.,* B*);*
- *The union of all Markov boundaries of* T *includes all variables that are adjacent with* T $(\{\mathrm{A},\mathrm{C}\})$.

Example 3 *Consider a Bayesian network shown in Figure 2. It involves n + 1 binary variables:* $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$ *and a response variable* T. *Variables* X_i *can be divided into* m *groups such that any two variables in a group contain context-independent equivalent information about* T. *Assume that* n *is divisible by* m. *Since there are* n/m *variables in each group, the total number of Markov boundaries is* $(\mathrm{n}/\mathrm{m})^m$. *Now assume that* k = n/m. *Then the total number of Markov boundaries is* k^m. *Since* $\mathrm{k} > 1$ *and* $\mathrm{m} = O(\mathrm{n})$*, it follows that the number of Markov boundaries grows exponentially in the number of variables in this example.*

Now we provide theoretical results that connect the concepts of Markov blanket and Markov boundary with the underlying causal graph. Theorem 4 was proved in the work by Neapolitan (2004) and Pearl (1988), Theorem 5 was proved in the work by Neapolitan (2004) and Tsamardinos and Aliferis (2003), and the proof of Theorem 6 is given in Appendix A.

Theorem 4 *If a joint probability distribution* $\mathbb{P}$ *satisfies the global Markov condition for directed graph* $\mathbb{G}$*, then the set of children, parents, and spouses of* T *is a Markov blanket of* T.

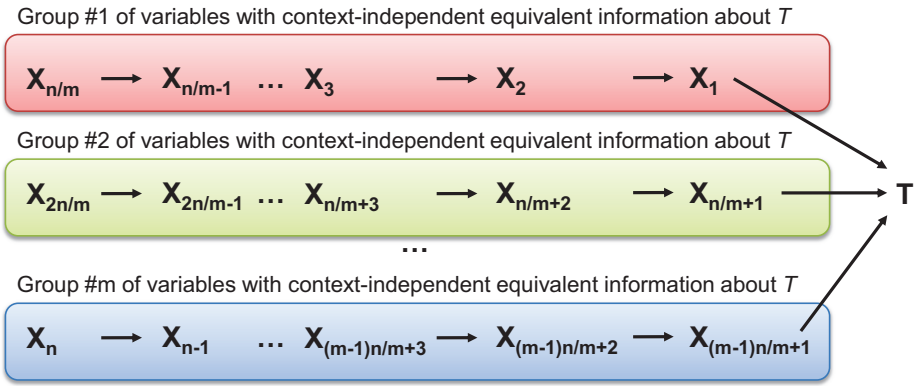


Figure 2: Graph of a Bayesian network used to demonstrate that the number of Markov boundaries can be exponential in the number of variables in the network. The network parameterization is provided in Table 5 in Appendix B. The response variable is T. All variables take values $\{0,1\}$. All variables X_i in each group provide context-independent equivalent information about T.

Theorem 5 *If a joint probability distribution $\mathbb{P}$ is DAG-faithful to $\mathbb{G}$, then the set of children, parents, and spouses of* T *is a unique Markov boundary of* T.

Theorem 6 *If a joint probability distribution $\mathbb{P}$ satisfies the global Markov condition for ancestral graph $\mathbb{G}$, then the set of children, parents, and spouses of* T*, and vertices connected with* T *or children of* T *by a bi-directed path (i.e., only with edges "$\leftrightarrow$") and their respective parents is a Markov blanket of* T.

A graphical illustration of Theorem 6 is provided in Figure 3.

Definition 15 ***Optimal predictor:*** *Given a data set $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, a learning algorithm $\mathbb{L}$, and a performance metric $\mathbb{M}$ to assess learner's models, a variable set $\boldsymbol{X} \subseteq \boldsymbol{V} \setminus \{\mathrm{T}\}$ is an optimal predictor of* T *if $\boldsymbol{X}$ maximizes the performance metric $\mathbb{M}$ for predicting* T *using learner $\mathbb{L}$ in the data set $\mathbb{D}$.*

The following theorem links together the definitions of Markov blanket and optimal predictor, and its proof is given in Appendix A.

Theorem 7 *If $\mathbb{M}$ is a performance metric that is maximized only when $P(\mathrm{T} \mid \boldsymbol{V} \setminus \{\mathrm{T}\})$ is estimated accurately*[3] *and $\mathbb{L}$ is a learning algorithm that can approximate any conditional probability distribution,*[4] *then $\boldsymbol{M}$ is a Markov blanket of* T *if and only if it is an optimal predictor of* T.

3. For example, $\mathbb{M}$ can be negative mean squared error between estimated and true values of $P(T \mid \boldsymbol{V} \setminus \{T\})$ (Tsamardinos and Aliferis, 2003).
4. For example, $\mathbb{L}$ can be feed-forward neural networks or support vector machines that are known to have universal approximation capabilities (Hammer and Gersmann, 2003; Pinkus, 1999; Scarselli and Chung Tsoi, 1998).

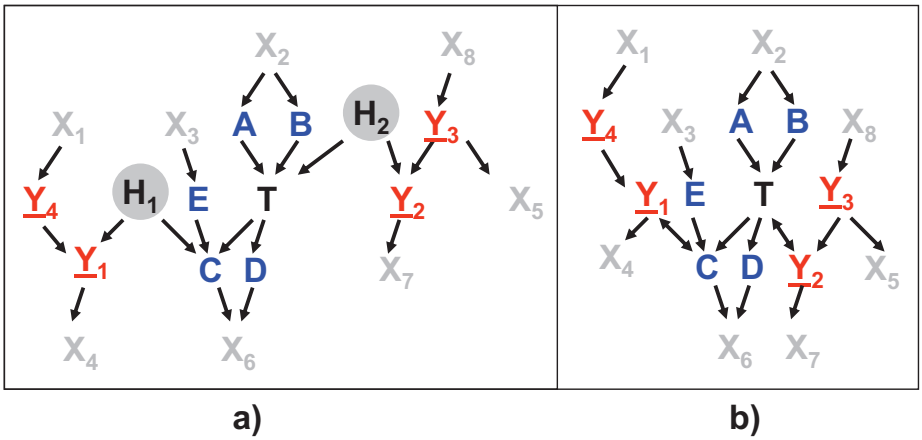


Figure 3: Graphical illustration of a Markov blanket in an ancestral graph. a) Data-generative DAG, variables H_1 and H_2 are latent. b) Corresponding ancestral graph. The set of parents, children, and spouses of T are shown in blue. Vertices connected with T or children of T by a bi-directed path and their respective parents are shown in red and are underlined. If the global Markov condition holds for the graph and joint probability distribution, a Markov blanket of T consists of vertices shown in blue and red. All grey vertices will be then independent of T conditioned on the Markov blanket.

2.5 Prior Algorithms for Learning a Single Markov Boundary

The Markov boundary algorithm IAMB is described in Figure 4 (Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a). Originally, this algorithm was proved to be correct (i.e., that it identifies a Markov boundary) if the joint probability distribution $\mathbb{P}$ is DAG-faithful to $\mathbb{G}$. Then it was proved to be correct when the composition property holds (Peña et al., 2007). The following theorem further relaxes conditions sufficient for correctness of IAMB, requiring that only the local composition property holds; the proof is given in Appendix A.

Theorem 8 *IAMB outputs a Markov boundary of* T *if the joint probability distribution* $\mathbb{P}$ *satisfies the local composition property with respect to* T.

Notice that IAMB identifies a Markov boundary of T by essentially implementing its definition and conditioning on the entire Markov boundary when testing variables for independence from the response T. Conditioning on the entire Markov boundary may become especially problematic in discrete data where the sample size required for high-confidence statistical tests of conditional independence grows exponentially in the size of the conditioning set. This in part motivated the development of the sample-efficient Markov boundary induction algorithmic family Generalized Local Learning, or GLL (Aliferis et al., 2010a). Figure 5 presents the Semi-Interleaved HITON-PC algorithm (Aliferis et al., 2010a), an instantiation of the GLL algorithmic family that we will use in the present paper. Originally, Semi-Interleaved HITON-PC was proved to correctly identify a set of parents and children of T in the Bayesian network $N =< \mathbb{G}, \mathbb{P} >$ if the joint probability distribution $\mathbb{P}$ is DAG-faithful to $\mathbb{G}$ and the so-called symmetry correction is not required (Aliferis et al., 2010a).

Algorithm **IAMB**

Input: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T.
Output: a Markov boundary $\boldsymbol{M}$ of T.

Phase I: Forward

1. Initialize $\boldsymbol{M}$ with an empty set
2. Initialize the set of eligible variables $\boldsymbol{E} \leftarrow \boldsymbol{V} \setminus \{T\}$
3. Repeat
4. $\quad Y \leftarrow \text{argmax}_{X \in \boldsymbol{E}} \text{Association}(T, X \mid \boldsymbol{M})$
5. $\quad \boldsymbol{E} \leftarrow \boldsymbol{E} \setminus \{Y\}$
6. $\quad$ If $T \not\perp\!\!\!\perp Y \mid \boldsymbol{M}$ then
7. $\quad\quad \boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{Y\}$
8. $\quad\quad \boldsymbol{E} \leftarrow \boldsymbol{V} \setminus \boldsymbol{M} \setminus \{T\}$
9. Until $\boldsymbol{E}$ is empty

Phase II: Backward

10. For each $X \in \boldsymbol{M}$
11. $\quad$ If $T \perp X \mid (\boldsymbol{M} \setminus \{X\})$ then
12. $\quad\quad \boldsymbol{M} \leftarrow \boldsymbol{M} \setminus \{X\}$
13. End
14. Output $\boldsymbol{M}$

Figure 4: IAMB algorithm.

The algorithm also retains its correctness for identification of a Markov boundary of T under more relaxed assumptions stated in Theorem 9 (proof is given in Appendix A).

Theorem 9 *Semi-Interleaved HITON-PC outputs a Markov boundary of* T *if there is a Markov boundary of* T *in the joint probability distribution* $\mathbb{P}$ *such that all its members are marginally dependent on* T *and are also conditionally dependent on* T*, except for violations of the intersection property that lead to context-independent information equivalence relations.*

Theorem 9 can be also restated and proved using sufficient assumptions that are motivated by the common assumptions in the causal discovery literature: (i) the joint probability distribution $\mathbb{P}$ and directed or ancestral graph $\mathbb{G}$ are locally adjacency faithful with respect to T with the exception of violations of the intersection property that lead to context-independent information equivalence relations; (ii) $\mathbb{P}$ satisfies the global Markov condition for $\mathbb{G}$; (iii) the set of vertices adjacent with T in $\mathbb{G}$ is a Markov blanket of T.

The proofs of correctness for the Markov boundary algorithms in Theorems 8 and 9 implicitly assume that the statistical decisions about dependence and independence are correct. This requirement is satisfied when the data set $\mathbb{D}$ is a sufficiently large i.i.d. (independent and identically distributed) sample of the underlying probability distribution $\mathbb{P}$. When the sample size is small, the statistical tests of independence will incur type I and II errors. This may affect the correctness of the algorithms output Markov boundary.

In the empirical experiments of this work, we use Semi-Interleaved HITON-PC without "symmetry correction" as a primary method for Markov boundary induction because prior research has

Algorithm **Semi-Interleaved HITON-PC** (without "symmetry correction")

Input: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T.
Output: a Markov boundary $\boldsymbol{M}$ of T.

Phase I: Forward

1. Initialize $\boldsymbol{M}$ with an empty set
2. Initialize the set of eligible variables $\boldsymbol{E} \leftarrow \boldsymbol{V} \setminus \{T\}$
3. Repeat
4. $\quad Y \leftarrow \text{argmax}_{X \in \boldsymbol{E}} \text{Association}(T, X)$
5. $\quad \boldsymbol{E} \leftarrow \boldsymbol{E} \setminus \{Y\}$
6. $\quad$ If there is no subset $\boldsymbol{Z} \subseteq \boldsymbol{M}$ such that $T \perp Y \mid \boldsymbol{Z}$ then
7. $\quad\quad \boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{Y\}$
8. Until $\boldsymbol{E}$ is empty

Phase II: Backward

9. For each $X \in \boldsymbol{M}$
10. $\quad$ If there is a subset $\boldsymbol{Z} \subseteq \boldsymbol{M} \setminus \{X\}$ such that $T \perp X \mid \boldsymbol{Z}$ then
11. $\quad\quad \boldsymbol{M} \leftarrow \boldsymbol{M} \setminus \{X\}$
12. End
13. Output $\boldsymbol{M}$

Figure 5: Semi-Interleaved HITON-PC algorithm (without "symmetry correction"), member of the Generalized Local Learning (GLL) algorithmic family. The algorithm is restated in a fashion similar to IAMB for ease of comparative understanding. Original pseudo-code is given in the work by Aliferis et al. (2010a).

demonstrated empirical superiority of this algorithm compared to the version with the "symmetry correction"; the GLL-MB family of algorithms (including Semi-Interleaved HITON-MB) that can identify Markov boundary members that are non-adjacent spouses of T (and thus may be marginally independent with T); IAMB algorithms (Tsamardinos et al., 2003a); and other comparator Markov boundary induction methods (Aliferis et al., 2010a,b).

3. Prior Algorithms for Learning Multiple Markov Boundaries and Variable Sets

Table 1 summarizes the properties of prior algorithms for learning multiple Markov boundaries and variable sets, while a detailed description of the algorithms and their theoretical analysis is presented in Appendix C. As can be seen, there is no algorithm that is simultaneously theoretically correct, complete, computationally and sample efficient, and does not rely on extensive parameterization. This was our motivation for introducing the TIE* algorithmic family that is described in Section 4.

We would like to note that *not all algorithms listed in Table 1 are designed for identification of Markov boundaries*; methods Resampling+RFE, Resampling+UAF, and IR-SPLR are designed for variable selection. However, sometimes variable sets output by these methods can approximate Markov boundaries, that is why we included these methods in our study (Aliferis et al., 2010a,b).

	Markov boundary identification (assuming faithfulness except for violations of the intersection property)		Parameterization: does not require prior knowledge of		Computationally efficient	sample efficient
	correct (identifies Markov boundaries)	complete (identifies all Markov boundaries)	the number of Markov boundaries/ variable sets	the size of Markov boundaries/ variable sets		
KIAMB	+	+	-	+	-	-
EGS-CMIM	-	-	-	-	-	+
EGS-NCMIGS	-	-	-	+/-	-	+
EGSG	-	-	-	+	-	+
Resampling+RFE	-	-	-	+	-	+
Resampling+UAF	-	-	-	+	-	+
IR-HITON-PC	+	-	+	+	+	+
IR-SPLR	-	-	+	+	+	+

Table 1: Prior algorithms for learning multiple Markov boundaries and variable sets. "$+$" means that the corresponding property is satisfied by a method, "$-$" means that the property is not satisfied, and "$+/-$" denotes cases where the property is satisfied under certain conditions.

4. TIE*: A Family of Multiple Markov Boundary Induction Algorithms

In this section we present a generative anytime algorithm TIE* (which is an acronym for "Target Information Equivalence") for learning from data all Markov boundaries of the response variable. This generative algorithm describes a family of related but not identical algorithms which can be seen as instantiations of the same broad algorithmic principles. We decided to state TIE* as a generative algorithm in order to facilitate a broader understanding of this methodology and devise formal conditions for correctness not only at the algorithm level but also at the level of algorithm family. The latter is achieved by specifying the general set of assumptions (*admissibility rules*) that apply to the generative algorithm and provide a set of flexible tools for constructing numerous algorithmic instantiations, each of which is guaranteed to be correct. This methodology thus significantly facilitates development of new correct algorithms for discovery of multiple Markov boundaries in various distributions.

4.1 Pseudo-Code and Trace

The pseudo-code of the TIE* generative algorithm is provided in Figure 6. On input TIE* receives (i) a data set $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T; (ii) a single Markov boundary induction algorithm $\mathbb{X}$; (iii) a procedure $\mathbb{Y}$ to generate data sets $\mathbb{D}^e$ from the so-called *embedded distributions* that are obtained by removing subsets of variables from the full set of variables $\boldsymbol{V}$ in the *original distribution* $\mathbb{P}$; and (iv) a criterion $\mathbb{Z}$ to verify Markov boundaries of T. The inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ are selected to be suitable for the distribution at hand and should satisfy admissibility rules stated in Figure 7 for correctness of the algorithm (see next two subsections for details). The algorithm outputs all Markov boundaries of T that exist in the distribution $\mathbb{P}$.

Generative algorithm **TIE***

Inputs:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T;
- Markov boundary induction algorithm $\mathbb{X}$;
- procedure $\mathbb{Y}$ to generate datasets from the embedded distributions;
- criterion $\mathbb{Z}$ to verify Markov boundaries of T.

(specific examples of inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ are given in subsection 4.2)

Output: all Markov boundaries of T that exist in $\mathbb{P}$.

1. Use algorithm $\mathbb{X}$ to learn a Markov boundary $\boldsymbol{M}$ of T from the dataset $\mathbb{D}$ for variables $\boldsymbol{V}$ (i.e., in the original distribution $\mathbb{P}$)
2. Output $\boldsymbol{M}$
3. Repeat
4. Use procedure $\mathbb{Y}$ to generate a dataset $\mathbb{D}^e$ from the embedded distribution by removing a subset of variables $\boldsymbol{G}$ from the full set of variables $\boldsymbol{V}$ in the original distribution (also denoted as $\mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$).
5. Use algorithm $\mathbb{X}$ to learn a Markov boundary $\boldsymbol{M}_{new}$ of T from the dataset $\mathbb{D}^e$
6. If $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution according to criterion $\mathbb{Z}$, output $\boldsymbol{M}_{new}$
7. Until all datasets $\mathbb{D}^e$ generated by procedure $\mathbb{Y}$ have been considered.

Figure 6: TIE* generative algorithm.

Admissibility rules for inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ of the TIE* algorithm

I. The Markov boundary induction algorithm $\mathbb{X}$ can correctly identify a Markov boundary of T both in the dataset $\mathbb{D}$ (from the original distribution) and in all datasets $\mathbb{D}^e$ (from the embedded distributions) that are generated by procedure $\mathbb{Y}$.

II. For every Markov boundary of T ($\boldsymbol{M}$) that exists in the original distribution, the procedure $\mathbb{Y}$ generates a dataset $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ such that $\boldsymbol{M}$ can be discovered by the Markov boundary induction algorithm $\mathbb{X}$ applied to the dataset $\mathbb{D}^e$.

III. The criterion $\mathbb{Z}$ can correctly verify that $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution.

Figure 7: Admissibility rules for inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ of the TIE* algorithm.

In step 1, TIE* uses a Markov boundary induction algorithm $\mathbb{X}$ to learn a Markov boundary $\boldsymbol{M}$ of T from the data set $\mathbb{D}$ for variables $\boldsymbol{V}$ (i.e., in the original distribution). Then $\boldsymbol{M}$ is output in step 2. In step 4, the algorithm uses a procedure $\mathbb{Y}$ to generate a data set $\mathbb{D}^e$ that spans over a subset of variables that participate in $\mathbb{D}$. The motivation is that $\mathbb{D}^e$ may lead to identification of a new Markov boundary of T that was previously "invisible" to a single Markov boundary induction algorithm because it was "masked" by another Markov boundary of T. Next, in step 5 the Markov boundary algorithm $\mathbb{X}$ is applied to $\mathbb{D}^e$, resulting in a Markov boundary $\boldsymbol{M}_{new}$ in the embedded distribution. If $\boldsymbol{M}_{new}$ is also a Markov boundary of T in the original distribution according to criterion $\mathbb{Z}$, then $\boldsymbol{M}_{new}$ is output (step 6). The loop in steps 3-7 is repeated until all data sets $\mathbb{D}^e$ generated by procedure $\mathbb{Y}$ have been considered.

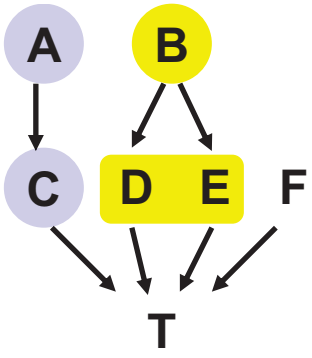


Figure 8: Graph of a causal Bayesian network used to trace the TIE* algorithm. The network parameterization is provided in Table 6 in Appendix B. The response variable is T. All variables take values $\{0,1\}$ except for B that takes values $\{0,1,2,3\}$. Variables A and C contain equivalent information about T and are highlighted with the same color. Likewise, two variables $\{D,E\}$ jointly and a single variable B contain equivalent information about T and thus are also highlighted with the same color.

Next we provide a high-level trace of the algorithm. Consider running an instance of the TIE* algorithm with admissible inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ implemented by an *oracle* in the data set $\mathbb{D}$ generated from the example causal Bayesian network shown in Figure 8.[5] The response variable T is directly caused by C, D, E, and F. The underlying distribution is such that variables A and C contain equivalent information about T; likewise two variables $\{D,E\}$ jointly and a single variable B contain equivalent information about T. In step 1 of TIE* (Figure 6), a Markov boundary induction algorithm $\mathbb{X}$ is applied to learn a Markov boundary of T, resulting in $\boldsymbol{M} = \{A,B,F\}$. Then $\boldsymbol{M}$ is output in step 2. In step 4, a procedure $\mathbb{Y}$ considers removing $\boldsymbol{G} = \{F\}$ and generates a data set $\mathbb{D}^e$ for variables $\boldsymbol{V} \setminus \boldsymbol{G}$. Then in step 5 the Markov boundary induction algorithm $\mathbb{P}$ is run on the data set $\mathbb{D}^e$. This yields a Markov boundary of T in the embedded distribution $\boldsymbol{M}_{new} = \{A,B\}$. The criterion $\mathbb{Z}$ in step 6 does not confirm that $\boldsymbol{M}_{new}$ is also Markov boundary of T in the original distribution; thus $\boldsymbol{M}_{new}$ is not output. The loop is run again. In step 4 the procedure $\mathbb{Y}$ considers removing $\boldsymbol{G} = \{A\}$ and generates a data set $\mathbb{D}^e$ for variables $\boldsymbol{V} \setminus \boldsymbol{G}$. The Markov boundary induction algorithm $\mathbb{X}$ in step 5 yields a Markov boundary of T in the embedded distribution $\boldsymbol{M}_{new} = \{C,B,F\}$. The criterion $\mathbb{Z}$ in step 6 confirms that $\boldsymbol{M}_{new}$ is also a Markov boundary in the original distribution, thus it is returned. Similarly, when the Markov boundary induction algorithm $\mathbb{X}$ is run on the data set $\mathbb{D}^e = \boldsymbol{V} \setminus \boldsymbol{G}$ where $\boldsymbol{G} = \{B\}$ or $\boldsymbol{G} = \{A,B\}$, two additional Markov boundaries of T in the original distribution, $\{A,D,E,F\}$ or $\{C,D,E,F\}$, respectively, are found and output. The algorithm terminates shortly. In total, four Markov boundaries of T are output by the algorithm: $\{A,B,F\}$, $\{C,B,F\}$, $\{A,D,E,F\}$ and $\{C,D,E,F\}$. These are exactly all Markov boundaries of T that exist in the distribution.

5. Specific examples of inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ are given in the next subsection and are omitted here in order to emphasize core algorithmic principles of TIE*.

4.2 Specific Instantiations

In this subsection we give several specific instantiations of the generative algorithm TIE* (Figure 6) and in the next subsection we prove their admissibility (i.e., that they satisfy rules stated in Figure 7). An instantiation of TIE* is specified by assigning its inputs $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ to well-defined algorithms.

Input $\mathbb{X}$: This is a Markov boundary induction algorithm. For example, we can use IAMB (Figure 4) or Semi-Interleaved HITON-PC (Figure 5) algorithms that were described in Section 2.5. Other sound Markov boundary induction algorithms can be used as well (Aliferis et al., 2010a, 2003a; Mani and Cooper, 2004; Peña et al., 2007; Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a,b).

Input $\mathbb{Y}$: This is a procedure to generate data sets from the embedded distributions that would allow identification of new Markov boundaries of T. Before we give specific examples of this procedure, it is worthwhile to understand its use in TIE*. The main principle of TIE* is to first identify a Markov boundary of T in the original distribution and then iteratively run a Markov boundary induction algorithm in data sets from the embedded distributions (that are obtained by removing subsets of variables from $\boldsymbol{M}$) in order to identify new Markov boundaries *in the original distribution*. Generating such data sets from the embedded distributions is the purpose of procedure $\mathbb{Y}$. The reason why we need to remove subsets of variables from the original data and rerun Markov boundary induction algorithm in the data set $\mathbb{D}^e = \mathbb{P}\,(\,\boldsymbol{V} \setminus \boldsymbol{G}\,)$ is because some variables "mask" Markov boundaries during operation of conventional single Markov boundary induction algorithms by rendering some of the Markov boundary members conditionally independent of T. One possible approach is to generate data sets by removing subsets of the original Markov boundary, or, more broadly, subsets from all currently identified Markov boundaries. The procedure termed IGS (which is an acronym for "Incremental Generation of Subsets") implements the above stated approach and is described in Figure 9.[6]

Below and in Table 2 we revisit the trace of TIE* that was given in the previous subsection, now focusing on the operation of the procedure IGS ($\mathbb{Y}$) from Figure 9. Recall that application of the Markov boundary induction algorithm in step 1 of TIE* resulted in $\boldsymbol{M} = \{A, B, F\}$. In step 4 of TIE*, the procedure IGS can generate data sets $\mathbb{D}^e = \mathbb{D}\,(\,\boldsymbol{V} \setminus \boldsymbol{G}\,)$ from the embedded distributions by removing any of the three possible subsets $\boldsymbol{G} = \{A\}$ or $\{B\}$ or $\{F\}$ from $\boldsymbol{V}$ (it will not consider larger subsets because of the requirement of the smallest subset size in step 1 of IGS, see Figure 9). Recall that next we considered a data set $\mathbb{D}^e$ obtained by removing $\boldsymbol{G} = \{F\}$ and identified via algorithm $\mathbb{X}$ a Markov boundary in the embedded distribution $\boldsymbol{M}_{new} = \{A, B\}$ that did not turn out to be a Markov boundary in the original distribution. When the procedure IGS is executed in the following iterations of steps 3-7, it will never generate data set $\mathbb{D}^e$ without $\{F\}$ because $\boldsymbol{G}^*_1 = \{F\}$ and we require that $\boldsymbol{G}$ does not include $\boldsymbol{G}^*_j$ for j = 1,...,m. In the next iteration, IGS can generate two possible data sets $\mathbb{D}^e$ by removing $\boldsymbol{G} = \{A\}$ or $\{B\}$ from $\boldsymbol{V}$. In order to be consistent with our previous trace, assume that the procedure IGS output a data set $\mathbb{D}^e$ obtained by removing $\boldsymbol{G} = \{A\}$ which led to identification of a new Markov boundary both in the original and embedded distribution

6. To retain simplicity of the TIE* pseudo-code (Figure 6), we implicitly assume that $\boldsymbol{M}_i, \boldsymbol{G}_i, \boldsymbol{G}^*_j$ are stored during operation of the generative algorithm TIE*. This can be implemented by setting a counter of all identified Markov boundaries in the original distribution (i) and a counter of all identified Markov boundaries in the embedded distribution that are not Markov boundaries the original distribution (j). Then the following assignments should be made: $\boldsymbol{M}_1 \leftarrow \boldsymbol{M}$ and $\boldsymbol{G}^*_1 \leftarrow \oslash$ after step 1 of TIE*; $\boldsymbol{M}_i \leftarrow \boldsymbol{M}_{new}$ and $\boldsymbol{G}^*_i \leftarrow \boldsymbol{G}^*$ in step 6 of TIE* if $\boldsymbol{M}_{new}$ is a Markov boundary in the original distribution; and $\boldsymbol{G}^*_j \leftarrow \boldsymbol{G}$ in step 6 of TIE* if $\boldsymbol{M}_{new}$ is not a Markov boundary in the original distribution.

Procedure *IGS* to generate datasets from the embedded distributions

// This is an instantiation of the procedure $\mathbb{Y}$ in the generative algorithm TIE*

Inputs[6]:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T;
- Markov boundaries $\boldsymbol{M}_1, ..., \boldsymbol{M}_n$ of T (in the original distribution) obtained so far by TIE* and ordered by the time of discovery from earliest ($\boldsymbol{M}_1$) to latest ($\boldsymbol{M}_n$);
- subsets $\boldsymbol{G}_1, ..., \boldsymbol{G}_n$ that were used in previous calls to IGS to generate datasets from the embedded distributions that led to discovery of the above Markov boundaries (we use $\boldsymbol{G}_1=\varnothing$);
- subsets $\boldsymbol{G}_1^*, ..., \boldsymbol{G}_m^*$ that were used in previous calls to IGS to generate datasets from the embedded distributions that did not lead to Markov boundaries in the original distribution.

1. Generate the smallest subset of variables $\boldsymbol{G}$: $\boldsymbol{G}_i \subset \boldsymbol{G} \subseteq (\boldsymbol{M}_i \cup \boldsymbol{G}_i)$ for some $i = 1, .., n$ that neither includes $\boldsymbol{G}_j^*$ nor coincides with $\boldsymbol{G}_k$ for any $j = 1, \ldots, m$ and $k = 1, \ldots, n$
2. $\mathbb{D}^e \leftarrow \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ // This is a dataset from the embedded distribution

Figure 9: Procedure IGS ($\mathbb{Y}$) to generate data sets from the embedded distributions Note that IGS is a procedure (not a function), and we assume that $\mathbb{D}^e$ and $\boldsymbol{G}$ are accessible in the scope of TIE*.

$\boldsymbol{M}_{new}=\{C, B, F\}$. When the procedure IGS is executed in the next iteration, it will generate a data set $\mathbb{D}^e$ by removing a subset $\boldsymbol{G} = \{B\}$ from $\boldsymbol{V}$ (all other subsets will have two or more variables and thus will not be considered). This would lead to identification of a new Markov boundary both in the original and embedded distribution $\boldsymbol{M}_{new} = \{A, D, E, F\}$. When the procedure IGS is executed in the next iteration, it can generate data sets $\mathbb{D}^e$ by removing $\boldsymbol{G} = \{A, B\}$ or $\{A, C\}$ or $\{B, D\}$ or $\{B, E\}$ from $\boldsymbol{V}$. Assume that the procedure generated a data set $\mathbb{D}^e$ by removing $\boldsymbol{G} = \{A, B\}$, which would lead to identification of a new Markov boundary both in the original and embedded distribution $\boldsymbol{M}_{new} = \{C, D, E, F\}$. Several more iterations will follow, but no new Markov boundaries in the original distribution will be identified (see Table 2 for one more iteration), and TIE* will terminate.

As it follows from the above example, we may have several possibilities for the subset $\boldsymbol{G}$ (and thus for defining a data set $\mathbb{D}^e$) in the procedure IGS and we need to define rules in order to select a single subset. We therefore provide three specific implementations of the procedure IGS:

• *IGS-Lex* ("Lex" stands for "lexicographical"): Procedure IGS from Figure 9 where one chooses a subset $\boldsymbol{G}$ with the smallest lexicographical order of its variables;

• *IGS-MinAssoc* ("MinAssoc" stands for "minimal association"): Procedure IGS from Figure 9 where one chooses a subset $\boldsymbol{G}$ with the smallest association with the response variable T;

• *IGS-MaxAssoc* ("MaxAssoc" stands for "maximal association"): Procedure IGS from Figure 9 where one chooses a subset $\boldsymbol{G}$ with the largest association with the response variable T.

The above three instantiations of the procedure IGS may lead to different traces of the TIE* algorithm, however the final output of the algorithm will be the same (it will discover all Markov boundaries of T).

Loop iteration (steps 3-7)	Procedure IGS (step 4)			Identified Markov boundary (MB) (step 5)	MB in *original distribution* (step 6)?
	Inputs	Possible subsets $\boldsymbol{G}$	Output $\mathbb{D}^e$		
#1	· $\boldsymbol{M}_1=\{A, B, F\}$ · $\boldsymbol{G}_1=\emptyset$	$\{A\}$, $\{B\}$, $\{F\}$	$\mathbb{D}(\boldsymbol{V} \setminus \{F\})$	$\{A, B\}$	NO
#2	· $\boldsymbol{M}_1$ · $\boldsymbol{G}_1$ · $\boldsymbol{G}_1^* = \{F\}$	$\{A\}$, $\{B\}$	$\mathbb{D}(\boldsymbol{V} \setminus \{A\})$	$\{C, B, F\}$	YES
#3	· $\boldsymbol{M}_1, \boldsymbol{M}_2=\{C, B, F\}$ · $\boldsymbol{G}_1, \boldsymbol{G}_2=\{A\}$ · $\boldsymbol{G}_1^*$	$\{B\}$	$\mathbb{D}(\boldsymbol{V} \setminus \{B\})$	$\{A, D, E, F\}$	YES
#4	· $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3=\{A, D, E, F\}$ · $\boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{G}_3=\{B\}$ · $\boldsymbol{G}_1^*$	$\{A, B\}$, $\{A, C\}$, $\{B, D\}$, $\{B, E\}$	$\mathbb{D}(\boldsymbol{V} \setminus \{A, B\})$	$\{C, D, E, F\}$	YES
#5	· $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3,$ $\boldsymbol{M}_4=\{C, D, E, F\}$ · $\boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{G}_3, \boldsymbol{G}_4=\{A, B\}$ · $\boldsymbol{G}_1^*$	$\{A, C\}$, $\{B, D\}$, $\{B, E\}$	$\mathbb{D}(\boldsymbol{V} \setminus \{A, C\})$	$\{B, F\}$	NO

Table 2: Part of the trace of TIE*, focusing on operation of the procedure $\mathbb{Y}$.

Input $\mathbb{Z}$: This is a criterion that can verify whether $\boldsymbol{M}_{new}$, a Markov boundary in the embedded distribution (that was found by application of the Markov boundary induction algorithm $\mathbb{X}$ in step 5 of TIE* to the data set $\mathbb{D}^e$) is also a Markov boundary in the original distribution. In other words, it is a criterion to verify the Markov boundary property of $\boldsymbol{M}_{new}$ in the original definition. For example, we can use the following two criteria given in Figures 10 and 11. Criterion Independence from Figure 10 is closely related to the definition of the Markov boundary, and essentially implies its verification. Criterion Predictivity from Figure 11 verifies Markov boundaries by assessing their predictive (classification or regression) performance using some learning algorithm and performance metric.

Appendix D provides two concrete admissible instantiations of the generative algorithm TIE* (admissibility follows from theoretical results presented in the next subsection). The instantiation in Figure 17 is obtained using $\mathbb{X}$ = Semi-Interleaved HITON-PC, $\mathbb{Y}$ = IGS, $\mathbb{Z}$ = Predictivity. The instantiation in Figure 18 is obtained using $\mathbb{X}$ = Semi-Interleaved HITON-PC, $\mathbb{Y}$ = IGS, $\mathbb{Z}$ = Independence. Appendix D also gives practical considerations for computer implementations of TIE*.

4.3 Analysis of the Algorithm Correctness

In this subsection we state theorems about correctness of TIE* and its specific instantiations that were described in the previous subsection and Appendix D. The proofs of all theorems are given in Appendix A.

First we show that the generative algorithm TIE* is sound and complete:

Criterion *Independence* to verify Markov boundaries

// This is an instantiation of the criterion $\mathbb{Z}$ in the generative algorithm TIE*

Inputs:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T;
- Markov boundary $\boldsymbol{M}$ of T in the original distribution;
- Markov boundary $\boldsymbol{M}_{new}$ of T in the embedded distribution.

Output:

- TRUE if $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution;
- FALSE if $\boldsymbol{M}_{new}$ is a not a Markov blanket of T in the original distribution.

If $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$, output TRUE; otherwise output FALSE.

Figure 10: Criterion Independence ($\mathbb{Z}$) to verify Markov boundaries.

Criterion *Predictivity* to verify Markov boundaries

// This is an instantiation of the criterion $\mathbb{Z}$ in the generative algorithm TIE*

Inputs:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T;
- Markov boundary $\boldsymbol{M}$ of T in the original distribution;
- Markov boundary $\boldsymbol{M}_{new}$ of T in the embedded distribution;
- learning algorithm $\mathbb{L}$ (to build a prediction model for T given data $\mathbb{D}$ for some subset of variables from $\boldsymbol{V}$);
- performance metric $\mathbb{M}$ (to assess the prediction model obtained by $\mathbb{L}$; larger values of this performance metric correspond to higher predictivity of the model).

Output:

- TRUE if $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution;
- FALSE if $\boldsymbol{M}_{new}$ is a not a Markov blanket of T in the original distribution.

1. $\hat{m}_1 \leftarrow$ performance estimate using metric $\mathbb{M}$ for prediction model obtained by $\mathbb{L}$ in data $\mathbb{D}$ for variables $\boldsymbol{M}$
2. $\hat{m}_2 \leftarrow$ performance estimate using metric $\mathbb{M}$ for prediction model obtained by $\mathbb{L}$ in data $\mathbb{D}$ for variables $\boldsymbol{M}_{new}$
3. If $\hat{m}_1 \leq \hat{m}_2$ (taking into account statistical uncertainty), output TRUE; otherwise output FALSE.

Figure 11: Criterion Predictivity ($\mathbb{Z}$) to verify Markov boundaries.

Theorem 10 *The generative algorithm TIE* outputs all and only Markov boundaries of* $\mathbf{T}$ *that exist in the joint probability distribution* $\mathbb{P}$ *if the inputs* $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ *are admissible (i.e., satisfy admissibility rules in Figure 7).*

Now we show that IAMB (Figure 4) and Semi-Interleaved HITON-PC (Figure 5) are admissible Markov boundary algorithms for TIE* under sufficient assumptions. In the case of the IAMB algorithm, the sufficient assumptions for TIE* admissibility are the same as sufficient assumptions for the general algorithm correctness (see Theorem 8). This leads to the following theorem.

Theorem 11 *IAMB is an admissible Markov boundary induction algorithm for TIE* (input* $\mathbb{X}$*) if the joint probability distribution* $\mathbb{P}$ *satisfies the local composition property with respect to* T.

However, the sufficient assumptions for the general correctness of Semi-Interleaved HITON-PC (Theorem 9) are not sufficient for TIE* admissibility and require further restriction. Specifically, we need to require that all members of *all* Markov boundaries retain marginal and conditional dependence on *T*, except for certain violations of the intersection property. This leads to the following theorem.

Theorem 12 *Semi-Interleaved HITON-PC is an admissible Markov boundary induction algorithm for TIE* (input* $\mathbb{P}$*) if all members of all Markov boundaries of* T *that exist in the joint probability distribution* $\mathbb{P}$ *are marginally dependent on* T *and are also conditionally dependent on* T*, except for violations of the intersection property that lead to context-independent information equivalence relations.*

The next theorem states that the procedure IGS (Figure 9) is admissible for TIE*:

Theorem 13 *Procedure IGS to generate data sets from the embedded distributions (input* $\mathbb{Y}$*) is admissible for TIE*.*

Finally we show that both criteria Independence (Figure 10) and Predictivity (Figure 11) for verification of Markov boundaries are admissible for TIE* and state sufficient assumptions for the latter criterion. The former criterion implicitly assumes correctness of statistical decisions, similarly to IAMB and Semi-Interleaved HITON-PC (see end of Section 2.5 for related discussion).

Theorem 14 *Criterion Independence to verify Markov boundaries (input* $\mathbb{Z}$*) is admissible for TIE**

Theorem 15 *Criterion Predictivity to verify Markov boundaries (input* $\mathbb{Z}$*) is admissible for TIE* if the following conditions hold: (i) the learning algorithm* $\mathbb{L}$ *can accurately approximate any conditional probability distribution, and (ii) the performance metric* $\mathbb{M}$ *is maximized only when P(* T$|$ $\boldsymbol{V}$ $\setminus$ {T}*) is estimated accurately.*

As mentioned in the beginning of Section 4, the generative nature of TIE* facilitates design of new algorithms for discovery of multiple Markov boundaries by simply instantiating TIE* with input components $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$. Furthermore, if $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$ are admissible, then TIE* will be sound and complete according to Theorem 10, otherwise the algorithm will be heuristic. For example, one can take an established Markov boundary induction algorithm, prove its admissibility, and then plug it into TIE* with admissible components $\mathbb{Y}$ and $\mathbb{Z}$ (e.g., ones presented above). This will yield a new correct algorithm and significant economies in the proof of its correctness because one has only to prove admissibility of new input components.

4.4 Complexity Analysis

We first note that the computational complexity of TIE* depends on a specific instantiation of its input components $\mathbb{X}$ (Markov boundary induction algorithm), $\mathbb{Y}$ (procedure for generating data sets from the embedded distributions) and $\mathbb{Z}$ (criterion for verifying Markov boundaries), and on the underlying joint probability distribution over a set variables $\boldsymbol{V}$. In this subsection we will consider the complexity of the following two specific instantiations of TIE*: ($\mathbb{X}$= Semi-Interleaved HITON-PC, $\mathbb{Y}$=IGS-Lex, $\mathbb{Z}$=Independence) and ($\mathbb{X}$= IAMB, $\mathbb{Y}$=IGS-Lex, $\mathbb{Z}$=Independence).

Since in our experiments we found that Markov boundary induction (with input component $\mathbb{X}$) was the most computationally expensive step in TIE* and accounted for $> 99\%$ of algorithm runtime, we will omit from consideration the complexity of components $\mathbb{Y}$ and $\mathbb{Z}$, and will use the complexity of component $\mathbb{X}$ to derive an estimate of the total computational complexity of TIE*. Following general practice in complexity analysis of Markov boundary and causal discovery algorithms, we measure computational complexity in terms of the number of statistical tests of conditional independence.[7] For completeness we also note that there exist efficient implementations of the G^2 test for discrete variables that can take only time $nlog(n)$ in the number of training instances n. The time for computation of Fishers Z-test for continuous variables is also bounded by a low order polynomial in n because this test essentially involves solution of a linear system. See the work by Aliferis et al. (2010a) and Anderson (2003) for more details and discussion.

As with all sound and complete computational causal discovery algorithms, discovery of all Markov boundaries (and even one Markov boundary) is worst-case intractable. However we are interested in the average-case complexity of TIE* in *real-life distributions* that is more instructive to consider. Complexities of Markov boundary induction algorithms IAMB and Semi-Interleaved HITON-PC are $O(|\boldsymbol{V}||\boldsymbol{M}|)$ and $O(|\boldsymbol{V}|2^{|\boldsymbol{M}|})$, respectively, assuming that the size of the candidate Markov boundary $\boldsymbol{M}$ obtained in the Forward phase is close to the size of the true Markov boundary obtained after the Backward phase (see Figures 4 and 5), which is typically the case in practice (Aliferis et al., 2010a; Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a). When TIE* is parameterized with the IGS procedure (as the component $\mathbb{Y}$) and there is only one Markov boundary $\boldsymbol{M}$ in the distribution, TIE* will invoke a Markov boundary induction algorithm $\mathbb{X}$, $|\boldsymbol{M}|+1$ number of times. Thus, the total computational complexity of TIE* in this case becomes $O(|\boldsymbol{V}||\boldsymbol{M}|^2)$ if $\mathbb{X}$ = IAMB and $O(|\boldsymbol{V}||\boldsymbol{M}|\,|\,2^{|\boldsymbol{M}|})$ if $\mathbb{X}$ = Semi-Interleaved HITON-PC.When N Markov boundaries with the average size $|\boldsymbol{M}|$ are present in the distribution, TIE* with IGS procedure will invoke a Markov boundary induction algorithm no more than $O(N2^{|\boldsymbol{M}|})$ times. Therefore, the total complexity of TIE* with the IGS procedure is $O(N2^{|\boldsymbol{M}|}|\boldsymbol{V}||\boldsymbol{M}|)$) when $\mathbb{X}$ = IAMB and $O(N|\boldsymbol{V}|2^{2|\boldsymbol{M}|})$ when $\mathbb{X}$ = Semi-Interleaved HITON-PC.

In practical applications of TIE* with Semi-Interleaved HITON-PC, we use an additional caching mechanism for conditional independence decisions, which alleviates the need to repeatedly conduct the same conditional independence tests during Markov boundary induction when we have only slightly altered the data set by removing a subset of variables $\boldsymbol{G}$. In this case, induction of the first Markov boundary still takes $O(|\boldsymbol{V}|2^{|\boldsymbol{M}|})$ independence tests, but all consecutive Markov boundaries typically require less than $O(|\boldsymbol{V}|)$ conditional independence tests. Thus, the overall complex-

7. Since we use negative p-values from a conditional independence test as the measure of association in IAMB and Semi-Interleaved HITON-PC (see Appendix D), we assume that complexity of computing an association is equal to the complexity of conditional independence testing.

Algorithm **iTIE***

Input: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T.

Output: all Markov boundaries of T that exist in $\mathbb{P}$.

1. Run steps 1-13 of Semi-Interleaved HITON-PC algorithm (Figure 5) with the following steps instead of steps 6 and 7:
   - (a) If there is no subset $\boldsymbol{Z} \subseteq \boldsymbol{M}$ such that $T \perp Y \mid \boldsymbol{Z}$ then
   - (b) $\quad \boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{Y\}$
   - (c) Else if $\boldsymbol{Z}$ exists and the following relations hold: $T \not\perp Y$, $T \not\perp \boldsymbol{Z}$, $T \perp \boldsymbol{Z} \mid Y$
   - (d) $\quad$ Record in Θ that Y and $\boldsymbol{Z}$ contain equivalent information with respect to T
2. Compute the Cartesian product of information equivalence relations for subsets of $\boldsymbol{M}$ that are stored in Θ to construct multiple Markov boundaries of T
3. Output multiple Markov boundaries of T

Figure 12: iTIE* algorithm, presented as a modification of Semi-Interleaved HITON-PC. Similar algorithms may be obtained by modification of other members of the GLL-PC algorithmic family (Aliferis et al., 2010a).

ity of TIE* with the IGS procedure and Semi-Interleaved HITON-PC becomes $O(|\boldsymbol{V}|2^{|\boldsymbol{M}|} + (N-1)|\boldsymbol{V}|2^{|\boldsymbol{M}|})$, or equivalently $O(N|\boldsymbol{V}|2^{|\boldsymbol{M}|})$.

Finally, in practice we use parameters *max-card* for IGS procedure in TIE* and *max-k* for Semi-Interleaved HITON-PC to limit the number of conditional independence tests (see Appendix D). Thus, complexity of TIE* with the IGS procedure becomes $O(N|\boldsymbol{V}||\boldsymbol{M}|^{max\text{-}card+1})$ when $\mathbb{X}$= IAMB and $O(|\boldsymbol{V}||\boldsymbol{M}|^{max\text{-}k} + (N-1)|\boldsymbol{V}||\boldsymbol{M}|^{max\text{-}card})$ when $\mathbb{X}$ = Semi-Interleaved HITON-PC.

4.5 A Simple and Fast Algorithm for Special Distributions

The TIE* algorithm allows to find all Markov boundaries when there are information equivalence relations between arbitrary *sets of variables*. A simpler and faster algorithm can be obtained by restricting consideration to distributions where all information equivalence relations follow from context-independent information equivalence relations between *individual variables*. The resulting algorithm is termed iTIE*(which is an acronym for "Individual Target Information Equivalence") and is described in Figure 12. As can be seen, iTIE* can be described as a modification to Semi-Interleaved HITON-PC (or GLL-PC in general).

Consider running the iTIE* algorithm on data $\mathbb{D}$ generated from the example causal Bayesian network shown in Figure 13. The response variable T is directly caused by C, D, F. The underlying distribution is such that variables A and C contain equivalent information about T; likewise variables B and D contain equivalent information about T. iTIE* starts by executing Semi-Interleaved HITON-PC with the modified steps 6 and 7. Assume that we are running the loop in steps 3-8 of Semi-Interleaved HITON-PC and currently $\boldsymbol{E} = \{C, D\}$ and $\boldsymbol{M} = \{A, B, F\}$; variables E and J were eliminated conditioned on F in previous iterations of the loop. In step 4 of Semi-Interleaved HITON-PC, the algorithm may select $Y = C$. Next the modified steps 6 and 7 of Semi-Interleaved HITON-PC proceed as described in Figure 12, namely: 1(a) we find that a subset $\boldsymbol{Z}$=$\{A\}$ renders T independent

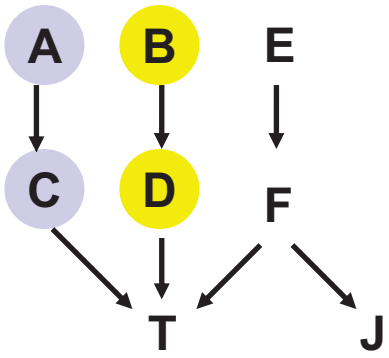


Figure 13: Graph of a causal Bayesian network used to trace the iTIE* algorithm. The network parameterization is provided in Table 7 in Appendix B. The response variable is T. All variables take values $\{0,1\}$. Variables A and C contain equivalent information about T and are highlighted with the same color. Likewise, variables B and D contain equivalent information about T and thus are also highlighted with the same color.

of $Y = C$; 1(c) T is marginally dependent on $Y = C, T$ is marginally dependent on $\boldsymbol{Z} = \{A\}$, and $Y = C$ renders T independent of $\boldsymbol{Z} = \{A\}$, thus 1(d) we record in Θ that $Y = C$ and $\boldsymbol{Z} = \{A\}$ contain equivalent information with respect to T. In the next iteration of the loop in steps 3- 8 of the modified Semi-Interleaved HITON-PC, we record in Θ that $Y = D$ and $\boldsymbol{Z} = \{B\}$ contain equivalent information with respect to T. The Backward phase in steps 9-13 of Semi-Interleaved HITON-PC does not result in variable eliminations in this example, thus we have $\boldsymbol{M} = \{A,B,F\}$. Finally, we build Cartesian product of information equivalence relations for subsets of $\boldsymbol{M}$ that are stored in Θ and obtain 4 Markov boundaries of T: $\{A,B,F\}, \{A,D,F\}, \{C,B,F\}$, and $\{C,D,F\}$.

The iTIE* algorithm correctly identifies all Markov boundaries under the following sufficient assumptions: (a) all equivalence relations in the underlying distribution follow from context-independent equivalence relations of *individual* variables, and (b) the assumptions of Theorem 12 hold. The proof of correctness of iTIE* can be obtained from the proofs of Theorems 9 and 12 and Lemma 1.

It is also important to notice that in some cases iTIE* can identify all Markov boundaries even if the above stated sufficient assumption (a) is violated; that is why we do not exclude the possibility that $\boldsymbol{Z}$ can be a set of variables in steps 1(c,d) of iTIE*. Consider a Bayesian network with the graph $C \begin{array}{c} \nearrow \\ \searrow \end{array} \begin{array}{c} A \\ B \end{array} \begin{array}{c} \searrow \\ \nearrow \end{array} T$ that is parameterized such that a variable C and the set of variables $\{A,B\}$ jointly contain context-independent equivalent information about T, and T is marginally dependent on A,B,C. Thus, there are two Markov boundaries of T in the joint probability distribution: $\{C\}$ and $\{A,B\}$. Now consider a situation when iTIE* first admits $\{A,B\}$ to $\boldsymbol{M}$ during execution of the modified Semi-Interleaved HITON-PC or another instance of GLL-PC. Then the step 1(c) of iTIE* will reveal that while $T \perp C \mid \{A,B\}$, the following relations hold $T \not\perp C$, $T \not\perp \{A,B\}$, and $T \perp \{A,B\} \mid C$. Thus, the algorithm will identify that C and $\{A,B\}$ contain equivalent information about T and will correctly find all Markov boundaries in the distribution. However, if iTIE* first admits C to $\boldsymbol{M}$, then the algorithm will output only one Markov boundary of T that consists of

a single variable C, because variables A and B, when considered separately, will be eliminated by conditioning on C and no equivalence relations will be found.

Notice that unlike TIE*, iTIE* does not rely on repeated invocation of a Markov boundary induction algorithm and instead extends Semi-Interleaved HITON-PC by potentially performing at most one additional independence test for each variable in $\boldsymbol{V}$ during the Forward phase, as shown in Figure 12.[8] This allows iTIE* to maintain computational complexity of the same order as Semi-Interleaved HITON-PC, namely, $O(|\boldsymbol{V}|2^{|\boldsymbol{M}|})$ conditional independence tests. As before, $|\boldsymbol{M}|$ denotes the average size of a Markov boundary and the above complexity bound assumes that the size of a candidate Markov boundary obtained in the Forward phase is close to the size of a true Markov boundary obtained at the end of the Backward phase (see Figure 5). In practical applications of iTIE*, we also use parameter *max-k* that limits the maximum size of a conditioning test, which brings complexity of iTIE* to $O(|\boldsymbol{V}||\boldsymbol{M}|^{max\text{-}k})$. Interestingly, iTIE* can efficiently identify all Markov boundaries in the distribution shown in Figure 2. This is due to the fact that the distribution in Figure 2 satisfies the assumption underlying iTIE* (i.e., that all information equivalences in a distribution follow from context-independent equivalences between individual variables) and thus allows it to capture all equivalence relationships between variables within groups in a single run of the Forward phase of the modified Semi-Interleaved HITON-PC. All Markov boundaries in the example in Figure 2 can then be reconstructed by taking the Cartesian product over sets of variables found to be equivalent with respect to T in step 2 of iTIE* (Figure 12).

For experiments reported in this work, we implemented and ran iTIE* based on the Causal Explorer code of Semi-Interleaved HITON-PC (Aliferis et al., 2003b; Statnikov et al., 2010) with values of parameters and statistical tests of independence that are described in Appendix D.

5. Empirical Experiments

In this section, we present experimental results obtained by applying methods for learning multiple Markov boundaries and variable sets on simulated and real data. The evaluated methods and their parameterizations are shown in Table 9 in Appendix E. These methods were chosen for our evaluation as they are the current state-of-the-art techniques for discovery of multiple Markov boundaries and variable sets. In order to study the behavior of these methods as a function of parameter settings, we considered several distinct parameterizations of each algorithm. In cases when parameter settings have been recommended by the authors of a method, we included these settings in our evaluation. A detailed description of parameters of prior methods for induction of multiple Markov boundaries and variable sets is provided in Appendix C.

All experiments involving assessment of classification performance were executed by hold-out validation or cross-validation (see below), whereby Markov boundaries and variable sets are discovered in a training subset of data samples (training set), classification models based on the above variables are also developed in the training set, and the reported performance of classification models is estimated in an independent testing set. Assessment of classification performance of the extracted Markov boundaries and variable sets was done using Support Vector Machines (SVMs) (Vapnik, 1998). We chose to use SVMs due to their excellent empirical performance across a wide range of application domains (especially with high-dimensional data and relatively small sample sizes), regularization capabilities, ability to learn both simple and complex classification

8. This is a test $T \perp \boldsymbol{Z} \mid Y$. Other necessary tests $T \not\perp Y$ and $T \not\perp \boldsymbol{Z}$ have been previously computed in step 4 of Semi-Interleaved HITON-PC algorithm, and their results can be retrieved from the cache.

functions, and tractable computational time (Cristianini and Shawe-Taylor, 2000; Schölkopf et al., 1999; Shawe-Taylor and Cristianini, 2004; Vapnik, 1998). When the response variable was multi-class, we applied SVMs in one-versus-rest fashion (Schölkopf et al., 1999). We used libSVM v.2.9.1 (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) implementation of SVMs in all experiments (Fan et al., 2005). Polynomial kernels were used in SVMs as they have shown good classification performance across the data domains considered in this study. The degree d of the polynomial kernel and the penalty parameter C of SVM were optimized by cross-validation on the training data. Each variable in a data set was scaled to $[0,1]$ range to facilitate SVM training. The scaling constants were computed on the training set of samples and then applied to the entire data set.

All experiments presented in this section were run on the Asclepius Compute Cluster at the Center for Health Informatics and Bioinformatics (CHIBI) at New York University Langone Medical Center (`http://www.nyuinformatics.org`) and the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University (`http://www.accre.vanderbilt.edu/`). For comparative purposes all experiments used exclusively the latest generation of Intel Xeon Nehalem (x86) processors. Overall, it took >50 years of single CPU time to complete all reported experiments.

5.1 Experiments with Simulated Data

Below we present an evaluation of methods for extraction of multiple Markov boundaries and variable sets in simulated data. Simulated data allows us to evaluate methods in a controlled setting where the underlying causal process and all Markov boundaries of the response variable T are known exactly. Two data sets were used in this evaluation. One of these data sets, referred to as $TIED$, was previously used in an international causality challenge (Statnikov and Aliferis, 2010b). $TIED$ contains 30 variables, including the response variable T. The underlying causal graph and its parameterization are given in the work by Statnikov and Aliferis (2010b). There are 72 distinct Markov boundaries of T. Each Markov boundary contains 5 variables: variable X_{10} and one variable from each of the four subsets $\{X_1, X_2, X_3, X_{11}\}$, $\{X_5, X_9\}$, $\{X_{12}, X_{13}, X_{14}\}$ and $\{X_{19}, X_{20}, X_{21}\}$. Another simulated data set, referred to as $TIED1000$, contains 1,000 variables in total and was generated by the causal process of $TIED$ augmented with an additional 970 variables that have no association with T. $TIED1000$ has the same set of Markov boundaries of T as $TIED$. $TIED1000$ allows us to study the behavior of different methods for learning multiple Markov boundaries and variable sets in an environment where the fraction of variables carrying relevant information about T is small.

For each of the two data sets, 750 observations were used for discovery of Markov boundaries/variable sets and training of the SVM classification models of the response variable T (with the goal to predict its values from the inferred Markov boundary variables), and an independent testing set of 3,000 observations was used for evaluation of the models' classification performance.

All methods for extracting multiple Markov boundaries and variable sets were assessed based on the following six performance criteria:

I. *The number of distinct Markov boundaries/variable sets output by the method.*

II. *The average size of an output Markov boundary/variable set (number of variables).*

III. *The number of true Markov boundaries identified exactly, that is, without false positives and false negatives.*[9]

IV. *The average Proportion of False Positives (PFP) in the output Markov boundaries/variable sets.*[10]

V. *The average False Negative Rate (FNR) in the output Markov boundaries/variable sets.*[11]

VI. *The average classification performance (weighted accuracy) over all output Markov boundaries/variable sets.*[12] We also compared the average classification performance of the SVM models with the maximum a posteriori classifier in the true Bayesian network (denoted as MAP-BN) using the same data sample.

Technical details about computing performance criteria III-V are given in Appendix E.

The results presented in Figure 14 in the manuscript, and Tables 10 and 11 and Figure 19 in Appendix E show that only TIE* and iTIE* identified *exactly* all and only true Markov boundaries of T in both simulated data sets, and their classification performance with the SVM classifier was statistically comparable to performance of the MAP-BN classifier. None of the comparator methods, regardless of the number of Markov boundaries/variable sets output, were able to identify *exactly* any of the 72 true Markov boundaries, except for Resampling+RFE (without statistical comparison) and IR-HITON-PC that identified *exactly* 1-2 out of 72 true Markov boundaries, depending on the data set. Overall prior methods had either large proportion of false positives or large false negative rate, and often their classification performance was significantly worse that the performance of the MAP-BN classifier. However, in some cases the classification performance of other methods was comparable to the MAP-BN classifier, regardless of the number of Markov boundaries identified *exactly*. This can be attributed to (i) the relative insensitivity of the SVM classifiers to false positives, (ii) connectivity in the underlying graph that compensates false negatives with other weakly relevant variables, and (iii) differences between the employed classification performance metric (weighted accuracy) and the metric which is maximized by the Markov boundary variables (that requires accurate estimation of $\mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$, which is a harder task than maximizing proportions of correct classification in the weighted accuracy metric). Thus, we remind the reader that a high classification performance is often a necessary but not sufficient condition for correct identification of Markov boundaries. Detailed discussion of the performance of comparator methods is given in Appendix E.

5.2 Experiments with Real Data

For evaluation of methods for learning multiple Markov boundaries and variable sets in real data, we used 13 data sets that cover a broad range of application domains (clinical outcome prediction, gene expression, proteomics, drug discovery, text categorization, digit recognition, ecology and finance), dimensionalities (from 86 to over 100,000), and sample sizes (from hundreds to thousands) that are representative of those appearing in practical applications. These data sets have recently been used

9. False positives are variables that do not belong to any true Markov boundary of T in the causal graph, but are included in a Markov boundary/variable set extracted by some method. False negatives are variables that belong to a true Markov boundary of T, but are absent in the extracted Markov boundary/variable set.
10. PFP is the number of false positives in an output Markov boundary/variable set divided by its size.
11. FNR is the number of false negatives in an output Markov boundary/variable set divided by the size of the *true* Markov boundary.
12. Given that the response variable T had four possible values, classification performance was measured by the weighted accuracy metric that allows to measure classification performance independent of class priors and can be applied to multiclass responses (Guyon et al., 2006). In brief, weighted accuracy is obtained by computing proportion of correct classifications in each class and combining these proportions by weighting using prior probabilities in each class.

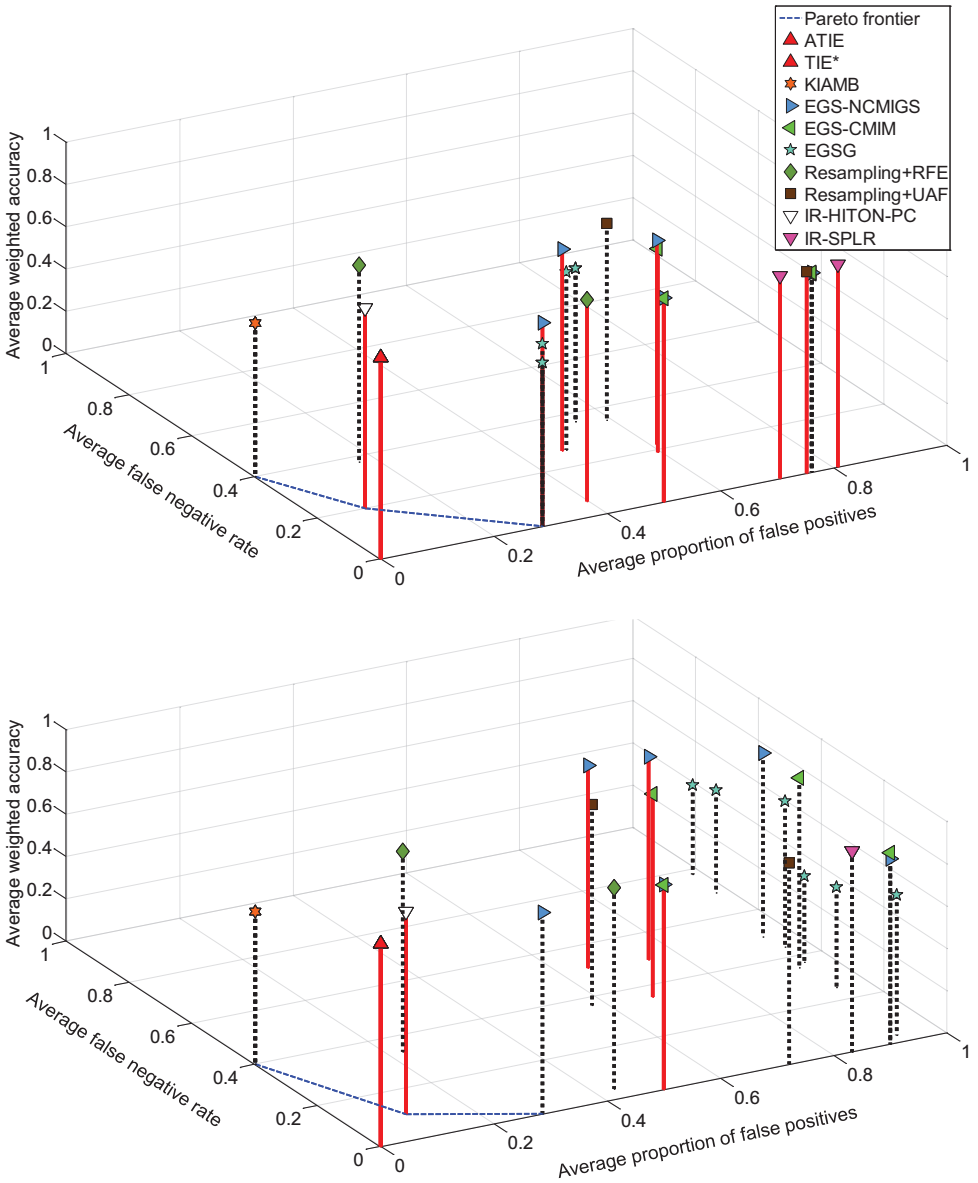


Figure 14: Results for average classification performance (weighted accuracy), average false negative rate, and average proportion of false positives that were obtained in *TIED* (top figure) and *TIED1000* (bottom figure) data sets. The style and color of a vertical line connecting each point with the plane shows whether the average SVM classification performance of a method is statistically comparable with the MAP-BN classifier in the same data sample (red solid line) or not (black dotted line). The Pareto frontier was constructed based on the average false negative rate and the average proportion of false positives over the comparator methods (i.e., non-TIE*). Results of TIE* and iTIE* were identical in both data sets.

in a broad benchmark (Aliferis et al., 2010a) of the current state-of-the-art single Markov boundary induction and feature selection methods, which is another reason why we chose to use the same data in this study. The data sets are described in detail in Table 12 in Appendix E. The data sets were preprocessed (imputed, discretized, etc.) as described in the work by Aliferis et al. (2010a).

In data sets with relatively large sample sizes (> 600), classification performance of the output Markov boundaries and variable sets was estimated by holdout validation with 75% of samples used for Markov boundary/variable set induction and SVM classifier training, and the remaining 25% of samples used for estimation of classification performance. In small-sample data sets, 10- fold cross-validation was used instead. Markov boundary/variable set induction and classifier training were both performed on the training sets from the 10-fold cross-validation design, with classification performance being subsequently estimated on the respective testing sets.

Evaluation of Markov boundary/variable selection methods in real data is challenging due to the lack of knowledge of the true Markov boundaries. In practical applications, however, the interest typically lies in the most compact subsets of variables that give the highest classification performance for reasonable and widely used classifiers (Guyon and Elisseeff, 2003). This consideration motivated the following two primary evaluation criteria (with the averages taken over all Markov boundaries/variable sets output by each method):

I. *The average Proportion of Variables (PV) in the output Markov boundaries/variable sets.*[13]

II. *The average classification performance (AUC) of the output Markov boundaries/variable sets.*[14]

In addition to the above two primary criteria, in some problems we are also interested in extracting as many of the maximally compact and predictive variable sets (i.e., optimal solutions to the variable selection problem) as possible. Therefore, we also considered a third criterion in our evaluation:

III. *The number of distinct Markov boundaries/variable sets output by each method (N).*

We note that criterion I (PV) on its own can be optimized independently of the actual classification problem by taking small subsets of variables (e.g., 1 or 2 variables in each subset) to be the presumed Markov boundaries of the response variable T. Criterion I may therefore be biased towards methods that output Markov boundaries/variable sets of a user-defined size (e.g., some parameterizations of EGS-NCMIGS). Similarly, criterion III (N) can be maximized independently of the response T by simply taking all $2^{|\boldsymbol{V}|-1}-1$ non-empty subsets of the variable set $\boldsymbol{V} \setminus \{T\}$ to be the presumed Markov boundaries of T. This criterion could be potentially biased towards Markov boundary/variable set extraction methods that output a number of Markov boundaries specified by a user-defined parameter (e.g., EGSG) rather than by a data driven process (e.g., TIE*). Criterion II (AUC) served as a modulator for criteria I and III, because high performance on the latter two criteria does not necessarily imply high classification performance.

We also ranked all methods on each of the three criteria averaged over all 13 real data sets, as described in Appendix E. The ranks incorporated permutation-based statistical comparison of difference in the performance of algorithms, in order to ensure that methods with statistically comparable performance receive the same rank.

13. The PV of an output Markov boundary/variable set measures its compactness and is defined as the number of variables in the output Markov boundary/variable set divided by the total number of variables in the data set.
14. Classification performance was measured using area under ROC curve (AUC) (Fawcett, 2003), because all response variables were binary.

Finally, given ranks on the individual criteria I (PV) and II (AUC), we defined a combined (PV, AUC) ranking criterion which reflects the ability of methods to find Markov boundaries in real data. This is because Markov boundaries are expected to maximize performance of the classifiers with universal approximation capabilities (maximize AUC of SVMs in our study) and be of minimal size (minimize PV in our study) (Tsamardinos and Aliferis, 2003). The combined (PV, AUC) criterion was defined as follows: First, the ranks on the individual criteria PV and AUC were normalized to the $[0, 1]$ interval to account for varying rank ranges that resulted from ties in performance. Second, the normalized ranks on the two criteria were averaged. Third, the resulting averages were used to establish a new ranking of methods, aided by a permutation-based testing approach to ensure that methods with statistically comparable performance receive the same rank (see Appendix E).

Other alternative combined (PV, AUC) ranking criteria, for example, one that performs ranking based on some combination of *raw* PV and AUC scores, can also be used for performance assessment in our study. We have confirmed that the best performing method remains the same when either combining normalized ranks of PV and AUC (our criterion) or raw scores of PV and AUC (alternative criterion) by an average function. This can be evidenced from Figure 15 and Tables 3 and 4 which are discussed below.

The results of experiments are presented in Figure 15 and Tables 3 and 4.[15] Figure 15 shows a 2-dimensional plot of PV versus AUC and a 3-dimensional plot of PV versus AUC versus the number of extracted distinct Markov boundaries or variable sets (N). Each point in Figure 15 corresponds to the results of one of the methods considered in this evaluation, averaged over all 13 data sets. The Pareto frontier shown in Figure 15 was constructed based on the two primary evaluation criteria PV and AUC over the prior methods (i.e., non-TIE*). Methods on the Pareto frontier are such that no other non-TIE* method had both lower PV and higher AUC when averaged over all data sets. For ease of visualization, results on all variables (i.e., without variable selection) were omitted from Figure 15. When all variables were used for classification, the average PV and AUC were 100% and 0.902, respectively. These results did not alter the Pareto set of prior methods in Figure 15 and are reported in Table 13, 14 and 15 in Appendix E. The results averaged over all data sets are shown in Table 3. The results for all methods in each data set individually are presented in Table 13, 14 and 15 in Appendix E. Ranks of the methods were computed as described above and are shown in Table 4.

As can be seen in Figure 15 and Tables 3 and 4, none of the prior methods had both more compact Markov boundaries or variable sets (lower PV) and better classification performance (higher AUC) than TIE*. This is evidenced by TIE*s performance laying beyond the Pareto frontier constructed over the prior methods in Figure 15. While a few methods had comparable or slightly higher AUC (Table 3), their Markov boundaries or variable sets were substantially larger with the average PV reaching as high as 41% (see Resampling+UAF in Table 3). In contrast, Markov boundaries output by TIE* were much more compact with an average PV of 2.3%. On the other hand, methods that had PV lower than TIE* also had lower AUC. KIAMB, for example, had a PV of 1% and an AUC of about 0.8, which was 7-8% lower than the AUC of TIE*. Overall, TIE* ranked first out of 15 on the combined (PV, AUC) criterion. Please see Appendix E for a detailed discussion of the results of prior methods.

It is worth noting that use of the AUC metric for verification of Markov boundaries in the Predictivity criterion of TIE* can result in some spurious multiplicity of the output Markov boundaries.

15. We did not include iTIE* in this comparison, because we anticipated that it will be outperformed by TIE* due to its broader distributional assumptions than the ones of iTIE*.

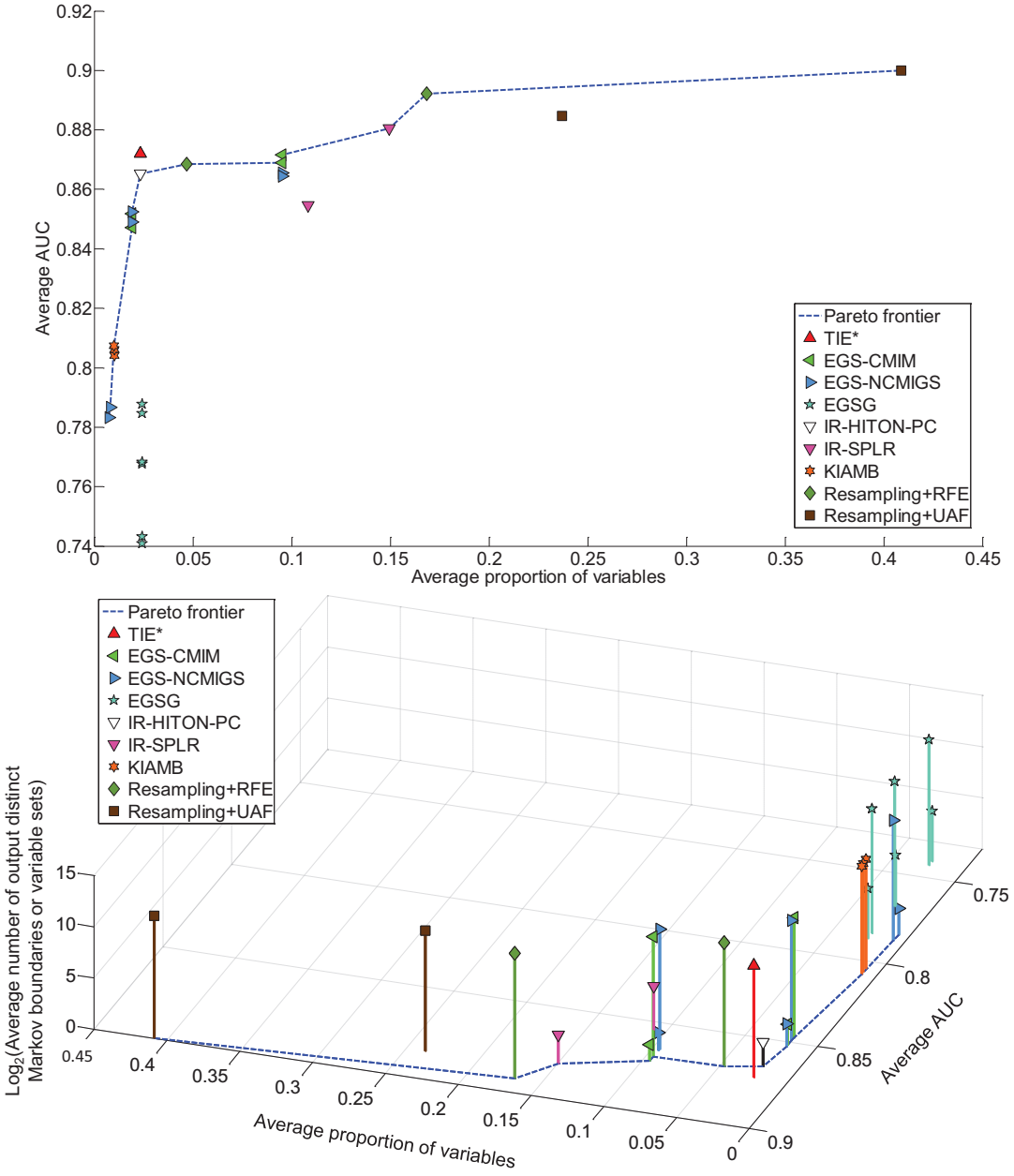


Figure 15: Average performance of the evaluated methods across 13 real data sets. The Pareto frontier was constructed based on the average proportion of variables and the average AUC over the prior methods (i.e., non-TIE*). Detailed results are provided in Tables 3 and 4.

Method		Average		
		N	PV	AUC
TIE*	*max-k* = 3, $\alpha = 0.05$	1993	0.023	0.872
KIAMB	Number of runs = 5000, $\alpha = 0.05$, $K = 0.7$	1688	0.010	0.804
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.8$	1552	0.010	0.806
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.9$	1461	0.010	0.807
EGS-NCMIGS	$l = 7$, $\delta = 0.015$	6	0.007	0.783
	$l = 7$, $K = 10$	5	0.019	0.853
	$l = 7$, $K = 50$	3	0.095	0.865
	$l = 5000$, $\delta = 0.015$	3402	0.008	0.787
	$l = 5000$, $K = 10$	3395	0.019	0.849
	$l = 5000$, $K = 50$	3364	0.095	0.864
EGS-CMIM	$l = 7$, $K = 10$	4	0.019	0.852
	$l = 7$, $K = 50$	3	0.095	0.872
	$l = 5000$, $K = 10$	3394	0.019	0.847
	$l = 5000$, $K = 50$	3363	0.095	0.869
EGSG	Number of Markov boundaries = 30, $t = 5$	30	0.024	0.788
	Number of Markov boundaries = 30, $t = 10$	30	0.024	0.768
	Number of Markov boundaries = 30, $t = 15$	30	0.024	0.741
	Number of Markov boundaries = 5,000, $t = 5$	4634	0.024	0.785
	Number of Markov boundaries = 5,000, $t = 10$	4879	0.024	0.768
	Number of Markov boundaries = 5,000, $t = 15$	4936	0.024	0.743
Resampling+RFE	without statistical comparison	4896	0.168	0.892
	with statistical comparison ($\alpha = 0.05$)	4371	0.047	0.868
Resampling+UAF	without statistical comparison	4033	0.409	0.900
	with statistical comparison ($\alpha = 0.05$)	3548	0.237	0.885
IR-HITON-PC	*max-k* = 3, $\alpha = 0.05$	5	0.023	0.865
IR-SPLR	without statistical comparison	7	0.149	0.881
	with statistical comparison ($\alpha = 0.05$)	20	0.108	0.855

Table 3: Number of distinct Markov boundaries or variable sets identified by the evaluated methods (N), proportion of variables in them (PV) and their classification performance (AUC) averaged across all 13 real data sets for each method. The color of highlighting signifies relative performance on each criterion with dark red corresponding to the best performance and light yellow to the worst. See Table 4 for ranks of methods that also incorporate formal statistical comparison of the observed differences between methods.

This can happen due to a possible mismatch between subsets of variables that lead to maximization of the AUC metric for a given classifier and those that render the response variable T conditionally independent of all other variables (thus effectively optimizing a metric that requires accurate estimation of $\mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$). Consider an example where only a subset of variables from some Markov boundary is sufficient to obtain the same AUC as the entire Markov boundary. Suppose there are in total five variables $\{A, B, C, D, T\}$ in the data set and $\boldsymbol{M}_1 = \{A, B, C, D\}$ is the only Markov boundary of the response variable T. Suppose also that the subset $\boldsymbol{M}_2 = \{A, B, C\}$ yields the same classification performance as the Markov boundary $\boldsymbol{M}_1$ according to the AUC metric. Once TIE* discovers the Markov boundary $\boldsymbol{M}_1 = \{A, B, C, D\}$, it will consider removing $\{D\}$, as well as other

subsets of $\boldsymbol{M}_1$, to discover other possible Markov boundaries. After removing subset $\{D\}$ from the data, TIE* would identify $\boldsymbol{M}_2 = \{A, B, C\}$ as a candidate Markov boundary to be verified by the Predictivity criterion. Because $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ have the same classification performance (AUC), $\boldsymbol{M}_2$ will be admitted as a Markov boundary by the Predictivity criterion. In order to control for possible presence of such spurious Markov boundaries in the output of TIE*, we performed an additional analysis of its output whereby for each data set, we considered only those Markov boundaries that were not proper subsets of any other Markov boundary extracted by TIE* in the same data set. We refer to such Markov boundaries as minimal. The average number of minimal Markov boundaries identified by TIE* was 1,484 (versus the average number of all Markov boundaries identified by TIE* equal to 1,993). The average size (2.3% PV) and classification performance (0.872 AUC) of the minimal Markov boundaries were statistically indistinguishable from the results obtained on all Markov boundaries identified by TIE* and so were the ranks on the PV, AUC and (PV, AUC) criteria.

In summary, TIE* extracted multiple compact Markov boundaries with high classification performance and surpassed all other methods on the combined (PV, AUC) criterion. Since the data-generative process in experiments with real data sets is unknown, a question that arises is: *do multiple Markov boundaries exist in real data?* Prior work using the same data has established that performance patterns of single Markov boundaries identified by Semi-Interleaved HITON-PC (an instantiation of the GLL framework) are highly consistent with the Markov boundary induction theory and that GLL algorithms dominated an extensive panel of prior state-of-the-art Markov boundary and variable selection methods in terms of compactness and classification performance (Aliferis et al., 2010a). In this paper, we showed that TIE* parameterized with Semi-Interleaved HITON-PC as the base Markov boundary induction algorithm was able to identify multiple compact Markov boundaries with consistently high classification performance in real data. For example, in the ACPJ_Etiology data set, TIE* identified 5,330 distinct Markov boundaries (and 4,263 minimal ones) that on average contained 18 variables out of 28,228 and had an AUC of 0.91. Out of all prior methods for learning multiple Markov boundaries and variable sets applied to the same data set, Resampling+UAF had the highest classification performance with an AUC of 0.93, which was statistically non-distinguishable from TIE*, while variable sets extracted by Resampling+UAF, on average, were more than two orders of magnitude larger and contained 3,883 variables. A similar pattern can be observed in the Dexter data set where TIE* identified 4,791 distinct Markov boundaries (and 3,498 minimal ones) with an average size of 17 variables out of 19,999 and an AUC of 0.96. The best performer among prior methods in the same data was EGS-CMIM with Markov boundaries containing 50 variables each and an average AUC of 0.98, the latter being statistically non-distinguishable from TIE*. The compactness of Markov boundaries extracted by TIE* coupled with their high classification performance provides strong evidence that there are indeed multiple Markov boundaries in many real-life problem domains.

6. Discussion

This section summarizes main findings, reiterates key principles of TIE* efficiency, demonstrates how the generative algorithm TIE* can be configured for optimal results, presents limitations of this study, and outlines directions for future research.

Method		Rank			
		N	PV	AUC	(PV, AUC)
TIE*	*max-k* = 3, α = 0.05	4	5	2	1
KIAMB	Number of runs = 5000, α = 0.05, K = 0.7	4	2	4	5
	Number of runs = 5000, α = 0.05, K = 0.8	4	2	4	5
	Number of runs = 5000, α = 0.05, K = 0.9	4	2	4	5
EGS-NCMIGS	l = 7, δ = 0.015	6	1	4	3
	l = 7, K = 10	6	5	3	6
	l = 7, K = 50	6	9	3	11
	l = 5000, δ = 0.015	3	2	4	5
	l = 5000, K = 10	3	4	3	4
	l = 5000, K = 50	3	9	3	11
EGS-CMIM	l = 7, K = 10	6	5	3	6
	l = 7, K = 50	6	9	2	8
	l = 5000, K = 10	3	3	3	2
	l = 5000, K = 50	3	9	2	8
EGSG	Number of Markov boundaries = 30, t = 5	5	6	4	10
	Number of Markov boundaries = 30, t = 10	5	6	4	10
	Number of Markov boundaries = 30, t = 15	5	6	5	13
	Number of Markov boundaries = 5,000, t = 5	2	9	4	14
	Number of Markov boundaries = 5,000, t = 10	2	8	4	12
	Number of Markov boundaries = 5,000, t = 15	1	7	5	15
Resampling+RFE	without statistical comparison	2	10	2	9
	with statistical comparison (α = 0.05)	2	9	3	11
Resampling+UAF	without statistical comparison	3	11	1	7
	with statistical comparison (α = 0.05)	3	10	2	9
IR-HITON-PC	*max-k* = 3, α = 0.05	6	5	3	6
IR-SPLR	without statistical comparison	6	10	2	9
	with statistical comparison (α = 0.05)	5	9	3	11

Table 4: Ranks of methods based on individual and combined criteria. Smaller ranks correspond to better methods according to each criterion. As described in text, ranks were obtained using formal statistical comparison of the observed differences between methods; that is why they do not necessarily range between 1 and 27 (total number of tested methods).

6.1 Main Findings

There are two major contribution of this study. First, we presented TIE*, a generative anytime algorithm for discovery of multiple Markov boundaries. TIE* is sound under well-defined sufficient conditions and can be practically applied to high-dimensional data sets with relatively small sample. We performed a theoretical analysis of the algorithm correctness and derived estimates of its computational complexity. To make our paper valuable for practitioners, we provided several specific instantiations of the generative algorithm TIE* and described their implementation details.

Second, we conducted an empirical comparison of TIE* with 26 state-of-the-art methods for discovery of multiple Markov boundaries and variable sets. The empirical study was performed on 2 simulated data sets with exactly known Markov boundaries and 13 real data sets from a diversity

of application domains. We found that unlike prior methods, TIE* identifies exactly all true Markov boundaries in simulated data, and in real data it yields Markov boundaries with simultaneously better classification performance and smaller number of variables compared to prior methods.

Other notable contributions of this work include: (i) developing a deeper theoretical understanding of distributions with multiple Markov boundaries of the same variable (Sections 2.2-2.4), (ii) theoretical analysis of prior state-of-the-art algorithms for discovery of multiple Markov boundaries and variable sets (Appendix C), (iii) a novel simple and fast algorithm iTIE* for learning multiple Markov boundaries in special distributions (Section 4.5), and (iv) evidence that multiple Markov boundaries exist in real data (Section 5.2).

6.2 Key Principles of TIE* Efficiency

We will illustrate key principles of TIE* efficiency using a simple example. Consider a distribution that spans over variables $\boldsymbol{M} = \{T, X_1, X_2, X_3, X_4, X_5, Y_1, Y_2, Z_1, \ldots, Z_{1000}\}$ and contains two Markov boundaries of T: $\boldsymbol{M}_1 = \{X_1, X_2, X_3, X_4, X_5\}$ and $\boldsymbol{M}_2 = \{X_1, X_2, X_3, X_4, Y_1, Y_2\}$, because X_5 and $\{Y_1, Y_2\}$ contain context-independent equivalent information about T. Assuming that we can apply a standard single Markov boundary induction algorithm to identify $\boldsymbol{M}_1$, one naive approach to discover multiple Markov boundaries in this distribution is to exhaustively consider whether a variable subset in $\boldsymbol{M}_1$ can be substituted with a variable subset in $\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \{T\}$ to obtain a new Markov boundary. In this example we will have to substitute 31 non-empty subsets in $\boldsymbol{M}_1$ with approximately $2^{1002} - 1$ non-empty subsets of $\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \{T\}$ (the latter number being orders of magnitude larger than the number of atoms in the universe). This approach is clearly computationally prohibitive in high-dimensional data sets. *The first core efficiency principle* in TIE* is to avoid explicit search of all possible subsets of $\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \{T\}$ and repeatedly run a fast Markov boundary induction algorithm on the data for variables in $\boldsymbol{V} \setminus \boldsymbol{G}$, where $\boldsymbol{G}$ is a subset of the previously found Markov boundaries. In the example stated above, this would lead to running a Markov boundary induction algorithm $2^7 = 128$ times (because there are 7 members in the union of all Markov boundaries) to find all Markov boundaries that exist in the distribution. *The second core efficiency principle* in TIE* dictates to consider removing from $\boldsymbol{V}$ only certain subsets $\boldsymbol{G}$ of the previously found Markov boundaries. Specifically, we consider only subsets $\boldsymbol{G}$ that do not include a subset of variables $\boldsymbol{G}^*$ (i.e., $\boldsymbol{G}^* \not\subset \boldsymbol{G}$) that did not result in discovery of a Markov boundary when the Markov boundary induction algorithm has been previously run on the data for variables in $\boldsymbol{V} \setminus \boldsymbol{G}^*$. Coupled with the heuristic to first generate subsets $\boldsymbol{G}$ of the smallest size, this principle can significantly decrease the number of runs of the Markov boundary induction algorithm. In the example stated above, this principle as exemplified in IGS procedure would lead to running a single Markov boundary induction algorithm only 8 times in order to find all Markov boundaries that exist in the distribution. Specifically, we will have to consider $\boldsymbol{G} = \varnothing, \{X_1\}, \{X_2\}, \{X_3\}, \{X_4\}, \{X_5\}, \{X_5, Y_1\}$, and $\{X_5, Y_2\}$. We would not need to consider $\boldsymbol{G} = \{X_1, X_2\}$ because its subset ($\boldsymbol{G}^* = \{X_1\}$ or $\{X_2\}$) did not lead to discovery of any Markov boundary when the algorithm was run on the data for variables in $\boldsymbol{V} \setminus \boldsymbol{G}^*$. Finally, since very fast single Markov boundary induction algorithms have been recently introduced (Aliferis et al., 2010a, 2003a; Peña et al., 2007; Tsamardinos et al., 2003a,b), the overall TIE* operation is very fast.

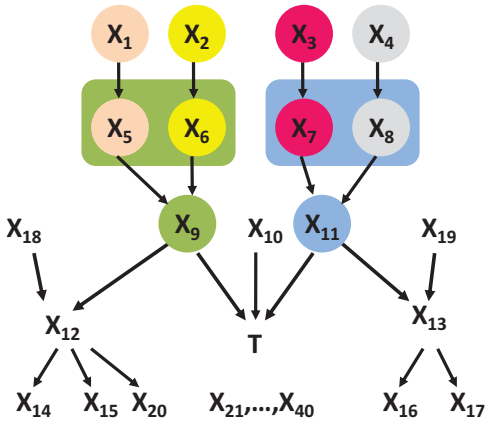


Figure 16: Graph of a causal Bayesian network used to trace the TIE* algorithm. The network parameterization is provided in Table 8 in Appendix B. The response variable is T. All variables take values $\{0,1\}$. Variables that contain equivalent information about T are highlighted with the same color, for example, variables X_1 and X_5 provide equivalent information about T; variable X_9 and each of the four variable sets $\{X_5,X_6\}$, $\{X_1,X_2\}$, $\{X_1,X_6\}$, $\{X_5,X_2\}$ provide equivalent information about T.

6.3 The Generative Nature of TIE* Allows to Configure the Algorithm for Optimal Results

TIE* is a generative algorithm that can be instantiated differently for different distributions. For example, distributions that violate the local composition property with respect to T for members of Markov boundaries (e.g., when T is defined as a parity function of its Markov boundary members that are unrelated and have balanced priors) are incompatible with the assumptions of Markov boundary induction algorithms IAMB and Semi-Interleaved HITON-PC that were considered in this work. The generative nature of TIE* suggests to use an admissible Markov boundary induction algorithm that is suitable for the distribution at hand.

Consider running TIE* algorithm on data $\mathbb{D}$ generated from the example causal Bayesian network shown in Figure 16. There are 25 distinct Markov boundaries of T in this distribution. Each of these Markov boundaries contains 3 or 5 variables: (i) X_9 or $\{X_5,X_6\}$ or $\{X_1,X_2\}$ or $\{X_1,X_6\}$ or $\{X_5,X_2\}$, (ii) X_{10}, and (iii) X_{11} or $\{X_7,X_8\}$ or $\{X_3,X_4\}$ or $\{X_3,X_8\}$ or $\{X_7,X_4\}$. The local composition property with respect to T is violated here because $T = \text{XOR}(X_9,X_{10},X_{11})$. To illustrate applicability to such distributions, we ran TIE* with a Markov boundary induction algorithm SVM-FSMB (Brown et al., 2012; Tsamardinos and Brown, 2008) as input component $\mathbb{X}$ and $\mathbb{Y}$ = IGS-Lex, $\mathbb{Z}$ = Predictivity. In brief, SVM-FSMB works by first extracting features from the polynomial SVM feature space that have largest SVM weights and then running a Markov boundary induction algorithm Semi-Interleaved HITON-MB in the SVM feature space on the constructed features. This allows SVM-FSMB to circumvent the requirement for the local composition property. We found that in a sufficiently large sample size ($\geq$ 2,000), TIE* can discover all 25 true Markov boundaries with only 1 false positive in each extracted Markov boundary. This showcases how the generative nature of TIE* allows to optimally configure the algorithm for the distribution at hand.

6.4 Limitations and Open Problems

The empirical evaluation of TIE* performed in this study used 13 real data sets from a diversity of application domains and provided evidence about existence of multiple Markov boundaries in real-life data, primarily based on compactness of output variable sets and high classification performance. The absence of knowledge about the true Markov boundaries in real data sets is a limitation of the study, which is in our opinion mitigated by strong empirical evidence for existence of multiple Markov boundaries.

Related to the above, the present work does not address the source of multiplicity of Markov boundaries induced in real data. In other words, we do not separate intrinsic multiplicity of Markov boundaries (that exists in the underlying probability distribution) from apparent multiplicity due to various factors including (but not limited to) small sample size, hidden variables, correlated measurement noise, and artifacts of normalization and/or data pre-processing (Statnikov and Aliferis, 2010a).

Also, as we have pointed out, the use of the AUC metric for verification of Markov boundaries in the Predictivity criterion of TIE* can result in a small percentage of spurious Markov boundaries in the output of the algorithm. This can happen due to a possible mismatch between subsets of variables that lead to maximization of the AUC metric for a given classifier and those that render the response variable T conditionally independent of all other variables (thus effectively optimizing a metric that requires accurate estimation of $\mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$). In this paper we experimented with one approach to reduce spurious multiplicity of TIE* by filtering extracted Markov boundaries to the minimal ones. A more conventional approach to this problem is to augment the Markov boundary induction method with an additional backward wrapping step (Aliferis et al., 2010a; Kohavi and John, 1997). However, backward wrappers are prone to overfitting because they evaluate a large number of classifier models with various variable subsets (Aliferis et al., 2010a), thus negatively affecting generalizability of TIE*. We have conducted preliminary experiments with a backward wrapping method applied on 13 real data sets, and indeed the results revealed a significant reduction in classification performance, as theoretically expected. We believe that it is still worthwhile to explore more sophisticated wrapping strategies (especially ones that guard against overfitting) in order to optimize the output of a Markov boundary inducer for a specific performance metric and classifier.

Finally, another limitation of this study is that we included in empirical experiments both algorithms for discovery of multiple Markov boundaries and algorithms for discovery of multiple variable sets. Even though the latter family of algorithms are not theoretically designed for Markov boundary induction, many researchers use them (Pellet and Elisseeff, 2008). This motivated us to include in our study methods for selection of multiple variable sets.

6.5 Directions for Future Research

In addition to addressing open problems outlined in the previous subsection, there are several promising directions for future research.

First, it is interesting to routinely apply TIE* to discover multiple Markov boundaries in various application domains. This would allow one to learn whether some problem domains are more prone to multiplicity of Markov boundaries than others. These results would instruct data-analysts about potential existence of many more solutions and can form guidelines for performing analysis in such data.

Second, it is important to extend existing causal graph discovery methods to take into account violations of the intersection property that lead to multiple Markov boundaries. For example, recent work was able to modify the PC algorithm to account for information equivalence relations between variables (Lemeire et al., 2010). However, many more algorithms remain to be improved upon.

Third, a useful direction for future research is to improve computational efficiency and run time of TIE* by using high-performance computers with parallel and/or distributed architectures. We have previously designed parallel versions of Markov boundary induction algorithms (Aliferis et al., 2010b, 2002) and in some cases were able to achieve more than linear increase of computational efficiency. At face value, this suggests that modifications of TIE* that run on parallel/distributed architectures can discover multiple Markov boundaries in domains where TIE*'s run time was prohibitive.

Acknowledgments

The empirical evaluation was supported in part by grants R01 LM011179-01A1 and R56 LM007948-04A1 from the NLM/NIH and 1UL1RR029893 from the NCRR/NIH. The authors are also grateful to Efstratios Efstathiadis, Frank E. Harrell Jr., Carie Lee Kennedy, and Eric Peskin for help with providing access and running experiments on high performance computing facilities. Finally, the authors would like to thank Alexander V. Alekseyenko, Mikael Henaff, and Yindalon Aphinyanaphongs for their useful comments on the manuscript.

Appendix A. Proofs of Theorems and Lemmas

Proof Lemma 1 : Assume that $\boldsymbol{M} \cap \boldsymbol{M}_{new} = \boldsymbol{N}$. Then it follows that $\boldsymbol{M}=\boldsymbol{N} \cup \boldsymbol{Y}$ and $\boldsymbol{M}_{new} = \boldsymbol{N} \cup \boldsymbol{Z}$. Since $\boldsymbol{M}$ is a Markov blanket, $T \perp (\boldsymbol{V} \backslash \{T\} \backslash (\boldsymbol{N} \cup \boldsymbol{Y})) \mid (\boldsymbol{N} \cup \boldsymbol{Y})$. By the self-conditioning property, it follows that $T \perp (\boldsymbol{V} \backslash \{T\}) \mid (\boldsymbol{N} \cup \boldsymbol{Y})$. The previous independence relation is equivalent to $T \perp ((\boldsymbol{V} \backslash \{T\} \backslash \boldsymbol{Z}) \cup \boldsymbol{Z}) \mid (\boldsymbol{N} \cup \boldsymbol{Y})$. By the weak union property, $T \perp (\boldsymbol{V} \backslash \{T\} \backslash \boldsymbol{Z}) \mid (\boldsymbol{N} \cup \boldsymbol{Y} \cup \boldsymbol{Z})$. By the self-conditioning property, $T \perp (\boldsymbol{V} \backslash \{T\}) \mid (\boldsymbol{N} \cup \boldsymbol{Y} \cup \boldsymbol{Z})$. Equivalently, we can rewrite the previous relation as $T \perp (\boldsymbol{V} \backslash \{T\}) \mid ((\boldsymbol{N} \cup \boldsymbol{Y}) \cup (\boldsymbol{N} \cup \boldsymbol{Z}))$. Since $\boldsymbol{Z}$ and $\boldsymbol{Y}$ provide context independent equivalent information about T and by the self-conditioning property $T \perp (\boldsymbol{N} \cup \boldsymbol{Y}) \mid (\boldsymbol{N} \cup \boldsymbol{Z})$. By the contraction property, $T \perp (\boldsymbol{V} \backslash \{T\}) \mid ((\boldsymbol{N} \cup \boldsymbol{Y}) \cup (\boldsymbol{N} \cup \boldsymbol{Z}))$ and $T \perp (\boldsymbol{N} \cup \boldsymbol{Y}) \mid (\boldsymbol{N} \cup \boldsymbol{Z})$ imply that $T \perp ((\boldsymbol{V} \backslash \{T\}) \cup (\boldsymbol{N} \cup \boldsymbol{Y})) \mid (\boldsymbol{N} \cup \boldsymbol{Z})$. This is equivalent to $T \perp (\boldsymbol{V} \backslash \{T\}) \mid (\boldsymbol{N} \cup \boldsymbol{Z})$. By the decomposition property this implies that $\boldsymbol{M}_{new} = \boldsymbol{N} \cup \boldsymbol{Z}$ is also a Markov blanket of T. (Q.E.D.) ■

Proof Lemma 2 : By definition of the Markov blanket, $T \perp (\boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}) \mid \boldsymbol{M}$. By the self-conditioning property, it follows that $T \perp (\boldsymbol{V} \backslash \{T\}) \mid \boldsymbol{M}$. Since $(\boldsymbol{V} \backslash \{T\}) = (\boldsymbol{V} \backslash \{T\}) \cup \boldsymbol{M}_{new}$ and according to the weak union property, $T \perp (\boldsymbol{V} \backslash \{T\} \backslash \boldsymbol{M}_{new}) \mid (\boldsymbol{M} \cup \boldsymbol{M}_{new})$. By the self-conditioning property, it follows that $T \perp (\boldsymbol{V} \backslash \{T\}) \mid (\boldsymbol{M} \cup \boldsymbol{M}_{new})$. Since $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$ and $T \perp (\boldsymbol{V} \backslash \{T\}) \mid (\boldsymbol{M} \cup \boldsymbol{M}_{new})$, the contraction property implies that $T \perp ((\boldsymbol{V} \backslash \{T\}) \cup \boldsymbol{M}) \mid \boldsymbol{M}_{new}$. Next, since $(\boldsymbol{V} \backslash \{T\}) = (\boldsymbol{V} \backslash \{T\}) \cup \boldsymbol{M}$, it follows that $T \perp (\boldsymbol{V} \backslash \{T\}) \mid \boldsymbol{M}_{new}$. By the decomposition property this implies that $\boldsymbol{M}_{new}$ is a Markov blanket of T. (Q.E.D.) ■

Proof Theorem 6 : Given an ancestral graph $\mathbb{G} = < \boldsymbol{V}, \mathbb{E} >$, let $\boldsymbol{M}$ denote the set containing all parents and children of T and every variable X connected to T by a path from T to X in $\mathbb{G}$ such that: (i) the first edge on the path is either bi-directed or away from T, (ii) all other edges except the last are bi-directed, and (iii) the last edge is either bi-directed or is away from X. Note that spouses of T satisfy the above conditions and are therefore included in $\boldsymbol{M}$.

We first show that set $\boldsymbol{M}$ m-separates T and every other variable $Y \in \boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}$. To see this, suppose that $\boldsymbol{M}$ does not m-separate T from some variable $Y \in \boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}$. Then, there must exist a path p connecting Y and T that is not blocked by $\boldsymbol{M}$. By definition of $\boldsymbol{M}$, Y cannot be directly connected to T and not be in $\boldsymbol{M}$. Additionally, path p cannot be through parents of T, its spouses, or parents of variables connected to T or its children by bi-directed paths, because any such variable would act as a non-collider that is in $\boldsymbol{M}$ and would therefore block the path p. The only remaining possibility is for path p to contain a variable $X \in \boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}$ that is a child of a variable $Z \in \boldsymbol{M}$ that is either (i) a child of T, or (ii) connected to T by a bi-directed path, or (iii) connected to a child of T by a bi-directed path. However, in this case, variable Z would be a non-collider on path p and would therefore block it. It follows that set $\boldsymbol{M}$ m-separates T and every other variable $Y \in \boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}$.

From the definition of the global Markov condition it follows that every m-separation relation in $\mathbb{G}$ implies conditional independence in every joint probability distribution $\mathbb{P}$ that satisfies the global Markov condition for $\mathbb{G}$. Thus, we have $T \perp Y \mid \boldsymbol{M}$ in $\mathbb{P}$ for every variable $Y \in \boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}$, from which it follows that $\boldsymbol{M}$ is a Markov blanket of T. (Q.E.D.) ■

Proof Theorem 7 : First we prove that any Markov blanket of T is an optimal predictor of T. If $\boldsymbol{M}$ is a Markov blanket of T, then by definition it is the optimal predictor of T because $\mathrm{P}(T \mid \boldsymbol{M}) = \mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$ and this distribution can be accurately approximated by $\mathbb{L}$, which implies that $\mathbb{M}$ will be maximized.

Now we prove that any optimal predictor of T is a Markov blanket of T. Assume that $\boldsymbol{X} \subseteq \boldsymbol{V} \backslash \{T\}$ is an optimal predictor of T but it is not a Markov blanket of T. This implies that, $\mathrm{P}(T \mid \boldsymbol{X}) \neq \mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$. By definition, $\boldsymbol{V} \backslash \{T\}$ is always a Markov blanket of T. By first part of the theorem, $\boldsymbol{V} \backslash \{T\}$ is an optimal predictor of T similarly to $\boldsymbol{X}$. Therefore, the following should hold: $\mathrm{P}(T \mid \boldsymbol{X}) = \mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$. This contradicts the assumption that $\boldsymbol{X}$ is not a Markov blanket of T. Therefore, $\boldsymbol{X}$ is a Markov blanket of T. (Q.E.D.) ■

Proof Theorem 8 : First we prove that $\boldsymbol{M}$ is a Markov blanket of T at the end of Phase I. Suppose it is not, that is, $T \not\perp (\boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}) \mid \boldsymbol{M}$. By the local composition property with respect to T, there exists $Y \in (\boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\})$ such that $T \not\perp Y \mid \boldsymbol{M}$. This contradicts the exit condition from the loop in step 9 that states that $\boldsymbol{E}$ should be empty, which can be the case if and only if for every $Y \in (\boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\})$, $T \perp Y \mid \boldsymbol{M}$. Therefore, $\boldsymbol{M}$ is a Markov blanket of T at the end of Phase I.

Next we prove that $\boldsymbol{M}$ remains a Markov blanket of T at the end of Phase II. Assume that a variable $Y \in \boldsymbol{M}$ can be rendered independent from T by conditioning on the remaining variables in $\boldsymbol{M}$, that is, $T \perp Y \mid (\boldsymbol{M} \backslash \{Y\})$. From Phase I it follows that $T \perp (\boldsymbol{V} \backslash \boldsymbol{M} \backslash \{T\}) \mid \boldsymbol{M}$. The above two independence relations by the contraction property imply that $T \perp (\boldsymbol{V} \backslash (\boldsymbol{M} \backslash \{Y\}) \backslash \{T\}) \mid (\boldsymbol{M} \backslash \{\boldsymbol{Y}\})$. Thus, $\boldsymbol{M}$ is a Markov blanket of T at the end of Phase II of the algorithm.

Finally we prove that $\boldsymbol{M}$ is a Markov boundary of T at the end of Phase II. Suppose it is not and thus there exists $\boldsymbol{N} \subset \boldsymbol{M}$ that is a Markov blanket of T. Let $Y \in \boldsymbol{M} \backslash \boldsymbol{N}$ and $\boldsymbol{Z} \subseteq (\boldsymbol{V} \backslash \boldsymbol{N} \backslash \{T\} \backslash \{Y\})$. By definition of the Markov blanket, $T \perp (\boldsymbol{V} \backslash \boldsymbol{N} \backslash \{T\}) \mid \boldsymbol{N}$. By the decomposition property,

$T \perp (\boldsymbol{Z} \cup \{Y\}) \mid \boldsymbol{N}$. The latter independence relation implies $T \perp Y \mid (\boldsymbol{N} \cup \boldsymbol{Z})$ by the weak union property. Therefore, any variable $Y \in \boldsymbol{M} \backslash \boldsymbol{N}$ would be removed by the algorithm in step 12 which contradicts the assumption that the algorithm output $\boldsymbol{M}$ and $\boldsymbol{N} \subset \boldsymbol{M}$ is another Markov blanket of T. Therefore, $\boldsymbol{M}$ is a Markov boundary of T at the end of Phase II. (Q.E.D.) ■

Proof Theorem 9 : First we prove that the set $\boldsymbol{M}$ is a Markov blanket of T at the end of Phase I. Because of the assumptions of the theorem, there are only two reasons for existence of a subset $\boldsymbol{Z}$ that renders Y independent of T: either Y is a non-Markov boundary member or there is a violation of the intersection property that leads to context-independent information equivalence relations. The former case does not compromise the Markov blanket property of $\boldsymbol{M}$, thus we consider only the latter case. For example, we can consider the following situation $T \perp Y \mid \boldsymbol{Z}$, $T \perp \boldsymbol{Z} \mid Y$ and $T \not\perp (\{Y\} \cup \boldsymbol{Z})$ that led to removal of Y. From Lemma 1 we know that if Y is a member of some Markov blanket $\boldsymbol{M}_1 = \boldsymbol{N} \cup \{Y\}$, then $\boldsymbol{M}_2 = \boldsymbol{N} \cup \boldsymbol{Z}$ is also a Markov blanket of T because Y and $\boldsymbol{Z}$ contain context-independent equivalent information about T. Therefore the set $\boldsymbol{M}$ is a Markov blanket of T at the end of Phase I.

The proofs that $\boldsymbol{M}$ remains a Markov blanket of T at the end of Phase II and that $\boldsymbol{M}$ is a Markov boundary of T at the end of Phase II are similar to the ones in IAMB algorithm (Theorem 8) and will not be repeated here. (Q.E.D.) ■

Proof Theorem 10 : TIE* will output only Markov boundaries of T when the inputs $\mathbb{X}$ and $\mathbb{Z}$ are admissible (see Figure 7). Assume that there exists a Markov boundary $\boldsymbol{W}$ that is not output by TIE*. Because of admissibility of inputs $\mathbb{X}$ and $\mathbb{Z}$ (Figure 7), $\boldsymbol{M}_{new} = \boldsymbol{W}$ was not identified in step 5 of the algorithm. However, because of admissibility of input $\mathbb{Y}$ (Figure 7), in some iteration of the algorithm in step 4 a data set $\mathbb{D}^e$ will be generated where a Markov boundary $\boldsymbol{W}$ can be discovered by $\mathbb{X}$ in step 5. The admissibility of input $\mathbb{Z}$ implies that $\boldsymbol{W}$ will be successfully verified and output in step 6. Therefore, a contradiction is reached, and TIE* would never miss Markov boundaries. (Q.E.D.) ■

Proof Theorem 11 : Since (i) all variables from each embedded distribution belong to the original distribution, and (ii) the joint probability distribution of variables in each embedded distribution is the same as marginal in the original one, the local composition property with respect to T also holds in each embedded distribution. Therefore according to Theorem 8, IAMB will correctly identify a Markov boundary in every embedded distribution. Thus, IAMB is an admissible Markov boundary induction algorithm for TIE*. (Q.E.D.) ■

Proof Theorem 12 : The proof follows from fact that assumptions of Theorem 9 are satisfied in each embedded distribution that contains a Markov boundary of T. Thus, Semi-Interleaved HITON-PC is an admissible Markov boundary induction algorithm for TIE*. (Q.E.D.) ■

Proof Theorem 13 : The procedure IGS is executed iteratively in TIE* and generates data sets $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \backslash \boldsymbol{G})$ from the embedded distributions by removing subsets $\boldsymbol{G}$ from the full set of variables $\boldsymbol{V}$. Such procedure is admissible if it uses as $\boldsymbol{G}$ all possible subsets of $\boldsymbol{V}$. This is because eventually the procedure will generate a data set $\mathbb{D}^e$ for every Markov boundary of T such that each data set

contains all members of only one Markov boundary and thus a single Markov boundary induction algorithm $\mathbb{X}$ can discover it. By similar argument, the procedure to generate embedded distributions is admissible if it uses as $\boldsymbol{G}$ all possible subsets of all Markov boundaries. Notice that in IGS, $\boldsymbol{G}$ is constructed iteratively from all possible subsets of the previously found Markov boundaries with the following modification in order to increase efficiency of TIE* (see Section 6.2). If we find that for some subset $\boldsymbol{G}^*$ a data set $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \backslash \boldsymbol{G}^*)$ leads to a Markov boundary $\boldsymbol{M}_{new}$ in the embedded distribution (as determined in step 5 of TIE*) that is not a Markov boundary in the original distribution (as determined in step 6 of TIE*), then IGS does not consider generating data sets $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \backslash \boldsymbol{G})$ where $\boldsymbol{G}$ includes $\boldsymbol{G}^*$. Below we prove by contradiction that this modification does not compromise admissibility of IGS.

Assume that there is $\boldsymbol{W}$ that is a Markov boundary of T in the original distribution and it was not output by TIE* because $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \backslash \boldsymbol{G}^+)$ for some $\boldsymbol{G}^+ : \boldsymbol{G}^+ \supset \boldsymbol{G}^*$ has not been generated by IGS.

- Since $\boldsymbol{W}$ is a Markov blanket of T in the original distribution and $\boldsymbol{M}_{new}$ is not, Theorem 7 implies that performance of a learning algorithm $\mathbb{L}$ (that can approximate any conditional probability distribution) for prediction of T measured by the metric $\mathbb{M}$ (that is maximized only when $\mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$ is estimated accurately) is larger for $\boldsymbol{W}$ than for $\boldsymbol{M}_{new}$.
- Since $\boldsymbol{W}$ satisfies $T \perp (\boldsymbol{V} \backslash \boldsymbol{W} \backslash \{T\}) \mid \boldsymbol{W}$ by the definition of Markov blanket, decomposition property implies that $T \perp (\boldsymbol{V} \backslash \boldsymbol{W} \backslash \boldsymbol{G}^* \backslash \{T\}) \mid \boldsymbol{W}$, that is, $\boldsymbol{W}$ similarly to $\boldsymbol{M}_{new}$ is a Markov blanket of T in the embedded distribution after removal of $\boldsymbol{G}^*$. Therefore by Theorem 7, performance of a learning algorithm $\mathbb{L}$ (that can approximate any conditional probability distribution) for prediction of T measured by metric $\mathbb{M}$ (that is maximized only when $\mathrm{P}(T \mid \boldsymbol{V} \backslash \{T\})$ is estimated accurately) should be the same for $\boldsymbol{W}$ and $\boldsymbol{M}_{new}$.

The above two points are contradictory, thus $\boldsymbol{W}$ does not exist. (Q.E.D.) ■

Proof Theorem 14 : Consider that there exists a set of variables $\boldsymbol{M}_{new} \subseteq \boldsymbol{V} \setminus \{T\}$ such that $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$. Since $\boldsymbol{M}$ is a Markov boundary of T in the original distribution, it is also a Markov blanket of T in the original distribution. From Lemma 2 we know that $\boldsymbol{M}_{new}$ is a Markov blanket of T in the original distribution. Since $\boldsymbol{M}_{new}$ is a Markov boundary of T in the embedded distribution and it is a Markov blanket of T in the original distribution, it is also a Markov boundary of T in the original distribution. (Q.E.D.) ■

Proof Theorem 15 : The proof that this criterion can identify whether $\boldsymbol{M}_{new}$ is a Markov blanket of T in the original distribution or not follows from Theorem 7. If $\boldsymbol{M}_{new}$ is a Markov blanket of T in the original distribution, it is also a Markov boundary of T in the original distribution because $\boldsymbol{M}_{new}$ is a Markov boundary of T in the embedded distribution. (Q.E.D.) ■

Appendix B. Parameterizations of Example Structures

This appendix provides parameterizations of example structures from the manuscript that are shown in Tables 5, 6, 7, and 8.

Conditional probability table for the response variable T:

$P(T \mid X_1, X_{n/m+1},\ldots X_{(m-1)n/m+1})$	$(X_1= 0, X_{n/m+1} = 0,\ldots X_{(m-1)n/m+1} = 0)$	$(X_1= 0, X_{n/m+1} = 0,\ldots X_{(m-1)n/m+1} = 1)$	...	$(X_1= 1, X_{n/m+1} = 1,\ldots X_{(m-1)n/m+1} = 1)$
$T = 0$	0.2	0.8		0.2
$T = 1$	0.8	0.2		0.8

Conditional probability tables for any pair of variables X_j and X_k belonging to the same group i:

$P(X_j \mid X_k)$	$X_k = 0$	$X_k = 1$
$X_j = 0$	1.0	0.0
$X_j = 1$	0.0	1.0

Table 5: Parameterization of the Bayesian network shown in Figure 2.

$P(A)$	
$A = 0$	0.6
$A = 1$	0.4

$P(B)$	
$B = 0$	0.3
$B = 1$	0.2
$B = 2$	0.3
$B = 3$	0.2

$P(C \mid A)$	$A = 0$	$A = 1$
$C = 0$	0.0	1.0
$C = 1$	1.0	0.0

$P(F)$	
$F = 0$	0.3
$F = 1$	0.7

$P(D \mid B)$	$B = 0$	$B = 1$	$B = 2$	$B = 3$
$D = 0$	1.0	1.0	0.0	0.0
$D = 1$	0.0	0.0	1.0	1.0

$P(E \mid B)$	$B = 0$	$B = 1$	$B = 2$	$B = 3$
$E = 0$	1.0	0.0	1.0	0.0
$E = 1$	0.0	1.0	0.0	1.0

$P(T \mid C, D, E, F)$	$(C=0, D=0, E=0, F=0)$	$(C=0, D=0, E=0, F=1)$	$(C=0, D=0, E=1, F=0)$	...	$(C=1, D=1, E=1, F=1)$
$T = 0$	0.9	0.1	0.9		0.1
$T = 1$	0.1	0.9	0.1		0.9

Table 6: Parameterization of the causal Bayesian network shown in Figure 8.

Appendix C. Description and Theoretical Analysis of Prior Algorithms for Learning Multiple Markov Boundaries and Variable Sets

This appendix provides description and theoretical analysis of prior algorithms for learning multiple Markov boundaries and variable sets.

C.1 Stochastic Markov Boundary Algorithms: KIAMB

Reference: The work by Peña et al. (2007).

Description: Recall that the IAMB algorithm (Figure 4) requires only the local composition property for its correctness (per Theorem 8) which is compatible with the existence of multiple Markov boundaries of the response variable T. However, due to IAMB's reliance on a greedy deterministic strategy for adding variables into the (candidate) Markov boundary in Phase I (Forward), the algorithm can identify only a single Markov boundary of T. KIAMB addresses this

P(A)	
$A = 0$	0.6
$A = 1$	0.4

P(B)	
$B = 0$	0.9
$B = 1$	0.1

P($C \mid A$)	$A = 0$	$A = 1$
$C = 0$	0.0	1.0
$C = 1$	1.0	0.0

P(E)	
$E = 0$	0.3
$E = 1$	0.7

P($D \mid B$)	$B = 0$	$B = 1$
$D = 0$	1.0	0.0
$D = 1$	0.0	1.0

P($F \mid E$)	$E = 0$	$E = 1$
$F = 0$	0.8	0.3
$F = 1$	0.2	0.7

P($J \mid F$)	$F = 0$	$F = 1$
$J = 0$	0.7	0.7
$J = 1$	0.3	0.3

P($T \mid C$, D, F)	(C=0, D=0, F=0)	(C=0, D=0, F=1)	(C=0, D=1, F=0)	...	(C=1, D=1, F=1)
$T = 0$	0.9	0.1	0.9	...	0.1
$T = 1$	0.1	0.9	0.1	...	0.9

Table 7: Parameterization of the causal Bayesian network shown in Figure 13.

X_1: P(X_1=0) = 0.5	$X_8 = X_4$	X_{15}: P(X_{15}=0|X_{12}=0) = 0.3 P(X_{15}=0|X_{12}=1) = 0.1
X_2: P(X_2=0) = 0.5	X_9 = OR(X_5, X_6)	X_{16}: P(X_{16}=0|X_{13}=0) = 0.2 P(X_{16}=0|X_{13}=1) = 0.5
X_3: P(X_3=0) = 0.5	X_{10}: P(X_{10}=0) = 0.5	X_{17}: P(X_{17}=0|X_{13}=0) = 0.6 P(X_{17}=0|X_{13}=1) = 0.4
X_4: P(X_4=0) = 0.5	X_{11} = OR(X_7, X_8)	X_{18}: P(X_{18}=0) = 0.5
$X_5 = 1 - X_1$	X_{12}: P(X_{12}=0|X_{18}=0, X_9=0) = 0.4 P(X_{12}=0|X_{18}=0, X_9=1) = 0.5 P(X_{12}=0|X_{18}=1, X_9=0) = 0.5 P(X_{12}=0|X_{18}=1, X_9=1) = 0.6	X_{19}: P(X_{18}=0) = 0.5
$X_6 = X_2$	X_{13}: P(X_{13}=0|X_{11}=0, X_{19}=0) = 0.4 P(X_{13}=0|X_{11}=0, X_{19}=1) = 0.6 P(X_{13}=0|X_{11}=1, X_{19}=0) = 0.5 P(X_{13}=0|X_{11}=1, X_{19}=1) = 0.5	X_{20}: P(X_{20}=0|X_{12}=0) = 0.5 P(X_{20}=0|X_{12}=1) = 0.2
$X_7 = 1 - X_3$	X_{14}: P(X_{14}=0|X_{12}=0) = 0.2 P(X_{14}=0|X_{12}=1) = 0.4	X_i: P(X_i=0) = 0.5, i = 21,…,40.
T = XOR(X_9, X_{10}, X_{11})		

Table 8: Parameterization of the causal Bayesian network shown in Figure 16. All variables are binary and take values $\{0, 1\}$.

limitation of IAMB by employing a stochastic search heuristic that repeatedly disrupts the order in which variables are selected for inclusion into the Markov boundary, thereby introducing a chance of discovering alternative Markov boundaries of T. KIAMB allows the user to control the trade-off between stochasticity and greediness of the search by setting the value of a single parameter

$K \in [0,1]$. Specifically, instead of picking the conditionally maximally associated variable Y from the set $\boldsymbol{E}$ in step 4 of IAMB, in KIAMB a maximally associated variable is selected from a randomly chosen subset of all the associated variables outside the current Markov boundary $\boldsymbol{M}$. The size of this subset relative to the size of the complete set of associated variables is determined by parameter K. Setting K equal to 0 results in a purely stochastic search where a single randomly chosen associated variable is added into $\boldsymbol{M}$ on each iteration in Phase I. Setting K equal to 1 results exactly in IAMB algorithm with its greedy deterministic search.

Analysis: KIAMB correctly identifies Markov boundaries assuming the local composition property. Theoretically, KIAMB can identify all Markov boundaries if given the chance to explore a large enough number of different sequences of additions of associated variables into the current Markov boundary in Phase I. However, KIAMB is computationally inefficient, because a large fraction of its runs may yield previously identified Markov boundaries. For example, suppose the causal graph consists of 11 variables: a response variable T and variables $X_1,\ldots,X_{10}$ such that $T \leftarrow X_{10} \leftarrow X_9 \leftarrow \cdots \leftarrow X_1$ and each $X_i (i = 1,\ldots,10)$ contains equivalent information about T and is significantly associated with it. Thus, there are 10 Markov boundaries $\{X_1\},\ldots,\{X_{10}\}$ of T in this distribution. Suppose also that parameter K was set equal to 0.7, which would mean that in Phase I, KIAMB will first randomly select 7 variables out of 10 and will then select out of these 7 variables, one with the highest association with T. Because all variables in this example contain equivalent information about T, all variables will have equal association with T (Lemeire, 2007). Selection of a single variable for inclusion in the Markov boundary could then be done based on lexicographic ordering. There are 120 ways to select 7 variables out of 10, but 84 (or 70%) of such subsets of size 7 will contain variable X_1 that precedes all other variables in lexicographic ordering. Therefore, on average, we can expect 70% of the runs of KIAMB to return Markov boundary $\{X_1\}$ in this example. In order for KIAMB to identify Markov boundary $\{X_1\}$, variables X_1, X_2, X_3 must not be among the 7 randomly selected variables. On average, this would happen in only roughly 0.8% of the runs of KIAMB. Note also that in the above scenario, KIAMB will not be able to discover Markov boundaries $\{X_5\},\ldots,\{X_{10}\}$, because there is no way to select 7 variables out of 10 and avoid including at least one variable from the subset $\{X_1,\ldots,X_4\}$. KIAMB could eventually discover all 10 Markov boundaries if instead of lexicographic ordering, ties were broken by random selection, or alternatively if parameter K was set equal to a smaller value. In both of these cases, however, the probability that KIAMB will discover all 10 Markov boundaries after 10 runs is only about 0.04%, indicating that a large number of runs may be necessary to recover all 10 Markov boundaries. Thus, in order to produce the complete set of Markov boundaries, the value of parameter K and the number of runs of KIAMB must be determined based on the topology of the causal graph and the number of Markov boundaries of T, neither of which are known in real-world causal discovery applications. Finally, KIAMB suffers from the same sample inefficiency as IAMB, which arises from conditioning on the entire Markov boundary when testing variables for independence from the response variable T.

C.2 Stochastic Markov Boundary Algorithms: EGS-CMIM and EGS-NCMIGS

Reference: The work by Liu et al. (2010b).

Description: These algorithms attempt to extract multiple Markov boundaries by repeatedly invoking single Markov boundary extraction methods CMIM (Fleuret, 2004) and NCMIGS (Liu et al., 2010b), respectively. Conceptually, CMIM and NCMIGS are very similar and differ primarily in the

types of measures of association between variables. Both methods employ only a greedy forward selection strategy similar to Phase I of IAMB and rely on mutual information-based functions for measuring (conditional) association between variables and the response T. The algorithmic framework of CMIM and NCMIGS is as follows. First, all variables are ordered by decreasing association with the response T. A Markov boundary $\boldsymbol{M}$ is initialized to be the empty set. The t-th highest associated variable (where t is a user-defined parameter) is then added into $\boldsymbol{M}$ and an iterative addition of other variables begins. On each iteration, a new variable that maximizes the value of a selection criterion $J(X)$ (discussed below) is added to the Markov boundary $\boldsymbol{M}$. The algorithm stops once a termination condition is reached. CMIM terminates when the Markov boundary reaches a user-defined size k. NCMIGS offers two different stopping criteria that the user could choose from. The first stopping criterion is the same as in CMIM controlled by the parameter k. The other termination criterion alleviates the requirement of explicitly specifying the size of the Markov boundary and forces iterative selection to stop if the value of the selection criterion $J(X)$ changes from one iteration to the next by no more than δ (a user-defined parameter), that is, if $\mid J(X_i) - J(X_{i-1}) \mid \leq \delta$, where X_i denotes the variable selected for addition into $\boldsymbol{M}$ on the i-th iteration of NCMIGS.

CMIM employs an approximation to the conditional mutual information $I(X,T \mid \boldsymbol{M})$ as the selection criterion $J_{CMIM}(X)$ for adding variables into the Markov boundary. The approximation is achieved by conditioning on a single variable instead of the entire Markov boundary $\boldsymbol{M}$ (as in KIAMB), that is, $J_{CMIM}(X) = argmin_{Y \in \boldsymbol{M}} I(X,T \mid Y)$. NCMIGS uses a very similar selection criterion that is based on a *normalized* conditional mutual information $J_{NCMIGS}(X) = argmin_{Y \in \boldsymbol{M}} I(X,T \mid Y)/H(X,T)$, where $H(X,T)$ denotes the joint entropy of variable X and response T. Conditioning on a single variable instead of the entire Markov boundary makes CMIM and NCMIGS sample efficient by circumventing the problem of exponential growth in the number of parameters and sample size required for estimating the conditional mutual information $I(X,Y \mid \boldsymbol{M})$ in discrete data as the size of the Markov boundary $\boldsymbol{M}$ increases.

Analysis: Recall that EGS-CMIM (EGS-NCMIGS) extracts multiple Markov boundaries by calling CMIM (NCMIGS) with different values of the input parameter $t = 1, \ldots, l$, where l is a user-defined parameter that bounds from above the total number of Markov boundaries that will be output. Therefore, EGS-CMIM and EGS-NCMIGS require prior knowledge/estimate of the number of Markov boundaries. Note that while admissible values of t (and therefore of l) are by design bounded from above by the number of variables in the data, the actual number of true Markov boundaries may be much higher. There is also no guarantee that different values of t will yield different Markov boundaries, which makes these methods computationally inefficient (similarly to KIAMB). In addition, because CMIM and NCMIGS implement only forward selection and employ conditioning on a single variable, these methods are prone to inclusion of false positives in their output. False positives may enter a Markov boundary for two reasons: (i) when more than one variable from the current Markov boundary is required to establish independence of the response T from some other variable being considered for addition into the Markov boundary, and (ii) when some of the variables added into the Markov boundary are independent of the response T conditional on variables that were added in later iterations. Furthermore, the stopping criteria in CMIM and NCMIGS are heuristic, which may lead to an arbitrary number of false negatives in the output. This may happen, for instance, if the value of parameter k (size of a Markov boundary) is set smaller than the true size of the Markov boundary. The alternative stopping criterion of NCMIGS does not fully solve the problem of false negatives, because the absolute difference $\mid J(X_i) - J(X_{i-1}) \mid$ may be small, while the individual values $J(X_i)$ and $J(X_{i-1})$ of the selection criterion may still be large

indicating that the considered variable X_i is highly associated with the response T and that it may be too early to stop. In summary, EGS-CMIM and EGS-NCMIGS offer no formal guarantees of neither correctness nor completeness of their output, require prior knowledge/estimate of the number of Markov boundaries and their size, are computationally inefficient, but are sample efficient.

C.3 Variable Grouping Followed by Random Sampling of Variables from Each Group: EGSG

Reference: The work by Liu et al. (2010).

Description: EGSG uses normalized version of mutual information $J_{EGSG}(X,Y) = I(X,Y)/H(X,Y)$ for measuring pair-wise association between variables and partitions them into disjoint groups. Each group has a "centroid", which is the first variable that formed the group. A variable X is added into a group if (i) X has a higher association with the groups centroid C than with the response T (i.e., if $J_{EGSG}(X,C) \geq J_{EGSG}(X,T)$), and (ii) X has lower association with T than does C (i.e., if $J_{EGSG}(C,T) \geq J_{EGSG}(X,T)$). If no such group is found, then a new group is created with X as the groups centroid. Variables within a group are implicitly assumed to carry similar information about T. Under this assumption, it is sufficient to select one variable from each group to form a Markov boundary of T. In EGSG, one of the top t variables most associated with the response T is sampled at random from each group to form a single Markov boundary. Here, the value of parameter t is given by the user. In order to extract multiple Markov boundaries, the above sampling is repeated a number of times determined by the user.

Analysis: From the point of view of soundness and completeness, EGSG suffers from two major drawbacks. First, the number of Markov boundaries output by EGSG is an arbitrary parameter and is independent of the data-generating causal graph. Second, Markov boundaries output by EGSG may contain an arbitrary number of false positives as well as false negatives. False positives may appear, for instance, if a variable from one group is independent of the response T *conditional* on a variable from another group. EGSG does not test for conditional independence and could include both variables in a Markov boundary. Moreover, since only one variable is sampled from each group, false negatives may appear in the output of EGSG if several variables within a group in reality belong to the same Markov boundary. Therefore, no guarantees can be made regarding the correctness and completeness of the output of EGSG. The method is not computationally efficient for discovery of distinct Markov boundaries, because EGSG may produce the same Markov boundary multiple times due to random sampling of variables from each group. However, its computationally efficiency can be improved by constructing Markov boundaries from the Cartesian product of top-t members of each group. EGSG is sample efficient, because it does not conduct any conditional independence tests, but only computes pair-wise associations between variables.

C.4 Resampling-based Methods: Resampling+RFE and Resampling+UAF

Reference: The work by Ein-Dor et al. (2005), Michiels et al. (2005), Roepman et al. (2006) and Statnikov and Aliferis (2010a).

Description: In resampling-based methods, multiple variable sets are extracted by repeatedly applying a variable selection method to different bootstrap samples of the data (Ein-Dor et al., 2005; Michiels et al., 2005; Roepman et al., 2006; Statnikov and Aliferis, 2010a). The two variable selection methods employed in the resampling framework in this paper are Univariate Association Filtering (UAF) (Hollander and Wolfe, 1999; Statnikov et al., 2005) and Recursive Feature Elim-

ination (RFE) (Guyon et al., 2002). These methods implement only the backward selection akin to Phase II of IAMB. Namely, given a bootstrap sample, all variables are first ordered by decreasing association with the response T. UAF orders variables using p-values and test statistics from Kruskal-Wallis non-parametric ANOVA (Hollander and Wolfe, 1999). RFE orders variables by decreasing absolute values of the SVM weights (Guyon et al., 2002). Once all variables have been ordered, a portion of the least significant variables is removed, performance of the remaining variables for predicting the response T is evaluated, and this variable elimination process is recursively applied to the remaining variables. The smallest nested subset of variables with the maximum predictive performance is then output. The proportion of variables to be removed on each iteration is controlled by a user-defined parameter called "reduction coefficient".[16] Assessment of predictive performance can be performed by training and evaluating a classifier model (e.g., SVM). One can also use variants of UAF and RFE, where the smallest nested subset of variables with predictive performance statistically indistinguishable from the nominally maximum predictive performance is output. This often produces smaller variable sets than the former approach.

Analysis: Neither UAF nor RFE, which are at the core of resampling-based methods, offer formal guarantees of the correctness of their output, because both methods are based on a heuristic approach to finding the most predictive subset of variables and not the Markov boundary (Aliferis et al., 2010b). Therefore, neither Resampling+UAF nor Resampling+RFE are sound and complete for extraction of multiple Markov boundaries. Resampling+UAF and Resampling+RFE are also computationally inefficient, because runs of UAF and RFE on different bootstrap samples may produce identical variable sets, especially when the sample size is large. In addition, the number of runs is a user-defined parameter that requires prior knowledge of the number of Markov boundaries in the data. Both resampling techniques are sample efficient, because UAF does not rely on conditional independence tests and because RFE leverages SVMs regularized loss function that allows for parameter estimation in high-dimensional data with small sample sizes.

C.5 Iterative Removal Methods: IR-HITON-PC and IR-SPLR

Reference: The work by Natsoulis et al. (2005) and Statnikov and Aliferis (2010a).

Description: Iterative removal methods identify multiple Markov boundaries (IR-HITON-PC) or multiple variable sets (IR-SPLR) by repeatedly executing the following two steps. Step 1: Extract a Markov boundary/variable set $\boldsymbol{M}$ from the current set $\boldsymbol{W}$ of variables (initially $\boldsymbol{W} = \boldsymbol{V} \setminus \{T\}$). Step 2: If $\boldsymbol{M}$ is the first Markov boundary/variable set extracted or if its predictive performance is statistically indistinguishable from performance of the first Markov boundary/variable set, then output $\boldsymbol{M}$, remove all of its variables from further consideration ($\boldsymbol{W} \leftarrow \boldsymbol{W} \setminus \boldsymbol{M}$) and go to Step 1. Otherwise, terminate. IR-HITON-PC uses Semi-Interleaved HITON-PC as the base Markov boundary extraction method. IR-SPLR extracts variable sets using regularized Logistic regression with a L_1 norm penalty term, which induces sparsity in the regression coefficients. All variables with non-zero coefficients are taken to belong to an output variable set.

Analysis: IR-HITON-PC is correct because it uses Semi-Interleaved HITON-PC to identify Markov boundaries (per Theorem 9). On the other hand, IR-SPLR relies on a heuristic regression-based approach to finding the most predictive subset of variables and not the Markov boundary; thus this method has no theoretical guarantees for correct identification of Markov boundaries. Furthermore, neither iterative removal method is guaranteed to be complete, because these methods

16. Reduction coefficient $= 1.2$ means that every iteration retains $1/1.2 = 83\%$ of variables from the previous iteration.

An example of instantiated algorithm **TIE***

Inputs: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T.

Output: all Markov boundaries of T that exist in $\mathbb{P}$.

1. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}$ of T from the dataset $\mathbb{D}$ for variables $\boldsymbol{V}$ (i.e., in the original distribution $\mathbb{P}$)
2. Output $\boldsymbol{M}$
3. Repeat
4. Generate a dataset $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ from the embedded distribution by removing from the full set of variables $\boldsymbol{V}$ in the original distribution the smallest subset $\boldsymbol{G}$ of the so far discovered Markov boundaries of T such that:
   (i) $\boldsymbol{G}$ was not considered in the previous iterations of this step, and
   (ii) $\boldsymbol{G}$ does not include any subset of variables that was previously removed from $\boldsymbol{V}$ to yield a dataset $\mathbb{D}^e$ when $\boldsymbol{M}_{new}$ was found not to be a Markov boundary of T in the original distribution (per step 6)
5. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}_{new}$ of T from the dataset $\mathbb{D}^e$ (i.e., in the embedded distribution)
6. If the holdout validation estimate of predictivity of T for the SVM classifier model induced from data $\mathbb{D}$ using variables $\boldsymbol{M}_{new}$ is not statistically worse than the respective predictivity estimate for variables $\boldsymbol{M}$, then $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution and it is output by the algorithm
7. Until all datasets $\mathbb{D}^e$ generated in step 4 have been considered.

Figure 17: An example of instantiated TIE* algorithm. This algorithm was used in experiments with real data in Section 5.2.

output disjoint Markov boundaries or variable sets, while in general multiple Markov boundaries may share a number of variables. IR-HITON-PC and IR-SPLR neither require prior knowledge of the number of Markov boundaries nor their size, and these methods are computationally and sample efficient.

Appendix D. Details about the TIE* Algorithm

This appendix provides details about the generative TIE* algorithm.

D.1 Example Instantiations of the Generative Algorithm

Example instantiations of the generative algorithm TIE* are given in Figures 17 and 18.

D.2 Specific Implementation Details

We proceed below with details about TIE* implementations. We discuss Markov boundary induction algorithm ($\mathbb{X}$), procedure to generate data sets from the embedded distributions ($\mathbb{Y}$), and criterion to verify Markov boundaries of T ($\mathbb{Z}$).

Markov boundary induction algorithm IAMB (Figure 4): We used the Matlab implementation of the algorithm from the Causal Explorer toolkit (Aliferis et al., 2003b; Statnikov et al., 2010).

An example of instantiated algorithm **TIE***

Inputs: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a response variable T.

Output: all Markov boundaries of T that exist in $\mathbb{P}$.

1. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}$ of T from the dataset $\mathbb{D}$ for variables $\boldsymbol{V}$ (i.e., in the original distribution $\mathbb{P}$)
2. Output $\boldsymbol{M}$
3. Repeat
4. Generate a dataset $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ from the embedded distribution by removing from the full set of variables $\boldsymbol{V}$ in the original distribution the smallest subset $\boldsymbol{G}$ of the so far discovered Markov boundaries of T such that:
   (i) $\boldsymbol{G}$ was not considered in the previous iterations of this step, and
   (ii) $\boldsymbol{G}$ does not include any subset of variables that was previously removed from $\boldsymbol{V}$ to yield a dataset $\mathbb{D}^e$ when $\boldsymbol{M}_{new}$ was found not to be a Markov boundary of T in the original distribution (per step 6)
5. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}_{new}$ of T from the dataset $\mathbb{D}^e$ (i.e., in the embedded distribution)
6. If $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$, then $\boldsymbol{M}_{new}$ is a Markov boundary of T in the original distribution and it is output by the algorithm
7. Until all datasets $\mathbb{D}^e$ generated in step 4 have been considered.

Figure 18: An example of instantiated TIE* algorithm. This algorithm was used in experiments with simulated data in Section 5.1.

When the algorithm was run on discrete data, we assessed independence of variables with G^2 test at significance level $\alpha = 0.05$. In our implementation of G^2 test, we required at least 5 samples per cell in the contingency tables. For continuous data, one can use Fishers Z test to assess independence of variables. To measure association Association$(T, X \mid \boldsymbol{M})$ in step 4 of the algorithm we used negative p-values returned by the corresponding test of independence $T \perp X \mid \boldsymbol{M}$.[17] Since the IAMB algorithm can be run multiple times in TIE*, we programmed on top of the Causal Explorer code a caching method to store and retrieve results of conditional independence tests.

Markov boundary induction algorithm Semi-Interleaved HITON-PC (Figure 5): We used the Matlab implementation of the algorithm from the Causal Explorer toolkit (Aliferis et al., 2003b; Statnikov et al., 2010). Semi-Interleaved HITON-PC was implemented without so-called "symmetry correction" (Aliferis et al., 2010a). Similarly to IAMB, to assess independence of variables in discrete data we used G^2 test at $\alpha = 0.05$, and one can use Fisher's Z test for continuous data. To measure Association(T, X) in step 4 of the algorithm, we used negative p-values returned by the corresponding test of independence $T \perp X$. The parameter *max-k* which denotes the upper bound on the size of the conditioning set in Semi-Interleaved HITON-PC (i.e., the maximum size of the subset $\boldsymbol{Z}$ in steps 6 and 10 of the algorithm) was set equal to 3. The choice of this value for *max-k* parameter is justified by empirical performance in a variety of data distributions, as well as by sample size limitations in our data (Aliferis et al., 2010a,b). Since the Semi-Interleaved HITON-PC

17. For the Fishers Z test and G^2 test, p-value is inversely related to the test statistic, given a fixed degree of freedom. Thus, larger test statistics correspond to smaller p-values, and vice-versa.

algorithm can be run multiple times in TIE*, we programmed on top of the Causal Explorer codes a caching method to store and retrieve results of conditional independence tests.

Procedures IGS-Lex, IGS-MinAssoc, and IGS-MaxAssoc to generate data sets from the embedded distributions (Figure 9): These procedures were implemented by (i) constructing all subsets $\boldsymbol{G}$ such that $\{\boldsymbol{G}_i\} \subset \boldsymbol{G} \subseteq \{\boldsymbol{M}_i \cup \boldsymbol{G}_i\}$ and $|\boldsymbol{G}| \leq$ parameter *max-card*, (ii) excluding subsets that either include $\boldsymbol{G}_j^*$ or coincide with $\boldsymbol{G}_k$, (iii) considering first subsets with the smallest number of variables, and (iv) using a subset $\boldsymbol{G}$ with the either smallest lexicographical order of variables, or minimum association with T, or maximum association with T (depending on the employed procedure). The association with T was assessed with the appropriate statistical test, as described above for the Markov boundary induction algorithms. The parameter *max-card* was set equal to 4 in all experiments except for experiments with simulated data where it was set equal to 8. The purpose of this parameter is to trade off completeness of the TIE* output for execution speed. We also experimented with larger values of *max-card* until no more new Markov boundaries can be obtained.

Criterion Independence to verify Markov boundaries (Figure 10): This criterion was implemented using statistical tests that were described above for the Markov boundary induction algorithms. Since the Markov boundary in the original distribution ($\boldsymbol{M}$) and the examined Markov boundary in the embedded distribution ($\boldsymbol{M}_{new}$) are often significantly overlapping, we used a sample efficient implementation where we do not need to condition on the entire Markov boundary in the embedded distribution $\boldsymbol{M}_{new}$. Consider that $\boldsymbol{M} \cap \boldsymbol{M}_{new} = \boldsymbol{W}$, $\boldsymbol{M} \backslash \boldsymbol{M}_{new} = \boldsymbol{S}_1$, and $\boldsymbol{M}_{new} \backslash \boldsymbol{M} = \boldsymbol{S}_2$. Then context-independent information equivalence of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ implies information equivalence of $\boldsymbol{M} = \boldsymbol{S}_1 \cup \boldsymbol{W}$ and $\boldsymbol{M}_{new} = \boldsymbol{S}_2 \cup \boldsymbol{W}$. Therefore, it suffices to verify that $T \perp \boldsymbol{S}_1 \mid \boldsymbol{S}_2$ and $T \perp \boldsymbol{S}_2 \mid \boldsymbol{S}_1$ instead of $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$. This was the essence of our implementation of the Independence criterion for Markov boundary verification.

Criterion Predictivity to verify Markov boundaries (Figure 11): As a learning algorithm $\mathbb{L}$, we used linear support vector machines (SVMs) with default value of the penalty parameter $C = 1$ (Fan et al., 2005; Vapnik, 1998). As a performance metric $\mathbb{M}$, we used area under ROC curve (AUC) (Fawcett, 2003) and weighted accuracy (Guyon et al., 2006) for binary and multiclass responses, respectively. We estimated classification performance (using either AUC or weighted accuracy) by holdout validation (Weiss and Kulikowski, 1991), whereby $2/3$ of data samples were used for Markov boundary induction and classifier training and remaining $1/3$ for classifier testing. Statistical comparison of AUC estimates was performed using DeLong's test at $\alpha = 0.05$ (DeLong et al., 1988) and comparison of weighted accuracy estimates was performed by permutation-based testing with 10,000 permutations of the vectors of classifier predictions (Good, 2000). We also experimented with other SVM kernels and parameters in the criterion Predictivity, but the final results were similar because SVMs are used here only for *relative* assessment of the classifier performance (i.e., to compare performance of the Markov boundary $\boldsymbol{M}$ from the original distribution with performance of the new Markov boundary $\boldsymbol{M}_{new}$ from the embedded distribution). Final assessment of the classifier performance for induced Markov boundary variables was carried out using SVMs with polynomial kernel and parameters C and degree d optimized by holdout validation or cross-validation, as described in Section 5.

Appendix E. Additional Information about Empirical Experiments

This appendix provides additional information about empirical experiments.

E.1 Parameterizations of Methods in Empirical Experiments

Parameterizations of methods in empirical experiments are given in Table 9.

E.2 On Computation of Performance Criteria in Experiments with Simulated Data

Since the number of distinct Markov boundaries/variable sets extracted by a given method in our evaluation may differ from the number of true Markov boundaries in the causal graph, it is necessary to establish a matching between the true Markov boundaries and the extracted Markov boundaries/variable sets before computing values of criteria III-V. This matching was performed by finding a minimum-weight matching in a complete bipartite graph $\mathbb{G} =< \boldsymbol{V}_1 \cup \boldsymbol{V}_2, \boldsymbol{E} >$, where vertices in $\boldsymbol{V}_1$ corresponded to the true Markov boundaries and vertices in $\boldsymbol{V}_2$ corresponded to the extracted Markov boundaries/variable sets. The weight of an edge $(u,v) \in \boldsymbol{E}, u \in \boldsymbol{V}_1, v \in \boldsymbol{V}_2$, was set equal to the sum of PFP and FNR that would have resulted from matching the true Markov boundary u with the extracted Markov boundary/variable set v. The extracted Markov boundaries/variable sets that were not matched to any true Markov boundary did not participate in the computation of criteria III-V. A limitation of this approach to evaluation of different methods is that methods that are parameterized to produce a number of Markov boundaries/variable sets that is much larger than the number of true Markov boundaries could potentially show better performance on criteria III-V than methods/parameterizations that output only a few Markov boundaries/variable sets. In order to control for this effect, whenever a method allowed it, some of its parameterizations were targeted towards producing the same "large" number of Markov boundaries/variable sets (5,000 in our case). In addition, since the true Markov boundaries are unknown in practical applications, the average classification performance (criterion VI) was computed over *all* distinct Markov boundaries/variable sets extracted by a method. This way of computing the average classification performance, in a sense, counteracts the potential bias in criteria III-V towards methods that produce large numbers of Markov boundaries/variable sets, since if many of the extracted Markov boundaries/variable sets do not contain the variables truly relevant to prediction of T (i.e., members of its true Markov boundaries), the classification performance may suffer.

E.3 Additional Discussion of the Results of Experiments with Simulated Data

KIAMB did not identify any true Markov boundaries exactly due to this method's sample inefficiency arising from conditioning on the entire (candidate) Markov boundary. The average classification performance of Markov boundaries extracted by KIAMB was about 20% lower than of the MAP-BN classifier in both data sets.

Performance of EGS-NCMIGS and EGS-CMIM was very similar and varied widely depending on parameterization, with the average PFP ranging from 29% to 76% and average FNR ranging from 0% to 27% in $TIED$. The high ends (i.e., worse results) of these measures increased to 95% PFP and 51% FNR in $TIED1000$ demonstrating the sensitivity of these methods to the presence of irrelevant variables in the data. The alternative stopping criterion of EGS-NCMIGS helped reduce the PFP relative to other parameterizations, but failed to reduce the FNR. The other stopping criterion that requires the size K of Markov boundaries to be specified, was able to achieve 0% FNR in $TIED$ (for large enough K; see Table 10). This suggests that, even though the alternative stopping criterion has the advantage of not requiring prior knowledge of the size of Markov boundaries, it makes EGS-NCMIGS susceptible to premature termination as discussed in Appendix C. The aver-

age classification performance of Markov boundaries extracted by EGS-NCMIGS and EGS-CMIM was statistically comparable to the MAP-BN classification performance for all parameterizations except those with $K = 50$ in both data sets and also for $(l = 7, \delta = 0.015)$ in $TIED1000$. The reduction in classification performance relative to MAP-BN reached as high as 10% and was due to presence of false positives and false negatives in the extracted Markov boundaries.

EGSG proved to be extremely sensitive to the presence of irrelevant variables with PFP and FNR increasing across all parameterizations from highs of 55% PFP and 37% FNR in $TIED$ to uniformly above 93% PFP and high of 78% FNR in $TIED1000$. In addition, the average size of Markov boundaries extracted by EGSG increased almost 10-fold, from 7 in $TIED$ to 67 in $TIED1000$, while the number of variables conditionally dependent on T in the underlying network remained unchanged. Consistent with the theoretical analysis in Appendix C, these results demonstrate the lack of control for false positives as well as false negatives in the output of EGSG. Classification performance was sensitive to the values of parameter t, with increasing values resulting in degradation of classification performance in both data sets, which is due to the fact that as t increases, Markov boundaries extracted by EGSG increasingly resemble subsets of randomly selected variables from the complete set of variables. Classification performance of Markov boundaries extracted by EGSG was lower than performance of the MAP-BN classifier by 9-23% (depending on parameter settings) in $TIED$ and by 27-55% in $TIED1000$. In addition, classification performance in $TIED1000$ was lower than in $TIED$ uniformly across all parameterizations of EGSG.

Variable sets extracted by Resampling+UAF were 24-50% larger than those found by Resampling+RFE, which helped Resampling+UAF reach slightly lower FNR (by 3-6% in $TIED$ and by 2-4% in $TIED1000$), but also resulted in significantly higher PFP (by about 42-46% in $TIED$ and by 30-35% in $TIED1000$). The larger size of the extracted variable sets and higher PFP are likely due to UAF's ranking of variables based solely on univariate association with the response T, whereas RFE's ranking is "multivariate" in a sense that it takes into account not only each variable's individual classification performance, but also the information that other variables in the current nested subset carry about T (Guyon et al., 2002). In fact, Resampling+RFE produced more compact variable sets than Resampling+UAF in every simulated and real data set considered in this study. In simulated data, parameterizations of Resampling+RFE and Resampling+UAF with statistical comparison of classification performance estimates produced variable sets that were on average 60-70% smaller than those found by parameterizations without statistical comparison, resulting in about 20% decreases in PFP, but causing roughly 30-36% increases in FNR. The average classification performance of variable sets extracted by Resampling+RFE in simulated data was statistically indistinguishable from Resampling+UAF with similar parameterizations. The average classification performance of both methods parameterized without statistical comparison was comparable with performance of the MAP-BN classifier in $TIED$ and was slightly lower (by about 1-2%) in $TIED1000$. Parameterizations with statistical comparison underperformed the MAP-BN classifier by about 2-3% in both data sets. The results in both simulated data sets also show that the number of *distinct* variable sets out of the 5,000 extracted by each parameterization of Resampling+RFE and Resampling+UAF ranged from 0.24% to 50%, and hence roughly 99% to 50% of computational resources were spent retrieving the same variable sets multiple times.

IR-HITON-PC was able to identify *exactly* only a single true Markov boundary in both data sets. This was a direct consequence of a violation of the iterative removal's underlying assumption that the true Markov boundaries are disjoint sets of variables. All true Markov boundaries in $TIED$ and $TIED1000$ share variable X_{10}. However, once that variable was found to be in a Markov boundary

by an iterative removal method, it was then removed from further consideration thus preventing all other extracted Markov boundaries from containing this variable. Markov boundaries extracted by IR-HITON-PC had 8-10% PFP (depending on the data set) and 10-20% FNR. The low PFP was due to Semi-Interleaved HITON-PC's built-in control for the false discovery rate (Aliferis et al., 2010b), while the high FNR was a consequence of the iterative removal scheme. As a result of high FNR in $TIED$, the average classification performance of Markov boundaries extracted by IR-HITON-PC was about 2% lower than of the MAP-BN classifier in the same data set. The FNR was lower in $TIED1000$ than in $TIED$, which resulted in classification performance becoming statistically comparable with the MAP-BN performance.

IR-SPLR was not able to identify any true Markov boundaries exactly in neither $TIED$ or $TIED1000$. Each parameterization of IR-SPLR extracted only one variable set in both simulated data sets. Variable sets extracted by IR-SPLR in simulated data were 4-6 times larger than those found by IR-HITON-PC, which resulted in about 60-70% increase in PFP (depending on the data set), but zero FNR. The PFP of IR-SPLR did not increase significantly in $TIED1000$ relative to $TIED$, which demonstrates the often-cited benefit of the L_1-norm regularization, that is, its ability to exclude irrelevant variables from the model. Classification performance of the extracted variable sets was statistically comparable to the MAP-BN performance in $TIED$ and was about 2% lower in $TIED1000$ due to an increase in PFP.

E.4 Real Data sets Used in the Experiments

The list of real data sets used in the experiments is given in Table 12.

Method	*Parameterizations*	*References*
NOVEL		
TIE*	• Semi-Interleaved HITON-PC (without symmetry correction) with α = 0.05 and *max-k* = 3 was used for identification of Markov boundaries. Procedure IGS-Lex was used for generating data sets from the embedded distributions. Criteria Independence and Predictivity were used for verifying Markov boundaries in simulated and real data, respectively. See Appendix D for details.	Extended from Statnikov and Aliferis (2010a)
iTIE*	• $\alpha = 0.05$, *max-k* $= 3$.	Novel method
STOCHASTIC MARKOV BOUNDARY DISCOVERY		
KIAMB	• $\sharp$ of runs $= 5,000, \alpha = 0.05, K = 0.7$ • $\sharp$ of runs $= 5,000, \alpha = 0.05, \underline{K = 0.8}$ • $\sharp$ of runs $= 5,000, \alpha = 0.05, K = 0.9$	Peña et al. (2007)
EGS-CMIM	• $l = 7, \underline{K = 10}$ • $l = 7, \underline{K = 50}$ • $l = 5,000, \underline{K = 10}$ • $l = 5,000, \underline{K = 50}$	Liu et al. (2010b)
EGS-NCMIGS	• $\underline{l = 7, \delta = 0.015}$ • $l = 7, K = 10$ • $l = 7, K = 50$ • $l = 5000, \underline{\delta = 0.015}$ • $l = 5000, K = 10$ • $l = 5000, K = 50$	
VARIABLE GROUPING-BASED MARKOV BOUNDARY DISCOVERY		
EGSG	• $\sharp$ of Markov boundaries $= 30, t = 15$ • $\sharp$ of Markov boundaries $= 30, t = 10$ • $\sharp$ of Markov boundaries $= 30, t = 5$ • $\sharp$ of Markov boundaries $= 5000, \underline{t = 15}$ • $\sharp$ of Markov boundaries $= 5000, t = 10$ • $\sharp$ of Markov boundaries $= 5000, t = 5$	Liu et al. (2010)
RESAMPLING-BASED VARIABLE SELECTION		
Resampling+ RFE	• w/o statistical comparison of classification performance estimates • with statistical comparison at significance level = 0.05 All configurations used 5,000 bootstrap samples and a reduction coefficient of 1.2. Statistical comparison of classification performance estimates was performed using permutation-based testing (with 10,000 permutations) for weighted accuracy (Good, 2000) and DeLong's test (DeLong et al., 1988) for AUC.	Ein-Dor et al. (2005); Michiels et al. (2005); Roepman et al. (2006); Statnikov and Aliferis (2010a)
Resampling+ UAF	• w/o statistical comparison of classification performance estimates • with statistical comparison at significance level $\alpha = 0.05$ All configurations used 5,000 bootstrap samples and a reduction coefficient of 1.2. The same tests as in Resampling+RFE were used for statistical comparisons.	
ITERATIVE REMOVAL FOR VARIABLE SELECTION AND MARKOV BOUNDARY DISCOVERY		
IR-HITON-PC	• max-k $= 3, \alpha = 0.05$ This method runs Semi-Interleaved HITON-PC without symmetry correction. The same tests as in Resampling+RFE were used for statistical comparisons.	Natsoulis et al. (2005); Statnikov and Aliferis (2010a)
IR-SPLR	• w/o statistical comparison of classification performance estimates • with statistical comparison at significance level $\alpha = 0.05$ The regularization coefficient λ for each SPLR model was determined by holdout validation in training data. The same tests as in Resampling+RFE were used for statistical comparisons.	

Table 9: Parameterizations of methods for discovery of multiple Markov boundaries and variable sets. Parameter settings that have been recommended by the authors of prior methods are underlined.

Method		*I. Number of distinct MBs or VSs*	*II. Average size of extracted distinct MBs or VSs*	*III. Number of true MBs identified exactly*	*IV. Average proportion of false positives*	*V. Average false negative rate*	*VI. Weighted accuracy over all extracted MBs or VSs*		
							Average	*95% Interval*	
TIE*	*max-k* = 3, $\alpha = 0.05$	72	5.0	72	0.000	0.000	**0.951**	0.938	0.965
iTIE*	*max-k* = 3, $\alpha = 0.05$	72	5.0	72	0.000	0.000	**0.951**	0.938	0.965
KIAMB	Number of runs = 5000, $\alpha = 0.05$, $K = 0.7$	377	2.8	0	0.000	0.400	0.727	0.479	0.946
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.8$	377	2.8	0	0.000	0.400	0.727	0.479	0.946
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.9$	377	2.8	0	0.000	0.400	0.727	0.479	0.946
EGS-NCMIGS	$l = 7$, $\delta = 0.015$	6	7.0	0	0.286	0.000	**0.964**	0.963	0.965
	$l = 7$, $K = 10$	6	10.0	0	0.500	0.000	**0.964**	0.963	0.965
	$l = 7$, $K = 50$	6	21.0	0	0.762	0.000	0.941	0.937	0.943
	$l = 5000$, $\delta = 0.015$	24	7.3	0	0.469	0.267	**0.954**	0.843	0.967
	$l = 5000$, $K = 10$	20	10.0	0	0.610	0.220	**0.964**	0.954	0.970
	$l = 5000$, $K = 50$	9	21.0	0	0.762	0.000	0.944	0.937	0.954
EGS-CMIM	$l = 7$, $K = 10$	6	10.0	0	0.500	0.000	**0.963**	0.963	0.965
	$l = 7$, $K = 50$	6	21.0	0	0.762	0.000	0.939	0.937	0.942
	$l = 5000$, $K = 10$	20	10.0	0	0.595	0.190	**0.963**	0.951	0.969
	$l = 5000$, $K = 50$	9	21.0	0	0.762	0.000	0.943	0.937	0.954
EGSG	Number of Markov boundaries = 30, $t = 5$	30	7.0	0	0.476	0.267	0.840	0.605	0.968
	Number of Markov boundaries = 30, $t = 10$	30	7.0	0	0.548	0.367	0.722	0.379	0.962
	Number of Markov boundaries = 30, $t = 15$	30	7.0	0	0.548	0.367	0.722	0.379	0.962
	Number of Markov boundaries = 5,000, $t = 5$	1,997	7.0	0	0.286	0.000	0.863	0.620	0.965
	Number of Markov boundaries = 5,000, $t = 10$	3,027	7.0	0	0.286	0.000	0.774	0.500	0.965
	Number of Markov boundaries = 5,000, $t = 15$	3,027	7.0	0	0.286	0.000	0.774	0.500	0.965
Resampling+RFE	without statistical comparison	1,374	14.9	1	0.397	0.058	**0.955**	0.932	0.979
	with statistical comparison ($\alpha = 0.05$)	188	4.9	0	0.171	0.378	0.930	0.917	0.967
Resampling+UAF	without statistical comparison	184	20.8	0	0.752	0.000	**0.953**	0.934	0.966
	with statistical comparison ($\alpha = 0.05$)	19	8.4	0	0.592	0.347	0.930	0.917	0.938
IR-HITON-PC	*max-k* = 3, $\alpha = 0.05$	3	4.3	1	0.083	0.200	0.946	0.936	0.965
IR-SPLR	without statistical comparison	1	26.0	0	0.808	0.000	**0.958**	0.958	0.958
	with statistical comparison ($\alpha = 0.05$)	1	17.0	0	0.706	0.000	**0.959**	0.959	0.959

Table 10: Results obtained in simulated data set *TIED*. "MB" stands for "Markov boundary", and "VS" stands for "variable set". The 95% interval for weighted accuracy denotes the range in which weighted accuracies of 95% of the extracted Markov boundaries/variable sets fell. Classification performance of the MAP-BN classifier in the same data sample was 0.966 weighted accuracy. Highlighted in bold are results that are statistically comparable to the MAP-BN classification performance.

Method		*I. Number of distinct MBs or VSs*	*II. Average size of extracted distinct MBs or VSs*	*III. Number of true MBs identified exactly*	*IV. Average proportion of false positives*	*V. Average false negative rate*	*VI. Weighted accuracy over all extracted MBs or VSs*		
							Average	*95% Interval*	
TIE*	*max-k* = 3, α = 0.05	72	5.0	72	0.000	0.000	**0.957**	0.952	0.960
iTIE*	*max-k* = 3, α = 0.05	72	5.0	72	0.000	0.000	**0.957**	0.952	0.960
KIAMB	Number of runs = 5000, α = 0.05, K = 0.7	349	2.8	0	0.000	0.400	0.722	0.450	0.959
	Number of runs = 5000, α = 0.05, K = 0.8	349	2.8	0	0.000	0.400	0.722	0.450	0.959
	Number of runs = 5000, α = 0.05, K = 0.9	349	2.8	0	0.000	0.400	0.722	0.450	0.959
EGS-NCMIGS	l = 7, δ = 0.015	6	7.0	0	0.286	0.000	0.953	0.952	0.956
	l = 7, K = 10	6	10.0	0	0.500	0.000	**0.968**	0.967	0.969
	l = 7, K = 50	6	50.0	0	0.900	0.000	0.877	0.866	0.887
	l = 5000, δ = 0.015	995	8.0	0	0.648	0.508	**0.960**	0.950	0.968
	l = 5000, K = 10	990	10.0	0	0.747	0.494	**0.961**	0.952	0.968
	l = 5000, K = 50	950	50.0	0	0.949	0.494	0.868	0.857	0.882
EGS-CMIM	l = 7, K = 10	6	10.0	0	0.500	0.000	**0.967**	0.965	0.968
	l = 7, K = 50	6	50.0	0	0.900	0.000	0.904	0.895	0.915
	l = 5000, K = 10	990	10.0	0	0.676	0.353	**0.961**	0.953	0.967
	l = 5000, K = 50	950	50.0	0	0.935	0.353	0.897	0.885	0.910
EGSG	Number of Markov boundaries = 30, t = 5	30	67.0	0	0.958	0.440	0.688	0.383	0.850
	Number of Markov boundaries = 30, t = 10	30	67.0	0	0.977	0.693	0.485	0.253	0.769
	Number of Markov boundaries = 30, t = 15	30	67.0	0	0.984	0.780	0.422	0.246	0.739
	Number of Markov boundaries = 5,000, t = 5	5,000	67.0	0	0.927	0.028	0.662	0.441	0.850
	Number of Markov boundaries = 5,000, t = 10	5,000	67.0	0	0.944	0.250	0.476	0.248	0.780
	Number of Markov boundaries = 5,000, t = 15	5,000	67.0	0	0.953	0.369	0.406	0.247	0.710
Resampling+RFE	without statistical comparison	2,492	16.7	2	0.434	0.039	0.951	0.931	0.968
	with statistical comparison (α = 0.05)	214	6.0	0	0.225	0.336	0.947	0.917	0.964
Resampling+UAF	without statistical comparison	1,207	28.7	0	0.721	0.000	0.952	0.935	0.964
	with statistical comparison (α = 0.05)	12	7.8	0	0.577	0.367	0.949	0.931	0.959
IR-HITON-PC	*max-k* = 3, α = 0.05	2	5.0	1	0.100	0.100	**0.958**	0.958	0.959
IR-SPLR	without statistical comparison	1	30.0	0	0.833	0.000	0.949	0.949	0.949
	with statistical comparison (α = 0.05)	1	30.0	0	0.833	0.000	0.949	0.949	0.949

Table 11: Results obtained in simulated data set *TIED*1000. "MB" stands for "Markov boundary", and "VS" stands for "variable set". The 95% interval for weighted accuracy denotes the range in which weighted accuracies of 95% of the extracted Markov boundaries/variable sets fell. Classification performance of the MAP-BN classifier in the same data sample was 0.972 weighted accuracy. Highlighted in bold are results that are statistically comparable to the MAP-BN classification performance.

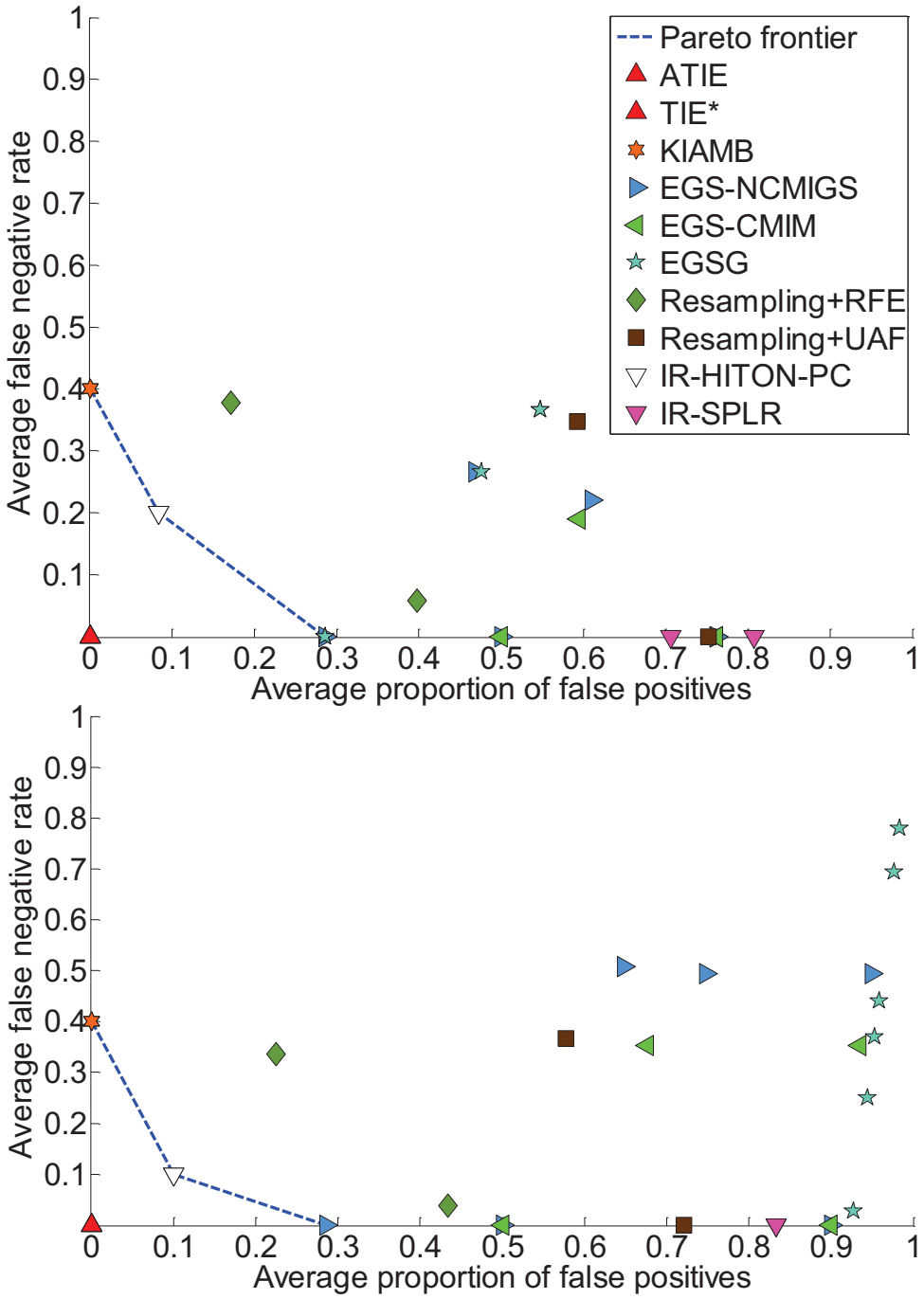


Figure 19: Results for average false negative rate and average proportion of false positives obtained in *TIED* (left) and *TIED*1000 (right) data sets. Results of TIE* and iTIE* were identical.

Name	Domain	♯ of samples	♯ of variables	Response type	Data type	CV design	References
Infant_Mortality	clinical	5,337	86	Death within the first year	Discrete	Holdout	Mani and Cooper (1999)
Ohsumed	Text	5,000	14,373	Relevant to neonatal diseases	Continuous	Holdout	Joachims (2002)
ACPJ_Etiology	Text	15,779	28,228	Relevant to etiology	Continuous	Holdout	Aphinyanaphongs et al. (2006)
Lymphoma	Gene Expression	227	7,399	3-year survival:dead vs. alive	Continuous	10-fold	Rosenwald et al. (2002)
Gisette	Digit recognition	7,000	5,000	4 vs. 9	Continuous	Holdout	NIPS 2003 Feature Selection Challenge Guyon et al. (2006)
Dexter	Text	600	19,999	Relevant to corporate acquisitions	Continuous	10-fold	NIPS 2003 Feature Selection Challenge Guyon et al. (2006)
Sylva	Ecology	14,394	216	Ponderosa vs. rest	Continuous	Holdout	WCCI 2006 Perf. Prediction Challenge
Ovarian_Cancer	Proteomics	216	2,190	Cancer vs. normal	Continuous	10-fold	Conrads et al. (2004)
Thrombin	Drug discovery	2,543	139,351	Binding to thrombin	Discrete	Holdout	KDD Cup 2001
Breast_Cancer	Gene Expression	286	17,816	ER+vs. ER-	Continuous	10-fold	Wang et al. (2005)
Hiva	Drug discovery	4,229	1,617	Activity to HIV AIDS infection	Discrete	Holdout	WCCI 2006 Perf.Prediction Challenge
Nova	Text	1,929	16,969	Political topics vs. religious	Discrete	Holdout	WCCI 2006 Perf. Prediction Chanllenge
Bankruptcy	Financial	7,063	147	Personal bankruptcy	Continuous	Holdout	Foster and Stine (2004)

Table 12: Real data sets used in the experiments.

E.5 On Computation of Performance Criteria in Experiments with Real Data

In order to rank all methods based on a given performance criterion, the average value of this criterion was first computed over all evaluation data sets for each method. The methods were then ordered from best to worst performing according to these averages. The best performer was assigned rank 1 and designated as the current "reference method". Performance of the next unranked method in the ordered list was compared to performance of the reference method using permutation-based testing at significance level 5% and with 10,000 permutations of the vectors of criterion values computed on each data set. If performance of the two methods was found to be statistically comparable, the unranked method received the same rank as the reference method. Otherwise, the next lowest rank was assigned to the unranked method and this method was designated as the new reference method. This process was repeated until each method was assigned a rank.

E.6 Additional Discussion of the Results of Experiments with Real Data

KIAMB produced some of the more compact Markov boundaries with the average PV of 1% and ranked second out of 11 by that criterion. Small sizes of the extracted Markov boundaries were, to a large extent, due to KIAMB's sample inefficiency resulting in inability to perform some of the required tests of independence as discussed in Appendix C. As a result, classification performance of Markov boundaries extracted by KIAMB was lower than of most other methods with KIAMB ranking 4 out of 5 by AUC. Consequently, KIAMB ranked 5 out of 15 on the (PV, AUC) criterion. Although, KIAMB was parameterized to produce 5,000 Markov boundaries, only about 30% of them were distinct, which means that 70% of computational time was spent on repeated retrieval of the same Markov boundaries.

EGS-NCMIGS with the alternative stopping criterion produced the smallest Markov boundaries at the expense of a significant reduction in AUC ($\sim 9\%$ below TIE*). Parameterizations of EGS-NCMIGS with the alternative stopping criterion ranked first and second out of 11 by PV and fourth out of 5 by AUC. Overall, performance of EGS-NCMIGS and EGS-CMIM varied widely depending on parameterization. Ranks of these methods ranged from 2 to 11 out of 15 on the (PV, AUC) criterion.

EGSG showed an overall poor performance, ranking between 10 and 15 out of 15 on (PV, AUC). Markov boundaries extracted by EGSG were larger than Markov boundaries identified by many other methods and had the lowest average classification performance.

Resampling+RFE and Resampling+UAF extracted variable sets that were the largest in comparison with other methods, but that also had highest classification performance. Resampling+RFE and Resampling+UAF ranked between 9 and 11 out of 11 by PV and between 1 and 3 by AUC. Notably, variable sets extracted by Resampling+UAF had an average PV between 24% and 41%, depending on parameterization. Resampling+RFE extracted more compact variable sets than Resampling+UAF in every data set, with the average PV between 5% and 17%. Due to poor performance on the PV criterion, Resampling+RFE and Resampling+UAF ranked in the mid to poor range on the combined (PV, AUC) criterion, scoring between 7 and 11 out of 15.

Iterative removal methods IR-HITON-PC and IR-SPLR extracted small numbers of Markov boundaries/variable sets and ranked between 5 and 6 out of 6 by that criterion. IR-HITON-PC produced more compact Markov boundaries than the variable sets of IR-SPLR. Markov boundaries extracted by IR-HITON-PC had an average PV of 2.3%, which was significantly smaller than the 11%-15% average PV of IR-SPLR. IR-HITON-PC method ranked 5 out of 11 by PV while IR-SPLR

methods ranked 9 and 10 by the same criterion. Despite the smaller average size of the extracted Markov boundaries, IR-HITON-PC ranked on par with IR-SPLR (parameterized with statistical comparison) by AUC, scoring third out of 5. Among all parameterizations of iterative removal methods, IR-SPLR without statistical comparison produced the largest variable sets, which helped this method reach a higher average classification performance and rank second out of 5 by AUC. Higher average PV of variable sets extracted by IR-SPLR caused these methods to rank 9 and 11 out of 15 on the combined (PV, AUC) criterion. IR-HITON-PC ranked sixth on the same criterion as a result of moderate ranks on PV and AUC.

Method		*Infant_Mortality*			*Ohsumed*			*ACPJ_Etiology*			*Lymphoma*			*Gisette*		
		N	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*
All variables		1	86	0.821	1	14,373	0.857	1	28,228	0.938	1	7,399	0.659	1	5,000	0.997
TIE*	*max-k* = 3, $\alpha = 0.05$	41	4	0.825	2,497	37	0.776	5,330	18	0.908	4,533	16	0.635	227	54	0.990
KIAMB	Number of runs = 5000, $\alpha = 0.05$, $K = 0.7$	67	4	0.753	250	7	0.651	1,354	9	0.884	88	3	0.562	5,000	8	0.871
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.8$	39	4	0.752	133	7	0.650	830	9	0.883	50	3	0.561	5,000	8	0.871
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.9$	17	4	0.752	58	7	0.648	414	9	0.884	23	3	0.561	5,000	8	0.871
EGS-NCMIGS	$l = 7$, $\delta = 0.015$	6	4	0.809	6	4	0.584	6	3	0.743	7	3	0.591	7	3	0.913
	$l = 7$, $K = 10$	3	10	0.874	1	10	0.691	3	10	0.780	5	10	0.615	7	10	0.952
	$l = 7$, $K = 50$	1	50	0.821	1	50	0.828	3	35	0.842	3	50	0.662	5	50	0.986
	$l = 5000$, $\delta = 0.015$	84	4	0.806	4,999	4	0.564	4,999	4	0.770	4,992	3	0.574	4,999	5	0.920
	$l = 5000$, $K = 10$	77	10	0.862	4,991	10	0.693	4,991	10	0.785	4,981	10	0.600	4,994	10	0.953
	$l = 5000$, $K = 50$	39	50	0.822	4,951	50	0.830	4,981	31	0.843	4,947	50	0.653	4,957	50	0.987
EGS-CMIM	$l = 7$, $K = 10$	2	10	0.865	1	10	0.696	2	10	0.915	6	10	0.577	7	10	0.956
	$l = 7$, $K = 50$	1	50	0.829	1	50	0.843	1	32	0.917	4	50	0.608	5	50	0.987
	$l = 5000$, $K = 10$	77	10	0.863	4,991	10	0.687	4,991	10	0.842	4,970	10	0.581	4,992	10	0.963
	$l = 5000$, $K = 50$	38	50	0.827	4,951	50	0.841	4,982	31	0.857	4,942	50	0.613	4,957	50	0.987
EGSG	Number of Markov boundaries = 30, $t = 5$	30	12	0.634	30	70	0.653	30	84	0.840	30	58	0.600	30	35	0.959
	Number of Markov boundaries = 30, $t = 10$	30	12	0.568	30	70	0.634	30	84	0.835	30	58	0.616	30	35	0.946
	Number of Markov boundaries = 30, $t = 15$	30	12	0.552	30	70	0.602	30	84	0.792	30	58	0.607	30	35	0.936
	Number of Markov boundaries = 5,000, $t = 5$	991	12	0.631	5,000	70	0.649	5,000	84	0.837	5,000	58	0.604	5,000	35	0.961
	Number of Markov boundaries = 5,000, $t = 10$	3,576	12	0.587	5,000	70	0.624	5,000	84	0.822	5,000	58	0.617	5,000	35	0.950
	Number of Markov boundaries = 5,000, $t = 15$	4,272	12	0.556	5,000	70	0.606	5,000	84	0.780	5,000	58	0.609	5,000	35	0.941
Resampling+RFE	without statistical comparison	4,230	17	0.825	4,942	3,889	0.846	5,000	2,441	0.924	4,919	1,293	0.634	4,948	697	0.997
	with statistical comparison ($\alpha = 0.05$)	3,222	9	0.814	5,000	914	0.836	5,000	308	0.864	4,962	45	0.587	5,000	134	0.995
Resampling+UAF	without statistical comparison	4,868	26	0.859	2,533	10,722	0.855	4,963	3,883	0.929	4,215	2,546	0.647	5,000	1,673	0.999
	with statistical comparison ($\alpha = 0.05$)	3,141	15	0.777	4,925	7,690	0.864	5,000	1,600	0.918	4,895	195	0.600	5,000	1,088	0.998
IR-HITON-PC	*max-k* = 3, $\alpha = 0.05$	1	5	0.857	2	40	0.778	4	22	0.875	12	10	0.593	3	64	0.990
IR-SPLR	without statistical comparison	1	8	0.835	1	176	0.829	4	123	0.885	16	456	0.577	1	466	0.996
	with statistical comparison ($\alpha = 0.05$)	1	2	0.828	3	122	0.728	5	26	0.844	139	47	0.572	1	261	0.996

(Continued on the next page)

Table 13: Results showing the number of distinct Markov boundaries or variable sets (N) extracted by each method, their average size in terms of the number of variables (S) and average classification performance (AUC) in each of 13 real data sets. The row labeled "All variables" shows performance of the entire set of variables available in each data set.

(Continued from the previous page)

Method		Dexter			Sylva			Ovarian_Cancer			Thrombin		
		N	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*
All variables		1	19,999	0.979	1	216	0.998	1	2,190	0.998	1	139,351	0.927
TIE*	*max-k* = 3, $\alpha = 0.05$	4,791	17	0.959	1,483	27	0.996	223	7	0.973	298	11	0.813
KIAMB	Number of runs = 5000, $\alpha = 0.05$, $K = 0.7$	299	5	0.882	4,429	8	0.949	285	4	0.925	4,936	6	0.771
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.8$	193	5	0.884	4,384	8	0.948	180	4	0.927	4,900	6	0.774
	Number of runs = 5000, $\alpha = 0.05$, $K = 0.9$	120	5	0.887	4,385	8	0.947	106	4	0.928	4,854	6	0.774
EGS-NCMIGS	$l = 7$, $\delta = 0.015$	6	4	0.839	4	5	0.960	6	5	0.951	7	3	0.781
	$l = 7$, $K = 10$	5	10	0.927	2	10	0.988	6	10	0.974	7	10	0.854
	$l = 7$, $K = 50$	4	50	0.971	1	50	0.998	3	50	0.986	7	12	0.760
	$l = 5000$, $\delta = 0.015$	4,998	5	0.840	213	5	0.954	2,188	6	0.956	4,999	4	0.779
	$l = 5000$, $K = 10$	4,991	10	0.927	207	10	0.987	2,183	10	0.971	4,996	10	0.858
	$l = 5000$, $K = 50$	4,951	50	0.970	167	50	0.998	2,144	50	0.988	4,997	14	0.764
EGS-CMIM	$l = 7$, $K = 10$	5	10	0.942	2	10	0.991	5	10	0.976	7	10	0.799
	$l = 7$, $K = 50$	3	50	0.979	1	50	0.997	3	50	0.991	7	12	0.711
	$l = 5000$, $K = 10$	4,991	10	0.943	207	10	0.992	2,182	10	0.973	4,999	10	0.856
	$l = 5000$, $K = 50$	4,951	50	0.979	167	50	0.998	2,144	50	0.992	5,000	14	0.720
EGSG	Number of Markov boundaries = 30, $t = 5$	30	76	0.857	30	12	0.803	30	12	0.953	30	29	0.776
	Number of Markov boundaries = 30, $t = 10$	30	76	0.791	30	12	0.810	30	12	0.940	30	29	0.817
	Number of Markov boundaries = 30, $t = 15$	30	76	0.749	30	12	0.744	30	12	0.930	30	29	0.757
	Number of Markov boundaries = 5,000, $t = 5$	5,000	76	0.854	4,997	12	0.792	4,878	12	0.951	5,000	29	0.758
	Number of Markov boundaries = 5,000, $t = 10$	5,000	76	0.787	5,000	12	0.803	4,990	12	0.936	5,000	29	0.815
	Number of Markov boundaries = 5,000, $t = 15$	5,000	76	0.746	5,000	12	0.752	4,996	12	0.927	5,000	29	0.749
Resampling+RFE	without statistical comparison	5,000	2,097	0.976	4,976	19	0.998	4,951	142	0.983	5,000	14,996	0.912
	with statistical comparison ($\alpha = 0.05$)	4,998	96	0.956	3,549	12	0.998	2,601	5	0.926	5,000	216	0.861
Resampling+UAF	without statistical comparison	3,561	15,491	0.976	3,944	44	0.998	4,372	424	0.980	4,998	74,521	0.933
	with statistical comparison ($\alpha = 0.05$)	4,992	14,064	0.972	2,842	23	0.998	972	13	0.955	5,000	25,217	0.916
IR-HITON-PC	*max-k* = 3, $\alpha = 0.05$	1	20	0.958	1	24	0.997	2	7	0.962	1	13	0.844
IR-SPLR	without statistical comparison	1	425	0.974	1	36	0.999	2	709	0.986	4	245	0.876
	with statistical comparison ($\alpha = 0.05$)	3	149	0.940	1	36	0.999	11	115	0.943	28	144	0.749

(Continued on the next page)

Table 14:

(Continued from the previous two pages)

Method		Breast_Cancer			Hiva			Nova			Bankruptcy		
		N	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*	*N*	*S*	*AUC*
All variables		1	17,816	0.914	1	1,617	0.716	1	16,969	0.981	1	147	0.940
TIE*	*max-k* = 3, α = 0.05	1,011	10	0.906	246	8	0.712	3,751	41	0.922	1,478	14	0.923
KIAMB	Number of runs = 5000, α = 0.05, K = 0.7	418	4	0.873	876	7	0.735	130	6	0.759	3,810	6	0.839
	Number of runs = 5000, α = 0.05, K = 0.8	255	4	0.874	439	7	0.754	57	6	0.764	3,713	5	0.836
	Number of runs = 5000, α = 0.05, K = 0.9	136	4	0.879	172	7	0.755	23	6	0.771	3,681	5	0.838
EGS-NCMIGS	l = 7, δ = 0.015	6	4	0.922	7	3	0.661	5	4	0.730	7	4	0.698
	l = 7, K = 10	6	10	0.926	7	10	0.750	2	10	0.815	6	10	0.936
	l = 7, K = 50	6	50	0.928	7	50	0.668	1	50	0.849	3	50	0.953
	l = 5000, δ = 0.015	4,994	4	0.911	1,616	4	0.696	4,998	5	0.735	145	4	0.721
	l = 5000, K = 10	4,985	10	0.926	1,610	10	0.760	4,991	10	0.780	139	10	0.936
	l = 5000, K = 50	4,973	50	0.927	1,570	50	0.658	4,951	50	0.846	101	50	0.953
EGS-CMIM	l = 7, K = 10	7	10	0.914	5	10	0.713	3	10	0.818	3	10	0.913
	l = 7, K = 50	6	50	0.902	5	50	0.727	1	50	0.886	1	50	0.954
	l = 5000, K = 10	4,978	10	0.906	1,608	10	0.724	4,992	10	0.780	139	10	0.901
	l = 5000, K = 50	4,966	50	0.907	1,577	50	0.729	4,951	50	0.897	99	50	0.954
EGSG	Number of Markov boundaries = 30, t = 5	30	205	0.893	30	17	0.705	30	89	0.751	30	9	0.815
	Number of Markov boundaries = 30, t = 10	30	205	0.886	30	17	0.660	30	89	0.722	30	9	0.754
	Number of Markov boundaries = 30, t = 15	30	205	0.890	30	17	0.633	30	89	0.687	30	9	0.752
	Number of Markov boundaries = 5,000, t = 5	5,000	205	0.892	5,000	17	0.701	5,000	89	0.751	4,373	9	0.819
	Number of Markov boundaries = 5,000, t = 10	5,000	205	0.888	5,000	17	0.652	5,000	89	0.720	4,856	9	0.786
	Number of Markov boundaries = 5,000, t = 15	5,000	205	0.889	5,000	17	0.638	5,000	89	0.682	4,905	9	0.787
Resampling+RFE	without statistical comparison	4,848	1,067	0.901	4,938	220	0.679	4,948	5,305	0.982	4,949	66	0.940
	with statistical comparison (α = 0.05)	2,922	10	0.894	4,598	13	0.646	5,000	1,261	0.966	4,972	38	0.946
Resampling+UAF	without statistical comparison	4,365	3,359	0.905	4,587	309	0.685	645	13,950	0.981	4,379	79	0.952
	with statistical comparison (α = 0.05)	1,295	42	0.917	4,250	36	0.663	3,185	12,503	0.978	628	47	0.946
IR-HITON-PC	*max-k* = 3, α = 0.05	12	9	0.890	23	6	0.673	2	49	0.920	3	15	0.910
IR-SPLR	without statistical comparison	47	159	0.892	10	129	0.661	1	10,289	0.981	1	69	0.956
	with statistical comparison (α = 0.05)	54	27	0.880	7	22	0.694	1	10,289	0.981	1	69	0.956

Table 15:

References

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using markov blanket induction for the prediction of protein-drug binding. *Technical Report DSL 02-06*, 2002.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Hiton: a novel markov blanket algorithm for optimal variable selection. *AMIA 2003 Annual Symposium Proceedings*, pages 21–25, 2003a.

C. F. Aliferis, I. Tsamardinos, A. Statnikov, and L. E. Brown. Causal explorer: a causal probabilistic network learning toolkit for biomedical discovery. *Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*, 2003b.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification. part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010a.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. 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, 2010b.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*, volume 3rd of *Wiley Series in Probability and Statistics*. Wiley-Interscience, Hoboken, N.J, 2003.

Y. Aphinyanaphongs, A. Statnikov, and C. F. Aliferis. A comparison of citation metrics to machine learning filters for the identification of high quality medline documents. *J.Am.Med.Inform.Assoc.*, 13(4):446–455, 2006.

L. Breiman. Statistical modeling: the two cultures. *Statistical Science*, 16(3):199–215, 2001.

L. E. Brown, I. Tsamardinos, and D. Hardin. To feature space and back: Identifying top-weighted features in polynomial support vector machine models. *Intelligent Data Analysis*, 16(4), 2012.

T. P. Conrads, V. A. Fusaro, S. Ross, D. Johann, V. Rajapakse, B. A. Hitt, S. M. Steinberg, E. C. Kohn, D. A. Fishman, G. Whitely, J. C. Barrett, L. A. Liotta, III Petricoin, E. F., and T. D. Veenstra. High-resolution serum proteomic features for ovarian cancer detection. *Endocr.Relat Cancer*, 11(2):163–178, 2004.

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

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, 2000.

E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3): 837–845, 1988.

E. Dougherty and M. Brun. On the number of close-to-optimal feature sets. *Cancer Informatics*, 2: 189–196, 2006.

L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21(2):171–178, 2005.

L. Ein-Dor, O. Zuk, and E. Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc.Natl.Acad.Sci.U.S.A*, 103(15):5923–5928, 2006.

R. E. Fan, P. H. Chen, and C. J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6(1889):1918, 2005.

T. Fawcett. Roc graphs: Notes and practical considerations for researchers. *Technical Report, HPL-2003-4, HP Laboratories*, 2003.

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

D. P. Foster and R. A. Stine. Variable selecion in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, 99(466):303–314, 2004.

C.N. Glymour and G.F. Copper. *Computation, Causation and Discovery*. AAAI Press, Menlo Park, Calif, 1991.

P. I. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, volume 2nd of *Springer series in statistics*. Springer, New York, 2000.

A. Gopnik and L. Schulz. *Causal Learning: Psychology, Philosophy, and Computation*. Oxford University Press, Oxford, 2007.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(1):1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications*. Studies in fuzziness and soft computing. Springer-Verlag, Berlin, 2006.

B. Hammer and K. Gersmann. A note on the universal approximation capability of support vector machines. *Neural Processing Letters*, 17(1):43–53, 2003.

M. Hollander and D. Wolfe. *Nonparametric statistical methods*, volume 2nd of *Wiley Series in Probability and Statistics*. Wiley, New York, NY, USA, 1999.

T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, 2002.

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

J. Lemeire. *Learning Causal Models of Multivariate Systems and the Value of It for the Performance Modeling of Computer Programs*. PhD thesis, 2007.

J. Lemeire, S. Meganck, and F. Cartella. Robust independence-based causal structure learning in absence of adjacency faithfulness. *Proccedings of the Fifth European Workshop on Probabilistic Graphical Models (PGM 2010)*, 2010.

H. Liu, L. Liu, and H. Zhang. Ensemble gene selection by grouping for microarray data classification. *J.Biomed.Inform.*, 43(1):81–87, 2010.

H. Liu, L. Liu, and H. Zhang. Ensemble gene selection for cancer classification. *Pattern Recognition*, 43(8):2763–2772, 2010b.

S. Mani and G. F. Cooper. A study in causal discovery from population-based infant birth and death records. *Proceedings of the AMIA Annual Fall Symposium*, 319, 1999.

S. Mani and G. F. Cooper. Causal discovery using a bayesian local causal discovery algorithm. *Medinfo 2004*, 11(Pt 1):731–735, 2004.

S. Michiels, S. Koscielny, and C. Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet*, 365(9458):488–492, 2005.

G. Natsoulis, Ghaoui L. El, G. R. Lanckriet, A. M. Tolley, F. Leroy, S. Dunlea, B. P. Eynon, C. I. Pearson, S. Tugendreich, and K. Jarnagin. Classification of a large microarray data set: algorithm comparison and analysis of drug signatures. *Genome Res.*, 15(5):724–736, 2005.

R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

J. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211 –232, 2007.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann Publishers, San Mateo, California, 1988.

J. Pearl. *Causality: Models, Reasoning, and Inference*, volume 2nd. Cambridge University Press, Cambridge, U.K, 2009.

J. P. Pellet and A. Elisseeff. Using markov blankets for causal structure learning. *The Journal of Machine Learning Research*, 9:1295–1342, 2008.

A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

J. Ramsey, J. Zhang, and P. Spirtes. Adjacency-faithfulness and conservative causal inference. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006)*, pages 401–408, 2006.

T. Richardson and P. Spirtes. *Automated Discovery of Linear Feedback Models*. MIT Press, Menlo Park, CA, 1999.

T. S. Richardson and P. Spirtes. Ancestral graph markov models. *Annals of Statistics*, 30(4):962–1030, 2002.

P. Roepman, P. Kemmeren, L. F. Wessels, P. J. Slootweg, and F. C. Holstege. Multiple robust signatures for detecting lymph node metastasis in head and neck cancer. *Cancer Res.*, 66(4): 2361–2366, 2006.

A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. Lopez-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, and L. M. Staudt. 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.

F. Scarselli and A. Chung Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1):15–37, 1998.

B. Schölkopf, C. J. C. Burges, and A. J. Smola. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, Mass, 1999.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

R. L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, 19 (12):1484–1491, 2003.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*, volume 2nd of *Adaptive computation and machine learning*. MIT Press, Cambridge, Mass, 2000.

A. Statnikov and C. F. Aliferis. Analysis and computational dissection of molecular signature multiplicity. *PLoS Computational Biology*, 6(5):e1000790, 2010a.

A. Statnikov and C. F. Aliferis. TIED: An Artificially Simulated Dataset with Multiple Markov Boundaries. *Journal of Machine Learning Research Workshop and Conference Proceedings, Volume 6: Causality: Objectives and Assessment (NIPS 2008)*, 6:249–256, 2010b.

A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643, 2005.

A. Statnikov, I. Tsamardinos, L. E. Brown, and C. F. Aliferis. *Causal Explorer: A Matlab Library of Algorithms for Causal Discovery and Variable Selection for Classification*. In Challenges in Machine Learning. Volume 2: Causation and Prediction Challenge. Edited by Guyon, I. and Aliferis, C. F. and Cooper, G. F. and Elisseeff, A. and Pellet, J. P. and Spirtes, P. and Statnikov, A. Microtome Publishing, Bookline, Massachusetts, 2010.

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

I. Tsamardinos and L. E. Brown. Markov blanket-based variable selection in feature space. *Technical Report DSL-08-01*, 2008.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 376–381, 2003a.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 673–678, 2003b.

V. N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

Y. Wang, J. G. Klijn, Y. Zhang, A. M. Sieuwerts, M. P. Look, F. Yang, D. Talantov, M. Timmermans, M. E. Meijer-van Gelder, J. Yu, T. Jatkoe, E. M. Berns, D. Atkins, and J. A. Foekens. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365(9460):671–679, 2005.

S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. M. Kaufmann Publishers, San Mateo, Calif, 1991.

Journal of Machine Learning Research 14 (2013) 567-599 Submitted 9/12; Revised 1/13; Published 2/13

Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
Benin school of Computer Science and Engineering
The Hebrew University
Jerusalem, 91904, Israel

Tong Zhang TZHANG@STAT.RUTGERS.EDU
Department of Statistics
Rutgers University
Piscataway, NJ, 08854, USA

Editor: Leon Bottou

Abstract

Stochastic Gradient Descent (SGD) has become popular for solving large scale supervised machine learning optimization problems such as SVM, due to their strong theoretical guarantees. While the closely related Dual Coordinate Ascent (DCA) method has been implemented in various software packages, it has so far lacked good convergence analysis. This paper presents a new analysis of Stochastic Dual Coordinate Ascent (SDCA) showing that this class of methods enjoy strong theoretical guarantees that are comparable or better than SGD. This analysis justifies the effectiveness of SDCA for practical applications.

Keywords: stochastic dual coordinate ascent, optimization, computational complexity, regularized loss minimization, support vector machines, ridge regression, logistic regression

1. Introduction

We consider the following generic optimization problem associated with regularized loss minimization of linear predictors: Let $x_1,\ldots,x_n$ be vectors in $\mathbb{R}^d$, let $\phi_1,\ldots,\phi_n$ be a sequence of scalar convex functions, and let $\lambda > 0$ be a regularization parameter. Our goal is to solve $\min_{w\in\mathbb{R}^d} P(w)$ where[1]

$$P(w) = \left[\frac{1}{n}\sum_{i=1}^{n}\phi_i(w^\top x_i) + \frac{\lambda}{2}\|w\|^2\right]. \tag{1}$$

For example, given labels $y_1,\ldots,y_n$ in $\{\pm 1\}$, the SVM problem (with linear kernels and no bias term) is obtained by setting $\phi_i(a) = \max\{0, 1-y_i a\}$. Regularized logistic regression is obtained by setting $\phi_i(a) = \log(1+\exp(-y_i a))$. Regression problems also fall into the above. For example, ridge regression is obtained by setting $\phi_i(a) = (a-y_i)^2$, regression with the absolute-value is obtained by setting $\phi_i(a) = |a-y_i|$, and support vector regression is obtained by setting $\phi_i(a) = \max\{0, |a-y_i|-\nu\}$, for some predefined insensitivity parameter $\nu > 0$.

Let w^* be the optimum of (1). We say that a solution w is ε_P-sub-optimal if $P(w) - P(w^*) \le \varepsilon_P$. We analyze the runtime of optimization procedures as a function of the time required to find an ε_P-sub-optimal solution.

1. Throughout this paper, we only consider the ℓ_2-norm.

A simple approach for solving SVM is stochastic gradient descent (SGD) (Robbins and Monro, 1951; Murata, 1998; Cun and Bottou, 2004; Zhang, 2004; Bottou and Bousquet, 2008; Shalev-Shwartz et al., 2007). SGD finds an ε_P-sub-optimal solution in time $\tilde{O}(1/(\lambda\varepsilon_P))$. This runtime does not depend on n and therefore is favorable when n is very large. However, the SGD approach has several disadvantages. It does not have a clear stopping criterion; it tends to be too aggressive at the beginning of the optimization process, especially when λ is very small; while SGD reaches a moderate accuracy quite fast, its convergence becomes rather slow when we are interested in more accurate solutions.

An alternative approach is dual coordinate ascent (DCA), which solves a *dual* problem of (1). Specifically, for each i let $\phi_i^* : \mathbb{R} \to \mathbb{R}$ be the convex conjugate of ϕ_i, namely, $\phi_i^*(u) = \max_z(zu - \phi_i(z))$. The dual problem is

$$\max_{\alpha\in\mathbb{R}^n} D(\alpha) \quad \text{where} \quad D(\alpha) = \left[\frac{1}{n}\sum_{i=1}^n -\phi_i^*(-\alpha_i) - \frac{\lambda}{2}\left\|\frac{1}{\lambda n}\sum_{i=1}^n \alpha_i x_i\right\|^2\right] . \tag{2}$$

The dual objective in (2) has a different dual variable associated with each example in the training set. At each iteration of DCA, the dual objective is optimized with respect to a single dual variable, while the rest of the dual variables are kept in tact.

If we define

$$w(\alpha) = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i x_i, \tag{3}$$

then it is known that $w(\alpha^*) = w^*$, where α^* is an optimal solution of (2). It is also known that $P(w^*) = D(\alpha^*)$ which immediately implies that for all w and α, we have $P(w) \geq D(\alpha)$, and hence the duality gap defined as

$$P(w(\alpha)) - D(\alpha)$$

can be regarded as an upper bound of the primal sub-optimality $P(w(\alpha)) - P(w^*)$.

We focus on a *stochastic* version of DCA, abbreviated by SDCA, in which at each round we choose which dual coordinate to optimize uniformly at random. The purpose of this paper is to develop theoretical understanding of the convergence of the duality gap for SDCA.

We analyze SDCA either for L-Lipschitz loss functions or for $(1/\gamma)$-smooth loss functions, which are defined as follows. Throughout the paper, we will use $\phi'(a)$ to denote a sub-gradient of a convex function $\phi(\cdot)$, and use $\partial\phi(a)$ to denote its sub-differential.

Definition 1 *A function $\phi_i : \mathbb{R} \to \mathbb{R}$ is L-Lipschitz if for all $a, b \in \mathbb{R}$, we have*

$$|\phi_i(a) - \phi_i(b)| \leq L|a - b|.$$

A function $\phi_i : \mathbb{R} \to \mathbb{R}$ is $(1/\gamma)$-smooth if it is differentiable and its derivative is $(1/\gamma)$-Lipschitz. An equivalent condition is that for all $a, b \in \mathbb{R}$, we have

$$\phi_i(a) \leq \phi_i(b) + \phi_i'(b)(a - b) + \frac{1}{2\gamma}(a - b)^2,$$

where ϕ_i' is the derivative of ϕ_i.

It is well-known that if $\phi_i(a)$ is $(1/\gamma)$-smooth, then $\phi_i^*(u)$ is γ strongly convex: for all $u, v \in \mathbb{R}$ and $s \in [0,1]$:

$$-\phi_i^*(su + (1-s)v) \geq -s\phi_i^*(u) - (1-s)\phi_i^*(v) + \frac{\gamma s(1-s)}{2}(u-v)^2.$$

Our main findings are: in order to achieve a duality gap of ε,

- For L-Lipschitz loss functions, we obtain the rate of $\tilde{O}(n + L^2/(\lambda\varepsilon))$.
- For $(1/\gamma)$-smooth loss functions, we obtain the rate of $\tilde{O}((n + 1/(\lambda\gamma))\log(1/\varepsilon))$.
- For loss functions which are almost everywhere smooth (such as the hinge-loss), we can obtain rate better than the above rate for Lipschitz loss. See Section 5 for a precise statement.

2. Related Work

DCA methods are related to decomposition methods (Platt, 1998; Joachims, 1998). While several experiments have shown that decomposition methods are inferior to SGD for large scale SVM (Shalev-Shwartz et al., 2007; Bottou and Bousquet, 2008), Hsieh et al. (2008) recently argued that SDCA outperform the SGD approach in some regimes. For example, this occurs when we need relatively high solution accuracy so that either SGD or SDCA has to be run for more than a few passes over the data.

However, our theoretical understanding of SDCA is not satisfying. Several authors (e.g., Mangasarian and Musicant, 1999; Hsieh et al., 2008) proved a linear convergence rate for solving SVM with DCA (not necessarily stochastic). The basic technique is to adapt the linear convergence of coordinate ascent that was established by Luo and Tseng (1992). The linear convergence means that it achieves a rate of $(1-\nu)^k$ after k passes over the data, where $\nu > 0$. This convergence result tells us that after an unspecified number of iterations, the algorithm converges faster to the optimal solution than SGD.

However, there are two problems with this analysis. First, the linear convergence parameter, ν, may be very close to zero and the initial unspecified number of iterations might be very large. In fact, while the result of Luo and Tseng (1992) does not explicitly specify ν, an examine of their proof shows that ν is proportional to the smallest nonzero eigenvalue of $X^\top X$, where X is the $n \times d$ data matrix with its i-th row be the i-th data point x_i. For example if two data points $x_i \neq x_j$ becomes closer and closer, then $\nu \to 0$. This dependency is problematic in the data laden domain, and we note that such a dependency does not occur in the analysis of SGD.

Second, the analysis only deals with the sub-optimality of the *dual* objective, while our real goal is to bound the sub-optimality of the *primal* objective. Given a dual solution $\alpha \in \mathbb{R}^n$ its corresponding primal solution is $w(\alpha)$ (see (3)). The problem is that even if α is ε_D-sub-optimal in the dual, for some small ε_D, the primal solution $w(\alpha)$ might be far from being optimal. For SVM, (Hush et al., 2006, Theorem 2) showed that in order to obtain a primal ε_P-sub-optimal solution, we need a dual ε_D-sub-optimal solution with $\varepsilon_D = O(\lambda\varepsilon_P^2)$; therefore a convergence result for dual solution can only translate into a primal convergence result with worse convergence rate. Such a treatment is unsatisfactory, and this is what we will avoid in the current paper.

Some analyses of stochastic coordinate ascent provide solutions to the first problem mentioned above. For example, Collins et al. (2008) analyzed an exponentiated gradient dual coordinate ascent algorithm. The algorithm analyzed there (exponentiated gradient) is different from the standard

DCA algorithm which we consider here, and the proof techniques are quite different. Consequently their results are not directly comparable to results we obtain in this paper. Nevertheless we note that for SVM, their analysis shows a convergence rate of $O(n/\varepsilon_D)$ in order to achieve ε_D-sub-optimality (on the dual) while our analysis shows a convergence of $O(n\log\log n + 1/\lambda\varepsilon)$ to achieve ε duality gap; for logistic regression, their analysis shows a convergence rate of $O((n+1/\lambda)\log(1/\varepsilon_D))$ in order to achieve ε_D-sub-optimality on the dual while our analysis shows a convergence of $O((n+1/\lambda)\log(1/\varepsilon))$ to achieve ε duality gap.

In addition, Shalev-Shwartz and Tewari (2009), and later Nesterov (2012) have analyzed randomized versions of coordinate descent for unconstrained and constrained minimization of smooth convex functions. Hsieh et al. (2008, Theorem 4) applied these results to the dual SVM formulation. However, the resulting convergence rate is $O(n/\varepsilon_D)$ which is, as mentioned before, inferior to the results we obtain here. Furthermore, neither of these analyses can be applied to logistic regression due to their reliance on the smoothness of the dual objective function which is not satisfied for the dual formulation of logistic regression. We shall also point out again that all of these bounds are for the dual sub-optimality, while as mentioned before, we are interested in the primal sub-optimality.

In this paper we derive new bounds on the duality gap (hence, they also imply bounds on the primal sub-optimality) of SDCA. These bounds are superior to earlier results, and our analysis only holds for randomized (stochastic) dual coordinate ascent. As we will see from our experiments, randomization is important in practice. In fact, the practical convergence behavior of (non-stochastic) cyclic dual coordinate ascent (even with a random ordering of the data) can be slower than our theoretical bounds for SDCA, and thus cyclic DCA is inferior to SDCA. In this regard, we note that some of the earlier analysis such as Luo and Tseng (1992) can be applied both to stochastic and to cyclic dual coordinate ascent methods with similar results. This means that their analysis, which can be no better than the behavior of cyclic dual coordinate ascent, is inferior to our analysis.

Recently, Lacoste-Julien et al. (2012) derived a stochastic coordinate ascent for structural SVM based on the Frank-Wolfe algorithm. Specifying one variant of their algorithm to binary classification with the hinge loss, yields the SDCA algorithm for the hinge-loss. The rate of convergence Lacoste-Julien et al. (2012) derived for their algorithm is the same as the rate we derive for SDCA with a Lipschitz loss function.

Another relevant approach is the Stochastic Average Gradient (SAG), that has recently been analyzed in Le Roux et al. (2012). There, a convergence rate of $\tilde{O}(n\log(1/\varepsilon))$ rate is shown, for the case of smooth losses, assuming that $n \geq \frac{8}{\lambda\gamma}$. This matches our guarantee in the regime $n \geq \frac{8}{\lambda\gamma}$.

The following table summarizes our results in comparison to previous analyses. Note that for SDCA with Lipschitz loss, we observe a faster practical convergence rate, which is explained with our refined analysis in Section 5.

Lipschitz loss

Algorithm	type of convergence	rate
SGD	primal	$\tilde{O}(\frac{1}{\lambda\varepsilon})$
online EG (Collins et al., 2008) (for SVM)	dual	$\tilde{O}(\frac{n}{\varepsilon})$
Stochastic Frank-Wolfe (Lacoste-Julien et al., 2012)	primal-dual	$\tilde{O}(n+\frac{1}{\lambda\varepsilon})$
SDCA	primal-dual	$\tilde{O}(n+\frac{1}{\lambda\varepsilon})$ or faster

Smooth loss

Algorithm	type of convergence	rate
SGD	primal	$\tilde{O}(\frac{1}{\lambda\epsilon})$
online EG (Collins et al., 2008) (for logistic regression)	dual	$\tilde{O}((n+\frac{1}{\lambda})\log\frac{1}{\epsilon})$
SAG (Le Roux et al., 2012) (assuming $n \geq \frac{8}{\lambda\gamma}$)	primal	$\tilde{O}((n+\frac{1}{\lambda})\log\frac{1}{\epsilon})$
SDCA	primal-dual	$\tilde{O}((n+\frac{1}{\lambda})\log\frac{1}{\epsilon})$

3. Basic Results

The generic algorithm we analyze is described below. In the pseudo-code, the parameter T indicates the number of iterations while the parameter T_0 can be chosen to be a number between 1 to T. Based on our analysis, a good choice of T_0 is to be $T/2$. In practice, however, the parameters T and T_0 are not required as one can evaluate the duality gap and terminate when it is sufficiently small.

Procedure SDCA$(\alpha^{(0)})$

Let $w^{(0)} = w(\alpha^{(0)})$
Iterate: for $t = 1, 2, \ldots, T$:
  Randomly pick i
  Find $\Delta\alpha_i$ to maximize $-\phi_i^*(-(\alpha_i^{(t-1)} + \Delta\alpha_i)) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1}\Delta\alpha_i x_i\|^2$
  $\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \Delta\alpha_i e_i$
  $w^{(t)} \leftarrow w^{(t-1)} + (\lambda n)^{-1}\Delta\alpha_i x_i$
Output (Averaging option):
  Let $\bar{\alpha} = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T}\alpha^{(t-1)}$
  Let $\bar{w} = w(\bar{\alpha}) = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T} w^{(t-1)}$
  return $\bar{w}$
Output (Random option):
  Let $\bar{\alpha} = \alpha^{(t)}$ and $\bar{w} = w^{(t)}$ for some random $t \in T_0+1, \ldots, T$
  return $\bar{w}$

We analyze the algorithm based on different assumptions on the loss functions. To simplify the statements of our theorems, we always assume the following:

1. For all i, $\|x_i\| \leq 1$
2. For all i and a, $\phi_i(a) \geq 0$
3. For all i, $\phi_i(0) \leq 1$

Theorem 2 *Consider Procedure SDCA with $\alpha^{(0)} = 0$. Assume that ϕ_i is L-Lipschitz for all i. To obtain a duality gap of $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \epsilon_P$, it suffices to have a total number of iterations of*

$$T \geq T_0 + n + \frac{4L^2}{\lambda\epsilon_P} \geq \max(0, \lceil n\log(0.5\lambda n L^{-2})\rceil) + n + \frac{20L^2}{\lambda\epsilon_P}\,.$$

Moreover, when $t \geq T_0$, we have dual sub-optimality bound of $\mathbb{E}[D(\alpha^*) - D(\alpha^{(t)})] \leq \epsilon_P/2$.

Remark 3 *If we choose the average version, we may simply take $T = 2T_0$. Moreover, we note that Theorem 2 holds for both averaging or for choosing w at random from $\{T_0+1,\ldots,T\}$. This means that calculating the duality gap at few random points would lead to the same type of guarantee with high probability. This approach has the advantage over averaging, since it is easier to implement the stopping condition (we simply check the duality gap at some random stopping points. This is in contrast to averaging in which we need to know T, T_0 in advance).*

Remark 4 *The above theorem applies to the hinge-loss function, $\phi_i(u) = \max\{0, 1-y_i a\}$. However, for the hinge-loss, the constant 4 in the first inequality can be replaced by 1 (this is because the domain of the dual variables is positive, hence the constant 4 in Lemma 22 can be replaced by 1). We therefore obtain the bound:*

$$T \geq T_0 + n + \frac{L^2}{\lambda\epsilon_P} \geq \max(0, \lceil n\log(0.5\lambda n L^{-2})\rceil) + n + \frac{5L^2}{\lambda\epsilon_P}\ .$$

Theorem 5 *Consider Procedure SDCA with $\alpha^{(0)} = 0$. Assume that ϕ_i is $(1/\gamma)$-smooth for all i. To obtain an expected duality gap of $\mathbb{E}[P(w^{(T)}) - D(\alpha^{(T)})] \leq \epsilon_P$, it suffices to have a total number of iterations of*

$$T \geq \left(n + \tfrac{1}{\lambda\gamma}\right)\log((n+\tfrac{1}{\lambda\gamma})\cdot\tfrac{1}{\epsilon_P}).$$

Moreover, to obtain an expected duality gap of $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \epsilon_P$, it suffices to have a total number of iterations of $T > T_0$ where

$$T_0 \geq \left(n + \tfrac{1}{\lambda\gamma}\right)\log((n+\tfrac{1}{\lambda\gamma})\cdot\tfrac{1}{(T-T_0)\epsilon_P}).$$

Remark 6 *If we choose $T = 2T_0$, and assume that $T_0 \geq n + 1/(\lambda\gamma)$, then the second part of Theorem 5 implies a requirement of*

$$T_0 \geq \left(n + \tfrac{1}{\lambda\gamma}\right)\log(\tfrac{1}{\epsilon_P}),$$

which is slightly weaker than the first part of Theorem 5 when ϵ_P is relatively large.

Remark 7 *Bottou and Bousquet (2008) analyzed the runtime of SGD and other algorithms from the perspective of the time required to achieve a certain level of error on the test set. To perform such analysis, we also need to take into account the* estimation error, *namely, the additional error we suffer due to the fact that the training examples defining the regularized loss minimization problem are only a finite sample from the underlying distribution. The estimation error of the primal objective behaves like $\Theta\left(\frac{1}{\lambda n}\right)$ (see Shalev-Shwartz and Srebro, 2008; Sridharan et al., 2009). Therefore, an interesting regime is when $\frac{1}{\lambda n} = \Theta(\epsilon)$. In that case, the bound for both Lipschitz and smooth functions would be $\tilde{O}(n)$. However, this bound on the estimation error is for the worst-case distribution over examples. Therefore, another interesting regime is when we would like $\epsilon \ll \frac{1}{\lambda n}$, but still $\frac{1}{\lambda n} = O(1)$ (following the practical observation that $\lambda = \Theta(1/n)$ often performs well). In that case, smooth functions still yield the bound $\tilde{O}(n)$, but the dominating term for Lipschitz functions will be $\frac{1}{\lambda\epsilon}$.*

Remark 8 *The runtime of SGD is $\tilde{O}(\frac{1}{\lambda\epsilon})$. This can be better than SDCA if $n \gg \frac{1}{\lambda\epsilon}$. However, in that case, SGD in fact only looks at $n' = \tilde{O}(\frac{1}{\lambda\epsilon})$ examples, so we can run SDCA on these n' examples and obtain basically the same rate. For smooth functions, SGD can be much worse than SDCA if $\epsilon \ll \frac{1}{\lambda n}$.*

4. Using SGD At The First Epoch

From the convergence analysis, SDCA may not perform as well as SGD for the first few epochs (each epoch means one pass over the data). The main reason is that SGD takes a larger step size than SDCA earlier on, which helps its performance. It is thus natural to combine SGD and SDCA, where the first epoch is performed using a modified stochastic gradient descent rule. We show that the expected dual sub-optimality at the end of the first epoch is $\tilde{O}(1/(\lambda n))$. This result can be combined with SDCA to obtain a faster convergence when $\lambda \gg \log n/n$.

We first introduce convenient notation. Let P_t denote the primal objective for the first t examples in the training set,

$$P_t(w) = \left[\frac{1}{t}\sum_{i=1}^{t}\phi_i(w^\top x_i) + \frac{\lambda}{2}\|w\|^2\right].$$

The corresponding dual objective is

$$D_t(\alpha) = \left[\frac{1}{t}\sum_{i=1}^{t} -\phi_i^*(-\alpha_i) - \frac{\lambda}{2}\left\|\frac{1}{\lambda t}\sum_{i=1}^{t}\alpha_i x_i\right\|^2\right].$$

Note that $P_n(w)$ is the primal objective given in (1) and that $D_n(\alpha)$ is the dual objective given in (2).

The following algorithm is a modification of SGD. The idea is to greedily decrease the dual sub-optimality for problem $D_t(\cdot)$ at each step t. This is different from DCA which works with $D_n(\cdot)$ at each step t.

Procedure Modified-SGD

Initialize: $w^{(0)} = 0$
Iterate: for $t = 1, 2, \ldots, n$:
  Find α_t to maximize $-\phi_t^*(-\alpha_t) - \frac{\lambda t}{2}\|w^{(t-1)} + (\lambda t)^{-1}\alpha_t x_t\|^2$.
  Let $w^{(t)} = \frac{1}{\lambda t}\sum_{i=1}^{t}\alpha_i x_i$
return α

We have the following result for the convergence of dual objective:

Theorem 9 *Assume that ϕ_i is L-Lipschitz for all i. In addition, assume that (ϕ_i, x_i) are iid samples from the same distribution for all $i = 1, \ldots, n$. At the end of Procedure Modified-SGD, we have*

$$\mathbb{E}[D(\alpha^*) - D(\alpha)] \leq \frac{2L^2\log(en)}{\lambda n}.$$

Here the expectation is with respect to the random sampling of $\{(\phi_i, x_i) : i = 1, \ldots, n\}$.

Remark 10 *When λ is relatively large, the convergence rate in Theorem 9 for modified-SGD is better than what we can prove for SDCA. This is because Modified-SGD employs a larger step size at each step t for $D_t(\alpha)$ than the corresponding step size in SDCA for $D(\alpha)$. However, the proof requires us to assume that (ϕ_i, x_i) are randomly drawn from a certain distribution, while this extra randomness assumption is not needed for the convergence of SDCA.*

Procedure SDCA with SGD Initialization

Stage 1: call Procedure Modified-SGD and obtain α
Stage 2: call Procedure SDCA with parameter $\alpha^{(0)} = \alpha$

Theorem 11 *Assume that ϕ_i is L-Lipschitz for all i. In addition, assume that (ϕ_i, x_i) are iid samples from the same distribution for all $i = 1, \ldots, n$. Consider Procedure SDCA with SGD Initialization. To obtain a duality gap of $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \epsilon_P$ at Stage 2, it suffices to have a total number of SDCA iterations of*

$$T \geq T_0 + n + \frac{4L^2}{\lambda \epsilon_P} \geq \lceil n \log(\log(en)) \rceil + n + \frac{20L^2}{\lambda \epsilon_P} .$$

Moreover, when $t \geq T_0$, we have duality sub-optimality bound of $\mathbb{E}[D(\alpha^*) - D(\alpha^{(t)})] \leq \epsilon_P/2$.

Remark 12 *For Lipschitz loss, ideally we would like to have a computational complexity of $O(n + L^2/(\lambda\epsilon_P))$. Theorem 11 shows that SDCA with SGD at first epoch can achieve no worst than $O(n \log(\log n) + L^2/(\lambda\epsilon_P))$, which is very close to the ideal bound. The result is better than that of vanilla SDCA in Theorem 2 when λ is relatively large, which shows a complexity of $O(n \log(n) + L^2/(\lambda\epsilon_P))$. The difference is caused by small step-sizes in the vanilla SDCA, and its negative effect can be observed in practice. That is, the vanilla SDCA tends to have a slower convergence rate than SGD in the first few iterations when λ is relatively large.*

Remark 13 *Similar to Remark 4, for the hinge-loss, the constant 4 in Theorem 11 can be reduced to 1, and the constant 20 can be reduced to 5.*

5. Refined Analysis For Almost Smooth Loss

Our analysis shows that for smooth loss, SDCA converges faster than SGD (linear versus sub-linear convergence). For non-smooth loss, the analysis does not show any advantage of SDCA over SGD. This does not explain the practical observation that SDCA converges faster than SGD asymptotically even for SVM. This section tries to refine the analysis for Lipschitz loss and shows potential advantage of SDCA over SGD asymptotically. Note that the refined analysis of this section relies on quantities that depend on the underlying data distribution, and thus the results are more complicated than those presented earlier. Although precise interpretations of these results will be complex, we will discuss them qualitatively after the theorem statements, and use them to explain the advantage of SDCA over SGD for non-smooth losses.

Although we note that for SVM, Luo and Tseng's analysis (Luo and Tseng, 1992) shows linear convergence of the form $(1-\nu)^k$ for dual sub-optimality after k passes over the data, as we mentioned, ν is proportional to the smallest nonzero eigenvalue of the data Gram matrix $X^\top X$, and hence can be arbitrarily bad when two data points $x_i \neq x_j$ becomes very close to each other. Our analysis uses a completely different argument that avoids this dependency on the data Gram matrix.

The main intuition behind our analysis is that many non-smooth loss functions are nearly smooth everywhere. For example, the hinge loss $\max(0, 1 - uy_i)$ is smooth at any point u such that uy_i is not close to 1. Since a smooth loss has a strongly convex dual (and the strong convexity of the dual

is directly used in our proof to obtain fast rate for smooth loss), the refined analysis in this section relies on the following refined dual strong convexity condition that holds for nearly everywhere smooth loss functions.

Definition 14 *For each i, we define $\gamma_i(\cdot) \geq 0$ so that for all dual variables a and b, and $u \in \partial\phi_i^*(-b)$, we have*

$$\phi_i^*(-a) - \phi_i^*(-b) + u(a-b) \geq \gamma_i(u)|a-b|^2. \tag{4}$$

For the SVM loss, we have $\phi_i(u) = \max(0, 1-uy_i)$, and $\phi_i^*(-a) = -ay_i$, with $ay_i \in [0,1]$ and $y_i \in \{\pm 1\}$. It follows that

$$\phi_i^*(-a) - \phi_i^*(-b) + u(a-b) = (b-a)y_i + u(a-b) = |uy_i - 1||a-b| \geq |uy_i - 1| \cdot |a-b|^2.$$

Therefore we may take $\gamma_i(u) = |uy_i - 1|$.

For the absolute deviation loss, we have $\phi_i(u) = |u - y_i|$, and $\phi^*(-a) = -ay_i$ with $a \in [-1,1]$. It follows that $\gamma_i(u) = |u - y_i|$.

Proposition 15 *Under the assumption of* (4)*. Let $\gamma_i = \gamma_i(w^{*\top} x_i)$, we have the following dual strong convexity inequality:*

$$D(\alpha^*) - D(\alpha) \geq \frac{1}{n}\sum_{i=1}^n \gamma_i |\alpha_i - \alpha_i^*|^2 + \frac{\lambda}{2}(w - w^*)^\top (w - w^*). \tag{5}$$

Moreover, given $w \in \mathbb{R}^d$ and $-a_i \in \partial\phi_i(w^\top x_i)$, we have

$$|(w^* - w)^\top x_i| \geq \gamma_i |a_i - \alpha_i^*|.$$

For SVM, we can take $\gamma_i = |w^{*\top} x_i y_i - 1|$, and for the absolute deviation loss, we may take $\gamma_i = |w^{*\top} x_i - y_i|$. Although some of γ_i can be close to zero, in practice, most γ_i will be away from zero, which means $D(\alpha)$ is strongly convex at nearly all points. Under this assumption, we may establish a convergence result for the dual sub-optimality.

Theorem 16 *Consider Procedure SDCA with $\alpha^{(0)} = 0$. Assume that ϕ_i is L-Lipschitz for all i and it satisfies* (5)*. Define $N(u) = \#\{i : \gamma_i < u\}$. To obtain a dual-suboptimality of $\mathbb{E}[D(\alpha^*) - D(\alpha^t)] \leq \varepsilon_D$, it suffices to have a total number of iterations of*

$$t \geq 2(n/s)\log(2/\varepsilon_D),$$

where $s \in [0,1]$ satisfies $\varepsilon_D \geq 8L^2(s/\lambda n)N(s/\lambda n)/n$.

Remark 17 *if $N(s/\lambda n)/n$ is small, then Theorem 16 is superior to Theorem 2 for the convergence of the dual objective function. We consider three scenarios. The first scenario is when $s = 1$. If $N(1/\lambda n)/n$ is small, and $\varepsilon_D \geq 8L^2(1/\lambda n)N(1/\lambda n)/n$, then the convergence is linear. The second scenario is when there exists s_0 so that $N(s_0/\lambda n) = 0$ (for SVM, it means that $\lambda n|w^{*\top} x_i y_i - 1| \geq s_0$ for all i), and since $\varepsilon_D \geq 8L^2(s_0/\lambda n)N(s_0/\lambda n)/n$, we again have a linear convergence of $(2n/s_0)\log(2/\varepsilon_D)$. In the third scenario, we assume that $N(s/\lambda n)/n = O[(s/\lambda n)^\nu]$ for some $\nu > 0$, we can take $\varepsilon_D = O((s/\lambda n)^{1+\nu})$ and obtain*

$$t \geq O(\lambda^{-1}\varepsilon_D^{-1/(1+\nu)}\log(2/\varepsilon_D)).$$

The $\log(1/\varepsilon_D)$ factor can be removed in this case with a slightly more complex analysis. This result is again superior to Theorem 2 for dual convergence.

The following result shows fast convergence of duality gap using Theorem 16.

Theorem 18 *Consider Procedure SDCA with $\alpha^{(0)} = 0$. Assume that ϕ_i is L-Lipschitz for all i and it satisfies* (4)*. Let $\rho \leq 1$ be the largest eigenvalue of the matrix $n^{-1}\sum_{i=1} x_i x_i^\top$. Define $N(u) = \#\{i : \gamma_i < u\}$. Assume that at time $T_0 \geq n$, we have dual suboptimality of $\mathbb{E}[D(\alpha^*) - D(\alpha^{(T_0)})] \leq \varepsilon_D$, and define*

$$\tilde{\varepsilon}_P = \inf_{\gamma>0}\left[\frac{N(\gamma)}{n}4L^2 + \frac{2\varepsilon_D}{\min(\gamma, \lambda\gamma^2/(2\rho))}\right],$$

then at time $T = 2T_0$, we have

$$\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \varepsilon_D + \frac{\tilde{\varepsilon}_P}{2\lambda T_0}.$$

If for some γ, $N(\gamma)/n$ is small, then Theorem 18 is superior to Theorem 2. Although the general dependency may be complex, the improvement over Theorem 2 can be more easily seen in the special case that $N(\gamma) = 0$ for some $\gamma > 0$. In fact, in this case we have $\tilde{\varepsilon}_P = O(\varepsilon_D)$, and thus

$$\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] = O(\varepsilon_D).$$

This means that the convergence rate for duality gap in Theorem 18 is linear as implied by the linear convergence of ε_D in Theorem 16.

6. Examples

We will specify the SDCA algorithms for a few common loss functions. For simplicity, we only specify the algorithms without SGD initialization. In practice, instead of complete randomization, we may also run in epochs, and each epoch employs a random permutation of the data. We call this variant SDCA-Perm.

Procedure SDCA-Perm$(\alpha^{(0)})$

Let $w^{(0)} = w(\alpha^{(0)})$
Let $t = 0$
Iterate: for epoch $k = 1, 2, \ldots$
  Let $\{i_1, \ldots, i_n\}$ be a random permutation of $\{1, \ldots, n\}$
  Iterate: for $j = 1, 2, \ldots, n$:
    $t \leftarrow t + 1$
    $i = i_j$
    Find $\Delta\alpha_i$ to increase dual (*)
    $\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \Delta\alpha_i e_i$
    $w^{(t)} \leftarrow w^{(t-1)} + (\lambda n)^{-1}\Delta\alpha_i x_i$
Output (Averaging option):
  Let $\bar{\alpha} = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T} \alpha^{(t-1)}$
  Let $\bar{w} = w(\bar{\alpha}) = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T} w^{(t-1)}$
  return $\bar{w}$
Output (Random option):
  Let $\bar{\alpha} = \alpha^{(t)}$ and $\bar{w} = w^{(t)}$ for some random $t \in T_0 + 1, \ldots, T$
  return $\bar{w}$

6.1 Lipschitz Loss

Hinge loss is used in SVM. We have $\phi_i(u) = \max\{0, 1 - y_i u\}$ and $\phi_i^*(-a) = -a y_i$ with $a y_i \in [0,1]$. Absolute deviation loss is used in quantile regression. We have $\phi_i(u) = |u - y_i|$ and $\phi_i^*(-a) = -a y_i$ with $a \in [-1,1]$.

For the hinge loss, step (*) in Procedure SDCA-Perm has a closed form solution as

$$\Delta\alpha_i = y_i \max\left(0, \min\left(1, \frac{1 - x_i^\top w^{(t-1)} y_i}{\|x_i\|^2/(\lambda n)} + \alpha_i^{(t-1)} y_i\right)\right) - \alpha_i^{(t-1)}.$$

For absolute deviation loss, step (*) in Procedure SDCA-Perm has a closed form solution as

$$\Delta\alpha_i = \max\left(-1, \min\left(1, \frac{y_i - x_i^\top w^{(t-1)}}{\|x_i\|^2/(\lambda n)} + \alpha_i^{(t-1)}\right)\right) - \alpha_i^{(t-1)}.$$

Both hinge loss and absolute deviation loss are 1-Lipschitz. Therefore, we expect a convergence behavior of no worse than

$$O\left(n \log n + \frac{1}{\lambda \varepsilon}\right)$$

without SGD initialization based on Theorem 2. The refined analysis in Section 5 suggests a rate that can be significantly better, and this is confirmed with our empirical experiments.

6.2 Smooth Loss

Squared loss is used in ridge regression. We have $\phi_i(u) = (u - y_i)^2$, and $\phi_i^*(-a) = -a y_i + a^2/4$. Log loss is used in logistic regression. We have $\phi_i(u) = \log(1 + \exp(-y_i u))$, and $\phi_i^*(-a) = a y_i \log(a y_i) + (1 - a y_i)\log(1 - a y_i)$ with $a y_i \in [0,1]$.

For squared loss, step (*) in Procedure SDCA-Perm has a closed form solution as

$$\Delta\alpha_i = \frac{y_i - x_i^\top w^{(t-1)} - 0.5\alpha_i^{(t-1)}}{0.5 + \|x_i\|^2/(\lambda n)}.$$

For log loss, step (*) in Procedure SDCA-Perm does not have a closed form solution. However, one may start with the approximate solution,

$$\Delta\alpha_i = \frac{(1 + \exp(x_i^\top w^{(t-1)} y_i))^{-1} y_i - \alpha_i^{(t-1)}}{\max(1, 0.25 + \|x_i\|^2/(\lambda n))},$$

and further use several steps of Newton's update to get a more accurate solution.

Finally, we present a smooth variant of the hinge-loss, as defined below. Recall that the hinge loss function (for positive labels) is $\phi(u) = \max\{0, 1 - u\}$ and we have $\phi^*(-a) = -a$ with $a \in [0,1]$. Consider adding to ϕ^* the term $\frac{\gamma}{2}a^2$ which yields the γ-strongly convex function

$$\tilde{\phi}_\gamma^*(a) = \phi^*(a) + \frac{\gamma}{2}a^2 \,.$$

Then, its conjugate, which is defined below, is $(1/\gamma)$-smooth. We refer to it as the *smoothed hinge-loss* (for positive labels):

$$\tilde{\phi}_\gamma(x) = \max_{a \in [-1,0]} \left[ax - a - \frac{\gamma}{2}a^2\right] = \begin{cases} 0 & x > 1 \\ 1 - x - \gamma/2 & x < 1 - \gamma \\ \frac{1}{2\gamma}(1-x)^2 & \text{otherwise} \end{cases}. \tag{6}$$

For the smoothed hinge loss, step (*) in Procedure SDCA-Perm has a closed form solution as

$$\Delta\alpha_i = y_i \max\left(0, \min\left(1, \frac{1 - x_i^\top w^{(t-1)} y_i - \gamma \alpha_i^{(t-1)} y_i}{\|x_i\|^2/(\lambda n) + \gamma} + \alpha_i^{(t-1)} y_i\right)\right) - \alpha_i^{(t-1)}.$$

Both log loss and squared loss are 1-smooth. The smoothed-hinge loss is $1/\gamma$ smooth. Therefore we expect a convergence behavior of no worse than

$$O\left(\left(n + \frac{1}{\gamma\lambda}\right)\log\frac{1}{\varepsilon}\right).$$

This is confirmed in our empirical experiments.

7. Proofs

We denote by $\partial\phi_i(a)$ the set of sub-gradients of ϕ_i at a. We use the notation $\phi_i'(a)$ to denote some sub-gradient of ϕ_i at a. For convenience, we list the following simple facts about primal and dual formulations, which will used in the proofs. For each i, we have

$$-\alpha_i^* \in \partial\phi_i(w^{*\top} x_i), \quad w^{*\top} x_i \in \partial\phi_i^*(-\alpha_i^*),$$

and

$$w^* = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^* x_i.$$

The proof of our basic results stated in Theorem 5 and Theorem 2 relies on the fact that for SDCA, it is possible to lower bound the expected increase in dual objective by the duality gap. This key observation is stated in Lemma 19. Note that the duality gap can be further lower bounded using dual suboptimality. Therefore Lemma 19 implies a recursion for dual suboptimality which can be solved to obtain the convergence of dual objective. We can then apply Lemma 19 again, and the convergence of dual objective implies an upper bound of the duality gap, which leads to the basic theorems. The more refined results in Section 4 and Section 5 use similar strategies but with Lemma 19 replaced by its variants.

7.1 Proof Of Theorem 5

The key lemma, which estimates the expected increase in dual objective in terms of the duality gap, can be stated as follows.

Lemma 19 *Assume that ϕ_i^* is γ-strongly-convex (where γ can be zero). Then, for any iteration t and any $s \in [0,1]$ we have*

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n}\,\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \left(\frac{s}{n}\right)^2 \frac{G^{(t)}}{2\lambda},$$

where

$$G^{(t)} = \frac{1}{n}\sum_{i=1}^n \left(\|x_i\|^2 - \frac{\gamma(1-s)\lambda n}{s}\right)\mathbb{E}[(u_i^{(t-1)} - \alpha_i^{(t-1)})^2],$$

and $-u_i^{(t-1)} \in \partial\phi_i(x_i^\top w^{(t-1)})$.

Proof Since only the i'th element of α is updated, the improvement in the dual objective can be written as

$$n[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] = \underbrace{\left(-\phi_i^*(-\alpha_i^{(t)}) - \frac{\lambda n}{2}\|w^{(t)}\|^2\right)}_{A} - \underbrace{\left(-\phi_i^*(-\alpha_i^{(t-1)}) - \frac{\lambda n}{2}\|w^{(t-1)}\|^2\right)}_{B}.$$

By the definition of the update we have for all $s \in [0,1]$ that

$$\begin{aligned} A &= \max_{\Delta\alpha_i} -\phi_i^*(-(\alpha_i^{(t-1)} + \Delta\alpha_i)) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1}\Delta\alpha_i x_i\|^2 \\ &\geq -\phi_i^*(-(\alpha_i^{(t-1)} + s(u_i^{(t-1)} - \alpha_i^{(t-1)}))) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1}s(u_i^{(t-1)} - \alpha_i^{(t-1)})x_i\|^2. \end{aligned} \quad (7)$$

From now on, we omit the superscripts and subscripts. Since ϕ^* is γ-strongly convex, we have that

$$\phi^*(-(\alpha + s(u-\alpha))) = \phi^*(s(-u) + (1-s)(-\alpha)) \leq s\phi^*(-u) + (1-s)\phi^*(-\alpha) - \frac{\gamma}{2}s(1-s)(u-\alpha)^2.$$

Combining this with (7) and rearranging terms we obtain that

$$\begin{aligned} A &\geq -s\phi^*(-u) - (1-s)\phi^*(-\alpha) + \frac{\gamma}{2}s(1-s)(u-\alpha)^2 - \frac{\lambda n}{2}\|w + (\lambda n)^{-1}s(u-\alpha)x\|^2 \\ &= -s\phi^*(-u) - (1-s)\phi^*(-\alpha) + \frac{\gamma}{2}s(1-s)(u-\alpha)^2 - \frac{\lambda n}{2}\|w\|^2 - s(u-\alpha)w^\top x \\ &\quad - \frac{s^2(u-\alpha)^2}{2\lambda n}\|x\|^2 \\ &= \underbrace{-s(\phi^*(-u) + uw^\top x)}_{s\phi(w^\top x)} + \underbrace{(-\phi^*(-\alpha) - \frac{\lambda n}{2}\|w\|^2)}_{B} + \frac{s}{2}\left(\gamma(1-s) - \frac{s\|x\|^2}{\lambda n}\right)(u-\alpha)^2 \\ &\quad + s(\phi^*(-\alpha) + \alpha w^\top x), \end{aligned}$$

where we used $-u \in \partial\phi(w^\top x)$ which yields $\phi^*(-u) = -uw^\top x - \phi(w^\top x)$. Therefore

$$A - B \geq s\left[\phi(w^\top x) + \phi^*(-\alpha) + \alpha w^\top x + \left(\frac{\gamma(1-s)}{2} - \frac{s\|x\|^2}{2\lambda n}\right)(u-\alpha)^2\right]. \quad (8)$$

Next note that

$$\begin{aligned} P(w) - D(\alpha) &= \frac{1}{n}\sum_{i=1}^n \phi_i(w^\top x_i) + \frac{\lambda}{2}w^\top w - \left(-\frac{1}{n}\sum_{i=1}^n \phi_i^*(-\alpha_i) - \frac{\lambda}{2}w^\top w\right) \\ &= \frac{1}{n}\sum_{i=1}^n \left(\phi_i(w^\top x_i) + \phi_i^*(-\alpha_i) + \alpha_i w^\top x_i\right). \end{aligned}$$

Therefore, if we take expectation of (8) w.r.t. the choice of i we obtain that

$$\frac{1}{s}\,\mathbb{E}[A - B] \geq \mathbb{E}[P(w) - D(\alpha)] - \frac{s}{2\lambda n} \cdot \underbrace{\frac{1}{n}\sum_{i=1}^n \left(\|x_i\|^2 - \frac{\gamma(1-s)\lambda n}{s}\right)\mathbb{E}(u_i - \alpha_i)^2}_{=G^{(t)}}.$$

We have obtained that

$$\frac{n}{s}\,\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \frac{s\,G^{(t)}}{2\lambda n}\ .$$

Multiplying both sides by s/n concludes the proof of the lemma. ■

We also use the following simple lemma:

Lemma 20 *For all* α, $D(\alpha) \leq P(w^*) \leq P(0) \leq 1$. *In addition,* $D(0) \geq 0$.

Proof The first inequality is by weak duality, the second is by the optimality of w^*, and the third by the assumption that $\phi_i(0) \leq 1$. For the last inequality we use $-\phi_i^*(0) = -\max_z(0 - \phi_i(z)) = \min_z \phi_i(z) \geq 0$, which yields $D(0) \geq 0$. ■

Equipped with the above lemmas we are ready to prove Theorem 5.

Proof [Proof of Theorem 5] The assumption that ϕ_i is $(1/\gamma)$-smooth implies that ϕ_i^* is γ-strongly-convex. We will apply Lemma 19 with $s = \frac{\lambda n\gamma}{1+\lambda n\gamma} \in [0,1]$. Recall that $\|x_i\| \leq 1$. Therefore, the choice of s implies that $\|x_i\|^2 - \frac{\gamma(1-s)\lambda n}{s} \leq 0$, and hence $G^{(t)} \leq 0$ for all t. This yields,

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n}\,\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})]\ .$$

But since $\varepsilon_D^{(t-1)} := D(\alpha^*) - D(\alpha^{(t-1)}) \leq P(w^{(t-1)}) - D(\alpha^{(t-1)})$ and $D(\alpha^{(t)}) - D(\alpha^{(t-1)}) = \varepsilon_D^{(t-1)} - \varepsilon_D^{(t)}$, we obtain that

$$\mathbb{E}[\varepsilon_D^{(t)}] \leq \left(1 - \tfrac{s}{n}\right)\mathbb{E}[\varepsilon_D^{(t-1)}] \leq \left(1 - \tfrac{s}{n}\right)^t \mathbb{E}[\varepsilon_D^{(0)}] \leq \left(1 - \tfrac{s}{n}\right)^t \leq \exp(-st/n) = \exp\left(-\frac{\lambda\gamma t}{1+\lambda\gamma n}\right)\ .$$

This would be smaller than ε_D if

$$t \geq \left(n + \tfrac{1}{\lambda\gamma}\right)\log(1/\varepsilon_D)\ .$$

It implies that

$$\mathbb{E}[P(w^{(t)}) - D(\alpha^{(t)})] \leq \frac{n}{s}\,\mathbb{E}[\varepsilon_D^{(t)} - \varepsilon_D^{(t+1)}] \leq \frac{n}{s}\,\mathbb{E}[\varepsilon_D^{(t)}]. \tag{9}$$

So, requiring $\varepsilon_D^{(t)} \leq \frac{s}{n}\varepsilon_P$ we obtain a duality gap of at most ε_P. This means that we should require

$$t \geq \left(n + \tfrac{1}{\lambda\gamma}\right)\log\left((n + \tfrac{1}{\lambda\gamma}) \cdot \tfrac{1}{\varepsilon_P}\right)\ ,$$

which proves the first part of Theorem 5.

Next, we sum (9) over $t = T_0, \ldots, T-1$ to obtain

$$\mathbb{E}\left[\frac{1}{T-T_0}\sum_{t=T_0}^{T-1}(P(w^{(t)}) - D(\alpha^{(t)}))\right] \leq \frac{n}{s(T-T_0)}\,\mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})].$$

Now, if we choose $\bar{w}, \bar{\alpha}$ to be either the average vectors or a randomly chosen vector over $t \in \{T_0+1, \ldots, T\}$, then the above implies

$$\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \frac{n}{s(T-T_0)} \mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})] \leq \frac{n}{s(T-T_0)} \mathbb{E}[\varepsilon_D^{(T_0)})].$$

It follows that in order to obtain a result of $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \varepsilon_P$, we only need to have

$$\mathbb{E}[\varepsilon_D^{(T_0)})] \leq \frac{s(T-T_0)\varepsilon_P}{n} = \frac{(T-T_0)\varepsilon_P}{n + \frac{1}{\lambda\gamma}}.$$

This implies the second part of Theorem 5, and concludes the proof. ■

7.2 Proof Of Theorem 2

Next, we turn to the case of Lipschitz loss function. We rely on the following lemma.

Lemma 21 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be an L-Lipschitz function. Then, for any α s.t. $|\alpha| > L$ we have that $\phi^*(\alpha) = \infty$.*

Proof Fix some $\alpha > L$. By definition of the conjugate we have

$$\begin{aligned}
\phi^*(\alpha) &= \sup_x [\alpha x - \phi(x)] \\
&\geq -\phi(0) + \sup_x [\alpha x - (\phi(x) - \phi(0))] \\
&\geq -\phi(0) + \sup_x [\alpha x - L|x - 0|] \\
&\geq -\phi(0) + \sup_{x>0} (\alpha - L)\, x = \infty \,.
\end{aligned}$$

Similar argument holds for $\alpha < -L$. ■

A direct corollary of the above lemma is:

Lemma 22 *Suppose that for all i, ϕ_i is L-Lipschitz. Let $G^{(t)}$ be as defined in Lemma 19 (with $\gamma = 0$). Then, $G^{(t)} \leq 4L^2$.*

Proof Using Lemma 21 we know that $|\alpha_i^{(t-1)}| \leq L$, and in addition by the relation of Lipschitz and sub-gradients we have $|u_i^{(t-1)}| \leq L$. Thus, $(u_i^{(t-1)} - \alpha_i^{(t-1)})^2 \leq 4L^2$, and the proof follows. ■

We are now ready to prove Theorem 2.

Proof [Proof of Theorem 2] Let $G = \max_t G^{(t)}$ and note that by Lemma 22 we have $G \leq 4L^2$. Lemma 19, with $\gamma = 0$, tells us that

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{n} \mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \left(\frac{s}{n}\right)^2 \frac{G}{2\lambda}\,, \tag{10}$$

which implies that

$$\mathbb{E}[\varepsilon_D^{(t)}] \leq \left(1 - \tfrac{s}{n}\right) \mathbb{E}[\varepsilon_D^{(t-1)}] + \left(\tfrac{s}{n}\right)^2 \tfrac{G}{2\lambda}\,.$$

We next show that the above yields

$$\mathbb{E}[\varepsilon_D^{(t)}] \leq \frac{2G}{\lambda(2n+t-t_0)} \tag{11}$$

for all $t \geq t_0 = \max(0, \lceil n\log(2\lambda n\varepsilon_D^{(0)}/G)\rceil)$. Indeed, let us choose $s=1$, then at $t=t_0$, we have

$$\mathbb{E}[\varepsilon_D^{(t)}] \leq \left(1-\tfrac{1}{n}\right)^t \varepsilon_D^{(0)} + \tfrac{G}{2\lambda n^2}\tfrac{1}{1-(1-1/n)} \leq e^{-t/n}\varepsilon_D^{(0)} + \tfrac{G}{2\lambda n} \leq \tfrac{G}{\lambda n}\ .$$

This implies that (11) holds at $t=t_0$. For $t > t_0$ we use an inductive argument. Suppose the claim holds for $t-1$, therefore

$$\mathbb{E}[\varepsilon_D^{(t)}] \leq \left(1-\tfrac{s}{n}\right)\mathbb{E}[\varepsilon_D^{(t-1)}] + \left(\tfrac{s}{n}\right)^2 \tfrac{G}{2\lambda} \leq \left(1-\tfrac{s}{n}\right)\tfrac{2G}{\lambda(2n+t-1-t_0)} + \left(\tfrac{s}{n}\right)^2\tfrac{G}{2\lambda}.$$

Choosing $s = 2n/(2n-t_0+t-1) \in [0,1]$ yields

$$\begin{aligned}
\mathbb{E}[\varepsilon_D^{(t)}] &\leq \left(1-\tfrac{2}{2n-t_0+t-1}\right)\tfrac{2G}{\lambda(2n-t_0+t-1)} + \left(\tfrac{2}{2n-t_0+t-1}\right)^2\tfrac{G}{2\lambda} \\
&= \tfrac{2G}{\lambda(2n-t_0+t-1)}\left(1-\tfrac{1}{2n-t_0+t-1}\right) \\
&= \tfrac{2G}{\lambda(2n-t_0+t-1)}\tfrac{2n-t_0+t-2}{2n-t_0+t-1} \\
&\leq \tfrac{2G}{\lambda(2n-t_0+t-1)}\tfrac{2n-t_0+t-1}{2n-t_0+t} \\
&= \tfrac{2G}{\lambda(2n-t_0+t)}\ .
\end{aligned}$$

This provides a bound on the dual sub-optimality. We next turn to bound the duality gap. Summing (10) over $t = T_0+1,\ldots,T$ and rearranging terms we obtain that

$$\mathbb{E}\left[\frac{1}{T-T_0}\sum_{t=T_0+1}^{T}\left(P(w^{(t-1)}) - D(\alpha^{(t-1)})\right)\right] \leq \frac{n}{s(T-T_0)}\mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})] + \frac{sG}{2\lambda n}.$$

Now, if we choose $\bar{w}, \bar{\alpha}$ to be either the average vectors or a randomly chosen vector over $t \in \{T_0+1,\ldots,T\}$, then the above implies

$$\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \frac{n}{s(T-T_0)}\mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})] + \frac{sG}{2\lambda n}\ .$$

If $T \geq n+T_0$ and $T_0 \geq t_0$, we can set $s = n/(T-T_0)$ and combining with (11) we obtain

$$\begin{aligned}
\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] &\leq \mathbb{E}[D(\alpha^{(T)}) - D(\alpha^{(T_0)})] + \frac{G}{2\lambda(T-T_0)} \\
&\leq \mathbb{E}[D(\alpha^*) - D(\alpha^{(T_0)})] + \frac{G}{2\lambda(T-T_0)} \\
&\leq \frac{2G}{\lambda(2n-t_0+T_0)} + \frac{G}{2\lambda(T-T_0)}\ .
\end{aligned}$$

A sufficient condition for the above to be smaller than ε_P is that $T_0 \geq \frac{4G}{\lambda\varepsilon_P} - 2n + t_0$ and $T \geq T_0 + \frac{G}{\lambda\varepsilon_P}$. It also implies that $\mathbb{E}[D(\alpha^*) - D(\alpha^{(T_0)})] \leq \varepsilon_P/2$. Since we also need $T_0 \geq t_0$ and $T - T_0 \geq n$, the overall number of required iterations can be

$$T_0 \geq \max\{t_0, 4G/(\lambda\varepsilon_P) - 2n + t_0\}, \quad T - T_0 \geq \max\{n, G/(\lambda\varepsilon_P)\}.$$

We conclude the proof by noticing that $\varepsilon_D^{(0)} \leq 1$ using Lemma 20, which implies that $t_0 \leq \max(0, \lceil n \log(2\lambda n/G) \rceil)$. ■

7.3 Proof Of Theorem 9

We assume that (ϕ_t, x_t) are randomly drawn from a distribution D, and define the population optimizer

$$w_D^* = \operatorname*{argmin}_w P_D(w), \qquad P_D(w) = \mathbb{E}_{(\phi,x)\sim D}\left[\phi(w^\top x) + \frac{\lambda}{2}\|w\|^2\right].$$

By definition, we have $P(w^*) \leq P(w_D^*)$ for any specific realization of $\{(\phi_t, x_t) : t = 1, \ldots, n\}$. Therefore

$$\mathbb{E}\, P(w^*) \leq \mathbb{E}\, P(w_D^*) = \mathbb{E}\, P_D(w_D^*),$$

where the expectation is with respect to the choice of examples, and note that both $P(\cdot)$ and w^* are sample dependent.

After each step t, we let $\alpha^{(t)} = [\alpha_1, \ldots, \alpha_t]$, and let $-u \in \partial\phi_{t+1}(x_{t+1}^\top w^{(t)})$. We have, for all t,

$$\begin{aligned}
&(t+1)D_{t+1}(\alpha^{(t+1)}) - tD_t(\alpha^{(t)}) = -\phi_{t+1}^*(-\alpha_{t+1}^{(t+1)}) - (t+1)\frac{\lambda}{2}\|w^{(t+1)}\|^2 + t\frac{\lambda}{2}\|w^{(t)}\|^2 \\
&= -\phi_{t+1}^*(-\alpha_{t+1}^{(t+1)}) - \frac{1}{2(t+1)\lambda}\|\lambda t w^{(t)} + \alpha_{t+1}^{(t+1)} x_{t+1}\|^2 + \frac{1}{2t\lambda}\|\lambda t w^{(t)}\|^2 \\
&\geq -\phi_{t+1}^*(-u) - \frac{1}{2(t+1)\lambda}\|\lambda t w^{(t)} + u x_{t+1}\|^2 + \frac{1}{2t\lambda}\|\lambda t w^{(t)}\|^2 \\
&= -\phi_{t+1}^*(-u) - \frac{t}{t+1}x_{t+1}^\top w^{(t)}\, u + \frac{1}{2\lambda}\left(\frac{1}{t} - \frac{1}{t+1}\right)\|\lambda t w^{(t)}\|^2 - \frac{u^2\|x_{t+1}\|^2}{2(t+1)\lambda} \\
&= -\phi_{t+1}^*(-u) - x_{t+1}^\top w^{(t)}\, u + \left(1 - \frac{t}{t+1}\right)x_{t+1}^\top w^{(t)}\, u + \frac{1}{2(t+1)\lambda}\left(\frac{\|\lambda t w^{(t)}\|^2}{t} - u^2\|x_{t+1}\|^2\right) \\
&= \phi_{t+1}(x_{t+1}^\top w^{(t)}) + \frac{1}{2(t+1)\lambda}\left(2\lambda x_{t+1}^\top w^{(t)}\, u + \frac{\|\lambda t w^{(t)}\|^2}{t} - u^2\|x_{t+1}\|^2\right) \\
&= \phi_{t+1}(x_{t+1}^\top w^{(t)}) + \frac{\lambda}{2}\|w^{(t)}\|^2 + \frac{1}{2(t+1)\lambda}\left(2\lambda x_{t+1}^\top w^{(t)}\, u - \|\lambda w^{(t)}\|^2 - u^2\|x_{t+1}\|^2\right) \\
&= \phi_{t+1}(w^{(t)\ \top} x_{t+1}) + \frac{\lambda}{2}\|w^{(t)}\|^2 - \frac{\|\lambda w^{(t)} - u x_{t+1}\|^2}{2(t+1)\lambda}\,.
\end{aligned}$$

The inequality above can be obtained by noticing that the choice of $-\alpha_{t+1}^{(t+1)}$ maximizes the dual objective. In the derivation of the equalities we have used basic algebra as well as the equation $-\phi_{t+1}^*(-u) - x_{t+1}^\top w^{(t)}\, u = \phi_{t+1}(x_{t+1}^\top w^{(t)})$ which follows from $-u \in \partial\phi_{t+1}(x_{t+1}^\top w^{(t)})$. Next we note that $\|\lambda w^{(t)} - u x_{t+1}\| \leq 2L$ (where we used the triangle inequality, the definition of $w^{(t)}$, and Lemma 21). Therefore,

$$(t+1)D_{t+1}(\alpha^{(t+1)}) - tD_t(\alpha^{(t)}) \geq \phi_{t+1}(w^{(t)\ \top} x_{t+1}) + \frac{\lambda}{2}\|w^{(t)}\|^2 - \frac{2L^2}{(t+1)\lambda}\,.$$

Taking expectation *with respect to the choice of the examples*, and note that the $(t+1)$'th example does not depend on $w^{(t)}$ we obtain that

$$\begin{aligned}
&\mathbb{E}[(t+1)D_{t+1}(\alpha^{(t+1)}) - tD_t(\alpha^{(t)})] \\
\geq &\mathbb{E}[P_D(w^{(t)})] - \frac{2L^2}{(t+1)\lambda} \geq \mathbb{E}[P_D(w_D^*)] - \frac{2L^2}{(t+1)\lambda} \\
\geq &\mathbb{E}[P(w^*)] - \frac{2L^2}{(t+1)\lambda} = \mathbb{E}[D(\alpha^*)] - \frac{2L^2}{(t+1)\lambda}.
\end{aligned}$$

Using Lemma 20 we know that $D_t(\alpha^{(t)}) \geq 0$ for all t. Therefore, by summing the above over t we obtain that

$$\mathbb{E}[nD(\alpha^{(n)})] \geq n\,\mathbb{E}[D(\alpha^*)] - \frac{2L^2\log(en)}{\lambda},$$

which yields

$$\mathbb{E}[D(\alpha^*) - D(\alpha^{(n)})] \leq \frac{2L^2\log(en)}{\lambda n}.$$

7.4 Proof Of Theorem 11

The proof is identical to the proof of Theorem 2. We just need to notice that at the end of the first stage, we have $\mathbb{E}\varepsilon_D^{(0)} \leq 2L^2\log(en)/(\lambda n)$. It implies that $t_0 \leq \max(0, \lceil n\log(2\lambda n \cdot 2L^2\log(en)/(\lambda nG))\rceil)$.

7.5 Proof Of Proposition 15

Consider any feasible dual variable α and the corresponding $w = w(\alpha)$. Since

$$w = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i x_i, \qquad w^* = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^* x_i,$$

we have

$$\lambda(w - w^*)^\top w^* = \frac{1}{n}\sum_{i=1}^n (\alpha_i - \alpha_i^*) w^{*\top} x_i.$$

Therefore

$$\begin{aligned}
&D(\alpha^*) - D(\alpha) \\
=&\frac{1}{n}\sum_{i=1}^n \left[\phi_i^*(-\alpha_i) - \phi_i^*(-\alpha_i^*) + (\alpha_i - \alpha_i^*) w^{*\top} x_i\right] + \frac{\lambda}{2}[w^\top w - w^{*\top} w^* - 2(w - w^*)^\top w^*] \\
=&\frac{1}{n}\sum_{i=1}^n \left[\phi_i^*(-\alpha_i) - \phi_i^*(-\alpha_i^*) + (\alpha_i - \alpha_i^*) w^{*\top} x_i\right] + \frac{\lambda}{2}(w - w^*)^\top (w - w^*).
\end{aligned}$$

Since $w^{*\top} x_i \in \partial\phi_i^*(-\alpha_i^*)$, we have

$$\phi_i^*(-\alpha_i) - \phi_i^*(-\alpha_i^*) + (\alpha_i - \alpha_i^*) w^{*\top} x_i \geq \gamma_i(\alpha_i - \alpha_i^*)^2.$$

By combining the previous two displayed inequalities, we obtain the first desired bound.

Next, we let $u = w^{*\top} x_i$, $v = w^\top x_i$. Since $-a_i \in \partial\phi_i(v)$ and $-\alpha_i^* \in \partial\phi_i(u)$, it follows that $u \in \partial\phi_i^*(-\alpha_i^*)$ and $v \in \partial\phi_i^*(-a_i)$. Therefore

$$\begin{aligned}
&|u-v| \cdot |\alpha_i^* - a_i| \\
=&\underbrace{[\phi_i^*(-a_i) - \phi_i^*(-\alpha_i^*) + u(a_i - \alpha_i^*)]}_{\geq 0} + \underbrace{[\phi_i^*(-\alpha_i^*) - \phi_i^*(-a_i) + v(\alpha_i^* - a_i)]}_{\geq 0} \\
\geq&\phi_i^*(-a_i) - \phi_i^*(-\alpha_i^*) + u(a_i - \alpha_i^*) \geq \gamma_i(u)|a_i - \alpha_i^*|^2.
\end{aligned}$$

This implies the second bound.

7.6 Proof Of Theorem 16

The following lemma is very similar to Lemma 19 with nearly identical proof, but it focuses only on the convergence of dual objective function using (5).

Lemma 23 *Assume that* (5) *is valid. Then for any iteration t and any $s \in [0,1]$ we have*

$$\mathbb{E}[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] \geq \frac{s}{2n}\,\mathbb{E}[D(\alpha^*) - D(\alpha^{(t-1)})] + \frac{3s\lambda}{4n}\|w^* - w^{(t-1)}\|^2 - \left(\frac{s}{n}\right)^2 \frac{G_*^{(t)}(s)}{2\lambda}\,,$$

where

$$G_*^{(t)}(s) = \frac{1}{n}\sum_{i=1}^n \left(\|x_i\|^2 - \frac{\gamma_i \lambda n}{s}\right) \mathbb{E}[(\alpha_i^* - \alpha_i^{(t-1)})^2].$$

Proof Since only the i'th element of α is updated, the improvement in the dual objective can be written as

$$n[D(\alpha^{(t)}) - D(\alpha^{(t-1)})] = \underbrace{\left(-\phi^*(-\alpha_i^{(t)}) - \frac{\lambda n}{2}\|w^{(t)}\|^2\right)}_{A_i} - \underbrace{\left(-\phi^*(-\alpha_i^{(t-1)}) - \frac{\lambda n}{2}\|w^{(t-1)}\|^2\right)}_{B_i}.$$

By the definition of the update we have for all $s \in [0,1]$ that

$$\begin{aligned}
A_i &= \max_{\Delta\alpha_i} -\phi^*(-(\alpha_i^{(t-1)} + \Delta\alpha_i)) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1}\Delta\alpha_i x_i\|^2 \\
&\geq -\phi^*(-(\alpha_i^{(t-1)} + s(\alpha_i^* - \alpha_i^{(t-1)}))) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1} s(\alpha_i^* - \alpha_i^{(t-1)})x_i\|^2.
\end{aligned}$$

We can now apply the Jensen's inequality to obtain

$$\begin{aligned}
A_i &\geq -s\phi_i^*(-\alpha_i^*) - (1-s)\phi_i^*(-\alpha_i^{(t-1)}) - \frac{\lambda n}{2}\|w^{(t-1)} + (\lambda n)^{-1} s(\alpha_i^* - \alpha_i^{(t-1)})x_i\|^2 \\
&= -s[\phi_i^*(-\alpha_i^*) - \phi_i^*(-\alpha_i^{(t-1)})] \underbrace{-\phi_i^*(-\alpha_i^{(t-1)}) - \frac{\lambda n}{2}\|w^{(t-1)}\|^2}_{B_i} - s(\alpha_i^* - \alpha_i^{(t-1)})x_i^\top w^{(t-1)} \\
&\quad - \frac{s^2(\alpha_i^* - \alpha_i^{(t-1)})^2}{2\lambda n}\|x_i\|^2.
\end{aligned}$$

By summing over $i = 1, \ldots, n$, we obtain

$$\begin{aligned}\sum_{i=1}^n [A_i - B_i] \geq & -s\sum_{i=1}^n [\phi_i^*(-\alpha_i^*) - \phi_i^*(-\alpha_i^{(t-1)})] - s\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)}) x_i^\top w^{(t-1)} \\ & - \frac{s^2}{2\lambda n}\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)})^2 \|x_i\|^2 \\ = & -s\sum_{i=1}^n [\phi_i^*(-\alpha_i^*) - \phi_i^*(-\alpha_i^{(t-1)}) + \lambda (w^* - w^{(t-1)})^\top w^{(t-1)}] \\ & - \frac{s^2}{2\lambda n}\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)})^2 \|x_i\|^2,\end{aligned}$$

where the equality follows from $\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)}) x_i = \lambda n (w^* - w^{(t-1)})$. By rearranging the terms on the right hand side using $(w^* - w^{(t-1)})^\top w^{(t-1)} = \|w^*\|^2/2 - \|w^{(t-1)}\|^2/2 - \|w^* - w^{(t-1)}\|^2/2$, we obtain

$$\begin{aligned}& \sum_{i=1}^n [A_i - B_i] \\ \geq & -s\sum_{i=1}^n \left[\phi_i^*(-\alpha_i^*) - \phi_i^*(-\alpha_i^{(t-1)}) + \frac{\lambda}{2}\|w^*\|^2 - \frac{\lambda}{2}\|w^{(t-1)}\|^2\right] + \frac{\lambda s n}{2}\|w^* - w^{(t-1)}\|^2 \\ & - \frac{s^2}{2\lambda n}\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)})^2 \|x_i\|^2 \\ = & sn[D(\alpha^*) - D(\alpha^{(t-1)})] + \frac{s\lambda n}{2}\|w^* - w^{(t-1)}\|^2 - \frac{s^2}{2\lambda n}\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)})^2 \|x_i\|^2.\end{aligned}$$

We can now apply (5) to obtain

$$\begin{aligned}\sum_{i=1}^n [A_i - B_i] \geq & \frac{sn}{2}[D(\alpha^*) - D(\alpha^{(t-1)})] + \frac{3s\lambda n}{4}\|w^* - w^{(t-1)}\|^2 \\ & - \frac{s^2}{2\lambda n}\sum_{i=1}^n (\alpha_i^* - \alpha_i^{(t-1)})^2 (\|x_i\|^2 - \gamma_i \lambda n/s).\end{aligned}$$

This implies the desired result. ■

Lemma 24 *Suppose that for all i, ϕ_i is L-Lipschitz. Let $G_*^{(t)}$ be as defined in Lemma 23. Then*

$$G_*^{(t)}(s) \leq \frac{4L^2 N(s/(\lambda n))}{n}.$$

Proof Similarly to the proof of Lemma 22, we know that $(\alpha_i^* - \alpha_i^{(t-1)})^2 \leq 4L^2$. Moreover, $\|x_i\|^2 \leq 1$, and $\|x_i\|^2 - \frac{\gamma_i \lambda n}{s} \leq 0$ when $\gamma_i \geq s/(\lambda n)$. Therefore there are no more than $N(s/(\lambda n))$ data points i such that $\|x_i\|^2 - \frac{\gamma_i \lambda n}{s}$ is positive. The desired result follows from these facts. ■

Proof [Proof of Theorem 16] Let $\varepsilon_D^{(t)} = \mathbb{E}[D(\alpha^*) - D(\alpha^{(t)})]$, and $G_*(s) = 4L^2 N(s/\lambda n)/n$. We obtain from Lemma 23 and Lemma 24 that

$$\varepsilon_D^{(t)} \leq (1 - s/(2n))\varepsilon_D^{(t-1)} + \left(\frac{s}{n}\right)^2 \frac{G_*(s)}{2\lambda}.$$

It follows that for all $t > 0$ we have

$$\begin{aligned}\varepsilon_D^{(t)} \leq & (1 - s/(2n))^t \varepsilon_D^{(0)} + \frac{1}{1 - (1 - s/(2n))} \left(\frac{s}{n}\right)^2 \frac{G_*(s)}{2\lambda} \\ \leq & e^{-st/2n} + \left(\frac{s}{n}\right) \frac{G_*(s)}{\lambda} \leq e^{-st/2n} + \varepsilon_D/2.\end{aligned}$$

It follows that when

$$t \geq (2n/s)\log(2/\varepsilon_D),$$

we have $\varepsilon_D^{(t)} \leq \varepsilon_D$. ■

7.7 Proof Of Theorem 18

Let $\varepsilon_D^{(t)} = \mathbb{E}[D(\alpha^*) - D(\alpha^{(t)})]$. From Proposition 15, we know that for all $t \geq T_0$:

$$\begin{aligned}\varepsilon_D^{(t)} \geq & \frac{1}{n}\sum_{i=1}^n \left[\gamma_i \mathbb{E}\,|\alpha_i^{(t)} - \alpha_i^*|^2 + \frac{\lambda}{2\rho}\mathbb{E}((w^{(t)} - w^*)^\top x_i)^2\right] \\ \geq & \frac{1}{n}\sum_{i=1}^n \left[\gamma_i \mathbb{E}\,|\alpha_i^{(t)} - \alpha_i^*|^2 + \frac{\lambda\gamma_i^2}{2\rho}\mathbb{E}(u_i^{(t-1)} - \alpha_i^*)^2\right],\end{aligned}$$

where $-u_i^{(t-1)} \in \partial\phi_i(x_i^\top w^{(t)})$. It follows that given any $\gamma > 0$, we have

$$\begin{aligned}& \frac{1}{n}\sum_{i=1}^n \mathbb{E}\,|\alpha_i^{(t)} - u_i^{(t-1)}|^2 \\ \leq & \frac{N(\gamma)}{n}\sup_i \mathbb{E}\,|\alpha_i^{(t)} - u_i^{(t-1)}|^2 + \frac{2}{n}\sum_{i:\gamma_i \geq \gamma}\left[\mathbb{E}\,|\alpha_i^{(t)} - \alpha_i^*|^2 + \mathbb{E}(u_i^{(t-1)} - \alpha_i^*)^2\right] \\ \leq & \frac{N(\gamma)}{n}\sup_i \mathbb{E}\,|\alpha_i^{(t)} - u_i^{(t-1)}|^2 + \frac{\frac{2}{n}\sum_{i=1}^n \left[\gamma_i \mathbb{E}\,|\alpha_i^{(t)} - \alpha_i^*|^2 + \frac{\lambda\gamma_i^2}{2\rho}\mathbb{E}(u_i^{(t-1)} - \alpha_i^*)^2\right]}{\min(\gamma, \lambda\gamma^2/(2\rho))} \\ \leq & \frac{N(\gamma)}{n} 4L^2 + \frac{2\varepsilon_D^{(t)}}{\min(\gamma, \lambda\gamma^2/(2\rho))},\end{aligned}$$

where Lemma 22 is used for the last inequality. Since γ is arbitrary and $\varepsilon_D^{(t)} \leq \varepsilon_D$, it follows that

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\,|\alpha_i^{(t)} - u_i^{(t-1)}|^2 \leq \tilde{\varepsilon}_P.$$

Now plug into Lemma 19, we obtain for all $t \geq T_0 + 1$:

$$\begin{aligned}
&\epsilon_D^{(t-1)} - \epsilon_D^{(t)} \\
\geq &\frac{s}{n}\,\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \left(\frac{s}{n}\right)^2 \frac{1}{2\lambda n}\sum_{i=1}^{n}\mathbb{E}[(u_i^{(t-1)} - \alpha_i^{(t-1)})^2] \\
\geq &\frac{s}{n}\,\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)})] - \left(\frac{s}{n}\right)^2 \frac{\tilde{\epsilon}_P}{2\lambda}.
\end{aligned}$$

By taking $s = n/T_0$, and summing over $t = T_0 + 1, \ldots, 2T_0 = T$, we obtain

$$\epsilon_D \geq \epsilon_D^{(T_0)} - \epsilon_D^{(T)} \geq \mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] - \frac{\tilde{\epsilon}_P}{2\lambda T_0}.$$

This proves the desired bound.

8. Experimental Results

In this section we demonstrate the tightness of our theory. All our experiments are performed with the smooth variant of the hinge-loss defined in (6), where the value of γ is taken from the set $\{0, 0.01, 0.1, 1\}$. Note that for $\gamma = 0$ we obtain the vanilla non-smooth hinge-loss.

In the experiments, we use ϵ_D to denote the dual sub-optimality, and ϵ_P to denote the primal sub-optimality (note that this is different than the notation in our analysis which uses ϵ_P to denote the duality gap). It follows that $\epsilon_D + \epsilon_P$ is the duality gap.

8.1 Data

The experiments were performed on three large data sets with very different feature counts and sparsity, which were kindly provided by Thorsten Joachims. The astro-ph data set classifies abstracts of papers from the physics ArXiv according to whether they belong in the astro-physics section; CCAT is a classification task taken from the Reuters RCV1 collection; and cov1 is class 1 of the covertype data set of Blackard, Jock & Dean. The following table provides details of the data set characteristics.

Data Set	Training Size	Testing Size	Features	Sparsity
astro-ph	29882	32487	99757	0.08%
CCAT	781265	23149	47236	0.16%
cov1	522911	58101	54	22.22%

8.2 Linear Convergence For Smooth Hinge-loss

Our first experiments are with ϕ_γ where we set $\gamma = 1$. The goal of the experiment is to show that the convergence is indeed linear. We ran the SDCA algorithm for solving the regularized loss minimization problem with different values of regularization parameter λ. Figure 1 shows the results. Note that a logarithmic scale is used for the vertical axis. Therefore, a straight line corresponds to linear convergence. We indeed observe linear convergence for the duality gap.

8.3 Convergence For Non-smooth Hinge-loss

Next we experiment with the original hinge loss, which is 1-Lipschitz but is not smooth. We again ran the SDCA algorithm for solving the regularized loss minimization problem with different values

of regularization parameter λ. Figure 2 shows the results. As expected, the overall convergence rate is slower than the case of a smoothed hinge-loss. However, it is also apparent that for large values of λ a linear convergence is still exhibited, as expected according to our refined analysis. The bounds plotted are based on Theorem 2, which are slower than what we observe, as expected from the refined analysis in Section 5.

8.4 Effect Of Smoothness Parameter

We next show the effect of the smoothness parameter. Figure 3 shows the effect of the smoothness parameter on the rate of convergence. As can be seen, the convergence becomes faster as the loss function becomes smoother. However, the difference is more dominant when λ decreases.

Figure 4 shows the effect of the smoothness parameter on the zero-one test error. It is noticeable that even though the non-smooth hinge-loss is considered a tighter approximation of the zero-one error, in most cases, the smoothed hinge-loss actually provides a lower test error than the non-smooth hinge-loss. In any case, it is apparent that the smooth hinge-loss decreases the zero-one test error faster than the non-smooth hinge-loss.

8.5 Cyclic vs. Stochastic vs. Random Permutation

In Figure 5 we compare choosing dual variables at random with repetitions (as done in SDCA) vs. choosing dual variables using a random permutation at each epoch (as done in SDCA-Perm) vs. choosing dual variables in a fixed cyclic order (that was chosen once at random). As can be seen, a cyclic order does not lead to linear convergence and yields actual convergence rate much slower than the other methods and even worse than our bound. As mentioned before, some of the earlier analyses such as Luo and Tseng (1992) can be applied both to stochastic and to cyclic dual coordinate ascent methods with similar results. This means that their analysis, which can be no better than the behavior of cyclic dual coordinate ascent, is inferior to our analysis. Finally, we also observe that SDCA-Perm is sometimes faster than SDCA.

8.6 Comparison To SGD

We next compare SDCA to Stochastic Gradient Descent (SGD). In particular, we implemented SGD with the update rule $w^{(t+1)} = (1 - 1/t)w^{(t)} - \frac{1}{\lambda t}\phi_i'(w^{(t)\top} x_i)x_i$, where i is chosen uniformly at random and ϕ_i' denotes a sub-gradient of ϕ_i. One clear advantage of SDCA is the availability of a clear stopping condition (by calculating the duality gap). In Figure 6 and Figure 7 we present the primal sub-optimality of SDCA, SDCA-Perm, and SGD. As can be seen, SDCA converges faster than SGD in most regimes. SGD can be better if both λ is high and one performs a very small number of epochs. This is in line with our theory of Section 4. However, SDCA quickly catches up.

In Figure 8 we compare the zero-one test error of SDCA, when working with the smooth hinge-loss ($\gamma = 1$) to the zero-one test error of SGD, when working with the non-smooth hinge-loss. As can be seen, SDCA with the smooth hinge-loss achieves the smallest zero-one test error faster than SGD.

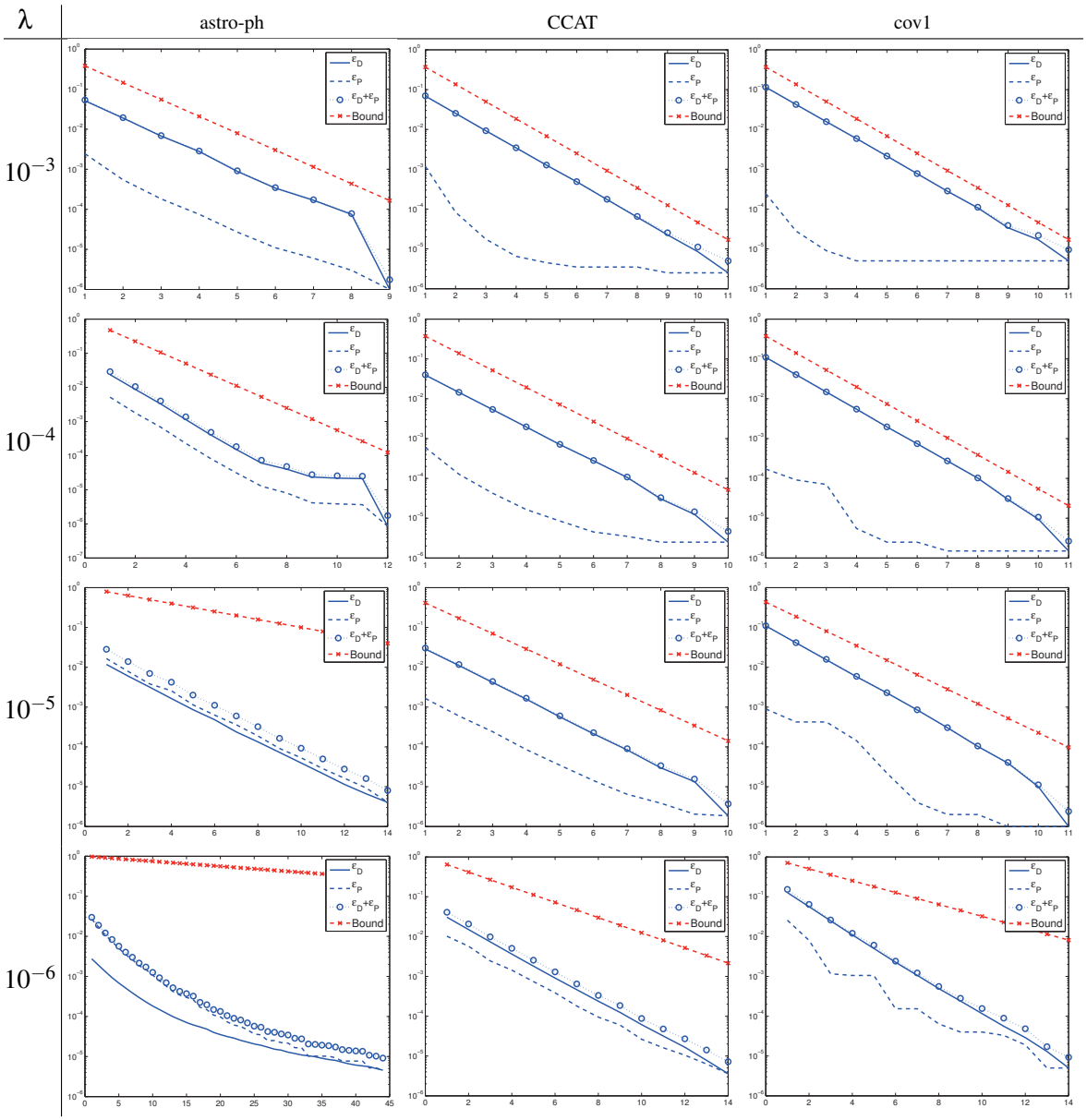


Figure 1: Experiments with the smoothed hinge-loss ($\gamma = 1$). The primal and dual sub-optimality, the duality gap, and our bound are depicted as a function of the number of epochs, on the astro-ph (left), CCAT (center) and cov1 (right) data sets. In all plots the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data).

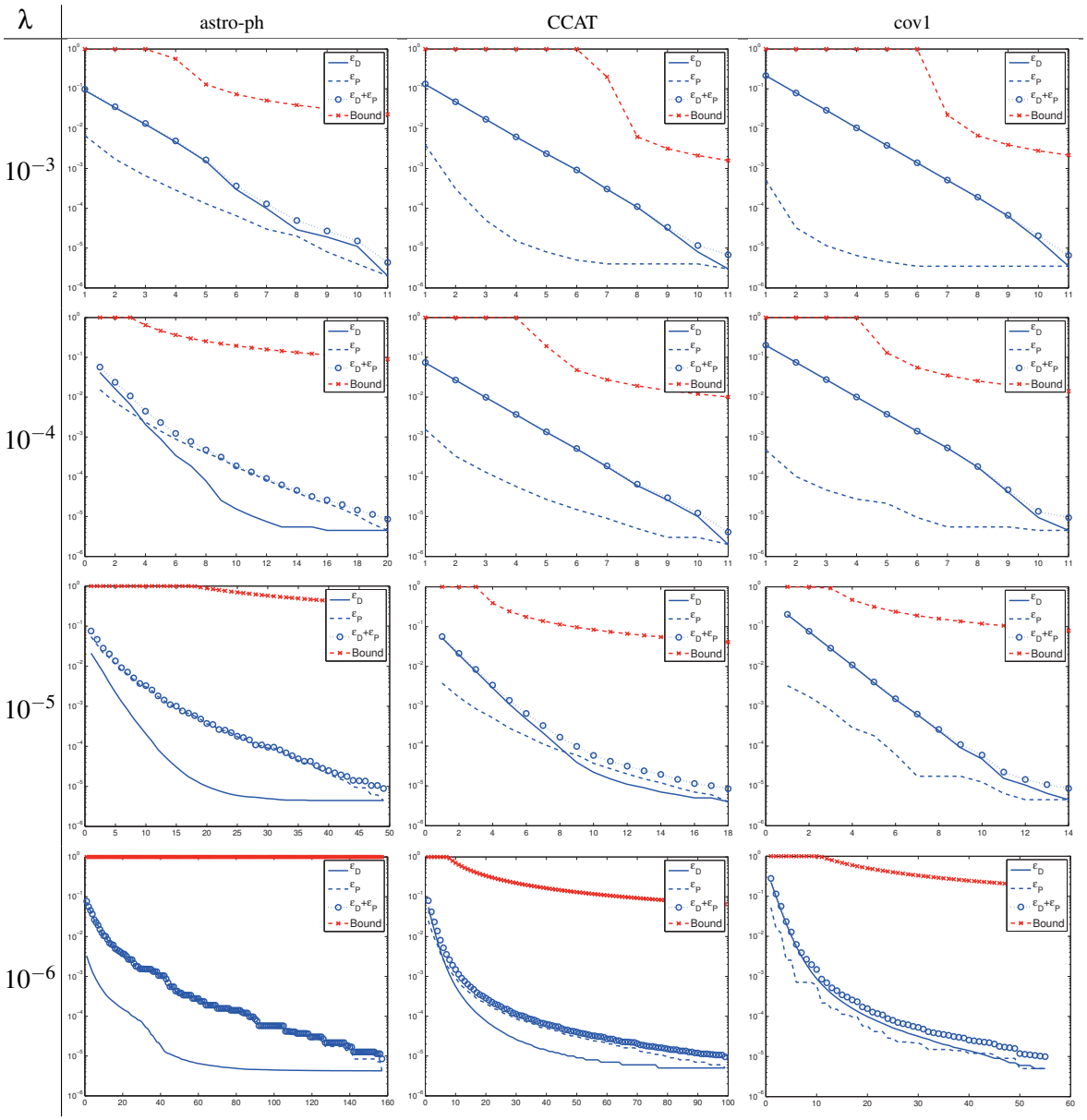


Figure 2: Experiments with the hinge-loss (non-smooth). The primal and dual sub-optimality, the duality gap, and our bound are depicted as a function of the number of epochs, on the astro-ph (left), CCAT (center) and cov1 (right) data sets. In all plots the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data).

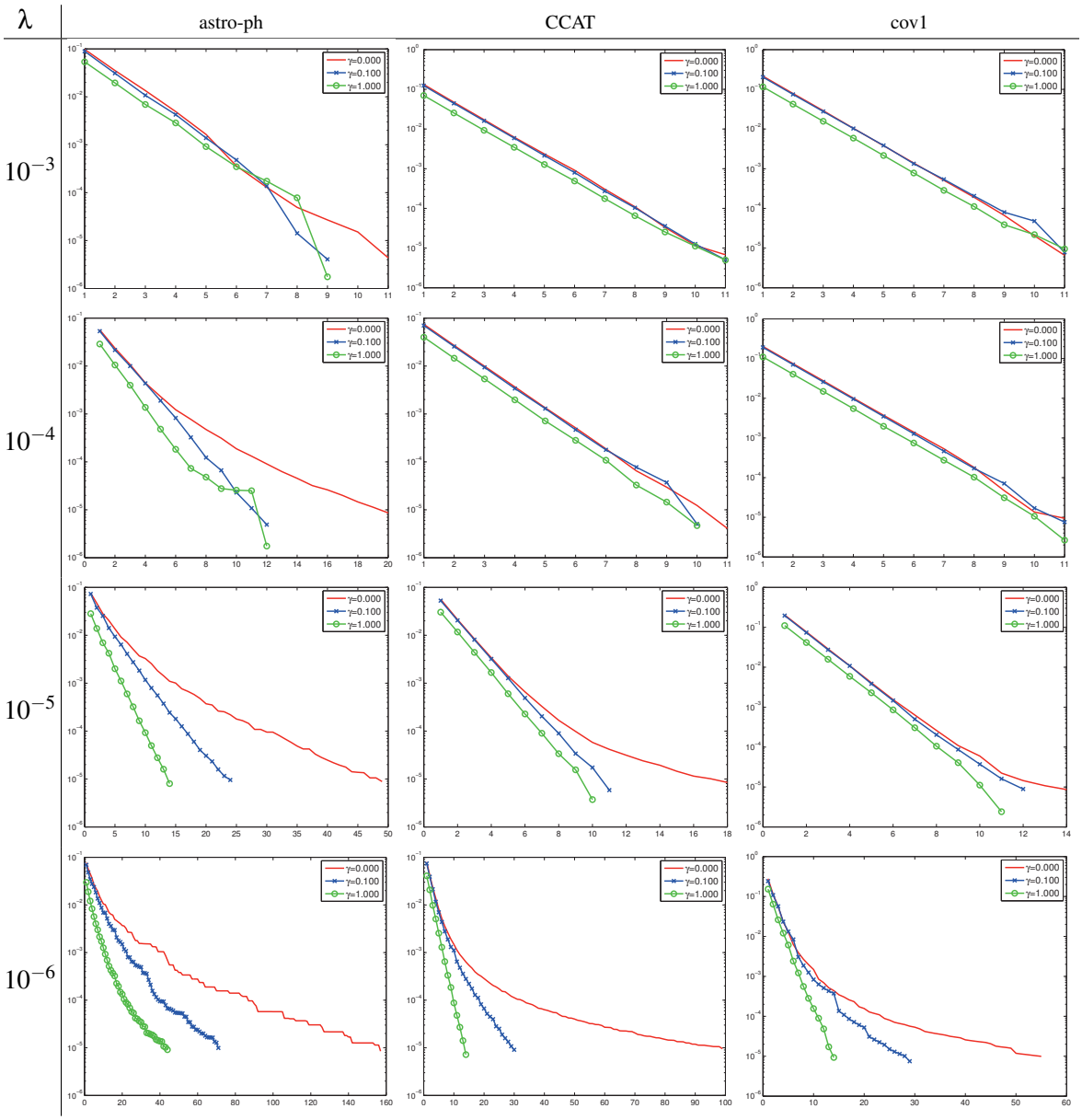


Figure 3: Duality gap as a function of the number of rounds for different values of γ.

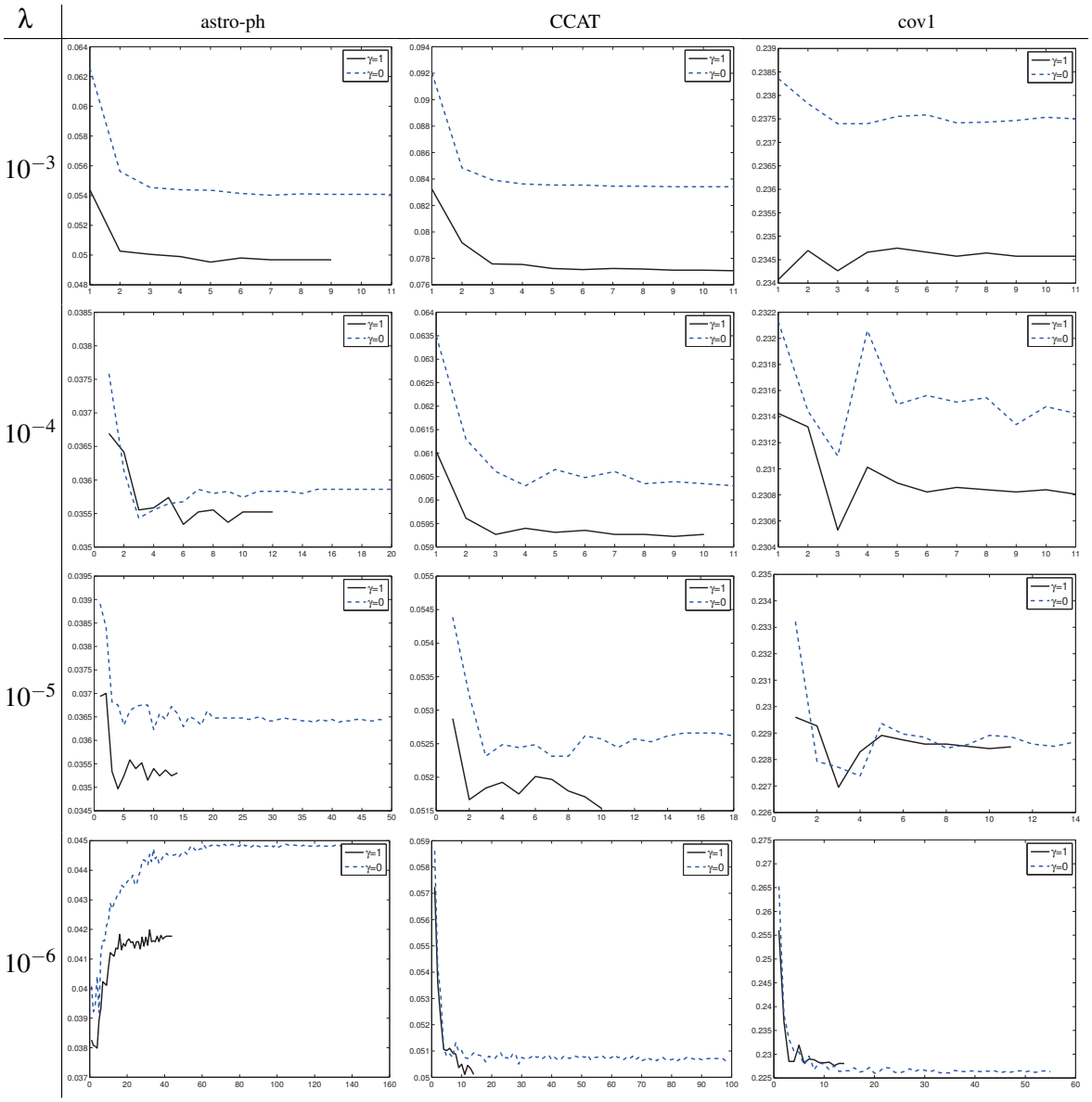


Figure 4: Comparing the test zero-one error of SDCA for smoothed hinge-loss ($\gamma = 1$) and non-smooth hinge-loss ($\gamma = 0$). In all plots the vertical axis is the zero-one error on the test set and the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data). We terminated each method when the duality gap was smaller than 10^{-5}.

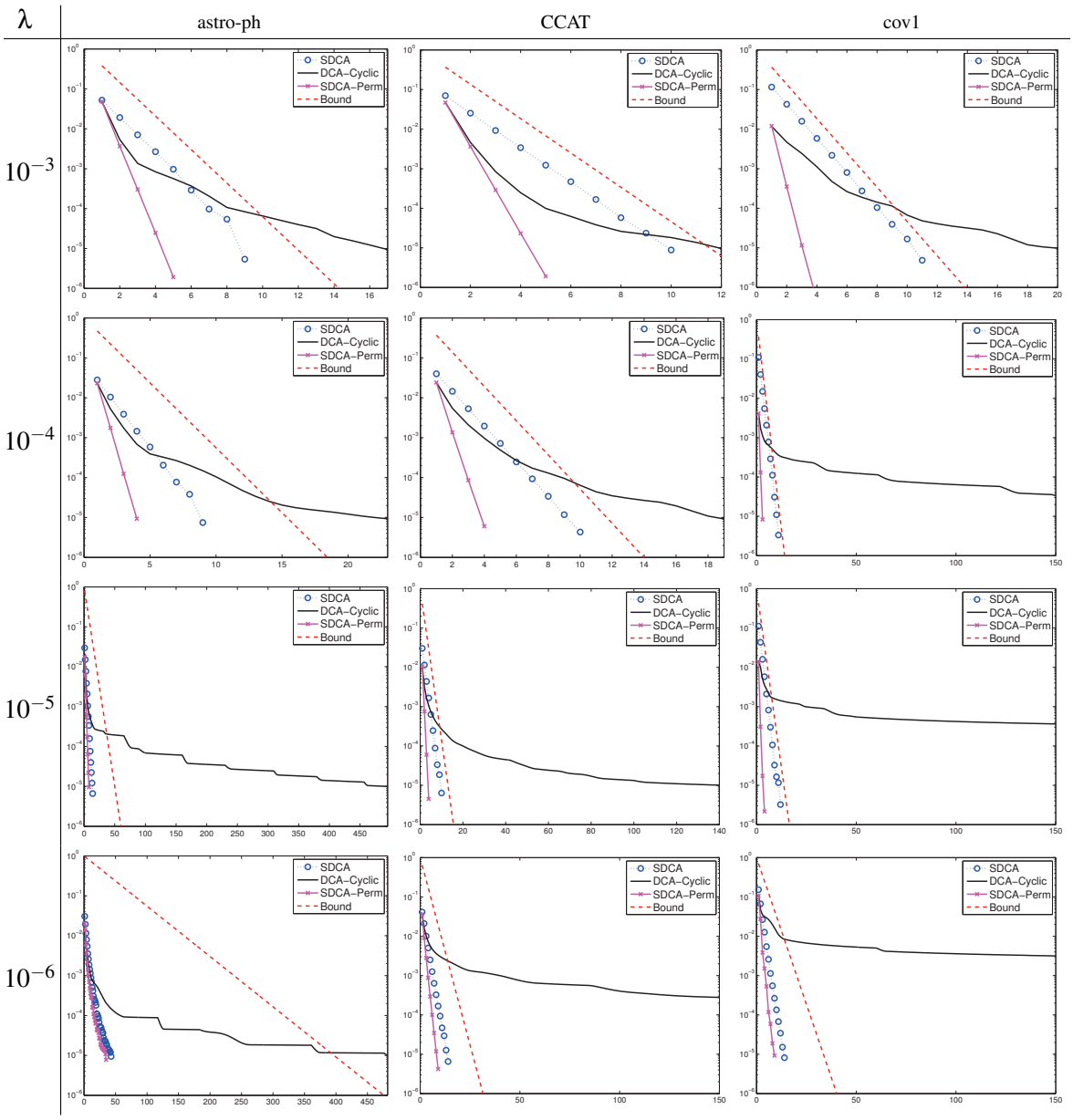


Figure 5: Comparing the duality gap achieved by choosing dual variables at random with repetitions (SDCA), choosing dual variables at random without repetitions (SDCA-Perm), or using a fixed cyclic order. In all cases, the duality gap is depicted as a function of the number of epochs for different values of λ. The loss function is the smooth hinge loss with $\gamma = 1$.

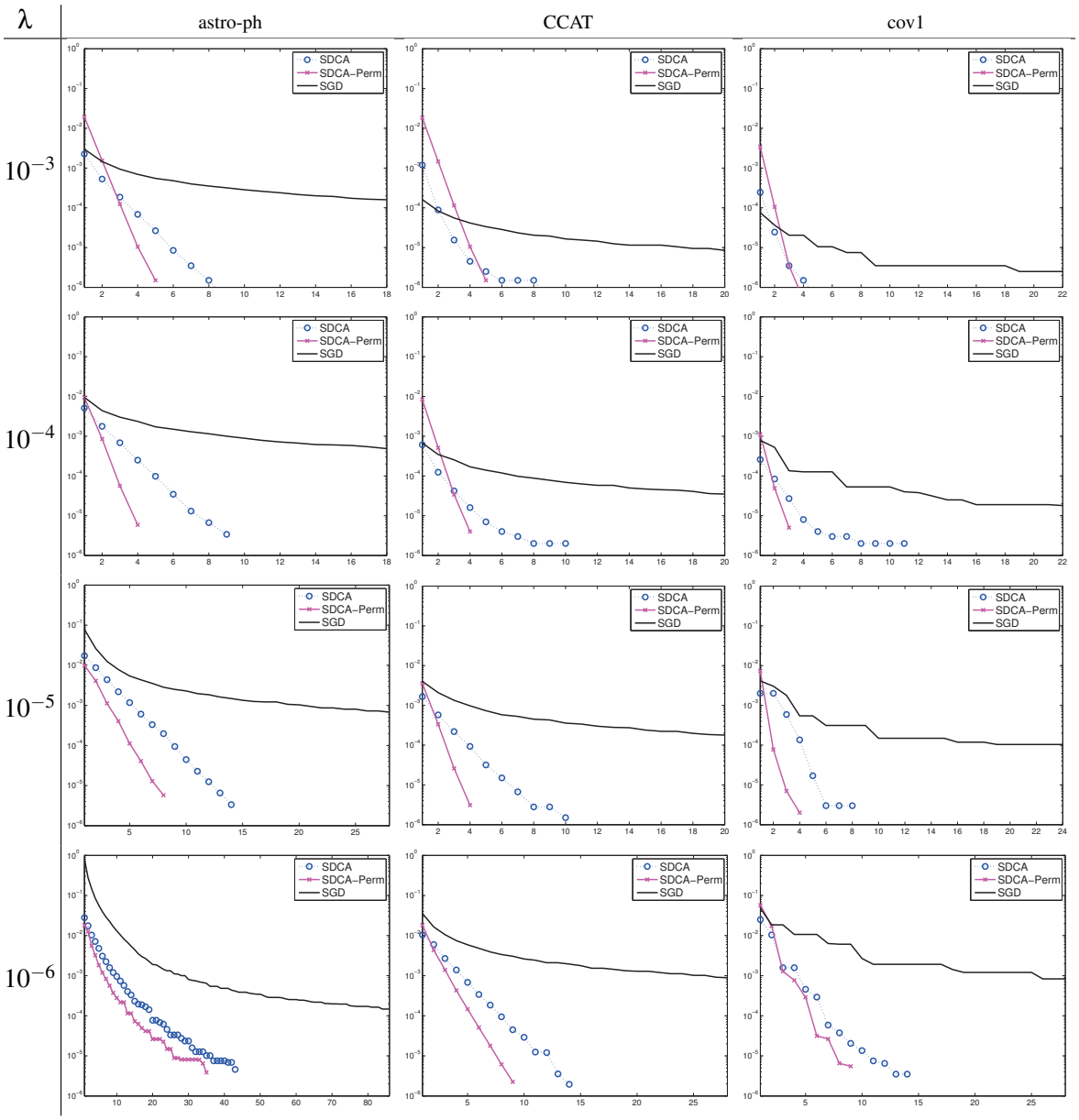


Figure 6: Comparing the primal sub-optimality of SDCA and SGD for the smoothed hinge-loss ($\gamma = 1$). In all plots the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data).

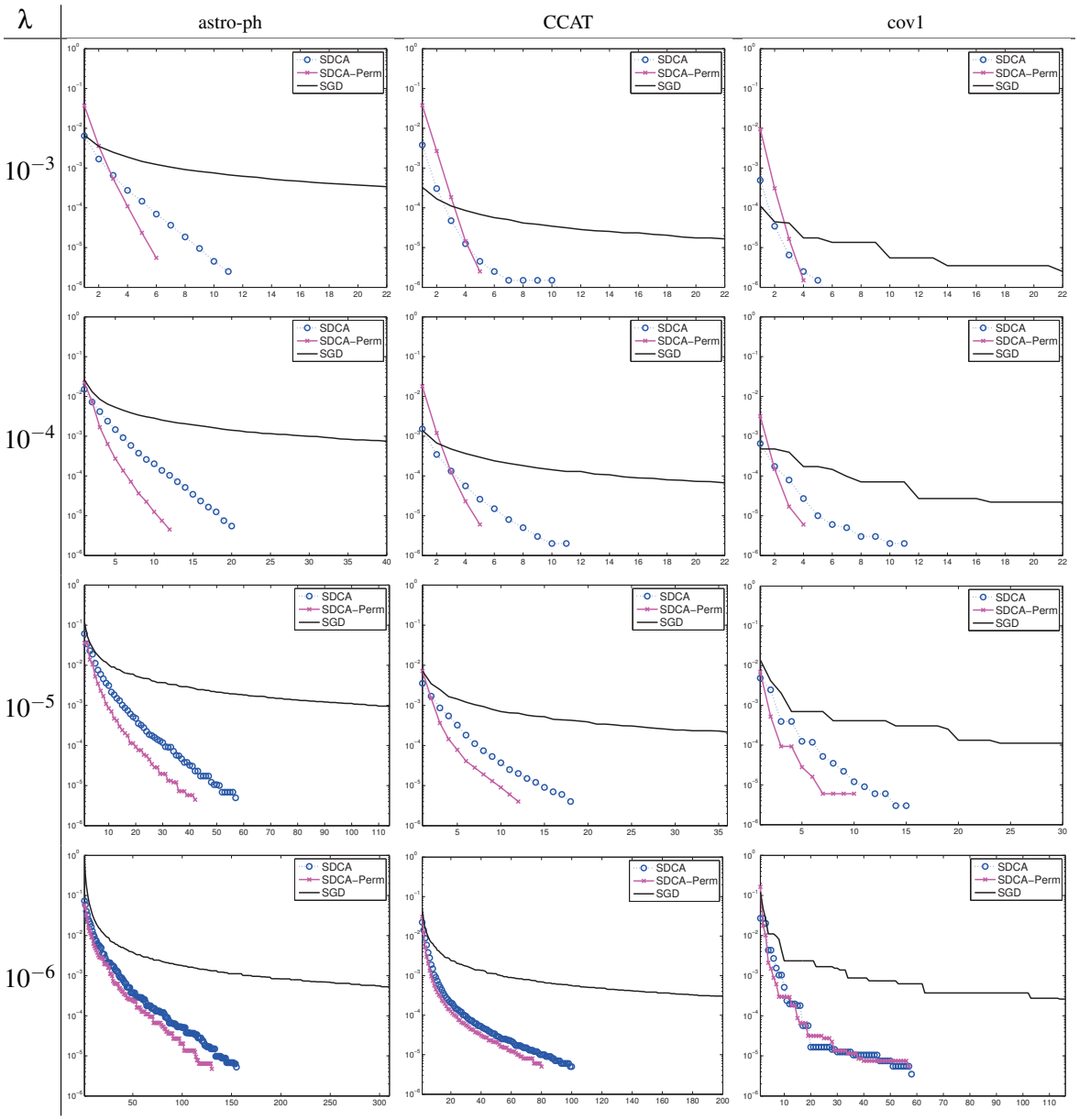


Figure 7: Comparing the primal sub-optimality of SDCA and SGD for the non-smooth hinge-loss ($\gamma = 0$). In all plots the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data).

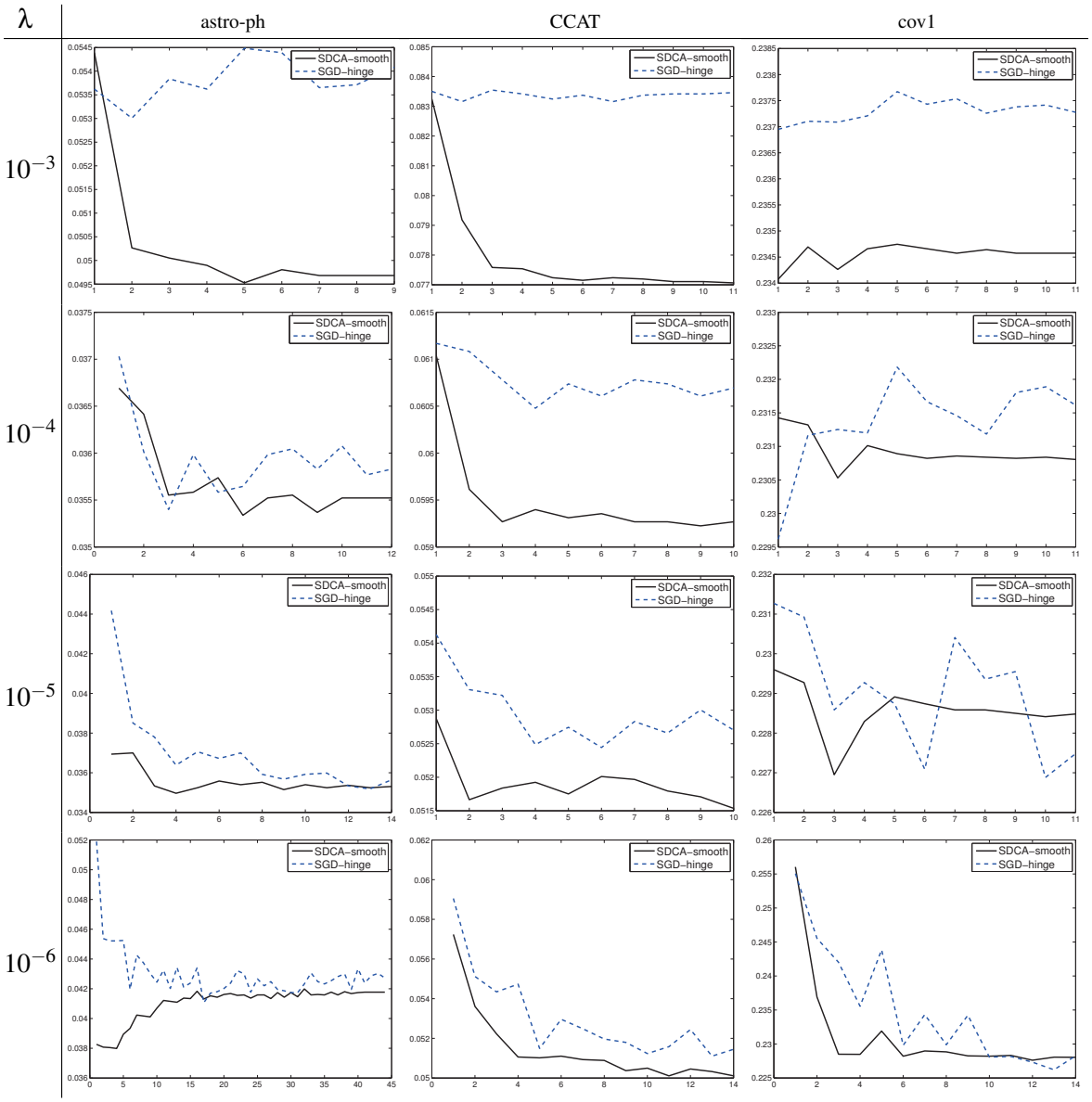


Figure 8: Comparing the test error of SDCA with the smoothed hinge-loss ($\gamma = 1$) to the test error of SGD with the non-smoothed hinge-loss. In all plots the vertical axis is the zero-one error on the test set and the horizontal axis is the number of iterations divided by training set size (corresponding to the number of epochs through the data). We terminated SDCA when the duality gap was smaller than 10^{-5}.

Acknowledgments

Shai Shalev-Shwartz acknowledges the support of the Israeli Science Foundation grant number 598-10. Tong Zhang acknowledges the support of NSF grant DMS-1007527 and NSF grant IIS-1016061.

References

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168, 2008.

M. Collins, A. Globerson, T. Koo, X. Carreras, and P. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008.

Y. Le Cun and L. Bottou. Large scale online learning. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, volume 16, page 217. MIT Press, 2004.

C.J. Hsieh, K.W. Chang, C.J. Lin, S.S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 408–415, 2008.

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.

T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Stochastic block-coordinate frank-wolfe optimization for structural svms. *arXiv preprint:1207.4747*, 2012.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems (NIPS): Proceedings of the 2012 Conference*, 2012. arXiv preprint:1202.6258.

Z.Q. Luo and P. Tseng. On the convergence of coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.*, 72:7–35, 1992.

O. Mangasarian and D. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10, 1999.

N. Murata. A statistical study of on-line learning. *Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK*, 1998.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

J. C. Platt. Fast training of Support Vector Machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

S. Shalev-Shwartz and N. Srebro. SVM optimization: Inverse dependence on training set size. In *International Conference on Machine Learning (ICML)*, pages 928–935, 2008.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for l_1 regularized loss minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 117, 2009.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2007.

K. Sridharan, N. Srebro, and S. Shalev-Shwartz. Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004.

Journal of Machine Learning Research 14 (2013) 601-623 Submitted 10/11; Revised 11/12; Published 2/13

Optimal Discovery with Probabilistic Expert Advice: Finite Time Analysis and Macroscopic Optimality

Sébastien Bubeck SBUBECK@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ, 08544, USA

Damien Ernst DERNST@ULG.AC.BE
Department of Electrical Engineering and Computer Science
University of Liège, Institut Montefiore, B28
B-4000 Liège, Belgium

Aurélien Garivier AURELIEN.GARIVIER@MATH.UNIV-TOULOUSE.FR
Institut de Mathématiques de Toulouse
Université Paul Sabatier
118, route de Narbonne
F-31062 Toulouse Cedex 9, France

Editor: Nicolo Cesa-Bianchi

Abstract

We consider an original problem that arises from the issue of security analysis of a power system and that we name optimal discovery with probabilistic expert advice. We address it with an algorithm based on the optimistic paradigm and on the Good-Turing missing mass estimator. We prove two different regret bounds on the performance of this algorithm under weak assumptions on the probabilistic experts. Under more restrictive hypotheses, we also prove a macroscopic optimality result, comparing the algorithm both with an oracle strategy and with uniform sampling. Finally, we provide numerical experiments illustrating these theoretical findings.

Keywords: optimal discovery, probabilistic experts, optimistic algorithm, Good-Turing estimator, UCB

1. Introduction

In this paper we consider the following problem: Let $\mathcal{X}$ be a set, and $A \subset \mathcal{X}$ be a set of interesting elements in $\mathcal{X}$. One can access $\mathcal{X}$ only through requests to a finite set of probabilistic experts. More precisely, when one makes a request to the i^{th} expert, the latter draws independently at random a point from a fixed probability distribution P_i over $\mathcal{X}$. One is interested in discovering rapidly as many elements of A as possible, by making sequential requests to the experts.

1.1 Motivation

The original motivation for this problem arises from the issue of real-time security analysis of a power system. This problem often amounts to identifying in a set of credible contingencies those that may indeed endanger the security of the power system and perhaps lead to a system collapse with catastrophic consequences (e.g., an entire region, country may be without electrical power for

hours). Once those dangerous contingencies have been identified, the system operators usually take preventive actions so as to ensure that they could mitigate their effect on the system in the likelihood they would occur. Note that usually, the dangerous contingencies are very rare with respect to the non dangerous ones. A straightforward approach for tackling this security analysis problem is to simulate the power system dynamics for every credible contingency so as to identify those that are indeed dangerous. Unfortunately, when the set of credible contingencies contains a large number of elements (say, there are more than 10^5 credible contingencies) such an approach may not possible anymore since the computational resources required to simulate every contingency may excess those that are usually available during the few (tens of) minutes available for the real-time security analysis. One is therefore left with the problem of identifying within this short time-frame a maximum number of dangerous contingencies rather than all of them. The approach proposed in Fonteneau-Belmudes (2012) and Fonteneau-Belmudes et al. (2010) addresses this problem by building first very rapidly what could be described as a probability distribution P over the set of credible contingencies that points with significant probability to contingencies which are dangerous. Afterwards, this probability distribution is used to draw the contingencies to be analyzed through simulations. When the computational resources are exhausted, the approach outputs the contingencies found to be dangerous. One of the main shortcoming of this approach is that usually P points only with a significant probability to a few of the dangerous contingencies and not all of them. This in turn makes this probability distribution not more likely to generate after a few draws new dangerous contingencies than for example a uniform one. The dangerous contingencies to which P points to with a significant probability depend however strongly on the set of (sometimes arbitrary) engineering choices that have been made for building it. One possible strategy to ensure that more dangerous contingencies can be identified within a limited budget of draws would therefore be to consider $K > 1$ sets of engineering choices to build K different probability distributions P_1, P_2, …, P_K and to draw the contingencies from these K distributions rather than only from a single one. This strategy raises however an important question to which this paper tries to answer: how should the distributions be selected for being able to generate with a given number of draws a maximum number of dangerous contingencies? We consider the specific case where the contingencies are sequentially drawn and where the distribution selected for generating a contingency at one instant can be based on the past distributions that have been selected, the contingencies that have been already drawn and the results of the security analyses (dangerous/non dangerous) for these contingencies. This corresponds exactly to the optimal discovery problem with expert advice described above. We believe that this framework has many other possible applications, such as for example web-based content access.

1.2 Setting and Notation

In this paper we restrict our attention to finite or countably infinite sets $\mathcal{X}$. We denote by K the number of experts. For each $i \in \{1,\dots,K\}$, we assume that $(X_{i,n})_{n\geq 1}$ are random variables with distribution P_i such that the $(X_{i,n})_{i,n}$ are independent. Sequential discovery with probabilistic expert advice can be described as follows: at each time step $t \in \mathbb{N}^*$, one picks an index $I_t \in \{1,\dots,K\}$, and one observes $X_{I_t,n_{I_t,t}}$, where

$$n_{i,t} = \sum_{s\leq t} \mathbb{1}\{I_s = i\} \, .$$

The goal is to choose the $(I_t)_{t\geq 1}$ so as to observe as many elements of A as possible in a fixed horizon t, that is to maximize the number of interesting items found after t requests

$$F(t) = \sum_{x\in A} \mathbb{1}\left\{x \in \{X_{1,1},\dots,X_{1,n_{1,t}},\dots,X_{K,1},\dots,X_{K,n_{K,t}}\}\right\}. \tag{1}$$

Note in particular that it is of no interest to observe twice the same same element of A. The index I_{t+1} may be chosen according to past observations: it is a (possibly randomized) function of $(I_1, X_{I_1,1},\dots,I_t,X_{I_t,n_{I_t,t}})$.

An easier quantity to analyze than the number of interesting items found $F(t)$ is the waiting time $T(\lambda)$, $\lambda \in (0,1)$, which is the time at which the strategy has a missing mass of interesting items smaller than λ on every experts, that is

$$T(\lambda) = \inf\left\{t : \forall i \in \{1,\dots,K\}, P_i(A \setminus \{X_{1,1},\dots,X_{1,n_{1,t}},\dots,X_{K,1},\dots,X_{K,n_{K,t}}\}) \leq \lambda\right\}. \tag{2}$$

While we shall derive a general strategy that can be used without any assumption on the probabilistic experts, for the mathematical analysis of the waiting time $T(\lambda)$ we make the following assumption:

(i) non-intersecting supports: $A \cap \mathrm{supp}(P_i) \cap \mathrm{supp}(P_j) = \emptyset$ for $i \neq j$.

Furthermore we will also derive some results under the following more restrictive assumptions:

(ii) finite supports with the same cardinality: $|\mathrm{supp}(P_i)| = N, \forall i \in \{1,\dots,K\}$,

(iii) uniform distributions: $P_i(x) = \frac{1}{N}, \forall x \in \mathrm{supp}(P_i), \forall i \in \{1,\dots,K\}$.

1.3 Contribution and Content of the Paper

This paper contains the description of a generic algorithm for the optimal discovery problem with probabilistic expert advice, and a theoretical analysis of its properties. In Section 2, we first depict our strategy, termed Good-UCB. This algorithm relies on the *optimistic paradigm*, which led to the UCB (Upper Confidence Bound) algorithm for multi-armed bandits, see Auer et al. (2002) and Garivier and Cappé (2011). It relies also on a finite-time analysis of the Good-Turing estimator for the missing mass. We also derive in Section 2 two different regret bounds under the non-intersecting assumption (i): we first show that $F^{UCB}(t)$ (the number of interesting items found by Good-UCB) is larger than $F^*(t)$ (the number of interesting items found by an oracle strategy), up to a term of order $\sqrt{Kt\log(t)}$. We argue that such a bound does not capture all the fine properties of Good-UCB: indeed, on the contrary to the multi-armed bandit problem, here the regret $F^*(t) - F(t)$ remains bounded for any reasonable strategy. This can be understood as a *restoring property* of the game: if a policy makes a sub-optimal choice at some given time t, then in the future it will have better opportunities than the optimal policy. This key feature of our problem prevents the regret from growing too much. To analyze this phenomenon, we complete our first bound by a second regret analysis—the main result of the paper—which states roughly that with high probability, $T_{UCB}(\lambda)$ (the waiting time for the strategy Good-UCB) is *uniformly* (in λ) smaller than $T^*(\lambda')$ (the smallest possible waiting time), for some λ' close to λ and up to a small additional term, see Theorem 5 for a more precise statement. We emphasize that these regret bounds are both completely distribution-free and explicit.

In Section 3 we propose to investigate the behavior of Good-UCB in a *macroscopic limit* sense, that is we make assumptions [(i), (ii), (iii)] and we consider the limit when the size of the set $\mathcal{X}$ grows to infinity while maintaining a constant proportion of interesting items. In this scenario we show that Good-UCB is macroscopically optimal, in the sense that the normalized waiting time of Good-UCB tends to the normalized smallest possible waiting time. We also derive a formula for this latter quantity and we show that it is equal to $\sum_{i:q_i>\lambda} \log \frac{q_i}{\lambda}$, where q_i is the limiting proportion of interesting items on expert i. This macroscopic limit also allows to easily assess the performance of different strategies, and we show that for example the normalized waiting time of uniform sampling tends to $K \max_{1\leq i\leq K} \log \frac{q_i}{\lambda}$, which proves that this strategy is macroscopically suboptimal, unless all experts have the same number of interesting items.

Finally, Section 4 reports experimental results that show that the Good-UCB algorithm performs very well, even in a setting where assumptions (i), (ii) and (iii) are not satisfied. The appendix contains some technical proofs, together with a more detailed discussion on oracle strategies in the macroscopic limit and on the relation between the waiting time T defined in (2) and the number of items found F defined in (1), proving in particular that optimality in terms of waiting time is equivalent to optimality in terms of number of items found.

2. The Good-UCB Algorithm

We describe here the Good-UCB strategy. This algorithm is a sequential method estimating at time t, for each expert $i \in \{1,\dots,K\}$, the total probability of the interesting items that remain to be discovered through requests to expert i. This estimation is done by adapting the so-called Good-Turing estimator for the missing mass. Then, instead of simply using the distribution with highest estimated missing mass, which proves hazardous, we make use of the *optimistic paradigm*—see Bubeck and Cesa-Bianchi (2012, Chapter 2, and references therein)—a heuristic principle well-known in reinforcement learning, which entails to prefer using an *upper-confidence bound* (UCB) of the missing mass instead. At a given time step, the Good-UCB algorithm simply makes a request to the expert with highest upper-confidence bound on the missing mass at this time step.

We start in Section 2.1 with the Good-Turing estimator and a brief study of its concentration properties. Then we describe precisely the Good-UCB strategy in Section 2.2. Next we proceed to the theoretical analysis of Good-UCB and we start in Section 2.3 where we describe an oracle strategy (that we shall use as a comparator) that we prove to be optimal under assumption (i). In Section 2.4 we show that one can obtain a standard regret bound of order $\sqrt{t}$ when one compares the number of items $F^{UCB}(t)$ found by Good-UCB to the number of items $F^*(t)$ found by the oracle. This bound is not completely satisfactory (as we explain in Section 2.4), and our main result—a 'non-linear' regret bound—is proved in Section 2.5.

2.1 Estimating the Missing Mass

Our algorithm relies on an estimation at each step of the probability of obtaining a new interesting item by making a request to a given expert. A similar issue was addressed by I. Good and A. Turing as part of their efforts to crack German ciphers for the Enigma machine during World War II. In this subsection, we describe a version of the Good-Turing estimator adapted to our problem. Let Ω be a discrete set, and let A be a subset of interesting elements of Ω. Assume that $X_1,\dots,X_n$ are elements

of Ω drawn independently under the same distribution P, and define for every $x \in \Omega$:

$$O_n(x) = \sum_{m=1}^{n} \mathbb{1}\{X_m = x\}, \quad Z_n(x) = \mathbb{1}\{O_n(x) = 0\}, \quad U_n(x) = \mathbb{1}\{O_n(x) = 1\} .$$

Let $R_n = \sum_{x \in A} Z_n(x) P(x)$ denote the missing mass of the interesting items, and let $U_n = \sum_{x \in A} U_n(x)$ be the number of elements of A that have been seen exactly once (in linguistics, they are often called *hapaxes*). The idea of the Good-Turing estimator—see Good (1953), see also McAllester and Schapire (2000); Orlitsky et al., and references therein—is to estimate the (random) "missing mass" R_n, which is the total probability of all the interesting items that do not occur in the sample $X_1, \ldots, X_n$, by the "fraction of hapaxes $\hat{R}_n = U_n/n$. This estimator is well-known in linguistics, for instance in order to estimate the number of words in some language, see Gale and Sampson (1995). We shall use the following tight bound on the estimation error. We emphasize the fact that the following bound holds true *independently of the underlying distribution P*.

Proposition 1 *With probability at least* $1 - \delta$*,*

$$\hat{R}_n - \frac{1}{n} - (1+\sqrt{2})\sqrt{\frac{\log(4/\delta)}{n}} \leq R_n \leq \hat{R}_n + (1+\sqrt{2})\sqrt{\frac{\log(4/\delta)}{n}} .$$

Proof For self-containment, we first show that $\mathbb{E}R_n - \mathbb{E}\hat{R}_n \in \left[-\frac{1}{n}, 0\right]$; this result is well known, see for example Theorem 1 in McAllester and Schapire (2000):

$$\begin{aligned}
\mathbb{E}R_n - \mathbb{E}\hat{R}_n &= \sum_{x \in A} \left[P(x)\left(1 - P(x)\right)^n - \frac{1}{n} \times nP(x)\left(1 - P(x)\right)^{n-1} \right] \\
&= -\frac{1}{n} \sum_{x \in A} P(x) \times nP(x)\left(1 - P(x)\right)^{n-1} \\
&= -\frac{1}{n} \mathbb{E}\left[\sum_{x \in A} P(x) U_n(x) \right] \in \left[-\frac{1}{n}, 0 \right] .
\end{aligned}$$

Next we apply the inequality of McDiarmid (1989) to $\hat{R}_n$ as follows. The random variable $\hat{R}_n$ is a function of the independent observations $X_1, \ldots, X_n$ such that, denoting $\hat{R}_n = f(X_1, \ldots, X_n)$, modifying just one observation has limited impact: $\forall l \in \{1, \ldots, n\}, \forall (x_1, \ldots, x_n, x_l') \in \Omega^{n+1}$,

$$\left| f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{l-1}, x_l', x_{l+1}, \ldots, x_n) \right| \leq \frac{2}{n}.$$

Thus one gets that, with probability at least $1 - \delta$,

$$\left| \hat{R}_n - \mathbb{E}[\hat{R}_n] \right| \leq \sqrt{\frac{2\log(2/\delta)}{n}} .$$

Finally we extract the following result from Theorem 10 and Theorem 16 in McAllester and Ortiz (2003): with probability at least $1 - \delta$,

$$\left| R_n - \mathbb{E}[R_n] \right| \leq \sqrt{\frac{\log(2/\delta)}{n}} .$$

which concludes the proof. ■

2.2 The Good-UCB Algorithm

Following the example of the well-known Upper-Confidence Bound procedure for multi-armed bandit problems, we propose Algorithm 1, which we call *Good-UCB* in reference to the estimator it relies on. For each arm $i \in \{1,\dots,K\}$, the index at time t of Good-UCB corresponds to the estimate

$$\hat{R}_{i,n_{i,t-1}} = \frac{1}{n_{i,t-1}} \sum_{x \in A} \mathbb{1}\left\{ \sum_{s=1}^{n_{i,t-1}} \mathbb{1}\{X_{i,s} = x\} = 1 \text{ and } \sum_{j=1}^{K} \sum_{s=1}^{n_{j,t-1}} \mathbb{1}\{X_{j,s} = x\} = 1 \right\}$$

of the missing mass

$$\sum_{x \in A \setminus \left\{ X_{I_1,n_{I_1,1}},\dots,X_{I_{t-1},n_{I_{t-1},t-1}} \right\}} P_i(x) \tag{3}$$

inflated by a confidence bonus of order $\sqrt{\log(t)/n_{i,t-1}}$. Good-UCB relies on a tuning parameter C which is discussed below.

Algorithm 1 Good-UCB

For $1 \le t \le K$ choose $I_t = t$.
for $t \ge K+1$ **do**
  Choose $I_t = \arg\max_{1 \le i \le K} \left\{ \hat{R}_{i,n_{i,t-1}} + C\sqrt{\frac{\log(4t)}{n_{i,t-1}}} \right\}$
  Observe X_t distributed as P_{I_t} and update the missing mass estimates accordingly
end for

The Good-UCB algorithm is designed to work without any assumption on the probabilistic experts. However for the analysis we shall make the non-intersecting supports assumption (i). Indeed without this assumption the missing mass of a given expert i depends explicitly on the outcomes of *all* requests (and not only requests to expert i), see (3), which makes the analysis significantly more difficult. On the other hand under assumption (i) one can define the missing mass of expert i after n pulls without any reference to the other arms, and it takes the following simple form:

$$R_{i,n} = \sum_{x \in A \setminus \{X_{i,1},\dots,X_{i,n}\}} P_i(x)\,. \tag{4}$$

Note that while the theoretical analysis will be carried out under assumption (i), we show in Section 4 that Good-UCB performs well in practice even when this assumption is not met.

2.3 The Closed-loop Oracle Policy

In this section we define a policy that we shall use as a benchmark to study the properties of Good-UCB. We assume hereafter that assumption (i) is satisfied (in particular we shall use the notation defined in (4)). The Oracle Closed-Loop policy, denoted OCL in the following, makes a request at time t to the expert

$$I_t^* = \arg\max_{1 \le i \le K} R_{i,n_{i,t-1}^*}\,, \text{ where } n_{i,t}^* = \sum_{s=1}^{t} \mathbb{1}\{I_s^* = i\}\,.$$

In words, OCL greedily selects the expert that maximizes the probability of finding a new interesting item. The next lemma shows that this greedy procedure is in fact optimal (in expectation) under assumption (i). The proof is given in the appendix.

For any given policy π, let $F^\pi(t)$ be the number of items found at time t with π, I_t^π be the expert chosen by π at time t, and $n_{i,t}^\pi = \sum_{s=1}^t \mathbb{1}\{I_s^\pi = i\}$ be the number of requests made by π to expert i up to time t.

Lemma 2 *Let π be an arbitrary policy, and $t \geq 1$. Then*

$$\mathbb{E}F^\pi(t) \leq \mathbb{E}F^*(t).$$

The optimality of OCL crucially relies on assumption (i). Consider for example the following problem instance: $\mathcal{X} = \{1,2,3,4\}$, $A = \{1,2,3\}$, $K = 3$, $\nu_1 = \delta_1$, $\nu_2 = \frac{2}{5}(\delta_1 + \delta_2) + \frac{1}{5}\delta_4$, and $\nu_3 = \frac{2}{5}(\delta_1 + \delta_3) + \frac{1}{5}\delta_4$ and $t = 2$. In this case OCL first chooses expert 1, and then (say) expert 2: this yields $F^*(2) = 1 + 2/5 = 7/5$. But the strategy π consisting in choosing first expert 2, and then expert 3, is readily seen to have expected return $\mathbb{E}F^\pi(2) = 2/5 \times (1 + 2/5) + 2/5 \times (1 + 4/5) + 1/5 \times 4/5 = 36/25 > 7/5$.

The next lemma is a technical result on OCL that shall prove to be very useful to derive a standard regret bound for Good-UCB. Its proof is also given in the appendix.

Lemma 3 *Let π be an arbitrary policy, and for $t \geq 1$ let*

$$\bar{I}_t = \arg\max_{1 \leq i \leq K} R_{i, n_{i,t-1}^\pi} .$$

Then

$$\mathbb{E}F^*(t) \leq \sum_{s=1}^t \mathbb{E}R_{\bar{I}_s, n_{\bar{I}_s, s-1}^\pi}.$$

2.4 Classical Analysis of the Good-UCB Algorithm

We provide here an upper bound on the expectation of $F^*(t) - F^{UCB}(t)$ which is completely distribution-free, and which depends only on the horizon t and on the number K of experts. This bound grows like $O(\sqrt{Kt\log(t)})$, which is a usual rate for a bandit problem. Indeed, thanks to Lemma 3, the analysis presented in this section follows the lines of classical regret analyses, see for instance Bubeck and Cesa-Bianchi (2012, and the references therein). Below, we discuss some differences between the discovery problem considered here and bandit problems, and we provide an alternative analysis of the Good-UCB algorithm which is more suited to understand its long-term behavior.

Theorem 4 *For any $t \geq 1$, under assumption (i), Good-UCB (with constant $C = (1+\sqrt{2})\sqrt{3}$) satisfies*

$$\mathbb{E}\left[F^*(t) - F^{UCB}(t)\right] \leq 17\sqrt{Kt\log(t)} + 20\sqrt{Kt} + K + K\log(t/K) .$$

Proof Consider the event

$$\xi = \Bigg\{ \forall i \in \{1,\ldots,K\}, \forall u > \sqrt{Kt}, \forall s \leq u, \\ \hat{R}_{i,s} - \frac{1}{s} - (1+\sqrt{2})\sqrt{\frac{3\log(4u)}{s}} \leq R_{i,s} \leq \hat{R}_{i,s} + (1+\sqrt{2})\sqrt{\frac{3\log(4u)}{s}} \Bigg\} .$$

Using Proposition 1 and an union bound, one obtains $\mathbb{P}(\xi) \geq 1 - \sqrt{\frac{K}{t}}$, and thus

$$\mathbb{E}\left[(F^*(t) - F^{UCB}(t))(1 - \mathbb{1}_\xi)\right] \leq t\sqrt{\frac{K}{t}} = \sqrt{Kt}\,.$$

Let $u > \sqrt{Kt}$ and $\bar{I}_u = \arg\max_{1\leq i\leq K} R_{i,n_{i,u-1}}$ be defined as in Lemma 3. On the event ξ, one obtains by definition of I_u that

$$\begin{aligned}
R_{I_u,n_{I_u,u-1}} &\geq \hat{R}_{I_u,n_{I_u,u-1}} - \frac{1}{n_{I_u,u-1}} - (1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}} \\
&\geq \hat{R}_{I_u,n_{I_u,u-1}} + (1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}} - \frac{1}{n_{I_u,u-1}} - 2(1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}} \\
&\geq \hat{R}_{\bar{I}_u,n_{\bar{I}_u,u-1}} + (1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I^*_u,u-1}}} - \frac{1}{n_{I_u,u-1}} - 2(1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}} \\
&\geq R_{\bar{I}_u,n_{\bar{I}_u,u-1}} - \frac{1}{n_{I_u,u-1}} - 2(1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}}\,,
\end{aligned}$$

and thus

$$\begin{aligned}
R_{\bar{I}_u,n_{\bar{I}_u,u-1}} - R_{I_u,n_{I_u,u-1}} &\leq \frac{1}{n_{I_u,u-1}} + 2(1+\sqrt{2})\sqrt{\frac{3\log(4u)}{n_{I_u,u-1}}} \\
&\leq \frac{1}{n_{I_u,u-1}} + 2(1+\sqrt{2})\sqrt{\frac{3\log(4t)}{n_{I_u,u-1}}}\,.
\end{aligned}$$

Hence, using Lemma 3 and the above computation, one obtains

$$\begin{aligned}
\mathbb{E}\left[F^*(t) - F^{UCB}(t)\right] &\leq \sqrt{Kt} + \mathbb{E}\left[\sum_{u=1}^{t} \frac{1}{n_{I_u,u-1}} + 2(1+\sqrt{2})\sqrt{\frac{3\log(4t)}{n_{I_u,u-1}}}\right] \\
&= \sqrt{Kt} + \mathbb{E}\left[\sum_{i=1}^{K}\sum_{s=1}^{n_{i,t-1}} \frac{1}{s} + 2(1+\sqrt{2})\sqrt{\frac{3\log(4t)}{s}}\right] \\
&\leq \sqrt{Kt} + \mathbb{E}\left[\sum_{i=1}^{K} 1 + \log(n_{i,t-1}) + 4(1+\sqrt{2})\sqrt{3\log(4t)(n_{i,t-1}+1)}\right] \\
&\leq \sqrt{Kt} + K + K\log(t/K) + 4(1+\sqrt{2})\sqrt{3Kt\log(4t)}
\end{aligned}$$

by Jensen's inequality and the fact that $\sum_{i=1}^{K} n_{i,t-1} = t-1$. ■

The cumulative regret bound provided in Theorem 4 has a similar flavor as well known regret bounds for the multi-armed bandit problem. Unfortunately here, such bounds, by suggesting that the regret increases with t, do not represent completely the behavior of Good-UCB: as we shall see

in the experiments, the difference between $F^*(t)$ and $F^{UCB}(t)$ is bounded and tends to 0 as t tends to infinity (indeed, ultimately any reasonable strategy will find all the interesting items). Theorem 4 provides insight into the properties of Good-UCB only for 'small' values of t.

The weakness of Theorem 4 and its analysis is that, by using the upper bound of Lemma 3, one ignores the *restoring property* of the game: if a policy makes a sub-optimal choice at some given time t, then it will have better opportunities than OCL in the future, which prevents the regret from growing too much. In the next section we provide a completely different analysis of Good-UCB that takes advantage of this restoring property. This results in a non-standard regret bound, which differs from usual results in the multi-armed bandit literature.

Let us make one more comment about the bound of Theorem 4. On the contrary to the multi-armed bandit, the discovery problem discussed in this paper has a 'natural' time scale: if the horizon t is too small, then even OCL will not be able to discover a significant proportion of interesting items, while if t is too large then any reasonable strategy will find almost all interesting items. To go around this issue we find it more elegant to study the waiting time $T(\lambda)$ (see (2)) which yields a sort of automatic normalization of the time scale.

2.5 Time-uniform Analysis of the Good-UCB Algorithm

In this section we analyze the waiting time of Good-UCB under assumption (i). We shall derive a non-linear regret bound as follows. For a fixed $\lambda \in (0,1)$ we consider the number of requests $T_{UCB}(\lambda)$ that Good-UCB needs to make in order to have a missing mass of interesting items smaller than λ on each expert, see (2). We also consider the omniscient oracle strategy that minimizes this number of requests, given the knowledge of λ and the sequence of answers to the requests $(X_{i,s})_{1\le i\le K, s\ge 1}$. We denote by $T^*(\lambda)$ the corresponding number of requests for this omniscient oracle strategy. (Note that this strategy is even more powerful than the OCL studied in the previous sections.) We now prove that with high probability, $T_{UCB}(\lambda)$ is smaller than $T^*(\lambda')$, for some λ' close to λ and up to a small additional term.

Theorem 5 *Let $c > 0$ and $S \ge 1$. Under assumption (i), Good-UCB (with constant $C = (1+\sqrt{2})\sqrt{c+2}$) satisfies with probability at least $1 - \frac{K}{cS^c}$, for any $\lambda \in (0,1)$,*

$$T_{UCB}(\lambda) \le T^* + KS\log\left(8T^* + 16KS\log(KS)\right),$$

$$\text{where} \quad T^* = T^*\left(\lambda - \frac{3}{S} - 2(1+\sqrt{2})\sqrt{\frac{c+2}{S}}\right).$$

Informally this bound shows that Good-UCB slightly lags behind the omniscient oracle strategy. Under more restrictive assumptions on the experts it is possible to obtain a more explicit bound by studying the variations of T. In the next section we take another route and we show that the above upper bound can be used to prove a clear qualitative property for Good-UCB, namely its *macroscopic optimality*.

Proof Recall that we work under assumption (i), and we run Good-UCB with parameter $C = (1+\sqrt{2})\sqrt{c+2}$, for some positive constant c. After t pulls, the missing mass estimate of expert i is:

$$\hat{R}_{i,t} = \frac{1}{t}\sum_{x\in A} \mathbb{1}\left\{1 = \sum_{s=1}^{t} \mathbb{1}\{X_{i,s} = x\}\right\}.$$

We consider the following event:

$$\xi = \Bigg\{ \forall i \in \{1,\dots,K\}, \forall t > S, \forall s \leq t, \\ \hat{R}_{i,s} - \frac{1}{s} - (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4t)}{s}} \leq R_{i,s} \leq \hat{R}_{i,s} + (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4t)}{s}} \Bigg\}.$$

Using Proposition 1 and an union bound, one obtains $\mathbb{P}(\xi) \geq 1 - \frac{K}{cS^c}$. In the following we work on the event ξ. Recall that $T^*(\lambda)$ (respectively $T_{UCB}(\lambda)$) is the time at which the omniscient oracle strategy (respectively the Good-UCB strategy) attains a missing mass smaller than λ on all experts. Note that $T^*(\lambda)$ and $T_{UCB}(\lambda)$ are functions of $(X_{i,s})_{1\leq i\leq K, s\geq 1}$. In particular one can write:

$$T_{UCB}(\lambda) = \min\left\{ t \geq 1 : \forall i \in \{1,\dots,K\}, R_{i,n_{i,t}} \leq \lambda \right\},$$
$$T^*(\lambda) = \sum_{i=1}^{K} T_i^*(\lambda), \text{ where } T_i^*(\lambda) = \min\left\{ t \geq 1 : R_{i,t} \leq \lambda \right\}.$$

Let

$$U(\lambda) = \min\left\{ t \geq 1 : \forall i \in \{1,\dots,K\}, \hat{R}_{i,n_{i,t}} + (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4t)}{n_{i,t}}} \leq \lambda \right\}.$$

Let $S' \geq S$ to be defined later. On the event ξ one clearly gets $T_{UCB}(\lambda) \leq \max(S', U(\lambda))$. Moreover the following inequalities hold true if $U(\lambda) > S'$ (see below for an explanation of each inequality)

$$\begin{aligned} R_{i,n_{i,U(\lambda)}} &\geq \hat{R}_{i,n_{i,U(\lambda)}} - \frac{1}{n_{i,U(\lambda)}} - (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{n_{i,U(\lambda)}}} \\ &\geq \hat{R}_{i,n_{i,U(\lambda)}-1} - \frac{3}{n_{i,U(\lambda)}} - (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{n_{i,U(\lambda)}}} \\ &\geq \left(\lambda - (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{n_{i,U(\lambda)}-1}}\right) - \frac{3}{n_{i,U(\lambda)}} - (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{n_{i,U(\lambda)}}} \\ &\geq \lambda - \frac{3}{n_{i,U(\lambda)}} - 2(1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{n_{i,U(\lambda)}-1}}. \end{aligned}$$

The first inequality comes from the fact that we are on event ξ and we assume $U(\lambda) > S'$. The second inequality uses the fact that when we make a request to an expert, the number of items uniquely seen on this expert can drop by at most one, and thus we get

$$s\hat{R}_{i,s} \geq (s-1)\hat{R}_{i,s-1} - 1 \geq s\hat{R}_{i,s-1} - 2.$$

The third inequality is the key step of the proof. Consider the time step t such that $n_{i,t} = n_{i,U(\lambda)} - 1$ and $n_{i,t+1} = n_{i,U(\lambda)}$. Since $t < U(\lambda)$ we know that one of the expert satisfies $\hat{R}_{j,n_{j,t}} + (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4t)}{n_{j,t}}} > \lambda$. Moreover, since Good-UCB is run with constant $C = (1+\sqrt{2})\sqrt{c+2}$ and since we make a request to expert i at time t, we know that it maximizes the Good-UCB index, and thus $\hat{R}_{i,n_{i,t}} + (1+\sqrt{2})\sqrt{\frac{(c+2)\log(4t)}{n_{i,t}}} > \lambda$. Using that $t \leq U(\lambda)$ completes the proof of the third inequality. The fourth inequality is trivial.

We just proved that if $n_{i,U(\lambda)} > S'$ then

$$R_{i,n_{i,U(\lambda)}} \geq \lambda - \frac{3}{S'} - 2(1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{S'}},$$

which clearly implies

$$n_{i,U(\lambda)} \leq T_i^* \left(\lambda - \frac{3}{S'} - 2(1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{S'}} \right).$$

Thus in any case we have proved that

$$n_{i,U(\lambda)} \leq S' + T_i^* \left(\lambda - \frac{3}{S'} - 2(1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{S'}} \right),$$

which implies

$$\begin{aligned} U(\lambda) &\leq KS' + T^* \left(\lambda - \frac{3}{S'} - 2(1+\sqrt{2})\sqrt{\frac{(c+2)\log(4U(\lambda))}{S'}} \right) \\ &\leq KS\log(4U(\lambda)) + T^* \left(\lambda - \frac{3}{S} - 2(1+\sqrt{2})\sqrt{\frac{c+2}{S}} \right), \end{aligned}$$

where the last inequality follows by taking $S' = S\log(4U(\lambda))$. Finally using Lemma 9 (in the appendix) and $T_{UCB}(\lambda) \leq \max(S', U(\lambda))$ ends the proof. ■

3. Macroscopic Limit

In the previous section we derived a very general non-linear regret bound for Good-UCB. Here we shall study the behavior of Good-UCB under more restrictive assumptions on the experts, but it will allow us to derive a clear qualitative statement about its performance, and it also permits easier comparison with other strategies such as uniform sampling. In this section we shall add the two following assumptions in addition to assumption (i):

(ii) finite supports with the same cardinality: $|\mathrm{supp}(P_i)| = N, \forall i \in \{1,\dots,K\}$,

(iii) uniform distributions: $P_i(x) = \frac{1}{N}, \forall x \in \mathrm{supp}(P_i), \forall i \in \{1,\dots,K\}$.

These assumptions are primarily made in order to be able to assess the performance of the optimal strategy. In this setting it is convenient to re-parameterize slightly the problem (in particular we make explicit the dependency on N for reasons that will appear later). Let $\mathcal{X}^N = \{1,\dots,K\} \times \{1,\dots,N\}$, $A^N \subset \mathcal{X}^N$ the set of interesting items of $\mathcal{X}^N$, and $Q^N = |A^N|$ the number of interesting items. We assume that, for expert $i \in \{1,\dots,K\}$, P_i^N is the uniform distribution on $\{i\} \times \{1,\dots,N\}$. We also denote by $Q_i^N = \left|A^N \cap (\{i\} \times \{1,\dots,N\})\right|$ the number of interesting items accessible through requests to expert i. Without loss of generality, we assume in this section that $Q_1^N \geq Q_2^N \geq \cdots \geq Q_K^N$.

The macroscopic limit that we investigate in this section corresponds to the setting where N goes to infinity together with the Q_i^N in such a way that $Q_i^N/N \to q_i \in (0,1)$. For a given strategy we are interested in the time $T^N(\lambda)$ such that all experts have at most $N\lambda$ undiscovered interesting items. In particular we define $T^N_{UCB}(\lambda)$ (respectively $T^N_*(\lambda)$) to be the corresponding time for the Good-UCB strategy (respectively the oracle omniscient strategy). In the macroscopic limit we shall be particularly interested in normalized limit waiting time $\lim_{N\to+\infty} T^N(\lambda)/N$.

3.1 Macroscopic Behavior of the Oracle Closed-loop Strategy

In this section we shall derive an explicit upper bound on the macroscopic limit of T^N_* by studying the OCL strategy introduced in Section 2.3. Recall that at each time step, OCL makes a request to one of the experts with highest number of still undiscovered interesting items: the expert requested at time t is:

$$I_t \in \operatorname*{arg\,max}_{1\le i\le K} P_i\left(A \setminus \{X_{1,1},\dots,X_{1,n_{1,t}},\dots,X_{K,1},\dots,X_{K,n_{K,t}}\}\right) .$$

Theorem 6 *For every $\lambda \in (0,q_1)$, for every sequence $(\lambda^N)_N$ converging to λ as N goes to infinity, under assumption (i), (ii) and (iii), almost surely*

$$\lim_{N\to\infty} \frac{T^N_{OCL}(\lambda^N)}{N} = \sum_{i:q_i>\lambda} \log\frac{q_i}{\lambda} .$$

Proof Denote by B_i^N the set of interesting items in $\{1,\dots,N\}$ supported by P_i^N: $B_i^N = \{x \in \{1,\dots,N\} : (i,x) \in A^N\}$. Successive draws of expert i are denoted $(i,X_{i,1}^N),(i,X_{i,2}^N)\dots$ where the variables $(X_{i,n}^N)_{i,n}$ are assumed to be independent. Without loss of generality, we may assume that $N\lambda^N$ is a positive integer, for otherwise λ^N can be replaced by $\lceil N\lambda^N\rceil/N$. We denote by $(D_{i,k}^N)_{1\le k\le Q_i^N}$ the increasing sequence of the indices corresponding to draws for which new interesting items are discovered with expert i:

$$D_{i,1}^N = \min\left\{n \ge 1 : X_{i,n}^N \in B_i^N\right\}, \quad D_{i,2}^N = \min\left\{n \ge D_{i,1}^N : X_{i,n}^N \in B_i^N \setminus \left\{X_{i,D_{i,1}^N}^N\right\}\right\},\dots$$

We also define $S_{i,0}^N = 0$ and for $k \ge 1, S_{i,k}^N = D_{i,k}^N - D_{i,k-1}^N$. The random variables $S_{i,k}^N$ $(1 \le i \le K, k \ge 1)$ are independent with geometric distribution $\mathcal{G}((1+Q_i^N-k)/N)$.

At every step, the OCL should call the expert with maximal number of undiscovered interesting items. Hence, it can:

- first request expert 1 for $D^N_{1,Q_1^N-Q_2^N}$ steps;
- then, alternatively request
  - expert 1 for $S^N_{1,1+Q_1^N-Q_2^N}$ steps;
  - expert 2 for $S^N_{2,1}$ steps;
  - expert 1 for $S^N_{1,2+Q_1^N-Q_2^N}$ steps;
  - expert 2 for $S^N_{2,2}$ steps;
  - and so on, until there are only Q_3^N undiscovered interesting items on experts 1 and 2.

- and so on, including successively experts $3, 4, \ldots, K$ in the alternation.

Obviously,

$$T^N_{OCL}(\lambda^N) = \sum_{i:Q^N_i > N\lambda^N} D^N_{i,Q^N_i - N\lambda^N} \cdot$$

It suffices now to show that for every expert $i \in \{1, \ldots, K\}$, $D^N_{i,Q^N_i - N\lambda^N}/N$ converges almost surely to $\log(q_i/\lambda)$ as N goes to infinity. Write

$$W^N_{i,N\lambda^N} = \frac{1}{N}\left(D^N_{i,Q^N_i - N\lambda^N} - \mathbb{E}\left[D^N_{i,Q^N_i - N\lambda^N}\right]\right) = \frac{1}{N}\sum_{k=1}^{Q^N_i - N\lambda^N - 1}\left(S^N_{i,k} - \mathbb{E}\left[S^N_{i,k}\right]\right) . \tag{5}$$

For every positive integer d and for $k \in \{1, \ldots, N\lambda^N - 1\}$, elementary manipulations of the geometric distribution yield that

$$\mathbb{E}\left[\left(S^N_{i,k} - \mathbb{E}\left[S^N_{i,k}\right]\right)^d\right] \leq \mathbb{E}\left[\left(S^N_{i,N\lambda^N} - \mathbb{E}\left[S^N_{i,N\lambda^N}\right]\right)^d\right] \leq \frac{c(d)}{(\lambda^N)^d} \leq \frac{2c(d)}{\lambda^4}$$

for some positive constant $c(d)$ depending only on d, and for N large enough. Hence, taking (5) to the fourth power and developing yields

$$\mathbb{E}\left[\left(W^N_{i,N\lambda^N}\right)^4\right] \leq \frac{c'}{N^2\lambda^4}$$

for some positive constant c'. Using Markov's inequality together with the Borel-Cantelli lemma, this permits to show that W^N_{i,λ^N} converges almost surely to 0 as N goes to infinity. But

$$\frac{1}{N}\mathbb{E}\left[D^N_{i,Q^N_i - N\lambda^N}\right] = \frac{1}{Q^N_1} + \cdots + \frac{1}{N\lambda^N + 1} = \log\frac{Q^N_i}{N\lambda^N} - \varepsilon^N ,$$

with $0 \leq \varepsilon^N \leq 1/(N\lambda^N)$ according to Lemma 10, and thus

$$\frac{1}{N}\mathbb{E}\left[D^N_{i,Q^N_i - N\lambda^N}\right] \to \lim_{N\to\infty} \log\left(\frac{Q^N_i/N}{\lambda^N}\right) = \log(q_i/\lambda) ,$$

which concludes the proof. ■

3.2 Macroscopic Behavior of Uniform Sampling

In this section we study the simple uniform sampling strategy that cycles through the experts, that is, at time t uniform sampling makes a request to the $(t \mod [K])^{th}$ expert. This strategy is not macroscopically optimal unless all experts have the same number of interesting items. Furthermore the next proposition makes precise the extent of improvement of a macroscopic optimal strategy over uniform sampling. The proof follows the exact same steps than the proof of Theorem 6 and thus is omitted.

Proposition 7 *For every $\lambda \in (0, q_1)$, for every sequence $(\lambda^N)_N$ converging to λ as N goes to infinity, under assumption (i), (ii) and (iii), almost surely*

$$\lim_{N\to\infty} \frac{T^N_{US}(\lambda^N)}{N} = K \log\frac{q_1}{\lambda} .$$

3.3 Macroscopic Optimality of Good-UCB

Using the regret bound of Theorem 5 we obtain the following corollary that shows the asymptotic optimality of the Good-UCB algorithm in the macroscopic sense.

Corollary 8 *Take $C = (1+\sqrt{2})\sqrt{c+2}$ with $c > 3/2$ in Algorithm 1. Under assumption (i), (ii) and (iii), for every sequence $(\lambda^N)_N$ converging to λ as N goes to infinity, almost surely*

$$\limsup_{N\to+\infty} \frac{T^N_{UCB}(\lambda^N)}{N} \leq \sum_{i:q_i>\lambda} \log\frac{q_i}{\lambda} \,.$$

Proof Let $S^N = N^{2/3}$. First note that:

$$\ell^N \overset{def}{=} \lambda^N - \frac{3}{S^N} - 2(1+\sqrt{2})\sqrt{\frac{c+2}{S^N}} \to \lambda \quad \text{when } N \to \infty \,.$$

Thus, by Theorem 6, and the fact that the OCL strategy needs at least as much time as the omniscient oracle strategy in order to find the same number of items, there exists an event Ω of probability 1 on which

$$\limsup_{N\to+\infty} \frac{T^N_*\left(\ell^N\right)}{N} \leq \sum_{i:q_i>\lambda} \log\frac{q_i}{\lambda} \,.$$

Thus, according to Theorem 5, for each positive integer N there exists an event A_N of probability $P(A_N) \geq 1 - K/(cN^{2c/3})$ on which

$$\begin{aligned}\frac{T^N_{UCB}(\lambda^N)}{N} &\leq \frac{T^N_*\left(\ell^N\right)}{N} + \frac{KS^N}{N}\log\left(8T^N_*\left(\ell^N\right) + 16KS\log(KS^N)\right) \\ &= \frac{T^*_N\left(\ell^N\right)}{N} + O\left(\frac{\log(N)}{N^{1/3}}\right) \,.\end{aligned}$$

Using Borel-Cantelli's lemma and the fact that, with our choice of parameters, $\sum_N N^{-2c/3} < \infty$, we obtain that except maybe on the set (of probability 0) $\bar{\Omega} \cup \limsup \overline{A_N}$,

$$\limsup_{N\to\infty} \frac{T^N_{UCB}(\lambda^N)}{N} \leq \limsup_{N\to+\infty} \frac{T^N_*(\ell^N)}{N} \leq \sum_{i:q_i>\lambda} \log\frac{q_i}{\lambda} \,,$$

which ends the proof. ■

4. Simulations

We provide a few simulations illustrating the behavior of the Good-UCB algorithm and the asymptotic analysis above of Section 3. We first consider an example with $K = 7$ different sampling distributions satisfying assumptions [(i),(ii),(iii)], with respective proportions of interesting items $q_1 = 51.2\%, q_2 = 25.6\%, q_3 = 12.8\%, q_4 = 6.4\%, q_5 = 3.2\%, q_6 = 1.6\%$ and $q_7 = 0.8\%$.

We have chosen to display here the numbers of items found as a function of the number of draws (see (1)), instead of the times $T^N(\lambda^N)$, because they express more intuitively the discovering

possibilities of each algorithm. Note, however, that the correspondence between these two quantities is straightforward, especially in the macroscopic limit: For $\lambda \in (0, q_1)$ let

$$T(\lambda) = \sum_{i:q_i>\lambda} \log \frac{q_i}{\lambda} \,. \tag{6}$$

It is easy to show that the proportion of interesting items found by the OCL strategy after Nt draws converge to

$$F(t) = \sum_{i=1}^{K} \left(q_i - T^{-1}(t)\right)_+ \,. \tag{7}$$

Furthermore the latter expression is a lower bound for the corresponding proportion of interesting items found by the Good-UCB algorithm. Proposition 11, proved in the Appendix, provides a more explicit expression for F: denoting $q = \sum_{i=1}^{K} q_i$, there exists an increasing, $\{1,\dots,K\}$-valued function I such that, for each t,

$$F(t) = q - I(t)\underline{q}_{I(t)} \exp\left(-t/I(t)\right) \,,$$

where $\underline{q}_{I(t)}$ denotes the geometric mean of $q_1,\dots,q_{I(t)}$. This permits an explicit comparison of the macroscopic performance of the Good-UCB algorithm with uniform sampling: when all distributions are sampled equally often, the proportion of unseen interesting items at time t is smaller than

$$\sum_{i=1}^{K} q_i \exp(-t/K) = K\bar{q}_K \exp(-t/K) \,,$$

where $\bar{q}_K = (\sum_{i=1}^{K} q_i)/K$ is the arithmetic mean of the $(q_i)_i$. On the other hand, for the Good-UCB algorithm, the proportion of unseen interesting items at time t is smaller than

$$I(t)\underline{q}_{I(t)} \exp\left(-t/I(t)\right) \,.$$

The ratio of those two quantities is a decreasing function of time lower-bounded by $\bar{q}_K/\underline{q}_K \geq 1$, the ratio of the arithmetic mean with the geometric mean of the $(q_i)_i$. As expected, this ratio gets larger when the proportions of interesting items among experts becomes more unbalanced.

Figure 1 displays the number of items found as a function of time by the Good-UCB (solid), the OCL (dashed) and the uniform sampling scheme that alternates between experts (dotted). The results are presented for sizes $N = 128, N = 500, N = 1000$ and $N = 10000$, each time for one representative run (averaging over different runs removes the interesting variability of the process). We chose to plot the number of items found rather than the waiting time t as the former is easier to visualize while the latter was easier to analyze. In fact, macroscopic optimality in terms of number of items found could also be derived with the techniques of Section 3. Figure 1 also shows clearly the macroscopic convergence of Good-UCB to the OCL. Moreover, it can be seen that, even for very moderate values of N, the Good-UCB significantly outperforms uniform sampling even if it is clearly distanced by the OCL.

For these simulations, the parameter C of Algorithm Good-UCB has been taken equal to $1/2$, which is a rather conservative choice. In fact, it appears that during all rounds of all runs, all upper-confidence bounds did contain the actual missing mass. Of course, a bolder choice of C can only improve the performance of the algorithm, as long as the confidence level remains sufficient.

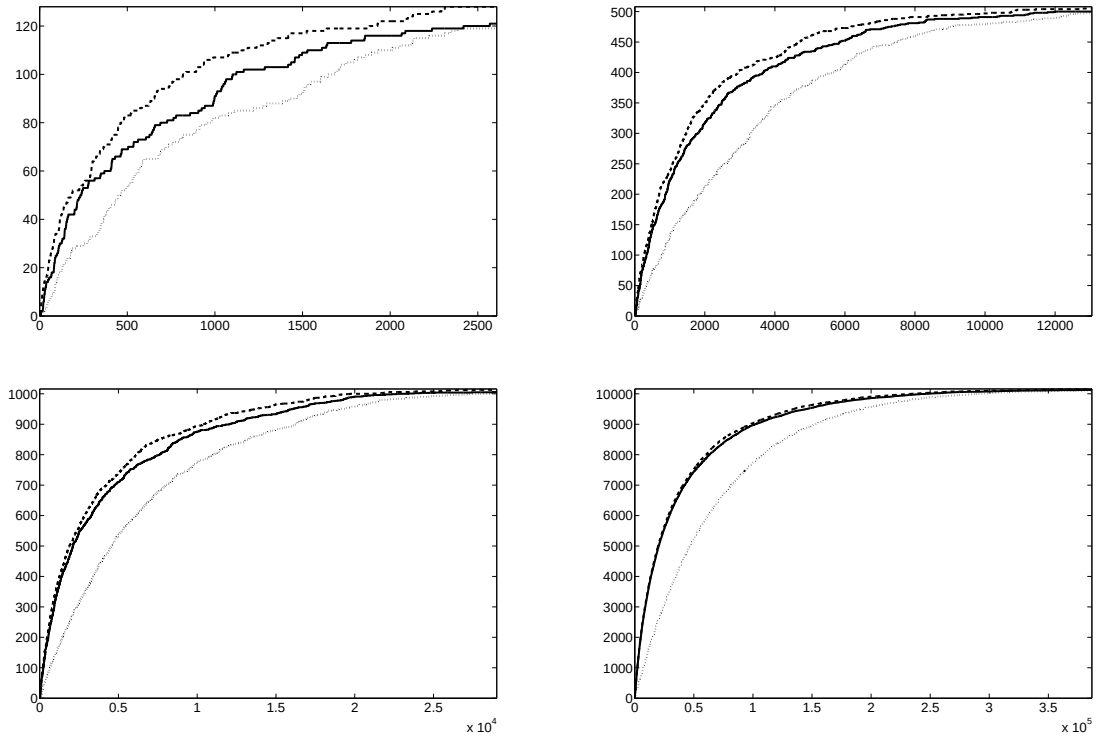


Figure 1: Number of items found by Good-UCB (solid), the OCL (dashed), and uniform sampling (dotted) as a function of time for sizes $N = 128, N = 500, N = 1000$ and $N = 10000$ in a 7-experts setting.

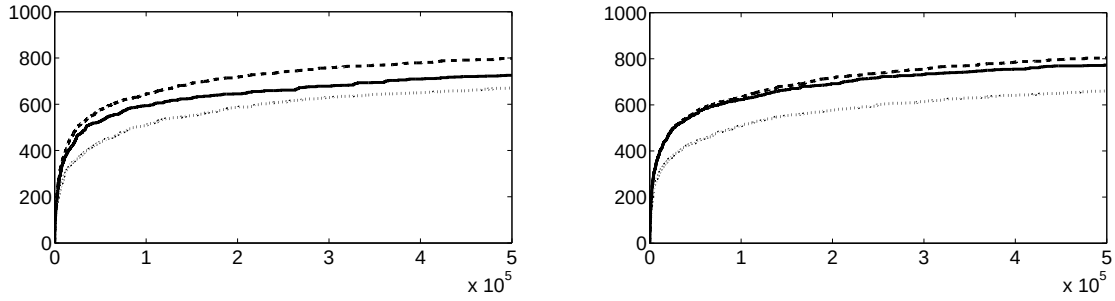


Figure 2: Number of prime numbers found by Good-UCB (solid), the OCL (dashed), and uniform sampling (dotted) as a function of time, using geometric experts with means $100, 300, 500, 700$ and 900, for $C = 0.1$ (left) and $C = 0.02$ (right).

In order to illustrate the efficiency of the Good-UCB algorithm in a more difficult setting, which does not satisfy any of the assumptions (i), (ii) and (iii), we also considered the following (artificial) example: $K = 5$ probabilistic experts draw independent sequences of geometrically distributed

random variables, with expectations 100, 300, 500, 700 and 900 respectively. The set of interesting items is the set of prime numbers. We compare the oracle closed-loop policy, Good-UCB and uniform sampling. The results are displayed in Figure 2. Even if the difference remains significant between Good-UCB and the OCL, the former still performs significantly better than uniform sampling during the entire discovery process. In this example, choosing a smaller parameter C seems to be preferable; this is due to the fact that the proportion of interesting items on each arm is low; in that case, it may be possible to show, by using tighter concentration inequalities, that the concentration of the Good-Turing estimator is actually better than suggested by Proposition 1. In fact, this experiment suggests that the value of C should be chosen smaller when the remaining missing mass is small.

Acknowledgments

We are especially thankful to one of the anonymous referees for suggesting to us to write Sections 2.3 and 2.4.

Appendix A.

Proof of lemma 2 We proceed by induction on t. For $t = 1$, the result is obvious. For $t > 1$, denote by $\bar{\pi}$ the policy choosing $I_1^{\bar{\pi}} = I_1^{\pi}$ and then playing like OCL for the $t-1$ remaining rounds. Denote $H_1 = (I_1^{\pi}, X_{I_1^{\pi},1})$, and $F^{\pi}(2:t)$ (respectively $F^{\bar{\pi}}(2:t)$) the number of interesting items found by policy π (respectively $\bar{\pi}$) between rounds 2 and t. Note that conditionally on H_1, $F^{\bar{\pi}}(2:t)$ corresponds to $F^*(t-1)$ in some modified problem (where one interesting item on expert I_1^{π} might have been removed from the set of interesting items). Thus one can apply the induction hypothesis to obtain

$$\mathbb{E}\left[F^{\pi}(2:t)|H_1\right] \leq \mathbb{E}\left[F^{\bar{\pi}}(2:t)|H_1\right].$$

Let us assume in the following that I_1^{π} is deterministic (we make this assumption only for sake of clarity, everything go through with a randomized choice of I_1^{π}). Then thanks to the above inequality one has

$$\mathbb{E}F^{\pi}(t) = R_{I_1^{\pi},0} + \mathbb{E}\left[F^{\pi}(2:t)\right] \leq R_{I_1^{\pi},0} + \mathbb{E}\left[F^{\bar{\pi}}(2:t)\right] = \mathbb{E}F^{\bar{\pi}}(t). \tag{8}$$

Now let

$$\tau = \min\{s \geq 1 : I_s^* = I_1^{\pi}\}.$$

On the event $\tau \leq t$, OCL and $\bar{\pi}$ observe exactly the same items during the t first rounds, and thus

$$F^{\bar{\pi}}(t)\mathbb{1}\{\tau \leq t\} = F^*(t)\mathbb{1}\{\tau \leq t\}. \tag{9}$$

On the other hand on the event $\tau > t$, $\bar{\pi}$ observe the same items between rounds 2 and t than OCL between rounds 1 and $t-1$, that is $F^{\bar{\pi}}(2:t)\mathbb{1}\{\tau > t\} = F^*(t-1)\mathbb{1}\{\tau > t\}$. Thanks to assumption (i), this implies (denoting $Y_1^*, \ldots, Y_t^*$ for the sequence of items observed by OCL),

$$F^{\bar{\pi}}(t)\mathbb{1}\{\tau > t\} = \left(\mathbb{1}\{X_{I_1^{\pi},1} \in A\} + F^*(t) - \mathbb{1}\{Y_t^* \in A \setminus \{Y_1^*, \ldots, Y_{t-1}^*\}\}\right)\mathbb{1}\{\tau > t\}. \tag{10}$$

By combining (8), (9) and (10), it only remains to show that

$$\mathbb{E}[\mathbb{1}\{X_{I_1^{\pi},1} \in A\}\mathbb{1}\{\tau > t\}] \leq \mathbb{E}[\mathbb{1}\{Y_t^* \in A \setminus \{Y_1^*, \ldots, Y_{t-1}^*\}\}\mathbb{1}\{\tau > t\}]. \tag{11}$$

Since $X_{I_1^\pi,1}$ is independent of $\mathbb{1}\{\tau > t\}$, one has $\mathbb{E}[\mathbb{1}\{X_{I_1^\pi,1} \in A\}\mathbb{1}\{\tau > t\}] = \mathbb{E}[R_{I_1^\pi,0}\mathbb{1}\{\tau > t\}]$. Moreover, noting that $\mathbb{1}\{\tau > t\}$, I_t^* and $R_{I_t^*,n^*_{I_t^*,t-1}}$ are measurable with respect to $H^*_{t-1} = \left(I_1^*, Y_1^*, \ldots, I_{t-1}^*, Y_{t-1}^*\right)$, one has

$$\mathbb{E}[\mathbb{1}\{Y_t^* \in A \setminus \{Y_1^*, \ldots, Y_{t-1}^*\}\}\mathbb{1}\{\tau > t\}] = \mathbb{E}[R_{I_t^*,n^*_{I_t^*,t-1}}\mathbb{1}\{\tau > t\}].$$

Finally remark that on the event $\tau > t$ one necessarily have that the remaining missing mass on the expert pulled at time t by OCL is larger than the initial missing mass of expert I_1^π, that is $R_{I_t^*,n^*_{I_t^*,t-1}}\mathbb{1}\{\tau > t\} \geq R_{I_1^\pi,0}\mathbb{1}\{\tau > t\}$, which concludes the proof of (11). ■

Proof of Lemma 3 Let $Y_s^\pi = X_{I_s^\pi,n_{I_s^\pi,s}}$ be the item observed by π at time step s, and $H_s^\pi = \left(I_1^\pi, Y_1^\pi, \ldots, I_{s-1}^\pi, Y_{s-1}^\pi\right)$ be the history of π prior to making the decision on time s. For any history $h_s = (i_1, y_1, \ldots, i_{s-1}, y_{s-1})$, let $F^*(t|h_s)$ be the number of newly discovered interesting items when running OCL from the history h_s for $t-s+1$ steps. 'From the history h_s' means that, prior to running OCL, the sequence of experts $i_1, \ldots, i_{s-1}$ has been chosen and has led to the observations $y_1, \ldots, y_{s-1}$. For $s' \geq s$ we shall also denote $I_{s'}^*(h_s)$ (respectively $Y_{s'}^*(h_s)$) the sequence of expert requests made by OCL starting at h_s (respectively the corresponding sequence of observed items). Note in particular that $\bar{I}_s$ defined in the statement of the lemma corresponds to $I_s^*(H_s^\pi)$. We shall also need τ_s to be the first time when OCL, running from history H_s^π, selects expert I_s^π, that is

$$\tau_s = \min\{s' \geq s : I_{s'}^*(H_s^\pi) = I_s^\pi\}\,,$$

and $\tau_s = +\infty$ if there is no interesting item to be found by expert I_s^π at time s.

We shall prove that

$$\mathbb{E}[F^*(t|H_s^\pi) - F^*(t|H_{s+1}^\pi)] \leq \mathbb{E}R_{\bar{I}_s,n^\pi_{\bar{I}_s,s-1}}, \tag{12}$$

which inductively yields the lemma since $F^*(t) = F^*(t|h_1)$ and $F^*(t|h_{t+1}) = 0$.

First let us consider the case when $\tau_s \leq t$. Then the observed items with OCL (running from H_s^π) between step s and t remains unchanged if one forces OCL to play I_s^π at time step s, that is

$$F^*(t|H_s^\pi)\mathbb{1}\{\tau_s \leq t\} = \mathbb{1}\{Y_s^\pi \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s \leq t\} + F^*(t|H_{s+1}^\pi)\mathbb{1}\{\tau_s \leq t\}.$$

On the other hand if $\tau_s > t$, the behavior of OCL will be the same if played for $t-s$ steps from H_s^π or from H_{s+1}^π, that is

$$F^*(t-1|H_s^\pi)\mathbb{1}\{\tau_s > t\} = F^*(t|H_{s+1}^\pi)\mathbb{1}\{\tau_s > t\}.$$

Moreover note that

$$F^*(t-1|H_s^\pi) = F^*(t|H_s^\pi) - \mathbb{1}\{Y_t^*(H_s^\pi) \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi, Y_s^*(H_s^\pi), \ldots, Y_{t-1}^*(H_s^\pi)\}\}.$$

Thus we proved so far that

$$\begin{aligned}
&F^*(t|H_s^\pi) - F^*(t|H_{s+1}^\pi) \\
&= \mathbb{1}\{Y_s^\pi \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s \leq t\} \\
&\quad + \mathbb{1}\{Y_t^*(H_s^\pi) \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi, Y_s^*(H_s^\pi), \ldots, Y_{t-1}^*(H_s^\pi)\}\}\mathbb{1}\{\tau_s > t\} \\
&\leq \mathbb{1}\{Y_s^\pi \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s \leq t\} + \mathbb{1}\{Y_t^*(H_s^\pi) \in A \setminus \{Y_1^\pi, \ldots, Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s > t\}.
\end{aligned}$$

Now remark that Y_s^π is independent of τ_s conditionally to H_s^π. Thus one immediately obtains

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}\{Y_s^\pi \in A \setminus \{Y_1^\pi,\dots,Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s \le t\}|H_s^\pi] \\
&= R_{I_s,n^\pi_{I_s,s-1}}\mathbb{E}[\mathbb{1}\{\tau_s \le t\}|H_s^\pi] \\
&\le R_{\bar{I}_s,n^\pi_{\bar{I}_s,s-1}}\mathbb{E}[\mathbb{1}\{\tau_s \le t\}|H_s^\pi].
\end{aligned}$$

Similarly $Y_t^*(H_s^\pi)$ is independent of $\mathbb{1}\{\tau_s > t\}$ conditionally to $(H_s^\pi, I_t^*(H_s^\pi))$ and thus

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}\{Y_t^*(H_s^\pi) \in A \setminus \{Y_1^\pi,\dots,Y_{s-1}^\pi\}\}\mathbb{1}\{\tau_s > t\}|H_s^\pi, I_t^*(H_s^\pi)] \\
&= \mathbb{E}[R_{I_t^*(H_s^\pi),n^\pi_{I_t^*(H_s^\pi),s-1}}|H_s^\pi, I_t^*(H_s^\pi)]\mathbb{E}[\mathbb{1}\{\tau_s > t\}|H_s^\pi, I_t^*(H_s^\pi)] \\
&\le R_{\bar{I}_s,n^\pi_{\bar{I}_s,s-1}}\mathbb{E}[\mathbb{1}\{\tau_s > t\}|H_s^\pi, I_t^*(H_s^\pi)].
\end{aligned}$$

Putting everything together one obtains (12), which concludes the proof. ■

Lemma 9 *Let $a > 0$, $b \ge 0.4$, and $x \ge e$, such that $x \le a + b\log x$. Then one has*

$$x \le a + b\log\left(2a + 4b\log(4b)\right).$$

Proof If $a \ge b\log x$ then $x \le 2a$ and thus $x \le a + b\log(2a)$. On the other hand if $a < b\log x$ then $x \le 2b\log x$ which easily implies $x \le 4b\log(4b)$ (indeed for $x \ge e$, $x \mapsto \frac{x}{\log x}$ is increasing and furthermore for $b \ge 0.4$ one can check that $4b\log(4b) > 2b\log(4b\log(4b))$) and thus $x \le a + b\log(4b\log(4b))$. In any case one has $x \le a + b\log\left(2a + 4b\log(4b)\right)$. ■

Lemma 10 *For all $1 \le k \le n$,*

$$-\frac{1}{k} + \log\frac{n}{k} \le \sum_{j=k+1}^{n}\frac{1}{j} \le \log\frac{n}{k}\,.$$

Proof The standard sum/integral comparison yields

$$\log\frac{n+1}{k+1} \le \sum_{j=k+1}^{n}\frac{1}{j} \le \log\frac{n}{k}\,,$$

but

$$\log\frac{n+1}{k+1} = \log\frac{n}{k} + \log\left(1+\frac{1}{n+1}\right) - \log\left(1+\frac{1}{k+1}\right) \ge \log\frac{n}{k} + 0 - \frac{1}{k}\,.$$

■

Appendix B. The Open-loop Oracle Policy

In this final section, we provide an macroscopic analysis of the open-loop oracle policy in the case of uniform sampling, that is under Hypotheses (i), (ii) and (iii). An open-loop policy must choose, for each horizon t, the respective numbers of requests $(n_1^N,\dots,n_K^N)$ for each distribution (so that $n_1^N+\cdots+n_K^N=t^N$) in advance. It appears here that, in the limit, the *oracle open-loop* (OOL) policy, which makes use of the parameters $(Q_1^N,\dots,Q_K^N)$, is as good as the OCL policy.

Let here $\underline{R}^N_{i,n_i^N}=(Q_i^N-F_i^N(n_i^N))/N$ be the proportion of interesting items not yet found with expert i after n_i^N requests. Suppose that $t^N/N\to t$, and that $n_i^N/N\to\nu_i$ as N goes to infinity; it is easily shown that, almost surely,

$$\lim_{N\to\infty}\underline{R}^N_{i,n_i^N}=\lim_{N\to\infty}\mathbb{E}\left[\underline{R}^N_{i,n_i^N}\right]=\lim_{N\to\infty}\frac{Q_i^N\left(1-\frac{1}{N}\right)^{n_i^N}}{N}=q_i\exp(-\nu_i)\,.$$

Hence, the proportion of interesting items found with the allocation $(n_1^N,\dots,n_K^N)$ almost surely converges to $\sum_{i=1}^K q_i(1-\exp(-\nu_i))$. Defining

$$r(\nu)=\sum_{i=1}^K q_i\exp(-\nu_i)\,,$$

it follows that finding the best macroscopic allocation reduces to the following constrained convex minimization problem:

$$\min_{\nu\in\mathbb{R}^K} r(\nu)\quad\text{such that }\nu_1+\cdots+\nu_K=t\text{ and }\forall i,\,\nu_i\geq 0\,.$$

The solution $r^*(t)$, reached at $\nu=\nu^*(t)$, is easily derived by classical optimization techniques:

Proposition 11 *For every $i\in\{1,\dots,K\}$, let $\underline{q}_i=\exp\left(1/i\times\sum_{k=1}^i\log q_k\right)$ denotes the geometric mean of $q_1,\dots,q_i$.*

1. *There exists $I(t)\in\{1,\dots,K\}$ such that*

$$\begin{cases}\forall i\leq I(t), & \nu_i^*(t)=\frac{t}{I(t)}+\log\frac{q_i}{\underline{q}_{I(t)}}\\ \forall i>I(t), & \nu_i^*(t)=0\,.\end{cases}$$

   Hence,

$$r^*(t)=I(t)\underline{q}_{I(t)}\exp\left(-\frac{t}{I(t)}\right)+\sum_{i>I(t)}q_i\,.$$

2. *There exists $1=t_1\leq\cdots\leq t_K<+\infty$ such that*

$$\forall t\in[t_i,t_{i+1}[,\ I(t)=i\,.$$

   The $(t_k)_k$ are such that

$$q_i+(i-1)\underline{q}_{i-1}\exp\left(-\frac{t_i}{i-1}\right)=i\underline{q}_i\exp\left(-\frac{t_i}{i}\right)\,.$$

   For instance, $t_1=\log(q_1/q_2)$.

Proof: Introduce the Lagrangian:

$$L(\nu_1,\ldots,\nu_K,\lambda,\mu_1,\ldots,\mu_K) = \sum_{i=1}^{K} q_i \exp\left(-\frac{\nu_i}{N}\right) + \lambda\left(\sum_{i=1}^{K}\nu_i\right) - \sum_{i=1}^{K}\mu_i\nu_i\,.$$

We need to find the solution of:

$$\begin{gathered}\forall i\in\{1,\ldots,M\},\ \ -q_i\exp(-\nu_i)+\lambda-\mu_i=0,\\ \sum_{i=1}^{K}\nu_i = t,\\ \forall i\in\{1,\ldots,M\},\ \mu_i\nu_i=0 \text{ and } \mu_i\geq 0\,.\end{gathered}$$

We first obtain that

$$\nu_i = \log q_i - \log(\lambda-\mu_i)\,.$$

Denoting $A=\{i:\nu_i>0\}$, and using that $i\in A \implies \mu_i=0$, we get

$$t = \sum_{i\in A}\log(q_i) - |A|\log(\lambda)\,,$$

from which we get

$$-\log(\lambda) = \frac{t}{|A|} - \frac{1}{|A|}\sum_{i\in A}\log q_i,$$

and then for all $i\in A$:

$$\nu_i = \log q_i + \frac{t}{|A|} - \frac{1}{|A|}\sum_{i\in A}\log q_i\,.$$

Next, observe that $\nu_i=0 \iff q_i>\lambda$: in fact, if $\nu_i=0$ then the first equation gives $-q_i+\lambda-\mu_i=0$, and $0\leq\mu_i=\lambda-q_i$. Conversely, if $\nu_i>0$ then $\mu_i=0$ and $\nu_i=\log(q_i/\lambda)>0$ implies $q_i>\lambda$. Thus, there exists $I(t)$ such that $A=\{1,\ldots,I(t)\}$, and for all $i\leq I(t)$,

$$\nu_i = \log\frac{q_i}{\underline{q}_{I(t)}} + \frac{t}{I(t)}\,.$$

Moreover,

$$\begin{aligned} r^*(t) &= r\left(\nu_1,\ldots,\nu_{I(t)},0,\ldots,0\right)\\ &= \sum_{i\leq I(t)} q_i\exp\left[-\left(\log\frac{q_i}{\underline{q}_{I(t)}}+\frac{t}{I(t)}\right)\right] + \sum_{i>I(t)} q_i\\ &= I(t)\underline{q}_{I(t)}\exp\left(-\frac{t}{I(t)}\right) + \sum_{i>I(t)} q_i\,.\end{aligned}$$

The instants $(t_i)_{1\leq i\leq K}$ are such that

$$(i-1)\underline{q}_{i-1}\exp\left(-\frac{t_i}{i-1}\right) + \sum_{k>i-1} q_k = i\underline{q}_i\exp\left(-\frac{t_i}{i}\right) + \sum_{k>i} q_k\,,$$

which is equivalent to

$$q_i + (i-1)\underline{q}_{i-1}\exp\left(-\frac{t_i}{i-1}\right) = i\underline{q}_i\exp\left(-\frac{t_i}{i}\right) .$$

For $i = 2$, this gives

$$0 = q_2 + q_1\exp(-\nu_2) - 2\sqrt{q_1 q_2}\exp\left(-\frac{\nu_2}{2}\right) = \left(\sqrt{q_2} - \sqrt{q_1\exp(-\nu_2)}\right)^2 ,$$

which leads to $t_1 = \log(q_1/q_2)$.

Theorem 12 *In the macroscopic limit, the proportion of items found by the open-loop oracle policy uniformly converges to the function F defined in Equation* (7)*.*

The proportion of interesting items found by the OOL policy is

$$q - r^*(t) = \sum_{i\leq I(t)}\left[q_i - \underline{q}_{I(t)}\exp\left(-\frac{t}{I(t)}\right)\right] = \sum_{i=1}^{K}(q_i - \Lambda(t))_+ ,$$

where $\Lambda(t) = \underline{q}_{I(t)}\exp\left(-\frac{t}{I(t)}\right) \in [0, q_{I(t)}]$. To conclude, it remains only to remark that $\Lambda = T^{-1}$, where T is defined in Equation (6). In fact, if λ is such that $q_{i_0+1} < \lambda \leq q_{i_0}$, then $I(T(\lambda)) = i_0$ and

$$\Lambda(T(\lambda)) = \underline{q}_{i_0}\exp\left(-\frac{T(\lambda)}{i_0}\right) = \exp\left(\frac{1}{i_0}\sum_{i\leq i_0}\log q_i\right)\exp\left(-\frac{\sum_{i\leq i_0}\log(q_i/\lambda)}{i_0}\right) = \lambda .$$

If $\lambda < q_K$, the same holds with $i_0 = K$.

References

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

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

F. Fonteneau-Belmudes. *Identification of Dangerous Contingencies for Large Scale Power System Security Assessment*. PhD thesis, University of Liège, 2012.

F. Fonteneau-Belmudes, D. Ernst, C. Druet, P. Panciatici, and L. Wehenkel. Consequence driven decomposition of large-scale power system security analysis. In *Proceedings of the 2010 IREP Symposium - Bulk Power Systems Dynamics and Control - VIII*, Buzios, Rio de Janeiro, Brazil, August 2010.

W.A. Gale and G. Sampson. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24rd Annual International Conference on Learning Theory*, 2011.

I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953. ISSN 0006-3444.

D. McAllester and L. Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *J. Mach. Learn. Res.*, 4:895–911, December 2003. ISSN 1532-4435.

D.A. McAllester and R.E. Schapire. On the convergence rate of Good-Turing estimators. In *COLT*, pages 1–6, 2000.

C. McDiarmid. On the method of bounded differences. In *Surveys in combinatorics, 1989 (Norwich, 1989)*, volume 141 of *London Math. Soc. Lecture Note Ser.*, pages 148–188. Cambridge Univ. Press, Cambridge, 1989.

A. Orlitsky, N.P. Santhanam, and J. Zhang. Always good Turing: Asymptotically optimal probability estimation. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 179+, Washington, DC, USA. IEEE Computer Society. ISBN 0-7695-2040-5.

Journal of Machine Learning Research 14 (2013) 625-628 Submitted 9/11; Revised 6/12; Published 2/13

A C++ Template-Based Reinforcement Learning Library: Fitting the Code to the Mathematics

Hervé Frezza-Buet[*] HERVE.FREZZA-BUET@SUPELEC.FR
Matthieu Geist MATTHIEU.GEIST@SUPELEC.FR
Supélec
2 rue Édouard Belin
57070 Metz, France

Editor: Mikio Braun

Abstract

This paper introduces the rllib as an original C++ template-based library oriented toward value function estimation. Generic programming is promoted here as a way of having a good fit between the mathematics of reinforcement learning and their implementation in a library. The main concepts of rllib are presented, as well as a short example.

Keywords: reinforcement learning, C++, generic programming

1. C++ Genericity for Fitting the Mathematics of Reinforcement Learning

Reinforcement learning (RL) is a field of machine learning that benefits from a rigorous mathematical formalism, as shown for example by Bertsekas (1995). Although this formalism is well accepted in the field, its translation into efficient computer science tools has surprisingly not led to any standard yet, as mentioned by Kovacs and Egginton (2011). The claim of this paper is that genericity enables a natural expression of the mathematics of RL. The rllib (2011) library implements this idea in the C++ language, where genericity relies on templates. Templates automate the re-writing of some generic code involving user types, offering a strong type checking at compile time that improves the code safety.

Using the rllib templates requires that the user-defined types fit some documented concepts. For example, some class C defining an agent should be designed so that C::state_type is the type used for the states, C::action_type is the type used for the actions, and the method C::action_type policy(const C::state_type& s) const is implemented, in order to compute the action to be performed in a given state. This concept definition specifies what is required for an agent mathematically. Note that C does not need to inherit from any kind of abstract rl::Agent class to be used by the rllib tools. It can be directly provided as a type argument to any rllib template requiring an argument fitting the concept of an agent, so that the re-written code actually compiles.

2. A Short Example

Let us consider the following toy-example. The state space contains values from 0 to 9, and actions consist in increasing or decreasing the value. When the value gets out of bounds, a reward is returned

∗. Also at UMI 2958 Georgia Tech / CNRS, 2-3, rue Marconi, 57070 Metz, France.

(-1 for bound 0, 1 for bound 9). Otherwise, a null reward is returned. Let us define this problem and run Sarsa. First, a simulator class fitting the concept `Simulator` described in the documentation is needed.

```
class Sim { // Our simulator class. No inheritance required.
private:
  int current; double r;
public:
  typedef int             phase_type  ; typedef int    observation_type;
  typedef enum {up,down} action_type ; typedef double reward_type;

  Sim(void) : current(0), r(0) {}
  void setPhase(const phase_type &s)  {current = s%10;}
  const observation_type& sense(void) const {return current;}
  reward_type reward(void) const {return r;}
  void timeStep(const action_type &a) {
    if(a == up) current++; else current--;
    if(current < 0) r=-1; else if(current > 9) r=1; else r=0;
    if(r != 0) throw rl::exception::Terminal("Out of range");
  }
};
```

Following the concept requirements, the class `Sim` naturally implements a sensor method `sense` that provides an observation from the current phase of the controlled dynamical system, and a method `timeStep` that computes a transition consecutive to some action. Note the use of exceptions for terminal states. For the sake of simplicity in further code, the following is added.

```
typedef Sim::phase_type                          S;
typedef Sim::action_type                         A;
typedef rl::Iterator<A,Sim::up,Sim::down> Aenum; // enumerate all actions.
typedef rl::SA<S,A>                              SA;
typedef rl::sa::Transition<S,A,double>     Transition; // i.e., (s,a,r,s',a')
```

As Sarsa computes some Q-values, a structure is needed to store these values. A 10×2 array could be used, but our library is oriented toward value function estimation. It means that the Q-function computed by Sarsa is an element taken in some parametrized set of functions from $S \times A$ to $\mathbb{R}$. The parameter used is a vector. Such Q-function fits the concept of `DerivableArchitecture`. Let us here consider a tabular representation of Q: a $|S| \times |A| = 20$ array is actually implemented, but it is viewed as some particular case of parametrized function (there is one dimension in the parameter vector for each Q-value in the array, that is, the parameter vector *is* the array). This leads to the following definition of `Q`. The explicit definition of the gradient of `Q` according to the parameter is required.

```
class Q { // Tabular representation of Q (theta contains the Q[s,a]).
public:
  typedef S           state_type          ; typedef A          action_type;
  typedef SA          sa_type             ; typedef SA         input_type;
  typedef Transition  sa_transition_type  ; typedef Transition transition_type;
  typedef gsl_vector* param_type          ; typedef double     output_type;

  param_type newParameterInstance(void) const {
    return gsl_vector_calloc(20); // 20 = |S|*|A|
  }
  output_type operator()(const param_type theta,
                         state_type s, action_type a) const {
    return gsl_vector_get(theta,2*s+a); // return the value q_theta(s,a).
  }
  void gradient(const param_type theta,
                state_type s,action_type a,
                param_type grad) const {
```

```
    gsl_vector_set_basis(grad,2*s+a); // ..001000... with 1 at [s,a]
  }
};
```

The simulator and the parametrized Q-function are the only classes to be defined, since they are problem-dependent. From these types, the Sarsa algorithm can be easily implemented from `rllib` templates, as well as different kinds of agents. Here, learning is performed by an ε-greedy agent, while testing is executed by a greedy agent.

```
class Param {
public:
  static double gamma(void)   {return .99;}
  static double alpha(void)   {return .05;}
  static double epsilon(void) {return 0.2;}
};
typedef rl::sa::SARSA<Q,Param>                                          Critic;
typedef rl::sa::ArgmaxFromAIteration<Critic,Aenum>                      ArgmaxCritic;
typedef rl::agent::Greedy<ArgmaxCritic>                                 TestAgent;
typedef rl::agent::online::EpsilonGreedy<ArgmaxCritic,Aenum,Param>      LearnAgent;
```

The `rllib` expresses that Sarsa provides a critic, offering a Q-function. As actions are discrete, the best action (i.e., $\text{argmax}_{a \in A} Q(s,a)$) can be found by considering all the actions sequentially. This is what `ArgmaxCritic` offers thanks to the action enumerator `Aenum`, in order to define greedy and ε-greedy agents. The main function then only consists in running episodes with the appropriate agents.

```
int main(int argc, char* argv[]) {
  Sim          simulator;              // This is what the agent controls.
  Transition   transition;             // This is some s,a,r,s',a' data.
  ArgmaxCritic critic;                 // This computes Q and argmax_a Q(s,a).
  LearnAgent   learner(critic);        // SARSA uses this agent to learn the policy.
  TestAgent    tester(critic);         // This behaves according to the critic.
  A            a;                      // Some action.
  S            s;                      // Some state.
  int          episode,length,step=0;

  for(episode = 0; episode < 10000; ++episode) { // Learning phase
    simulator.setPhase(rand()%10);
    rl::episode::sa::run_and_learn(simulator,learner,transition,0,length);
  }
  try { // Test phase
    simulator.setPhase(0);
    while(true) {
      s = simulator.sense(); a = tester.policy(s);
      step++; simulator.timeStep(a);
    }
  }
  catch(rl::exception::Terminal e) {std::cout << step << "␣steps." << std::endl;}
  return 0; // the message printed is ``10 steps.''
}
```

3. Features of the Library

Using the library requires to define the features that are specific to the problem (the simulator and the Q-function architecture in our example) from scratch, but with the help of concepts. Then, the specific features can be handled by generic code provided by the library to implement RL techniques with value function estimation.

Currently, Q-learing, Sarsa, KTD-Q, LSTD, and policy iteration are available, as well as a multi-layer perceptron architecture. Moreover, some benchmark problems (i.e., simulators) are also provided: the mountain car, the cliff walking, the inverted pendulum and the Boyan chain. Extending the library with new algorithms is allowed, since it consists in defining new templates. This is a bit more technical than only using the existing algorithms, but the structure of existing concepts helps, since it reflects the mathematics of RL. For example, concepts like `Feature`, for linear approaches mainly (i.e., $Q(s,a) = \theta^{\mathrm{T}}\varphi(s,a)$) and `Architecture` (i.e., $Q(s,a) = f_\theta(s,a)$ for more general approximation) orient the design toward functional approaches of RL. The algorithms implemented so far rely on the GNU Scientific Library (see `GSL`, 2011) for linear algebra computation, so the `GPL` licence of `GSL` propagates to the `rllib`.

4. Conclusion

The `rllib` relies only on the `C++` standard and the availability of the `GSL` on the system. It offers state-action function approximation tools for applying RL to real problems, as well as a design that fits the mathematics. The latter allows for extensions, but is also compliant with pedagogical purpose. The design of the `rllib` aims at allowing the user to build (using `C++` programming) its own experiment, using several algorithms, several agents, on-line or batch learning, and so on. Actually, the difficult part of RL is the algorithms themselves, not the script-like part of the experiment where things are put together (see the main function in our example). With a framework, in the sense of Kovacs and Egginton (2011), the experiment is not directly accessible to the user programs, since it is handled by some libraries in order to offer graphical interface or analyzing tools. The user code is then called by the framework when required. We advocate that allowing the user to call the `rllib` functionality at his/her convenience provides an open and extensible access to RL for students, researchers and engineers.

Last, the `rllib` fits the requirements expressed by Kovacs and Egginton (2011, Section 4.3): support of good scientific research, formulation compliant with the domain, allowing for any kind of agents and any kind of approximators, interoperability of components (the Q function of the example can be used for different algorithms and agents), maximization of run-time speed (use of `C++` and templates that inline massively the code), open source, etc. Extensions of `rllib` can be considered, for example for handling POMDPs, and contributions of users are expected. The use of templates is unfortunately unfamiliar to many programmers, but the effort is worth it, since it brings the code at the level of the mathematical formalism, increasing readability (by a rational use of typedefs) and reducing bugs. Even if the approach is dramatically different from existing frameworks, wrappings with frameworks can be considered in further development.

References

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd (2005-2007) edition, 1995.

`GSL`, 2011. `http://http://www.gnu.org/software/gsl`.

Tim Kovacs and Robert Egginton. On the analysis and design of software for reinforcement learning, with a survey of existing systems. *Machine Learning*, 84:7–49, 2011.

`rllib`, 2011. `http://ims.metz.supelec.fr/spip.php?article122`.

Journal of Machine Learning Research 14 (2013) 629-671 Submitted 3/12; Revised 10/12; Published 2/13

CODA: High Dimensional Copula Discriminant Analysis

Fang Han FHAN@JHSPH.EDU
Department of Biostatistics
Johns Hopkins University
Baltimore, MD 21205, USA

Tuo Zhao TOURZHAO@JHU.EDU
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA

Han Liu HANLIU@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Tong Zhang

Abstract

We propose a high dimensional classification method, named the *Copula Discriminant Analysis* (CODA). The CODA generalizes the normal-based linear discriminant analysis to the larger Gaussian Copula models (or the nonparanormal) as proposed by Liu et al. (2009). To simultaneously achieve estimation efficiency and robustness, the nonparametric rank-based methods including the Spearman's rho and Kendall's tau are exploited in estimating the covariance matrix. In high dimensional settings, we prove that the sparsity pattern of the discriminant features can be consistently recovered with the parametric rate, and the expected misclassification error is consistent to the Bayes risk. Our theory is backed up by careful numerical experiments, which show that the extra flexibility gained by the CODA method incurs little efficiency loss even when the data are truly Gaussian. These results suggest that the CODA method can be an alternative choice besides the normal-based high dimensional linear discriminant analysis.

Keywords: high dimensional statistics, sparse nonlinear discriminant analysis, Gaussian copula, nonparanormal distribution, rank-based statistics

1. Introduction

High dimensional classification is of great interest to both computer scientists and statisticians. Bickel and Levina (2004) show that the classical low dimensional normal-based linear discriminant analysis (LDA) is asymptotically equivalent to random guess when the dimension d increases fast compared to the sample size n, even if the Gaussian assumption is correct. To handle this problem, a sparsity condition is commonly added, resulting in many follow-up works in recent years. A variety of methods in sparse linear discriminant analysis, including the nearest shrunken centroids (Tibshirani et al., 2002; Wang and Zhu, 2007) and feature annealed independence rules (Fan and Fan, 2008), are based on a working independence assumption. Recently, numerous alternative approaches have been proposed by taking more complex covariance matrix structures into consideration (Fan et al., 2010; Shao et al., 2011; Cai and Liu, 2012; Mai et al., 2012).

A binary classification problem can be formulated as follows: suppose that we have a training set $\{(\boldsymbol{x}_i, y_i), i = 1, ..., n\}$ independently drawn from a joint distribution of $(\boldsymbol{X}, Y)$, where $\boldsymbol{X} \in \mathbb{R}^d$ and $Y \in \{0, 1\}$. The target of the classification is to determine the value of Y given a new data point $\boldsymbol{x}$. Let $\psi_0(\boldsymbol{x})$ and $\psi_1(\boldsymbol{x})$ be the density functions of $(\boldsymbol{X}|Y = 0)$ and $(\boldsymbol{X}|Y = 1)$, and the prior probabilities $\pi_0 = \mathbb{P}(Y = 0)$, $\pi_1 = \mathbb{P}(Y = 1)$. It is well known that the Bayes rule classifies a new data point $\boldsymbol{x}$ to the second class if and only if

$$\log \psi_1(\boldsymbol{x}) - \log \psi_0(\boldsymbol{x}) + \log(\pi_1/\pi_0) > 0. \tag{1}$$

Specifically, when $(\boldsymbol{X}|Y = 0) \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$, $(\boldsymbol{X}|Y = 1) \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $\pi_0 = \pi_1$, Equation (1) is equivalent to the following classifier:

$$g^*(\boldsymbol{x}) := I((\boldsymbol{x} - \boldsymbol{\mu}_a)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_d > 0),$$

where $\boldsymbol{\mu}_a := \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0}{2}$, $\boldsymbol{\mu}_d := \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$, and $I(\cdot)$ is the indicator function. It is well known then that for any linear discriminant rule with respect to $\boldsymbol{w} \in \mathbb{R}^d$:

$$g_{\boldsymbol{w}}(\boldsymbol{X}) := I((\boldsymbol{X} - \boldsymbol{\mu}_a)^T \boldsymbol{w} > 0), \tag{2}$$

the corresponding misclassification error is

$$C(g_{\boldsymbol{w}}) = 1 - \Phi\left(\frac{\boldsymbol{w}^T \boldsymbol{\mu}_d}{\sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}}}\right), \tag{3}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian. By simple calculation, we have

$$\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_d \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} C(g_{\boldsymbol{w}}),$$

and we denote by $\boldsymbol{\beta}^* := \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_d$. In exploring discriminant rules with a similar form as Equation (2), both Tibshirani et al. (2002) and Fan and Fan (2008) assume a working independence structure for Σ. However this assumption is often violated in real applications.

Alternatively, Fan et al. (2010) propose the Regularized Optimal Affine Discriminant (ROAD) approach. Let $\widehat{\boldsymbol{\Sigma}}$ and $\widehat{\boldsymbol{\mu}}_d$ be consistent estimators of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_d$. To minimize $C(g_{\boldsymbol{w}})$ in Equation (3), the ROAD minimizes $\boldsymbol{w}^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{w}$ with $\boldsymbol{w}^T \widehat{\boldsymbol{\mu}}_d$ restricted to be a constant value, that is,

$$\min_{\boldsymbol{w}^T \widehat{\boldsymbol{\mu}}_d = 1} \{\boldsymbol{w}^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{w}, \text{subject to} \, ||\boldsymbol{w}||_1 \leq c\}.$$

Later, Cai and Liu (2012) propose another version of the sparse LDA, which tries to make $\boldsymbol{w}$ close to the Bayes rule's linear term $\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_d$ in the ℓ_∞ norm (detailed definitions are provided in the next section), that is,

$$\min_{\boldsymbol{w}} \{ ||\boldsymbol{w}||_1, \text{subject to} \, ||\widehat{\boldsymbol{\Sigma}} \boldsymbol{w} - \widehat{\boldsymbol{\mu}}_d||_\infty \leq \lambda_n\}. \tag{4}$$

Equation (4) turns out to be a linear programming problem highly related to the Dantzig selector (Candes and Tao, 2007; Yuan, 2010; Cai et al., 2011).

Very recently, Mai et al. (2012) propose another version of the sparse linear discriminant analysis based on an equivalent least square formulation of the LDA. We will explain it in more details in Section 3. In brief, to avoid the "the curse of dimensionality", an ℓ_1 penalty is added in all three

methods to encourage a sparsity pattern of $\boldsymbol{w}$, and hence nice theoretical properties can be obtained under certain regularity conditions. However, though significant process has been made, all these methods require the normality assumptions which can be restrictive in applications.

There are three issues with regard to high dimensional linear discriminant analysis: (1) How to estimate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ accurately and efficiently (Rothman et al., 2008; Friedman et al., 2007; Ravikumar et al., 2009; Scheinberg et al., 2010); (2) How to incorporate the covariance estimator to classification (Fan et al., 2010; Shao et al., 2011; Cai and Liu, 2012; Witten and Tibshirani, 2011; Mai et al., 2012); (3) How to deal with non-Gaussian data (Lin and Jeon, 2003; Hastie and Tibshirani, 1996). In this paper, we propose a high dimensional classification method, named the Copula Discriminant Analysis (CODA), which addresses all the above three questions.

To handle non-Gaussian data, we extend the underlying conditional distributions of $(\boldsymbol{X}|Y=0)$ and $(\boldsymbol{X}|Y=1)$ from Gaussian to the larger nonparanormal family (Liu et al., 2009). A random variable $\boldsymbol{X}=(X_1,...,X_d)^T$ belongs to a nonparanormal family if and only if there exists a set of univariate strictly increasing functions $\{f_j\}_{j=1}^d$ such that $(f_1(X_1),...,f_d(X_d))^T$ is multivariate Gaussian.

To estimate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ robustly and efficiently, instead of estimating the transformation functions $\{f_j\}_{j=1}^d$ as Liu et al. (2009) did, we exploit the nonparametric rank-based correlation coefficient estimators including the Spearman's rho and Kendall's tau, which are invariant to the strictly increasing functions f_j. They have been shown to enjoy the optimal parametric rate in estimating the correlation matrix (Liu et al., 2012; Xue and Zou, 2012). Unlike previous analysis, a new contribution of this paper is that we provide an extra condition on the transformation functions which guarantees the fast rates of convergence of the marginal mean and standard deviation estimators, such that the covariance matrix can also be estimated with the parametric rate.

To incorporate the estimated covariance matrix into high dimensional classification, we show that the ROAD (Fan et al., 2010) is connected to the lasso in the sense that if we fix the second tuning parameter, these two problems are equivalent. Using this connection, we prove that the CODA is variable selection consistent.

Unlike the parametric cases, one new challenge for the CODA is that the rank-based covariance matrix estimator may not be positive semidefinite which makes the objective function nonconvex. To solve this problem, we first project the estimated covariance matrix into the cone of positive semidefinite matrices (using elementwise sup-norm). It can be proven that the theoretical properties are preserved in this way.

Finally, to show that the expected misclassification error is consistent to the Bayes risk, we quantify the difference between the CODA classifier $\widehat{g}^{npn}$ and the Bayes rule g^*. To this end, we measure the convergence rate of the estimated transformation function $\{\widetilde{f}_j\}_{j=1}^d$ to the true transformation function $\{f_j\}_{j=1}^d$. Under certain regularity conditions, we show that

$$\sup_{I_{n,\gamma}} |\widetilde{f}_j - f_j| = O_P\left(n^{-\frac{\gamma}{2}}\right), \quad \forall\, j \in \{1,2,\ldots,d\},$$

over an expanding site $I_{n,\gamma}$ determined by the sample size n and a parameter γ (detailed definitions will be provided later). $I_{n,\gamma}$ is set to go to $(-\infty,\infty)$. Using this result, we can show that:

$$\mathbb{E}(\mathcal{C}(\widehat{g}^{npn})) = \mathcal{C}(g^*) + o(1).$$

A related approach to our method has been proposed by Lin and Jeon (2003). They also consider the Gaussian copula family. However, their focus is on fixed dimensions. In contrast, this paper focuses on increasing dimensions and provides a thorough theoretical analysis.

The rest of this paper is organized as follows. In the next section, we briefly review the nonparanormal estimators (Liu et al., 2009, 2012). In Section 3, we present the CODA method. We give a theoretical analysis of the CODA estimator in Section 4, with more detailed proofs collected in the appendix. In Section 5, we present numerical results on both simulated and real data. More discussions are presented in the last section.

2. Background

We start with notations: for any two real values $a, b \in \mathbb{R}$, $a \wedge b := \min(a, b)$. Let $\mathbf{M} = [M_{jk}] \in \mathbb{R}^{d \times d}$ and $\boldsymbol{v} = (v_1, ..., v_d)^T \in \mathbb{R}^d$. Let $\boldsymbol{v}_{-j} := (v_1, ..., v_{d-1}, v_{j+1}, ..., v_d)^T$ and $\mathbf{M}_{-j,-k}$ be the matrix with $\mathbf{M}$'s j-th row and k-th column removed, $\mathbf{M}_{j,-k}$ be $\mathbf{M}$'s j-th row with the k-th column removed, $\mathbf{M}_{-j,k}$ be $\mathbf{M}$'s k-th column with the j-th row removed. Moreover, $\boldsymbol{v}$'s subvector with entries indexed by I is denoted by $\mathbf{v}_I$, $\mathbf{M}$'s submatrix with rows indexed by I and columns indexed by J is denoted by $\mathbf{M}_{IJ}$, $\mathbf{M}$'s submatrix with all rows and columns indexed by J is denoted by $\mathbf{M}_{\cdot J}$, $\mathbf{M}$'s submatrix with all rows and columns indexed by J is denoted by $\mathbf{M}_J$. For $0 < q < \infty$, we define

$$||\boldsymbol{v}_0|| := \text{card}(\text{support}(\boldsymbol{v})), \quad ||\boldsymbol{v}||_q := \left(\sum_{i=1}^{d} |v_i|^q \right)^{1/q}, \text{ and } ||\boldsymbol{v}||_\infty := \max_{1 \le i \le d} |v_i|.$$

We define the matrix $\ell_{\max}$ norm as the elementwise maximum value: $||\mathbf{M}||_{\max} := \max\{|M_{ij}|\}$ and the ℓ_∞ norm as $||\mathbf{M}||_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |M_{ij}|$. $\lambda_{\min}(\mathbf{M})$ and $\lambda_{\max}(\mathbf{M})$ are the smallest and largest eigenvalues of $\mathbf{M}$. We define the matrix operator norm as $||\mathbf{M}||_{\text{op}} := \lambda_{\max}(\mathbf{M})$.

2.1 The *nonparanormal*

A random variable $\boldsymbol{X} = (X_1, ..., X_d)^T$ is said to follow a *nonparanormal* distribution if and only if there exists a set of univariate strictly increasing transformations $f = \{f_j\}_{j=1}^d$ such that:

$$f(\boldsymbol{X}) = (f_1(X_1), ..., f_d(X_d))^T := \boldsymbol{Z} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\mu} = (\mu_1, ..., \mu_d)^T$, $\boldsymbol{\Sigma} = [\Sigma_{jk}]$, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, $\boldsymbol{\Sigma}^0 = [\Sigma^0_{jk}]$ are the mean, covariance, concentration and correlation matrices of the Gaussian distribution $\boldsymbol{Z}$. $\{\sigma_j^2 := \Sigma_{jj}\}_{j=1}^d$ are the corresponding marginal variances. To make the model identifiable, we add two constraints on f such that f preserves the population means and standard deviations. In other words, for $1 \le j \le d$,

$$\mathbb{E}(X_j) = \mathbb{E}(f_j(X_j)) = \mu_j; \;\; \text{Var}(X_j) = \text{Var}(f_j(X_j)) = \sigma_j^2.$$

In summary, we denote by such $\boldsymbol{X} \sim NPN(\boldsymbol{\mu}, \boldsymbol{\Sigma}, f)$. Liu et al. (2009) prove that the nonparanormal is highly related to the Gaussian Copula (Clemen and Reilly, 1999; Klaassen and Wellner, 1997).

2.2 Correlation Matrix and Transformation Functions Estimations

Liu et al. (2009) suggest a normal-score based correlation coefficient matrix to estimate $\mathbf{\Sigma}^0$. More specifically, let $\boldsymbol{x}_1,...,\boldsymbol{x}_n \in \mathbb{R}^d$ be n data point where $\boldsymbol{x}_i = (x_{i1},\ldots,x_{id})^T$. We define

$$\widetilde{F}_j(t;\delta_n,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) := T_{\delta_n}\left(\frac{1}{n}\sum_{i=1}^{n} I(x_{ij}\le t)\right), \tag{5}$$

to be the winsorized empirical cumulative distribution function of X_j. Here

$$T_{\delta_n}(x) := \begin{cases} \delta_n, & \text{if} \quad x < \delta_n, \\ x, & \text{if} \quad \delta_n \le x \le 1-\delta_n, \\ 1-\delta_n, & \text{if} \quad x > 1-\delta_n. \end{cases}$$

In particular, the empirical cumulative distribution function $\widehat{F}_j(t;\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) := \widetilde{F}_j(t;0,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$ by letting $\delta_n = 0$. Let $\Phi^{-1}(\cdot)$ be the quantile function of standard Gaussian, we define

$$\widetilde{f}_j(t) = \Phi^{-1}\big(\widetilde{F}_j(t)\big),$$

and the corresponding sample correlation estimator $\widehat{\mathbf{R}}^{ns} = [\widehat{R}^{ns}_{jk}]$ to be:

$$\widehat{R}^{\text{ns}}_{jk} := \frac{\frac{1}{n}\sum_{i=1}^{n}\widetilde{f}_j(x_{ij})\widetilde{f}_k(x_{ik})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\widetilde{f}_j^2(x_{ij})}\cdot\sqrt{\frac{1}{n}\sum_{i=1}^{n}\widetilde{f}_k^2(x_{ik})}}.$$

Liu et al. (2009) suggest to use the truncation level $\delta_n = \frac{1}{4n^{1/4}\sqrt{\pi\log n}}$ and prove that

$$||\widehat{\mathbf{R}}^{ns} - \mathbf{\Sigma}^0||_{\max} = O_p\left(\sqrt{\frac{\log d\log^2 n}{n^{1/2}}}\right).$$

In contrast, Liu et al. (2012) propose a different approach for estimating the correlations, called the *Nonparanormal* SKEPTIC. The Nonparanormal SKEPTIC exploits the Spearman's rho and Kendall's tau to directly estimate the unknown correlation matrix.

In specific, let r_{ij} be the rank of x_{ij} among $x_{1j},\ldots,x_{nj}$ and $\bar{r}_j = \frac{1}{n}\sum_{i=1}^{n} r_{ij}$. We consider the following two statistics:

$$\text{(Spearman's rho)}\ \widehat{\rho}_{jk} = \frac{\sum_{i=1}^{n}(r_{ij}-\bar{r}_j)(r_{ik}-\bar{r}_k)}{\sqrt{\sum_{i=1}^{n}(r_{ij}-\bar{r}_j)^2\cdot\sum_{i=1}^{n}(r_{ik}-\bar{r}_k)^2}},$$

$$\text{(Kendall's tau)}\ \widehat{\tau}_{jk} = \frac{2}{n(n-1)}\sum_{1\le i<i'\le n}\text{sign}\left(x_{ij}-x_{i'j}\right)\left(x_{ik}-x_{i'k}\right).$$

and the correlation matrix estimators:

$$\widehat{R}^{\rho}_{jk} = \begin{cases} 2\sin\left(\frac{\pi}{6}\widehat{\rho}_{jk}\right) & j\ne k \\ 1 & j=k \end{cases} \quad\text{and}\quad \widehat{R}^{\tau}_{jk} = \begin{cases} \sin\left(\frac{\pi}{2}\widehat{\tau}_{jk}\right) & j\ne k \\ 1 & j=k \end{cases}.$$

Let $\widehat{\mathbf{R}}^{\rho} = [\widehat{R}^{\rho}_{jk}]$ and $\widehat{\mathbf{R}}^{\tau} = [\widehat{R}^{\tau}_{jk}]$. Liu et al. (2012) prove the following key result:

Lemma 1 *For any $n \geq \frac{21}{\log d} + 2$, with probability at least $1 - 1/d^2$, we have*

$$||\widehat{\mathbf{R}}^{\rho} - \mathbf{\Sigma}^0||_{\max} \leq 8\pi\sqrt{\frac{\log d}{n}}.$$

For any $n > 1$, with probability at least $1 - 1/d$, we have

$$||\widehat{\mathbf{R}}^{\tau} - \mathbf{\Sigma}^0||_{\max} \leq 2.45\pi\sqrt{\frac{\log d}{n}}.$$

In the following we denote by $\widehat{\mathbf{S}}^{\rho} = [\widehat{S}^{\rho}_{jk}] = [\widehat{\sigma}_j\widehat{\sigma}_k\widehat{R}^{\rho}_{jk}]$ and $\widehat{\mathbf{S}}^{\tau} = [\widehat{S}^{\tau}_{jk}] = [\widehat{\sigma}_j\widehat{\sigma}_k\widehat{R}^{\tau}_{jk}]$, with $\{\widehat{\sigma}^2_j, j = 1, \ldots.d\}$ the sample variances, to be the Spearman's rho and Kendall's tau covariance matrix estimators. As the correlation matrix based on the Spearman's rho and Kendall's tau statistics have similar theoretical performance, in the following sections we omit the superscript ρ and τ and simply denote the estimated correlation and covariance matrices by $\widehat{\mathbf{R}}$ and $\widehat{\mathbf{S}}$.

The following theorem shows that $\widetilde{f}_j$ converges to f_j uniformly over an expanding interval with high probability. This theorem will play a key role in analyzing the classification performance of the CODA method. Here we note that a similar version of this theorem has been shown in Liu et al. (2012), but our result is stronger in terms of extending the region of I_n to be optimal (check the appendix for detailed discussions on it).

Theorem 2 *Let $g_j := f_j^{-1}$ be the inverse function of f_j. In Equation* (5)*, let $\delta_n = \frac{1}{2n}$. For any $0 < \gamma < 1$, we define*

$$I_n := \left[g_j\left(-\sqrt{2(1-\gamma)\log n}\right), g_j\left(\sqrt{2(1-\gamma)\log n}\right)\right],$$

then $\sup_{t \in I_n} |\widetilde{f}_j(t) - f_j(t)| = O_P\left(\sqrt{\frac{\log\log n}{n^{\gamma}}}\right)$.

3. Methods

Let $\boldsymbol{X}_0 \in \mathbb{R}^d$ and $\boldsymbol{X}_1 \in \mathbb{R}^d$ be two random variables with different means $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$ and the same covariance matrix $\mathbf{\Sigma}$. Here we do not pose extra assumptions on the distributions of $\boldsymbol{X}_0$ and $\boldsymbol{X}_1$. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_0}$ be n_0 data points i.i.d drawn from $\boldsymbol{X}_0$, $\boldsymbol{x}_{n_0+1}, \ldots, \boldsymbol{x}_n$ be n_1 data points i.i.d drawn from $\boldsymbol{X}_1$, $n = n_0 + n_1$. Denote by $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, $\boldsymbol{y} = (y_1, ..., y_n)^T = (-n_1/n, \ldots, -n_1/n, n_0/n, \ldots, n_0/n)^T$ with the first n_0 entries equal to $-n_1/n$ and the next n_1 entries equal to n_0/n. We have $n_0 \sim \text{Binomial}(n, \pi_0)$ and $n_1 \sim \text{Binomial}(n, \pi_1)$. In the sequel, without loss of generality, we assume that $\pi_0 = \pi_1 = 1/2$. The extension to the case where $\pi_0 \neq \pi_1$ is straightforward (Hastie et al., 2001).

Define

$$\widehat{\boldsymbol{\mu}}_0 = \frac{1}{n_0}\sum_{i:y_i=-n_1/n} \boldsymbol{x}_i, \quad \widehat{\boldsymbol{\mu}}_1 = \frac{1}{n_1}\sum_{i:y_i=n_0/n} \boldsymbol{x}_i, \quad \widehat{\boldsymbol{\mu}}_d = \widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0, \quad \widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_i \boldsymbol{x}_i,$$

$$\mathbf{S}_0 = \frac{1}{n_0}\sum_{i:y_i=-n_1/n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_0)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_0)^T, \quad \mathbf{S}_1 = \frac{1}{n_1}\sum_{i:y_i=n_0/n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)^T,$$

$$\mathbf{S}_b = \frac{1}{n}\sum_{i=0}^{1} n_i(\widehat{\boldsymbol{\mu}}_i - \widehat{\boldsymbol{\mu}})(\widehat{\boldsymbol{\mu}}_i - \widehat{\boldsymbol{\mu}})^T = \frac{n_0 n_1}{n^2}\widehat{\boldsymbol{\mu}}_d\widehat{\boldsymbol{\mu}}_d^T, \quad \mathbf{S}_w = \frac{n_0\mathbf{S}_0 + n_1\mathbf{S}_1}{n}.$$

3.1 The Connection between ROAD and Lasso

In this subsection, we first show that in low dimensions, LDA can be formulated as a least square problem. Motivated by such a relationship, we further show that in high dimensions, the lasso can be viewed as a special case of the ROAD. Such a connection between the ROAD and lasso will be further exploited to develop the CODA method.

When $d < n$, we define the population and sample versions of the LDA classifiers as:

$$\boldsymbol{\beta}^* = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d, \quad \widehat{\boldsymbol{\beta}}^* = \mathbf{S}_w^{-1}\widehat{\boldsymbol{\mu}}_d.$$

Similarly, a least square estimator has the formulation:

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmin}} ||\boldsymbol{y} - \beta_0 \mathbf{1} - \mathbf{X}\boldsymbol{\beta}||_2^2, \tag{6}$$

where $\mathbf{1} := (1, 1, \ldots, 1)^T$. The following lemma connects the LDA to simple linear regression. The proof is elementary, for self-containess, we include the proof here.

Lemma 3 *Under the above notations, $\widehat{\boldsymbol{\beta}} \propto \widehat{\boldsymbol{\beta}}^*$. Specifically, when $n_1 = n_0$, the linear discriminant classifier $g^*(x)$ is equivalent to the following classifier*

$$\widehat{l}(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad \widehat{\beta}_0 + \boldsymbol{x}^T\widehat{\boldsymbol{\beta}} > 0 \\ 0, & \text{otherwise}. \end{cases}.$$

Proof Taking the first derivatives of the right hand side of (6), we have $\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}$ satisfying:

$$n\beta_0 + (n_0\widehat{\boldsymbol{\mu}}_0 + n_1\widehat{\boldsymbol{\mu}}_1)^T\boldsymbol{\beta} = 0, \tag{7}$$

$$(n_0\widehat{\boldsymbol{\mu}}_0 + n_1\widehat{\boldsymbol{\mu}}_1)\beta_0 + (n\mathbf{S}_w + n_1\widehat{\boldsymbol{\mu}}_1\widehat{\boldsymbol{\mu}}_1^T + n_0\widehat{\boldsymbol{\mu}}_0\widehat{\boldsymbol{\mu}}_0^T)\boldsymbol{\beta} = \frac{n_0 n_1}{n}\widehat{\boldsymbol{\mu}}_d. \tag{8}$$

Combining Equations (7) and (8), we have

$$(\mathbf{S}_w + \mathbf{S}_b)\widehat{\boldsymbol{\beta}} = \frac{n_0 n_1}{n^2}\widehat{\boldsymbol{\mu}}_d.$$

Noticing that $\mathbf{S}_b\widehat{\boldsymbol{\beta}} \propto \widehat{\boldsymbol{\mu}}_d$, it must be true that

$$\mathbf{S}_w\widehat{\boldsymbol{\beta}} = \left(\frac{n_1 n_0}{n^2}\widehat{\boldsymbol{\mu}}_d - \mathbf{S}_b\widehat{\boldsymbol{\beta}}\right) \propto \widehat{\boldsymbol{\mu}}_d.$$

Therefore, $\widehat{\boldsymbol{\beta}} \propto \mathbf{S}_w^{-1}\widehat{\boldsymbol{\mu}}_d = \widehat{\boldsymbol{\beta}}^*$. This completes the proof of the first assertion.
Moreover, noticing that by (7), $\widehat{l}(\boldsymbol{x}) = 1$ is equivalent to

$$\widehat{\beta}_0 + \boldsymbol{x}^T\widehat{\boldsymbol{\beta}} = -\left(\frac{n_0\widehat{\boldsymbol{\mu}}_0 + n_1\widehat{\boldsymbol{\mu}}_1}{n}\right)^T\widehat{\boldsymbol{\beta}} + \boldsymbol{x}^T\widehat{\boldsymbol{\beta}} = (\boldsymbol{x} - \frac{n_1\widehat{\boldsymbol{\mu}}_1 + n_0\widehat{\boldsymbol{\mu}}_0}{n})^T\widehat{\boldsymbol{\beta}} > 0.$$

because $\widehat{\boldsymbol{\beta}} \propto \widehat{\boldsymbol{\beta}}^*$, $n_1 = n_0$ and $\operatorname{sign}(\widehat{\boldsymbol{\beta}}) = \operatorname{sign}(\widehat{\boldsymbol{\beta}}^*)$ (see Lemma 6 for details), we have $g^*(\boldsymbol{x}) = \widehat{l}(\boldsymbol{x})$. This proves the second assertion. ■

In the high dimensional setting, the following lemma shows that the ROAD is connected to the lasso.

Lemma 4 *We define:*

$$\widehat{\boldsymbol{\beta}}^{\lambda_n,\nu}_{ROAD} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{\beta}^T \mathbf{S}_w \boldsymbol{\beta} + \lambda_n ||\boldsymbol{\beta}||_1 + \frac{\nu}{2}(\boldsymbol{\beta}^T \widehat{\boldsymbol{\mu}}_d - 1)^2, \tag{9}$$

$$\widehat{\boldsymbol{\beta}}^{\lambda_n}_{*} = \underset{\boldsymbol{\beta}^T \widehat{\boldsymbol{\mu}}_d = 1}{\operatorname{argmin}} \frac{1}{2n} ||\boldsymbol{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}||_2^2 + \lambda_n ||\boldsymbol{\beta}||_1, \tag{10}$$

$$\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2n} ||\boldsymbol{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}||_2^2 + \lambda_n ||\boldsymbol{\beta}||_1, \tag{11}$$

where $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}_{n\times 1}\widehat{\boldsymbol{\mu}}^T$ *is the globally centered version of* $\mathbf{X}$*. We then have* $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{*} = \widehat{\boldsymbol{\beta}}^{\lambda_n,\infty}_{ROAD}$ *and* $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO} = \widehat{\boldsymbol{\beta}}^{\lambda_n,\nu^*}_{ROAD}$ *where* $\nu^* = \frac{n_1 n_0}{n^2}$.

Proof Noticing that the right hand side of Equation (11) has the form:

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO} &= \underset{\beta}{\operatorname{argmin}} \left(\frac{1}{2n} ||\boldsymbol{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}||_2^2 + \lambda_n ||\boldsymbol{\beta}||_1 \right) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2n}\boldsymbol{y}^T\boldsymbol{y} - \frac{n_1 n_0}{n^2}\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d + \frac{1}{2}\boldsymbol{\beta}^T(\mathbf{S}_w + \mathbf{S}_b)\boldsymbol{\beta} + \lambda_n ||\boldsymbol{\beta}||_1 \right) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\mathbf{S}_w\boldsymbol{\beta} + \frac{n_1 n_0}{2n^2}\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d\widehat{\boldsymbol{\mu}}_d^T\boldsymbol{\beta} - \frac{n_1 n_0}{n^2}\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d + \lambda_n ||\boldsymbol{\beta}||_1 \right) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\mathbf{S}_w\boldsymbol{\beta} + \frac{1}{2}\frac{n_1 n_0}{n^2}(\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d - 1)^2 + \lambda_n ||\boldsymbol{\beta}||_1 \right).
\end{aligned}$$

And similarly

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\lambda_n}_{*} &= \underset{\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d = 1}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\mathbf{S}_w\boldsymbol{\beta} + \frac{1}{2}\frac{n_1 n_0}{n^2}(\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d - 1)^2 + \lambda_n ||\boldsymbol{\beta}||_1 \right) \\
&= \underset{\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d = 1}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\mathbf{S}_w\boldsymbol{\beta} + \lambda_n ||\boldsymbol{\beta}||_1 \right).
\end{aligned}$$

This finishes the proof. ■

Motivated by the above lemma, later we will show that $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO}$ is already variable selection consistent.

3.2 Copula Discriminant Analysis

In this subsection we introduce the Copula Discriminant Analysis (CODA). We assume that

$$\boldsymbol{X}_0 \sim NPN(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}, f), \quad \boldsymbol{X}_1 \sim NPN(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}, f).$$

Here the transformation functions are $f = \{f_j\}_{j=1}^d$. In this setting, the corresponding Bayes rule can be easily calculated as:

$$g^{npn}(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad (f(\boldsymbol{x}) - \boldsymbol{\mu}_a)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

By Equation (12) the Bayes rule is the sign of the log odds: $(f(\boldsymbol{x})-\boldsymbol{\mu}_a)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d$. Therefore, similar to the linear discriminant analysis, if there is a sparsity pattern on $\boldsymbol{\beta}^* := \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d$, a fast rate is expected.

Inspired by Lemma 3 and Lemma 4, to recover the sparsity pattern, we propose the ℓ_1 regularized minimization equation:

$$\widehat{\boldsymbol{\beta}}_{npn} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2}\boldsymbol{\beta}^T\widehat{\mathbf{S}}\boldsymbol{\beta} + \frac{\nu}{2}(\boldsymbol{\beta}^T\widehat{\boldsymbol{\mu}}_d - 1)^2 + \lambda_n||\boldsymbol{\beta}||_1 \right). \tag{13}$$

Here $\widehat{\mathbf{S}} = n_0/n \cdot \widehat{\mathbf{S}}_0 + n_1/n \cdot \widehat{\mathbf{S}}_1$, $\widehat{\mathbf{S}}_0$ and $\widehat{\mathbf{S}}_1$ are the Spearman's rho/Kendall's tau covariance matrix estimators of $[\boldsymbol{x}_1,...,\boldsymbol{x}_{n_0}]^T$ and $[\boldsymbol{x}_{n_0+1},...,\boldsymbol{x}_n]^T$, respectively. When ν is set to be $\frac{n_0 n_1}{n^2}$, $\widehat{\boldsymbol{\beta}}_{npn}$ parallels the ℓ_1 regularization formulation shown in Equation (11); when ν goes to infinity, $\widehat{\boldsymbol{\beta}}_{npn}$ reduces to $\widehat{\boldsymbol{\beta}}^{\lambda_n}_*$ shown in Equation (10).

For any new data point $\boldsymbol{x} = (x_1,\ldots,x_d)^T$, reminding that the transforms f_j preserves the mean of x_j, we assign it to the second class if and only if

$$(\widehat{f}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}})^T\widehat{\boldsymbol{\beta}}_{npn} > 0,$$

where $\widehat{f}(\boldsymbol{x}) = (\widehat{f}_1(x_1),\ldots,\widehat{f}_d(x_d))^T$ with

$$\widehat{f}_j(x_j) = \left(n_0\widehat{f}_{0j}(x_j) + n_1\widehat{f}_{1j}(x_j) \right)/n, \quad \forall\ j \in \{1,\ldots,d\}.$$

Here $\widehat{f}_{0j}$ and $\widehat{f}_{1j}$ are defined to be:

$$\widehat{f}_{0j}(t) := \widehat{\mu}_0 + \widehat{S}_{jj}^{-1/2}\Phi^{-1}\left(\widetilde{F}_j(t;\delta_{n_0},\boldsymbol{x}_1,...,\boldsymbol{x}_{n_0}) \right),$$

and

$$\widehat{f}_{1j}(t) := \widehat{\mu}_1 + \widehat{S}_{jj}^{-1/2}\Phi^{-1}\left(\widetilde{F}_j(t;\delta_{n_1},\boldsymbol{x}_{n_0+1},...,\boldsymbol{x}_n) \right).$$

Here we use the truncation level $\delta_n = \dfrac{1}{2n}$. The corresponding classifier is named $\widehat{g}^{npn}$.

3.3 Algorithms

To solve the Equation (13), when ν is set to be $\frac{n_0 n_1}{n^2}$, Lemma 4 has shown that it can be formulated as a ℓ_1 regularized least square problem and hence popular softwares such as *glmnet* (Friedman et al., 2009, 2010) or *lars* (Efron et al., 2004) can be applied.

When ν goes to infinity, the Equation (13) reduces to the ROAD, which can be efficiently solved by the augmented Lagrangian method (Nocedal and Wright, 2006). More specifically, we define the augmented Lagrangian function:

$$\mathcal{L}(\boldsymbol{\beta},u) = \frac{1}{2}\boldsymbol{\beta}^T\widehat{\mathbf{S}}\boldsymbol{\beta} + \lambda\|\boldsymbol{\beta}\|_1 + \nu u(\widehat{\boldsymbol{\mu}}^T\boldsymbol{\beta} - 1) + \frac{\nu}{2}(\widehat{\boldsymbol{\mu}}^T\boldsymbol{\beta} - 1)^2,$$

where $u \in \mathbb{R}$ is the rescaled Lagrangian multiplier and $\nu > 0$ is the augmented Lagrangian multiplier. We can obtain the optimum to Equation (9) using the following iterative procedure. Suppose at the k-th iteration, we already have the solution $\boldsymbol{\beta}^{(k)}$, $u^{(k)}$, then at the $(k+1)$-th iteration,

• Step.1 Minimize $\mathcal{L}(\boldsymbol{\beta}, u)$ with respect to $\boldsymbol{\beta}$. It can be efficiently solved by coordinate descent. We rearrange

$$\widehat{\mathbf{S}} = \begin{pmatrix} \widehat{S}_{j,j} & \widehat{\mathbf{S}}_{j,-j} \\ \widehat{\mathbf{S}}_{-j,j} & \widehat{\mathbf{S}}_{-j,-j} \end{pmatrix}, \ \widehat{\boldsymbol{\mu}} = (\widehat{\mu}_j, \widehat{\boldsymbol{\mu}}_{-j}^T)^T$$

and $\boldsymbol{\beta} = (\beta_j, \boldsymbol{\beta}_{-j}^T)^T$. Then we can iteratively update β_j by the formula

$$\beta_j^{(k+1)} = \frac{\text{Soft}\left(\nu\widehat{\mu}_j \left(1 - u^{(k)} - \widehat{\boldsymbol{\mu}}_{-j}^T \boldsymbol{\beta}_{-j}^{(k)}\right) - \widehat{\mathbf{S}}_{j,-j}\boldsymbol{\beta}_{-j}^{(k)}, \ \lambda\right)}{\widehat{S}_{j,j} + \nu\widehat{\mu}_j^2},$$

where $\text{Soft}(x, \lambda) := \text{sign}(x)(|x| - \lambda)^+$. It is observed that a better empirical performance can be achieved by updating each β_j only once.

• Step.2 Update u using the formula

$$u^{(k+1)} = u^{(k)} + \widehat{\boldsymbol{\mu}}^T \boldsymbol{\beta}^{(k+1)} - 1.$$

This augmented Lagrangian method has provable global convergence. See Chapter 17 of Nocedal and Wright (2006) for discussions in details. Our empirical simulations show that this algorithm is more accurate than Fan et al. (2010)'s method.

To solve Equation (13), we also need to make sure that $\widehat{\mathbf{S}}$, or equivalently $\widehat{\mathbf{R}}$, is positive semidefinite. Otherwise, Equation (13) is not a convex optimization problem and the above algorithm may not even converge. Heuristically, we can truncate all of the negative eigenvalues of $\widehat{\mathbf{R}}$ to zero. In practice, we project $\widehat{\mathbf{R}}$ into the cone of the positive semidefinite matrices and find solution $\widetilde{\mathbf{R}}$ to the following convex optimization problem:

$$\widetilde{\mathbf{R}} = \underset{\mathbf{R} \succeq 0}{\arg\min} \|\widehat{\mathbf{R}} - \mathbf{R}\|_{\max}, \tag{14}$$

where $\ell_{\max}$ norm is chosen such that the theoretical properties in Lemma 1 can be preserved. In specific, we have the following corollary:

Lemma 5 *For all $t \geq 32\pi\sqrt{\frac{\log d}{n \log 2}}$, the minimizer $\widetilde{\mathbf{R}}$ to Equation* (14) *satisfies the following exponential inequality:*

$$\mathbb{P}(|\widetilde{R}_{jk} - \Sigma_{jk}^0| \geq t) \leq 2\exp\left(-\frac{nt^2}{512\pi^2}\right), \quad \forall\, 1 \leq j,k \leq d.$$

Proof Combining Equation (A.23) and Equation (A.28) of Liu et al. (2012), we have

$$\mathbb{P}(|\widehat{R}_{jk} - \Sigma_{jk}^0| > t) \leq 2\exp\left(-\frac{nt^2}{64\pi^2}\right).$$

Because $\boldsymbol{\Sigma}^0$ is feasible to Equation (14), $\widetilde{\mathbf{R}}$ must satisfy that:

$$||\widehat{\mathbf{R}} - \widetilde{\mathbf{R}}||_{\max} \leq ||\widehat{\mathbf{R}} - \boldsymbol{\Sigma}^0||_{\max}.$$

Using Pythagorean Theorem, we then have

$$\begin{aligned}
\mathbb{P}(|\widetilde{R}_{jk} - \Sigma^0_{jk}| \geq t) &\leq \mathbb{P}(|\widetilde{R}_{jk} - \widehat{R}_{jk}| + |\widehat{R}_{jk} - \Sigma^0_{jk}| \geq t) \\
&\leq \mathbb{P}(||\widetilde{\mathbf{R}} - \widehat{\mathbf{R}}||_{\max} + ||\widehat{\mathbf{R}} - \mathbf{\Sigma}^0||_{\max} \geq t) \\
&\leq \mathbb{P}(||\widehat{\mathbf{R}} - \mathbf{\Sigma}^0||_{\max} \geq t/2) \\
&\leq d^2 \exp\left(-\frac{nt^2}{256\pi^2}\right) \\
&\leq 2\exp\left(\frac{2\log d}{\log 2} - \frac{nt^2}{256\pi^2}\right).
\end{aligned}$$

Using the fact that $t \geq 32\pi\sqrt{\frac{\log d}{n \log 2}}$, we have the result. ■

Therefore, the theoretical properties in Lemma 1 also hold for $\widetilde{\mathbf{R}}$, only with a slight loose on the constant. In practice, it has been found that the optimization problem in Equation (14) can be formulated as the dual of a graphical lasso problem with the smallest possible tuning parameter that still guarantees a feasible solution (Liu et al., 2012). Empirically, we can use a surrogate projection procedure that computes a singular value decomposition of $\widehat{\mathbf{R}}$ and truncates all of the negative singular values to be zero. And then we define $\widetilde{\mathbf{S}} := [\widetilde{S}_{jk}] = [\widehat{\sigma}_j \widehat{\sigma}_k \widetilde{R}_{jk}]$ to be the projected Spearman's rho/Kendall's tau covariance matrices, which can be plugged into Equation (13) to obtain an optimum.

3.4 Computational Cost

Compared to the corresponding parametric methods like the ROAD and the least square formulation proposed by Mai et al. (2012), one extra cost of the CODA is the computation of $\widetilde{\mathbf{R}}$, which can be solved in two steps: (1) computing $\widehat{\mathbf{R}}$; (2) projecting $\widehat{\mathbf{R}}$ to the cone of the positive semidefinite matrices. In the first step, computing $\widehat{\mathbf{R}}$ requires the calculation of $d(d-1)/2$ pairwise Spearman's rho or Kendall's tau statistics. As shown in Christensen (2005) and Kruskal (1958), $\widehat{\mathbf{R}}$ can be computed with the cost $O(d^2 n \log n)$. In the second step, to obtain $\widetilde{\mathbf{R}}$ requires estimating a full path of estimates by implementing the graphical lasso algorithm. This approach shows good scalability to very high dimensional data sets (Friedman et al., 2007; Zhao et al., 2012). Moreover, in practice we can use a surrogate projection procedure, which can be solved by implementing the SVD decomposition of $\widehat{\mathbf{R}}$ once.

4. Theoretical Properties

In this section we provide the theoretical properties of the CODA method. We set $\nu = (n_0 n_1)/n^2$ in Equation (13). With such a choice of ν, we prove that the CODA method is variable selection consistent and has an oracle property. We define

$$\mathbf{C} := \mathbf{\Sigma} + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T/4. \tag{15}$$

To calculate $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO}$ in Equation (11), we define $\widetilde{\mathbf{\Sigma}}$:

$$\widetilde{\mathbf{\Sigma}} := \widetilde{\mathbf{S}} + \frac{n_0 n_1}{n^2} \cdot (\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)^T.$$

We then replace $\frac{1}{n}\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ with $\widetilde{\mathbf{\Sigma}}$ in Equation (11).

It is easy to see that $\widetilde{\mathbf{\Sigma}}$ is a consistent estimator of C. We define $\lfloor c \rfloor$ the greatest integer strictly less than the real number c. For any subset $T \subseteq \{1,2,\ldots,d\}$, let $\widetilde{\mathbf{X}}_T$ be the $n \times |T|$ matrix with the vectors $\{\widetilde{\mathbf{X}}_{\cdot i}, i \in T\}$ as columns. We assume that $\boldsymbol{\beta}^*$ is sparse and define $S := \{i \in 1,...,d | \beta_i^* \neq 0\}$ with $|S| = s$, $s \ll n$. we denote by

$$\boldsymbol{\beta}^{**} := \mathbf{C}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \quad \text{where } \beta^{**}_{\max} := \max_{j \in S}(|\beta_j^{**}|), \text{ and } \beta^{**}_{\min} := \min_{j \in S}(|\beta_j^{**}|).$$

Recalling that $\boldsymbol{\beta}^* = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d$, the next lemma claims that $\boldsymbol{\beta}^{**} \propto \boldsymbol{\beta}^*$ and therefore $\boldsymbol{\beta}^{**}$ is also sparse, and hence $\boldsymbol{\beta}^{**}_S = (\mathbf{C}_{SS})^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S$.

Lemma 6 *Let $\boldsymbol{\beta}^* = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d$. $\boldsymbol{\beta}^{**}$ is proportional to $\boldsymbol{\beta}^*$. Especially, we have*

$$\boldsymbol{\beta}^{**} = \frac{4\boldsymbol{\beta}^*}{4 + \boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d}.$$

Proof Using the Binomial inverse theorem (Strang, 2003), we have

$$\begin{aligned}\boldsymbol{\beta}^{**} &= (\mathbf{\Sigma} + \frac{1}{4}\boldsymbol{\mu}_d\boldsymbol{\mu}_d^T)^{-1}\boldsymbol{\mu}_d = \left(\mathbf{\Sigma}^{-1} - \frac{\frac{1}{4}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d\boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}}{1 + \frac{1}{4}\boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d}\right)\boldsymbol{\mu}_d = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d - \frac{\frac{1}{4}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d(\boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d)}{1 + \frac{1}{4}\boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d} \\ &= \left(1 - \frac{\boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d}{4 + \boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d}\right)\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d = \frac{4\boldsymbol{\beta}^*}{4 + \boldsymbol{\mu}_d^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_d}.\end{aligned}$$

This completes the proof. ■

We want to show that $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO}$ recovers the sparsity pattern of the unknown $\boldsymbol{\beta}^*$ with high probability. In the sequel, we use $\widehat{\boldsymbol{\beta}}$ to denote $\widehat{\boldsymbol{\beta}}^{\lambda_n}_{LASSO}$ for notational simplicity. We define the variable selection consistency property as:

Definition 7 (Variable Selection Consistency Property) *We say that a procedure has the variable selection consistency property $\mathcal{R}(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)$ if and only if there exists a λ_n and an optimal solution $\widehat{\boldsymbol{\beta}}$ such that $\widehat{\boldsymbol{\beta}}_S \neq 0$ and $\widehat{\boldsymbol{\beta}}_{S^c} = 0$.*

Furthermore, to ensure variable selection consistency, the following condition on the covariance matrix is imposed:

Definition 8 *A positive definite matrix C has the Irrepresentable Conditions (IC) property if*

$$||\mathbf{C}_{S^cS}(\mathbf{C}_{SS})^{-1}||_\infty := \psi < 1.$$

This assumption is well known to secure the variable selection consistency of the lasso procedure and we refer to Zou (2006), Meinshausen and Bühlmann (2006), Zhao and Yu (2007) and Wainwright (2009) for more thorough discussions.

The key to prove the variable selection consistency is to show that the marginal sample means and standard deviations converge to the population means and standard deviations in a fast rate for the nonparanormal. To get this result, we need extra conditions on the transformation functions $\{f_j\}_{j=1}^d$. For this, we define the Subgaussian Transformation Function Class.

Definition 9 (Subgaussian Transformation Function Class) *Let $Z \in \mathbb{R}$ be a random variable following the standard Gaussian distribution. The Subgaussian Transformation Function Class* $\mathrm{TF}(K)$ *is defined as the set of functions $g : \mathbb{R} \to \mathbb{R}$ which satisfies:*

$$\mathbb{E}|g(Z)|^m \leq \frac{m!}{2}K^m, \quad \forall\, m \in \mathbb{Z}^+.$$

Remark 10 *Here we note that for any function $g : \mathbb{R} \to \mathbb{R}$, if there exists a constant $L < \infty$ such that*

$$g(z) \leq L \quad \text{or} \quad g'(z) \leq L \quad \text{or} \quad g''(z) \leq L, \ \forall\, z \in \mathbb{R}, \tag{16}$$

then $g \in \mathrm{TF}(K)$ for some constant K. To show that, we have the central absolute moments of the standard Gaussian distribution satisfying, $\forall\, m \in \mathbb{Z}^+$:

$$\begin{aligned} &\mathbb{E}|Z|^m \leq (m-1)!! < m!!, \\ &\mathbb{E}|Z^2|^m = (2m-1)!! < m! \cdot 2^m. \end{aligned} \tag{17}$$

Because g satisfies the condition in Equation (16)*, using Taylor expansion, we have for any $z \in \mathbb{R}$,*

$$g(z) \leq |g(0)| + L \text{ or } |g(z)| \leq |g(0)| + L|z|, \text{ or } |g(z)| \leq |g(0)| + |g'(0)z| + Lz^2. \tag{18}$$

Combining Equations (17) *and* (18)*, we have $\mathbb{E}|g(Z)|^m \leq \frac{m!}{2}K^m$ for some constant K. This proves the assertion.*

The next theorem provides the variable selection consistency result of the proposed procedure. It shows that under certain conditions on the covariance matrix Σ and the transformation functions, the sparsity pattern of β^{**} can be recovered with a parametric rate.

Theorem 11 (Sparsity Recovery) *Let $\boldsymbol{X}_0 \sim NPN(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}, f)$, $\boldsymbol{X}_1 \sim NPN(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}, f)$. We assume that* $\mathbf{C}$ *in Equation* (15) *satisfies the IC condition and $||(\mathbf{C}_{SS})^{-1}||_\infty = D_{\max}$ for some $0 < D_{\max} < \infty$, $||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0||_\infty = \Delta_{\max}$ for some $0 < \Delta_{\max} < \infty$ and $\lambda_{\min}(\mathbf{C}_{SS}) > \delta$ for some constant $\delta > 0$. Then, if we have the additional conditions:*

Condition 1: λ_n is chosen such that

$$\lambda_n < \min\left\{\frac{3\beta^{**}_{\min}}{64 D_{\max}}, \ \frac{3\Delta_{\max}}{32}\right\};$$

Condition 2: Let $\sigma_{\max}$ be a constant such that

$$0 < 1/\sigma_{\max} < \min_j\{\sigma_j\} < \max_j\{\sigma_j\} < \sigma_{\max} < \infty, \max_j |\mu_j| \leq \sigma_{\max},$$

and $g = \{g_j := f_j^{-1}\}_{j=1}^d$ satisfies

$$g_j^2 \in \mathrm{TF}(K), \quad \forall\, j \in \{1, \ldots d\},$$

where $K < \infty$ is a constant,

then there exist positive constants c_0 and c_1 only depending on $\{g_j\}_{j=1}^d$, such that for large enough n

$$\begin{aligned}
&\mathbb{P}(\mathcal{R}(\mathbf{X},\boldsymbol{\beta}^{**},\lambda_n)) \\
&\geq 1-\underbrace{\left[2ds\cdot\exp\left(-\frac{c_0 n\varepsilon^2}{s^2}\right)+2d\cdot\exp\left(-\frac{4c_1 n\lambda_n^2(1-\psi-2\varepsilon D_{\max})^2}{(1+\psi)^2}\right)\right]}_{A} \\
&-\underbrace{\left[2s^2\exp\left(-\frac{c_0 n\varepsilon^2}{s^2}\right)+2s\exp(-c_1 n\varepsilon^2)\right]}_{B}-\underbrace{2s^2\exp(-\frac{c_0 n\delta^2}{4s^2})}_{C}-\underbrace{2\exp\left(-\frac{n}{8}\right)}_{D},
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{P}\left(||\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}-\boldsymbol{\beta}^{**}||_\infty\leq 228D_{\max}\lambda_n\right) \\
&\geq 1-\underbrace{\left[2s^2\exp\left(-\frac{c_0 n\varepsilon^2}{s^2}\right)+2s\exp(-c_1 n\varepsilon^2)\right]}_{B}-\underbrace{2\exp\left(-\frac{n}{8}\right)}_{D},
\end{aligned}\tag{19}$$

whenever ε satisfies that, for large enough n,

$$\begin{aligned}
64\pi\sqrt{\frac{\log d}{n\log 2}}\leq\varepsilon<\min\Bigg\{&1,\ \frac{1-\psi}{2D_{\max}},\ \frac{2\lambda_n(1-\psi)}{D_{\max}(4\lambda_n+(1+\psi)\Delta_{\max})}, \\
&\frac{\omega}{(3+\omega)D_{\max}},\ \frac{\Delta_{\max}\omega}{6+2\omega},\ \frac{4\lambda_n}{D_{\max}\Delta_{\max}},8\lambda_n^2\Bigg\}.
\end{aligned}$$

Here $\omega:=\frac{\beta_{\min}^{**}}{\Delta_{\max}D_{\max}}$ and $\delta\geq 128\pi s\sqrt{\frac{\log d}{n\log 2}}$.

Remark 12 *The above Condition 2 requires the transformation functions' inverse $\{g_j\}_{j=1}^d$ to be restricted such that the estimated marginal means and standard deviations converge to their population quantities exponentially fast. The exponential term A is set to control $\mathbb{P}(\widehat{\boldsymbol{\beta}}_{S^c}\neq 0)$, B is set to control $\mathbb{P}(\widehat{\boldsymbol{\beta}}_S=0)$, C is set to control $\mathbb{P}(\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_{SS})\leq 0)$ and D is set to control $\mathbb{P}(\frac{3}{16}\leq\frac{n_0 n_1}{n^2}\leq\frac{1}{4})$. Here we note that the key of the proof is to show that: (i) there exist fast rates for sample means and standard deviations converging to the population means and standard deviations for the nonparanormal; (ii) $\widetilde{\boldsymbol{\Sigma}}_{SS}$ is invertible with high probability. $\varepsilon\geq 64\pi\sqrt{\frac{\log d}{n\log 2}}$ is used to make sure that the Lemma 5 can be applied here.*

The next corollary provides an asymptotic result of the Theorem 11.

Corollary 13 *Under the same conditions as in Theorem 11, if we further have the following Conditions 3,4 and 5 hold:*

Condition 3: $D_{\max}$, $\Delta_{\max}$, ψ and δ are constants that do not scale with (n,d,s);

Condition 4: The triplet (n,d,s) admits the scaling such that

$$s\sqrt{\frac{\log d+\log s}{n}}\to 0 \quad \text{and} \quad \frac{s}{\beta^{**}_{\min}}\sqrt{\frac{\log d+\log s}{n}}\to 0;$$

Condition 5: λ_n scales with (n,d,s) such that

$$\frac{\lambda_n}{\beta^{**}_{\min}}\to 0 \quad \text{and} \quad \frac{s}{\lambda_n}\sqrt{\frac{\log d+\log s}{n}}\to 0,$$

then

$$\mathbb{P}(\mathcal{R}(\mathbf{X},\boldsymbol{\beta}^{**},\lambda_n))\to 1.$$

Remark 14 *Condition 3 is assumed to be true, in order to give an explicit relationship among (n,d,s) and λ_n. Condition 4 allows the dimension d to grow in an exponential rate of n, which is faster than any polynomial of n. Condition 5 requires that λ_n shrinks towards zero in a slower rate than $s\sqrt{\frac{\log d+\log s}{n}}$.*

In the next theorem, we analyze the classification oracle property of the CODA method. Suppose that there is an oracle, which classifies a new data point $\boldsymbol{x}$ to the second class if and only if

$$(f(\boldsymbol{x})-\boldsymbol{\mu}_a)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d>0. \tag{20}$$

In contrast, $\widehat{g}^{npn}$ will classify $\boldsymbol{x}$ to the second class if

$$(\widehat{f}(\boldsymbol{x})-\widehat{\boldsymbol{\mu}})^T\widehat{\boldsymbol{\beta}} \quad > \quad 0,$$

or equivalently,

$$(\widehat{f}(\boldsymbol{x})-\widehat{\boldsymbol{\mu}})^T\cdot\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1} \quad > \quad 0. \tag{21}$$

We try to quantify the difference between the "oracle" in Equation (20) and the empirical classifier in Equation (21). For this, we define the empirical and population classifiers as

$$\begin{aligned} G(\boldsymbol{x},\boldsymbol{\beta},f) &:= (f(\boldsymbol{x})-\boldsymbol{\mu}_a)^T\boldsymbol{\beta},\\ \widehat{G}(\boldsymbol{x},\boldsymbol{\beta},f) &:= (f(\boldsymbol{x})-\widehat{\boldsymbol{\mu}})^T\boldsymbol{\beta}. \end{aligned}$$

With the above definitions, we have the following theorem:

Theorem 15 *When the conditions in Theorem 11 hold such that $A+B+C\to 0$, furthermore b is a positive constant chosen to satisfy*

$$sn^{-c_2\cdot b}\to 0, \quad \text{where } c_2 \text{ is a constant depending only on the choice of } \{g_j\}_{j=1}^d,$$

then we have

$$\left|\widehat{G}\left(\boldsymbol{x},\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1},\widehat{f}\right)-G(\boldsymbol{x},\boldsymbol{\beta}^{**},f)\right|=O_P\left(s\beta^{**}_{\max}\sqrt{\frac{\log\log n}{n^{1-b/2}}}+sD_{\max}\lambda_n\left(\sqrt{\log n}+\Delta_{\max}\right)\right).$$

Remark 16 *Using* $\frac{n^2}{n_0 n_1}\widehat{\boldsymbol{\beta}}$ *instead of* $\widehat{\boldsymbol{\beta}}$ *is for the purpose of using the Equation* (19) *in Theorem 11.*

When the conditions in Corollary 13 hold, the rate in Theorem 15 can be written more explicitly:

Corollary 17 *When* $\beta_{\max}^{**}$ *is a positive constant which does not scale with* (n,d,s)*,*

$$\frac{\log s}{c_2 \log n} < b < \frac{4\log s}{\log n},$$

and the conditions in Corollary 13 hold, we have

$$\left| \widehat{G}\left(\boldsymbol{x}, \frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}, \widehat{f}\right) - G(\boldsymbol{x}, \boldsymbol{\beta}^{**}, f) \right| = O_P\left(s^2 \log n \cdot \sqrt{\frac{\log d + \log s}{n}} \right),$$

by choosing $\lambda_n \asymp s\sqrt{\dfrac{\log n(\log d + \log s)}{n}}$.

Remark 18 *Here we note that the conditions require that* $c_2 > 1/4$*, in order to give an explicit rate of the classifier estimation without including b. Theorem 15 or Corollary 17 can directly lead to the result on misclassification consistency. The key proof proceeds by showing that* $f(\boldsymbol{X})$ *satisfies a version of the "low noise condition" as proposed by Tsybakov (2004).*

Corollary 19 (Misclassification Consistency) *Let*

$$\mathcal{C}(g^*) := \mathbb{P}(Y \cdot \operatorname{sign}(G(\boldsymbol{X}, \boldsymbol{\beta}^{**}, f)) < 0) \quad \text{and} \quad \mathcal{C}(\widehat{g}) := \mathbb{P}(Y \cdot \operatorname{sign}(\widehat{G}(\boldsymbol{X}, \widehat{\boldsymbol{\beta}}, \widehat{f})) < 0 \,|\, \widehat{\boldsymbol{\beta}}, \widehat{f}),$$

be the misclassification errors of the Bayes classifier and the CODA classifier. Then if the conditions in Theorem 15 hold, and we have the addition assumption that

$$\log n \cdot \left(s\beta_{\max}^{**} \sqrt{\frac{\log\log n}{n^{1-b/2}}} + sD_{\max}\lambda_n \left(\sqrt{\log n} + \Delta_{\max} \right) \right) \to 0;$$

or if the conditions in Corollary 17 hold, and we have the additional assumption that

$$s^2 \log^2 n \cdot \sqrt{\frac{\log d + \log s}{n}} \to 0,$$

then we have

$$\mathbb{E}(\mathcal{C}(\widehat{g})) = \mathcal{C}(g^*) + o(1).$$

5. Experiments

In this section we investigate the empirical performance of the CODA method. We compare the following five methods:

- LS-LDA: the least square formulation for classification proposed by Mai et al. (2012);
- CODA-LS: the CODA using a similar optimization formulation as the LS-LDA;
- ROAD: the Regularized Optimal Affine Discriminant method (Fan et al., 2010);

- CODA-ROAD: the CODA using a similar optimization formulation as the LS-LDA
- SLR: the sparse logistic regression (Friedman et al., 2010).

We note that the main difference among the top four methods is that the covariance matrix $\mathbf{\Sigma}$ is estimated in different ways: the ROAD and LS-LDA both assume that data are Gaussian and use the sample covariance, which introduces estimation bias for non-Gaussian data and the resulting covariance matrix can be inconsistent to $\mathbf{\Sigma}$; in contrast, the CODA method exploits the Spearman's rho and Kendall's tau covariance matrices to estimate Σ. It enjoys a $O\left(\sqrt{\frac{\log d}{n}}\right)$ convergence rate in terms of ℓ_∞ norm. In the following, the Spearman's rho estimator is applied. The Kendall's tau estimator achieves very similar performance.

The LS-LDA, CODA-LS and SLR are implemented using the R package *glmnet* (Friedman et al., 2009). We use the augmented Lagrangian multiplier algorithm to solve ROAD and CODA-ROAD. Here in computing ROAD and CODA-ROAD, ν is set to be 10.

5.1 Synthetic Data

In the simulation studies, we randomly generate $n+1000$ class labels such that $\pi_1 = \pi_2 = 0.5$. Conditioning on the class labels, we generate d dimensional predictors $\boldsymbol{x}$ from nonparanormal distribution $NPN(\boldsymbol{\mu}_0, \mathbf{\Sigma}, f)$ and $NPN(\boldsymbol{\mu}_1, \mathbf{\Sigma}, f)$. Without loss of generality, we suppose $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\beta}^{\text{Bayes}} := \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1$ with $s := ||\boldsymbol{\beta}^{\text{Bayes}}||_0$. The data are then split to two parts: the first n data points as the training set and the next 1000 data points as the testing set. We consider twelve different simulation models. The choices of $n, d, s, \mathbf{\Sigma}, \boldsymbol{\beta}^{\text{Bayes}}$ are shown in Table 1. Here the first two schemes are sparse discriminant models with difference $\mathbf{\Sigma}$ and $\boldsymbol{\mu}_1$; Model 3 is practically sparse in the sense that its Bayes rule depends on all variables in theory but can be well approximated by sparse discriminant functions.

scheme	n	d	s	$\mathbf{\Sigma}$	$\boldsymbol{\beta}^{\text{Bayes}}$
Scheme 1	100	400	20	$\Sigma_{ij} = 0.5^{\lvert i-j\rvert}$	$0.342(1,\ldots,1,0,\ldots,0)^T$
Scheme 2	400	800	20	$\Sigma_{jj} = 1, \Sigma_{ij} = 0.5, i \neq j$	$0.176(1,\ldots,1,0,\ldots,0)^T$
Scheme 3	100	200	20	$\Sigma_{ij} = 0.6^{\lvert i-j\rvert}$	$0.198(1,\ldots,1,0.001,\ldots,0.001)^T$

Table 1: Simulation Models with different $n, d, s, \mathbf{\Sigma}$ and $\boldsymbol{\beta}^{\text{Bayes}}$ listed below.

Furthermore, we explore the effects of different transformation functions f by considering the following four types of the transformation functions:

- **Linear transformation:** $f_{linear} = (f^0, f^0, \ldots)$, where f^0 is the linear function;
- **Gaussian CDF transformation:** $f_{CDF} = (f^1, f^1, \ldots)$, where f^1 is the marginal Gaussian CDF transformation function as defined in Liu et al. (2009);
- **Power transformation:** $f_{power} = (f^2, f^2, \ldots)$, where f^2 is the marginal power transformation function as defined in Liu et al. (2009) with parameter 3;

- **Complex transformation:** $f_{complex} = (\underbrace{f^1, f^1, \ldots, f^1}_{s}, f^2, f^2, \ldots)$, where the first s variables are transformed through f^1, and the rest are transformed through f^2.

Then we obtain twelve models based on all the combinations of the three schemes of $n, d, s, \boldsymbol{\Sigma}, \boldsymbol{\beta}^{\text{Bayes}}$ and four transformation functions (linear, Gaussian CDF, power and complex). We note that the linear transformation f_{linear} is equivalent to no transformation. The Gaussian CDF transformation function is bounded and therefore preserves the theoretical properties of the CODA method. The power transformation function, on the other hand, is unbounded. We exploit f_{CDF}, f_{power} and $f_{complex}$ to separately illustrate how the CODA works when the assumptions in Section 4 hold, when these assumptions are mildly violated and when they are only violated for the variables X_j's with $(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0))_j = 0$.

Figures 1 to 3 summarize the simulation results based on 3,000 replications for twelve models discussed above. Here the means of misclassification errors in percentage are plotted against the numbers of extracted features to illustrate the performance of different methods across the whole regularization paths.

To further show quantitative comparisons among different methods, we use two penalty parameter selection criteria. First, we use an oracle penalty parameter selection criterion. Let $S := \text{support}(\boldsymbol{\beta}^{\text{Bayes}})$ be the set that contains the s discriminant features. Let $\widehat{S}_\lambda$ be the set of nonzero values in the estimated parameters using the regularization parameter λ in different methods. In this way, the number of false positives at λ is defined as $\text{FP}(\lambda) :=$ the number of features in $\widehat{S}_\lambda$ but not in S. The number of false negatives at λ is defined as $\text{FN}(\lambda) :=$ the number of features in S but not in $\widehat{S}_\lambda$. We further define the false positive rate (FPR) and false negative rate (FNR) as

$$\text{FPR}(\lambda) := \text{FP}(\lambda)/(d-s), \text{ and } \text{FNR}(\lambda) := \text{FN}(\lambda)/s.$$

Let Λ be the set of all regularization parameters used to create the full path. The oracle regularization parameter λ^* is defined as

$$\lambda^* := \underset{\lambda \in \Lambda}{\text{argmin}} \{\text{FPR}(\lambda) + \text{FNR}(\lambda)\}.$$

Using the oracle regularization parameter λ^*, the numerical comparisons of the five methods on the twelve models are presented in Table 2. Here Bayes is the Bayes risk and in each row the winning method is in bold. These results manifest how the five methods perform when data are either Gaussian or non-Gaussian.

Second, in practice, we propose a cross validation based approach in penalty parameter selection. In detail, for the training set, we randomly separate the data into ten folds with no overlap between each two parts. Each part has the same case and control data points. Each time we apply the above five methods to the combination of any nine folds, using a given set of regularization parameters. The parameters learned are then applied to predict the labels of the left one fold. We select the penalty parameter λ^*_{CV} to be the one that minimizes the averaged misclassification error. λ^*_{CV} is then applied to the test set. The numerical results are presented in Table 3. Here Bayes is the Bayes risk and in each row the winning method is in bold.

In the following we provide detailed analysis based on these numeric simulations.

5.1.1 NON-GAUSSIAN DATA

From Tables 2 and 3 and Figures 1 to 3, we observe that for different transformation functions f and different schemes of $\{n, d, s, \boldsymbol{\Sigma}, \boldsymbol{\beta}^{\text{Bayes}}\}$, CODA-ROAD and CODA-LS both significantly outperform

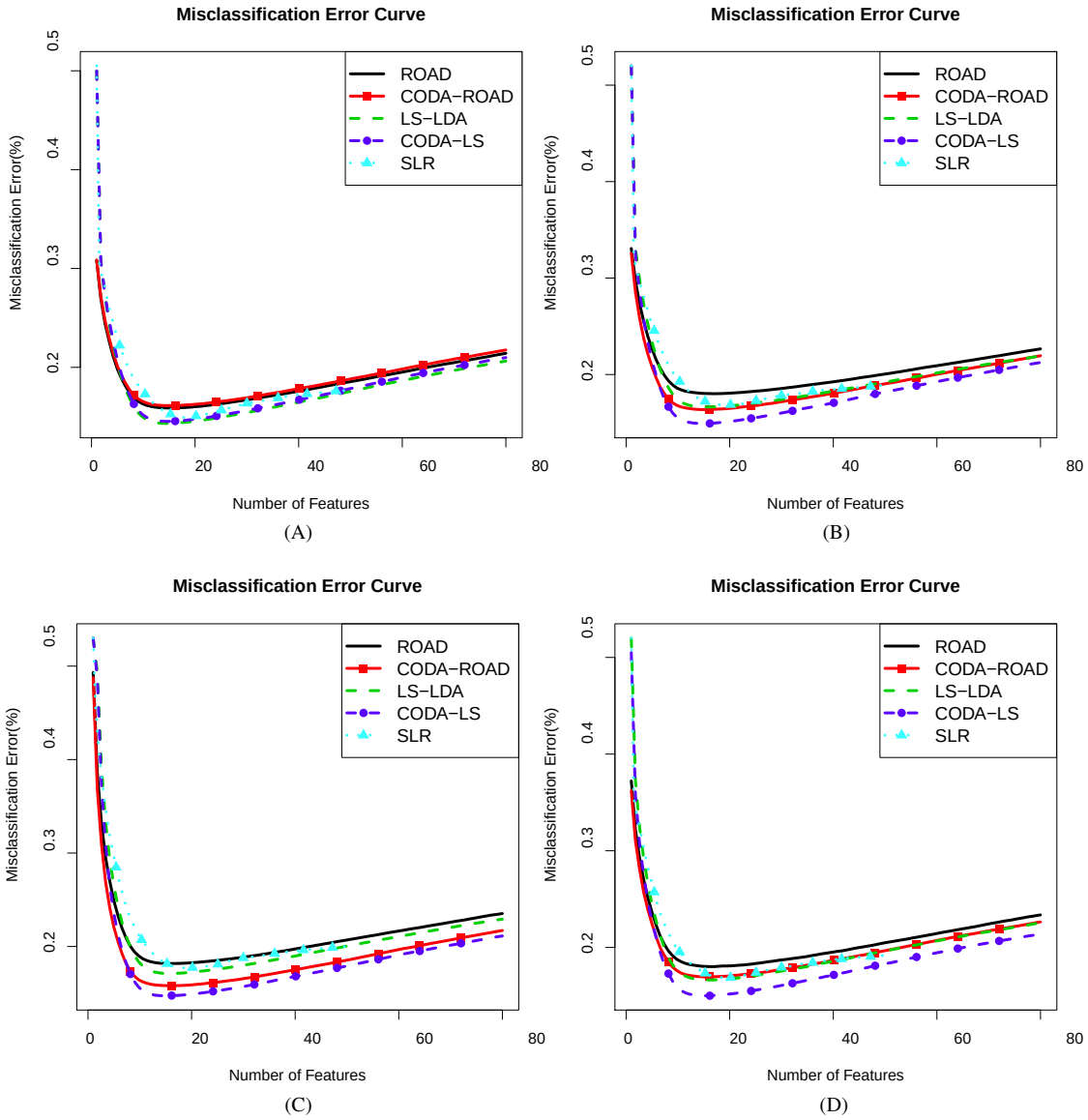


Figure 1: Misclassification error curves on Scheme 1 with four different transformation functions. (A) transformation function is f_{linear}; (B) transformation function is f_{CDF}; (C) transformation function is f_{power}; (D) transformation function is $f_{complex}$. The x-axis represents the numbers of features extracted by different methods; the y-axis represents the averaged misclassification errors in percentage of the methods on the testing data set based on 3,000 replications.

ROAD and LS-LDA, respectively. Secondly, for different transformation functions f, the differences between the two CODA methods (CODA-ROAD and CODA-LS) and their corresponding parametric methods (ROAD and LS-LDA) are comparable. This suggests that the CODA methods can beat the corresponding parametric methods when the sub-Gaussian assumptions for transformation functions

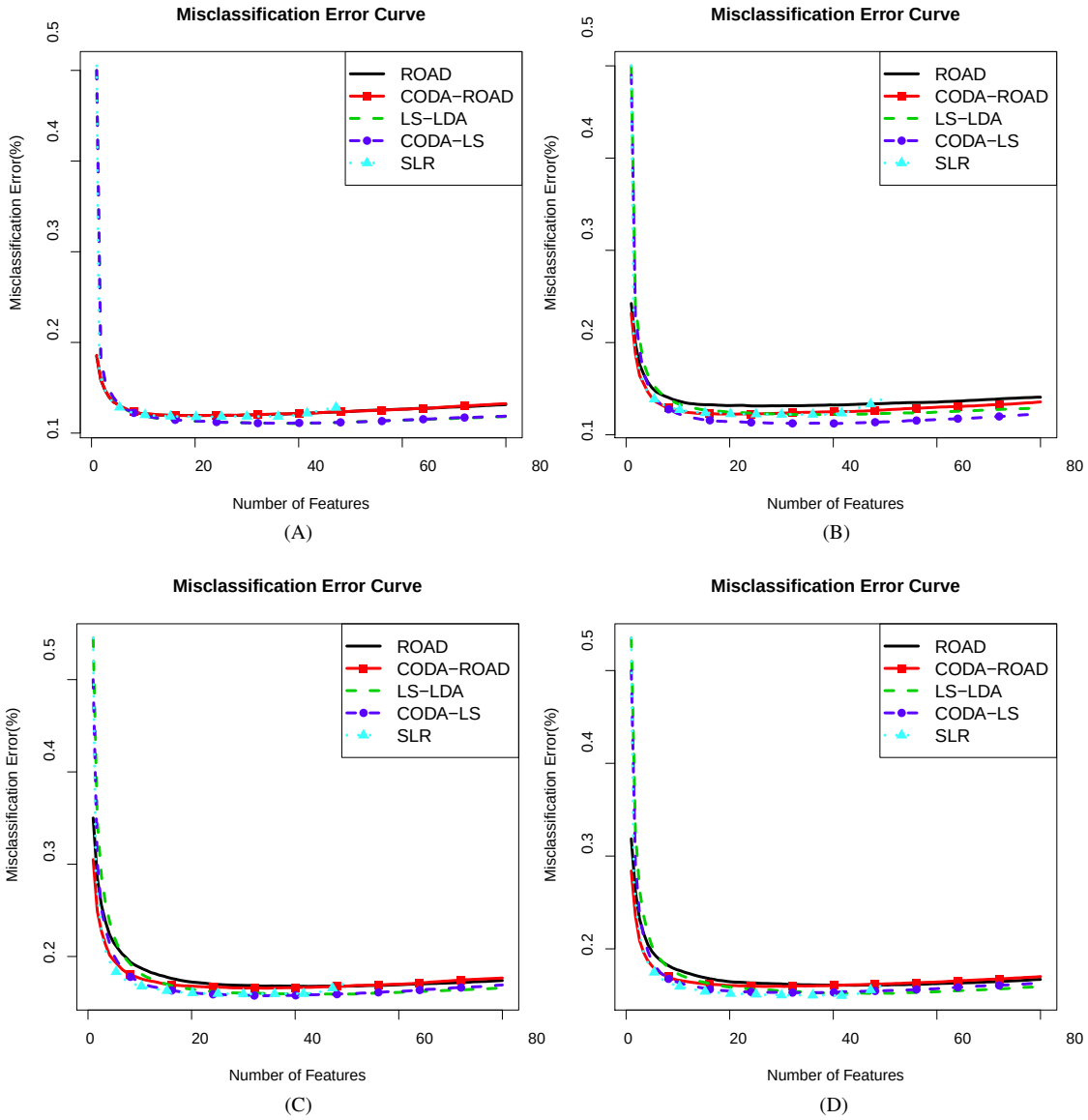


Figure 2: Misclassification error curves on Scheme 2 with four different transformation functions. (A) transformation function is f_{linear}; (B) transformation function is f_{CDF}; (C) transformation function is f_{power}; (D) transformation function is $f_{complex}$. The x-axis represents the numbers of features extracted by different methods; the y-axis represents the averaged misclassification errors in percentage of the methods on the testing data set based on 3,000 replications.

shown in Section 4 are mildly violated. Thirdly, the CODA methods CODA-LS and CODA-ROAD both outperform SLR frequently.

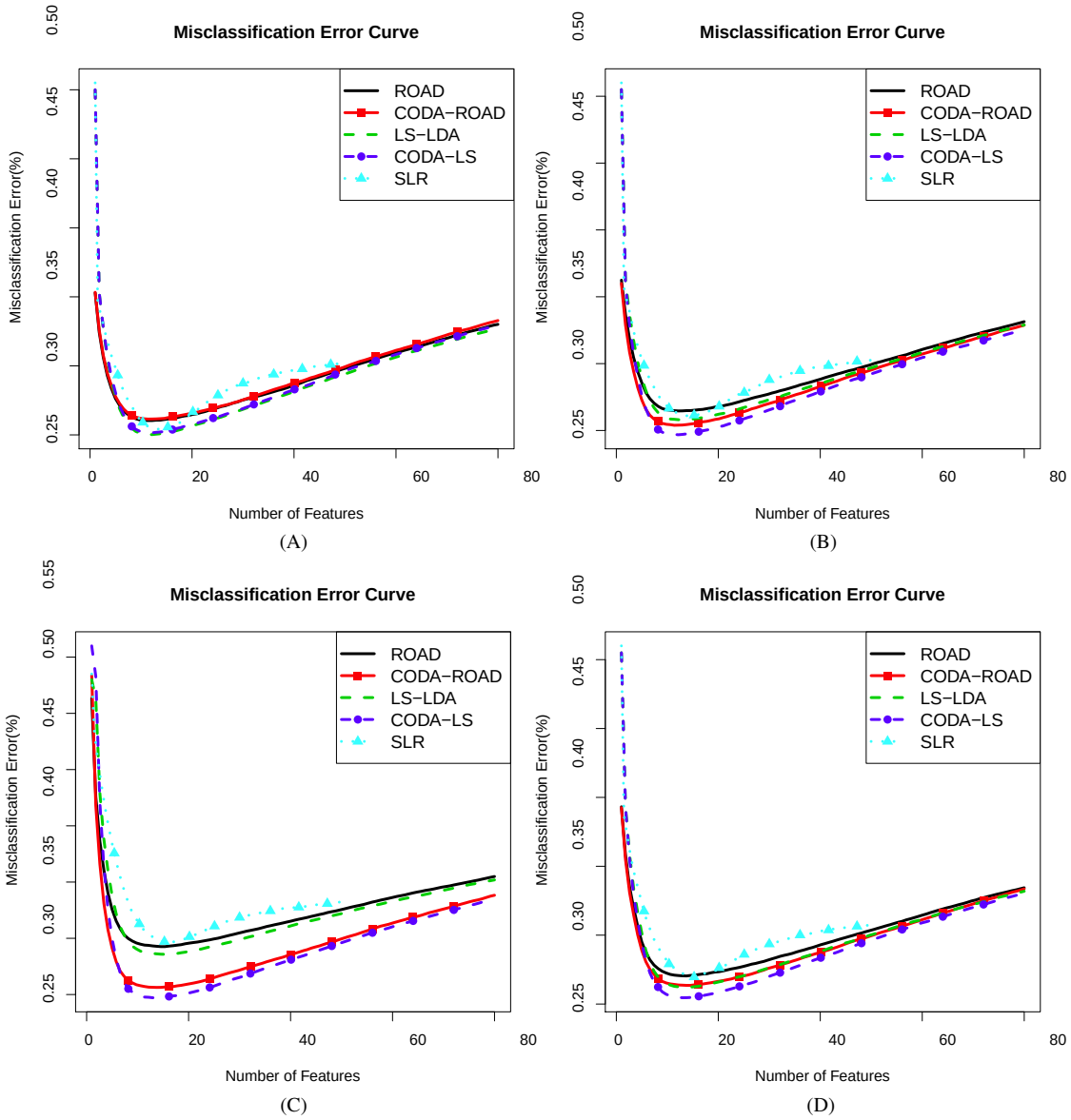


Figure 3: Misclassification error curves on Scheme 3 with four different transformation functions. (A) transformation function is f_{linear}; (B) transformation function is f_{CDF}; (C) transformation function is f_{power}; (D) transformation function is $f_{complex}$. The x-axis represents the numbers of features extracted by different methods; the y-axis represents the averaged misclassification errors in percentage of the methods on the testing data set based on 3,000 replications.

5.1.2 GAUSSIAN DATA

From Tables 2 and 3 and Figures 1 to 3, we observe that when the transformation function is f_{linear}, there is no significant differences between CODA-ROAD and ROAD, and between CODA-LS and LS-

Scheme	f	Bayes(%)	ROAD	CODA-ROAD	LS-LDA	CODA-LS	SLR
Scheme 1	f_{linear}	10.00	16.64(0.81)	16.90(0.84)	**15.09(0.52)**	15.29(0.53)	15.40(0.49)
	f_{CDF}	10.00	18.57(0.80)	17.17(0.84)	17.26(0.52)	**15.66(0.53)**	17.10(0.46)
	f_{power}	10.00	18.80(0.81)	16.51(0.86)	17.76(0.52)	**15.45(0.56)**	17.99(0.52)
	$f_{complex}$	10.00	18.68(0.84)	17.12(0.87)	17.40(0.54)	**15.78(0.53)**	17.26(0.57)
Scheme 2	f_{linear}	10.00	12.28(0.41)	12.34(0.38)	**11.46(0.28)**	11.48(0.28)	12.19(0.31)
	f_{cdf}	10.00	13.56(0.65)	12.83(0.72)	12.95(1.00)	**11.99(0.99)**	12.84(0.30)
	f_{power}	10.00	17.85(0.86)	17.38(0.65)	17.10(0.73)	16.65(0.50)	**16.50(0.33)**
	$f_{complex}$	10.00	16.89(1.39)	16.89(0.43)	17.20(2.43)	**16.77(0.33)**	16.91(0.47)
Scheme 3	f_{linear}	20.00	26.65(0.78)	26.69(0.77)	**25.59(0.63)**	25.70(0.64)	25.97(0.58)
	f_{cdf}	20.00	26.97(0.70)	26.03(0.78)	26.16(0.58)	**25.18(0.64)**	26.41(0.58)
	f_{power}	20.00	29.78(0.72)	26.07(0.87)	29.03(0.60)	**25.14(0.70)**	29.34(0.61)
	$f_{complex}$	20.00	27.54(0.71)	26.78(0.73)	26.70(0.62)	**25.87(0.59)**	26.87(0.57)

Table 2: Quantitative comparisons on different models with linear, Gaussian CDF, power, and complex transformations using the oracle penalty parameter selection criterion. The methods compared here are ROAD,CODA-ROAD,LS-LDA,CODA-LS and SLR. Here Bayes is the Bayes risk and the winning methods are in bold. The means of misclassification errors in percentage with their standard deviations in parentheses are presented. The results are based on 3,000 replications.

LDA. This suggests that the CODA methods can be an alternative choice besides the Gaussian-based high dimensional classification methods.

In summary, we observe that the CODA methods (CODA-LS in particular) have very good overall performance. The simulation results suggest that they can be an alternative choices besides their corresponding parametric methods. And the results also show that in our experiments the CODA methods can outperform their corresponding parametric methods when the sub-Gaussian assumptions for transformation functions are mildly violated.

5.2 Large-scale Genomic Data

In this section we investigate the performance of the CODA methods compared with the others using one of the largest microarray data sets (McCall et al., 2010). In summary, we collect in all 13,182 publicly available microarray samples from Affymetrixs HGU133a platform. The raw data contain 20,248 probes and 13,182 samples belonging to 2,711 tissue types (e.g., lung cancers, prostate cancer, brain tumor etc.). There are at most 1599 samples and at least 1 sample belonging to each tissue type. We merge the probes corresponding to the same gene. There are remaining 12,713 genes and 13,182 samples. The main purpose of this experiment is to compare the performance of different methods in classifying tissues.

We adopt the same idea of data preprocessing as in Liu et al. (2012). In particular, we remove the batch effect by applying the surrogate variable analysis proposed by Leek and Storey (2007). There are, accordingly, 12,713 genes left and the data matrix we are focusing is $12,713 \times 13,182$.

We then explore several tissue types with the largest sample size:

- Breast tumor, which has 1599 samples;

Scheme	f	Bayes(%)	ROAD	CODA-ROAD	LS-LDA	CODA-LS	SLR
Scheme 1	f_{linear}	10.00	16.86(0.77)	16.99(0.94)	**15.31(0.54)**	15.41(0.49)	15.44(0.51)
	f_{CDF}	10.00	18.86(0.83)	17.19(0.79)	17.39(0.68)	**16.16(0.63)**	17.42(0.64)
	f_{power}	10.00	19.13(0.91)	16.84(0.90)	17.91(0.61)	**15.92(0.66)**	18.13(0.62)
	$f_{complex}$	10.00	18.81(0.93)	17.73(0.89)	17.42(0.63)	**15.89(0.62)**	17.94(0.66)
Scheme 2	f_{linear}	10.00	12.58(0.52)	12.59(0.47)	**11.59(0.33)**	11.70(0.38)	12.19(0.29)
	f_{cdf}	10.00	13.97(0.74)	12.86(0.76)	13.36(1.05)	**12.08(1.03)**	13.03(0.32)
	f_{power}	10.00	18.23(0.76)	17.48(0.73)	17.11(0.77)	**16.85(0.59)**	16.86(0.37)
	$f_{complex}$	10.00	16.96(1.59)	16.74(0.61)	17.47(1.99)	**16.80(0.49)**	17.06(0.55)
Scheme 3	f_{linear}	20.00	26.83(0.88)	27.23(0.77)	**25.62(0.64)**	25.74(0.71)	26.21(0.63)
	f_{cdf}	20.00	27.13(0.81)	26.21(0.85)	26.76(0.64)	**25.23(0.61)**	26.43(0.69)
	f_{power}	20.00	30.17(0.85)	26.79(1.00)	29.03(0.73)	**25.15(0.78)**	29.85(0.63)
	$f_{complex}$	20.00	28.43(0.91)	26.82(0.77)	26.74(0.60)	**25.88(0.68)**	27.27(0.71)

Table 3: Quantitative comparisons on different models with linear, Gaussian CDF, power, and complex transformations using the cross validation based penalty parameter selection criterion. The methods compared here are ROAD,CODA-ROAD,LS-LDA,CODA-LS and SLR. Here Bayes is the Bayes risk and the winning methods are in bold. The means of misclassification errors in percentage with their standard deviations in parentheses are presented. The results are based on 3,000 replications.

- B cell lymphoma, which has 213 samples;
- Prostate tumor, which has 148 samples;
- Wilms tumor, which has 143 samples.

Different tissues have been believed to be associated with different sets of genes and microarray data have been heavily used to classify tissue types. See for example, Hans et al. (2004), Wang et al. (2008) and Huang and Chang (2007), among others. For each tissue type listed above, our target is to classify it from all the other tissue types. To this end, each time we randomly split the whole data to three parts: (i) the training set with 200 samples (equal size of case and control); (ii) the testing set with 1000 samples; (iii) the rest. We then run ROAD,CODA-ROAD,LS-LDA,CODA-LS on the training set and applying the learned parameters on the testing set. We repeat this for 1,000 times. The averaged misclassification errors in percentage versus the numbers of extracted features are illustrated in Figure 4. Quantitative results, with penalty parameter selected using the cross validation criterion, are presented in Table 4.

It can be observed that CODA-ROAD and CODA-LS have the best overall performance. Some biological discoveries have also been verified in this process. For example, the MYC gene has been discovered to be relevant to the b cell lymphoma (Lovec et al., 1994; Smith and Wickstrom, 1998) and has recently been found to be associated with the Wilms tumor (Ji et al., 2011). This gene is also constantly selected by the CODA methods in classifying b cell lymphoma and Wilms tumor with the rest.

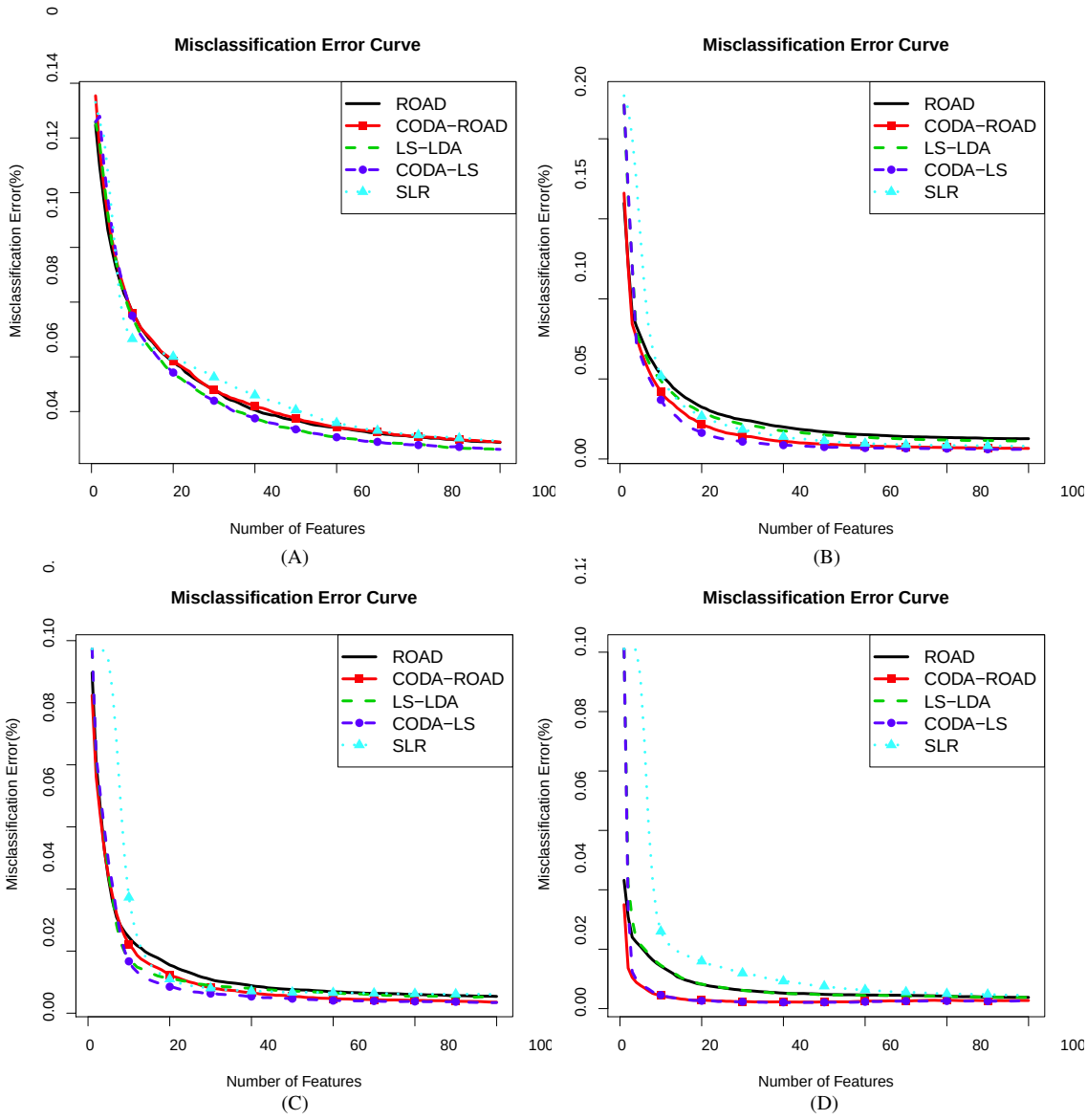


Figure 4: Misclassification error curves on the GPL96 data set. (A) Breast tumor; (B) B cell lymphoma; (C) Prostate tumor; (D) Wilms tumor. The x-axis represents the numbers of features extracted by different methods; the y-axis represents the averaged misclassification errors in percentage of the methods on the testing data set based on 1,000 replications.

5.3 Brain Imaging Data

In this section we investigate the performance of several methods on a brain imaging data set, the ADHD 200 data set (Eloyan et al., 2012). The ADHD 200 data set is a landmark study compiling over 1,000 functional and structural scans including subjects with and without attention deficit hyperactive disorder (ADHD). The current releases data are from 776 subjects: 491 controls and 285 children diagnosed with ADHD. Each has structural blood oxygen level dependent (BOLD)

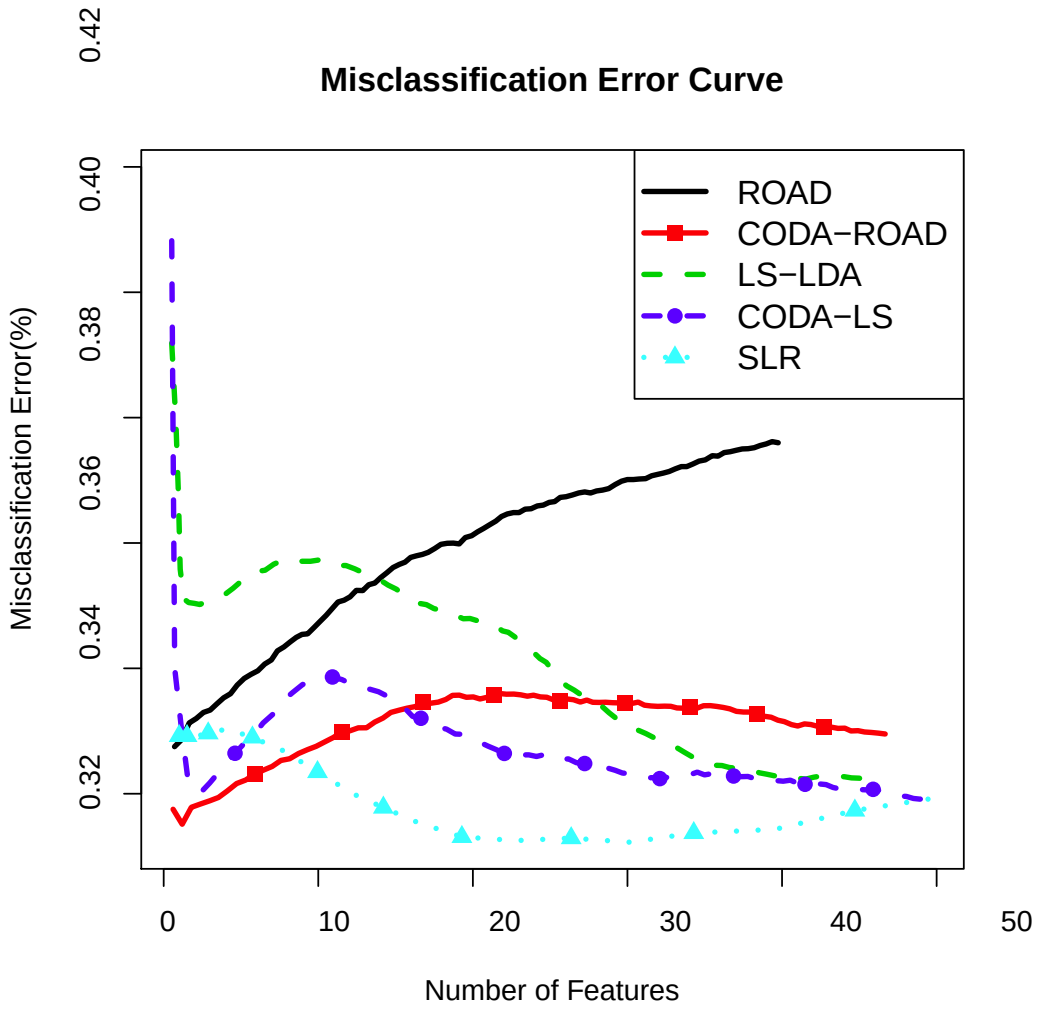


Figure 5: Misclassification error curves on the ADHD data set. The x-axis represents the numbers of features extracted by different methods; the y-axis represents the averaged misclassification errors in percentage of the methods on the testing data set based on 1,000 replications.

functional MRI scans. The data also include demographic variables as predictors. These include age, IQ, gender and handedness. We refer to Eloyan et al. (2012) for detailed data preprocessing procedures.

We construct our predictors by extracting voxels that broadly cover major functional regions of the cerebral cortex and cerebellum following Power et al. (2011). We also combine the information of the demographic variables, resulting to the final data matrix we will use with the dimension 268×776. The target is to differentiate the subjects with ADHD from those without ADHD.

To evaluate the performance of different methods, each time we randomly sample 155 data points unrepeatedly from the whole data. We then gather them together as the training set. The

Data	ROAD(%)	CODA-ROAD	LS-LDA	CODA-LS	SLR
Genomic (A)	0.29(0.17)	0.29(0.17)	0.26(0.16)	**0.25(0.15)**	0.29(0.18)
Genomic (B)	1.31(0.26)	0.69(0.15)	1.16(0.20)	**0.63(0.13)**	0.82(0.18)
Genomic (C)	0.56(0.13)	0.39(0.11)	0.55(0.15)	**0.37(0.12)**	0.62(0.17)
Genomic (D)	0.38(0.16)	0.23(0.08)	0.38(0.09)	**0.22(0.10)**	0.48(0.12)
ADHD	33.20(0.26)	31.89(0.27)	32.66(0.24)	32.25(0.24)	**31.73(0.21)**

Table 4: Quantitative comparisons on genomic and brain imaging data using the cross validation based penalty parameter selection criterion. The methods compared here are ROAD,CODA-ROAD,LS-LDA,CODA-LS and SLR. Here the winning methods are in bold. The means of misclassification errors in percentage with their standard deviations in parentheses are presented. Here "Genomic (A)" to "Genomic (D)" denote the breast tumor, b cell lymphoma, prostate tumor and Wilms tumor, 'ADHD' denotes the results in brain imaging data analysis.

rest are left as the testing set. We then run ROAD,CODA-ROAD,LS-LDA,CODA-LS on the training set and applying the learned parameters on the testing set. This is repeated for 1,000 times and the averaged misclassification errors in percentage versus the numbers of extracted features are illustrated in Figure 5. Quantitative results, with penalty parameter selected using the cross validation criterion, are presented in Table 4. In this data set, SLR performs the best, followed by CODA-LS and CODA-ROAD. Moreover, the CODA methods beat their corresponding parametric methods in this experiment. It can be observed in Table 4 that there is no significant difference between SLR and CODA-ROAD.

6. Discussions

In this paper a high dimensional classification method named the CODA (Copula Discriminant Analysis) is proposed. The main contributions of this paper include: (i) We relax the normality assumption of linear discriminant analysis through the nonparanormal (or Gaussian copula) modeling; (ii) We use the nonparanormal SKEPTIC procedure proposed by Liu et al. (2012) to efficiently estimate the model parameters; (iii) We build a connection of the ROAD and lasso and provide an approach to solve the problem that the rank-based covariance matrix may not be positive semidefinite; (iv) We provide sufficient conditions to secure the variable selection consistency with the parametric rate, and the expected misclassification error is consistent to the Bayes risk; (v) Careful experiments on synthetic and real data sets are conducted to support the theoretical claims.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their comments and suggestions which have led to valuable improvements of this paper. We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-1116730).

Appendix A. Proof of Theorem 2

To show that Theorem 2 holds, we need to provide several important lemmas using results of large deviation and empirical process. First, define $\phi(\cdot)$ and $\Phi(\cdot)$ to be the probability density function and cumulative distribution function of the standard Gaussian distribution. For any $x \in \mathbb{R}$, we denote by $x^+ = x \cdot I(x > 0)$ and $x^- = -x \cdot I(x < 0)$. By definition, $f_j(t) = \Phi^{-1}(F_j(t))$ and $g_j(u) := f_j^{-1}(u) = F_j^{-1}(\Phi(u))$. Here for notation simplicity, let $\widetilde{F}_j(t)$ and $\widehat{F}_j(t)$ be the abbreviations of $\widetilde{F}_j(t; 1/(2n), \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and $\widehat{F}_j(t; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ defined in Section 2.2.

The following lemma quantifies the region of the value $\widetilde{F}_j$ in I_n and shows that $\widetilde{F}_j$ is not truncated in I_n almost surely.

Lemma 20 (Liu et al., 2012) *We have for large enough n,*

$$\mathbb{P}\left(\frac{1}{n} \leq \widetilde{F}_j(t) \leq 1 - \frac{1}{n}, \text{ for all } t \in I_n\right) = 1.$$

With Lemma 20, we can now prove the following key lemma, which provides an uniform convergence rate on $\widetilde{F}_j(t)$ to $F_j(t)$. This result is mentioned in Liu et al. (2012), but without proof.

Lemma 21 *Consider a sequence of sub-intervals $[L_n^{(j)}, U_n^{(j)}]$ with both $L_n^{(j)} := g_j(\sqrt{\alpha \log n})$ and $U_n^{(j)} := g_j(\sqrt{\beta \log n}) \uparrow \infty$, then for any $0 < \alpha < \beta < 2$, for large enough n,*

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}} \sup_{L_n^{(j)} < t < U_n^{(j)}} \left| \frac{\widetilde{F}_j(t) - F_j(t)}{\sqrt{F_j(t)(1-F_j(t))}} \right| = C \text{ a.s.},$$

where $0 < C < 2\sqrt{2}$ is a constant.

Proof By Lemma 20, for large enough n,

$$\widetilde{F}_j(t) = \widehat{F}_j(t), \quad \text{for all } t \in I_n, \quad \text{almost surely.} \tag{22}$$

Given $\xi_1, \ldots, \xi_n$ a series of i.i.d random variables from Unif$(0,1)$ and define $\mathbb{G}_n(t) := \frac{1}{n}\sum I(\xi_i < t)$, it is easy to see that

$$\widehat{F}_j(t) = \mathbb{G}_n(F_j(t)) \text{ a.s..} \tag{23}$$

Define

$$\mathbb{U}_n(u) := \frac{\mathbb{G}_n(u) - u}{\sqrt{u(1-u)}}.$$

By Equation (22) and (23), it is easy to see that

$$\mathbb{U}_n(F_j(t)) = \frac{\widetilde{F}_j(t) - F_j(t)}{\sqrt{F_j(t)(1-F_j(t))}} \text{ a.s..} \tag{24}$$

By Theorem 1 in Section 2 (Chapter 16) of Shorack and Wellner (1986), we know that

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}} \sup_{0 \leq u \leq 1/2} (\mathbb{U}_n(u))^- = \sqrt{2} \text{ a.s..} \tag{25}$$

And by Theorem 2 in Section 3 (Chapter 16) of Shorack and Wellner (1986), for $a_n \to 0$ such that $\frac{\log\log(1/a_n)}{\log\log n} \to 1$, we have

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}} \sup_{a_n \le u \le 1/2} (\mathbb{U}_n(u))^+ = \sqrt{2} \text{ a.s..} \tag{26}$$

Combining Equation (25) and (26) together, we have

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}} \sup_{a_n \le u \le 1/2} |\mathbb{U}_n(u)| \le 2\sqrt{2} \text{ a.s..} \tag{27}$$

Furthermore, for any $u \in [0,1]$,

$$\mathbb{G}_n(1-u) = \frac{1}{n}\sum I(\xi_i < 1-u) = \frac{1}{n}\sum I(1-\xi_i \ge u) = 1 - \mathbb{G}_n(u),$$

which implies that

$$\mathbb{U}_n(1-u) = -\mathbb{U}_n(u).$$

Therefore, by Equation (27), for $a_n \downarrow 0$ such that $\frac{\log\log(1/a_n)}{\log\log n} \to 1$, we have

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}} \sup_{1/2 \le u \le 1-a_n} |\mathbb{U}_n(u)| \le 2\sqrt{2} \quad \text{a.s..} \tag{28}$$

Finally, choosing $a_n = 1 - F_j(U_n^{(j)})$, we have

$$a_n = 1 - \Phi(\sqrt{\beta \log n}) \approx n^{-\beta/2} \quad \text{and} \quad \frac{\log\log n^{\beta/2}}{\log\log n} \to 1,$$

so taking $a_n = 1 - F_j(U_n^{(j)})$ into Equation (28), the result follows by using Equation (24). ■

Proof [Proof of the Theorem 2] Finally, we prove the Theorem 2. By symmetry, we only need to conduct analysis on a sub-interval of $I_n^s \subset I_n$:

$$I_n^s := \left[g_j(0), g_j\left(\sqrt{2(1-\gamma)\log n}\right)\right].$$

We define a series $0 < \alpha < 1 < \beta_1 < \beta_2 < \ldots < \beta_\kappa$ and denote by $\beta_0 := \alpha$,

$$I_{0n} := \left[g_j(0), g_j(\sqrt{\alpha \log n})\right],$$

$$I_{1n} := \left[g_j(\sqrt{\alpha\log n}), g_j(\sqrt{\beta_1 \log n})\right], \quad \ldots \quad, I_{\kappa n} := \left[g_j(\sqrt{\beta_{\kappa-1}\log n}), g_j(\sqrt{\beta_\kappa \log n})\right].$$

For $i = 0, \ldots, \kappa$, we can rewrite

$$\sup_{t \in I_{in}} \left| \widetilde{f}_j(t) - f_j(t) \right| = \sup_{t \in I_{in}} \left| \Phi^{-1}(\widetilde{F}_j(t)) - \Phi^{-1}(F_j(t)) \right|.$$

By the mean value theorem, for some ξ_n such that

$$\xi_n \in \left[\min\{\widetilde{F}_j(g_j(\sqrt{\beta_{i-1}\log n})), F_j(g_j(\sqrt{\beta_{i-1}\log n}))\}, \max\{\widetilde{F}_j(g_j(\sqrt{\beta_i \log n})), F_j(g_j(\sqrt{\beta_i \log n}))\}\right],$$

we have

$$\sup_{t\in I_{in}} \left|\Phi^{-1}\left(\widetilde{F}_j(t)\right) - \Phi^{-1}\left(F_j(t)\right)\right| = \sup_{t\in I_{in}} \left|(\Phi^{-1})'(\xi_n)\left(\widetilde{F}_j(t) - F_j(t)\right)\right|. \tag{29}$$

Because Φ and Φ^{-1} are strictly increasing function, for large enough n, we have

$$(\Phi^{-1})'(\xi_n) \le (\Phi^{-1})'\left(\max\left\{F_j\left(g_j\left(\sqrt{\beta_i \log n}\right)\right), \widetilde{F}_j\left(g_j\left(\sqrt{\beta_i \log n}\right)\right)\right\}\right). \tag{30}$$

From Lemma 21, for large enough n, we have

$$\widetilde{F}_j(t) \le F_j(t) + 4\sqrt{\frac{\log\log n}{n}} \cdot \sqrt{1 - F_j(t)}.$$

In special, using the fact that $F_j(g_j(t)) = \Phi(t)$, we have

$$\begin{aligned}
\widetilde{F}_j(g_j(\sqrt{\beta_i \log n})) &\le F_j(g_j(\sqrt{\beta_i \log n})) + 4\sqrt{\frac{\log\log n}{n}} \cdot \sqrt{1 - F_j(g_j(\sqrt{\beta_i \log n}))} \\
&\le \Phi\left(\sqrt{\beta_i \log n} + 4\sqrt{\frac{\log\log n}{n^{1-\beta_i/2}}}\right).
\end{aligned}$$

The last inequality holds given Equation (B.4) to (B.12) in Liu et al. (2012).

Therefore,

$$\begin{aligned}
(\Phi^{-1})'(\widetilde{F}_j(g_j(\sqrt{\beta_i \log n}))) &\le \sqrt{2\pi}\exp\left(\frac{\left(\sqrt{\beta_i \log n} + 4\sqrt{\frac{\log\log n}{n^{1-\beta_i/2}}}\right)^2}{2}\right) \\
&\asymp (\Phi^{-1})'(F_j(g_j(\sqrt{\beta_i \log n}))).
\end{aligned}$$

Returning to Equation (30), we have

$$(\Phi^{-1})'(\xi_n) \le C(\Phi^{-1})'(F_j(g_j(\sqrt{\beta_i \log n}))) = \frac{C}{\phi(\sqrt{\beta_i \log n})} \le c_1 n^{\beta_i/2}, \tag{31}$$

where $C > 1$ and c_1 are generic constants. Specifically, when $i = 0$, using the Dvoretzky-Kiefer-Wolfowitz inequality (Massart, 1990; Dvoretzky et al., 1956), from Equation (29), we have

$$\sup_{t\in I_{0n}} \left|\Phi^{-1}(\widetilde{F}_j(t)) - \Phi^{-1}(F_j(t))\right| = O_P\left(\sqrt{\frac{\log\log n}{n^{1-\alpha}}}\right).$$

For any $i \in \{1, \ldots, \kappa\}$, using Lemma 21, for large enough n,

$$\begin{aligned}
\sup_{t\in I_{in}} \left|\widetilde{F}_j(t) - F_j(t)\right| &= O_P\left(\sqrt{\frac{\log\log n}{n}} \cdot \sqrt{1 - F_j\left(g_j(\sqrt{\beta_{i-1}\log n})\right)}\right) \\
&= O_P\left(\sqrt{\frac{\log\log n}{n}} \cdot \sqrt{\frac{n^{-\beta_{i-1}/2}}{\sqrt{\alpha \log n}}}\right) \\
&= O_P\left(\sqrt{\frac{\log\log n}{n^{\beta_{i-1}/2+1}}}\right).
\end{aligned} \tag{32}$$

Again, using Equation (31), we have

$$(\Phi^{-1})'(\xi_n) \leq C(\Phi^{-1})'(F_j(g_j(\sqrt{\beta_i \log n}))) = \frac{C}{\phi(\sqrt{\beta_i \log n})} \leq c_1 n^{\beta_i/2},$$

and applying Equation (32), we have

$$\sup_{t \in I_{in}} \left| \Phi^{-1}(\widetilde{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| = O_P\left(\sqrt{\frac{\log\log n}{n^{1+\beta_{i-1}/2-\beta_i}}} \right).$$

Chaining the inequalities together and choose

$$\beta_i = (2-(1/2)^i)(1-\gamma), \quad i \in \{0,1,\ldots,\kappa\},$$

we have for any $i \in \{0,1,\ldots,\kappa\}$,

$$\begin{aligned} 1-\alpha &= 1-(1-\gamma) = \gamma \quad \text{and} \\ 1+\beta_{i-1}/2-\beta_i &= 1+\left(1-\frac{1}{2^i}\right)(1-\gamma) - \left(2-\frac{1}{2^i}\right)(1-\gamma) = \gamma. \end{aligned}$$

And therefore, we have

$$\sup_{I_{0n} \cup \ldots \cup I_{\kappa n}} \left| \Phi^{-1}(\widetilde{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| = O_P\left(\sqrt{\frac{\log\log n}{n^{\gamma}}} \right),$$

while

$$I_{0n} \cup \ldots \cup I_{\kappa n} = \left[g_j(0), g_j\left(\sqrt{(2-2^{-\kappa})(1-\gamma)} \right) \right].$$

Taking $\kappa \uparrow \infty$, we have the result. ■

Appendix B. Proof of Theorem 11

To prove Theorem 11, we need the following three key lemmas. Lemma 22 claims that, under certain constraints on the transformation functions, there exist fast rates for the sample means and projected Spearman's rho/Kendall's tau covariance matrices converging to the population means and covariance matrix for the nonparanormal. Lemma 23 provides exponential inequalities for two estimators we are most interested in in analyzing the theoretical performance of the CODA. Lemma 25 claims that $\widetilde{\boldsymbol{\Sigma}}_{SS}$ is invertible with high probability.

Lemma 22 *For any $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ i.i.d drawn from $\boldsymbol{X}$, where $\boldsymbol{X} \sim NPN(\boldsymbol{\mu},\boldsymbol{\Sigma},f)$, $0 < 1/\sigma_{\max} < \min_j\{\sigma_j\} < \max_j\{\sigma_j\} < \sigma_{\max} < \infty$, $\max_j |\mu_j| \leq \sigma_{\max}$ and $g := f^{-1}$ satisfies $g_j^2 \in TF(K)$, $j = 1,\ldots d$ for some constant $K < \infty$, we have for any $t \geq 32\pi\sqrt{\frac{\log d}{n\log 2}}$,*

$$\mathbb{P}\left(|\widetilde{S}_{jk} - \Sigma_{jk}| > t \right) \leq 2\exp(-c_0' n t^2), \tag{33}$$

$$\mathbb{P}\left(|\widehat{\mu}_j - \mu_j| > t \right) \leq 2\exp(-c_1' n t^2), \tag{34}$$

where c_0' and c_1' are two constants only depending on the choice of $\{g_j\}_{j=1}^d$.

Proof Because $\sigma_{\max}$ is a constant which does not scale with (n,d,s), without loss of generality we can assume that $K \geq 1$, $\boldsymbol{\mu} = \mathbf{0}$ and $\mathrm{diag}(\boldsymbol{\Sigma}) = \mathbf{1}$. The key is to prove that the high order moments of each X_j and X_j^2 will not grow very fast.

We only focus on $j = 1$ and the results can be generalized to $j = 2,3,\ldots,d$. Define $Z := f_1(X_1) \sim N(0,1)$. We have $\forall\ m \in \mathbb{Z}^+$, because $g_1^2 \in \mathrm{TF}(K)$ for some constant K,

$$\mathbb{E}|X_1^2|^m = \mathbb{E}|g_1^2(Z)|^m \leq \frac{m!}{2}K^m.$$

Therefore, by Lemma 5.7 of van de Geer (2000), $\widehat{\sigma}_1^2$ goes to σ_1^2 exponentially fast. To show that the Equation (34) holds, we have

$$\begin{aligned}
&\mathbb{E}|X_1|^m = \mathbb{E}|X_1^2|^{m/2} \leq \frac{(m/2)!}{2}K^{m/2} < \frac{m!}{2}K^m, \quad \text{if } m \text{ is even,}\\
&\mathbb{E}|X_1|^m \leq 1 + \mathbb{E}|X_1|^m I(|X_1| \geq 1) \leq 1 + \mathbb{E}(|X_1|^{m+1} I(|X_1| \geq 1))\\
&\qquad \leq 1 + \mathbb{E}|X_1|^{m+1} \leq 1 + \frac{\left(\frac{m+1}{2}\right)!}{2}K^{\frac{m+1}{2}} < \frac{m!}{2}(2K+2)^m, \quad \text{if } m \text{ is odd.}
\end{aligned}$$

Therefore, again by Lemma 5.7 of van de Geer (2000), $\widehat{\mu}_1$ goes to μ_1 exponentially fast.

Similarly we can prove that $\mathbb{P}(|\widehat{\sigma}_j - \sigma_j| \geq t) = O(\exp(-cnt^2))$ for the generic constant c. Therefore, to prove that Equation (33) holds, the only thing left is to show that combining $\widehat{\sigma}_j, \widehat{\sigma}_k$ with $\widetilde{R}_{jk}$ does not change the rate. Actually, suppose that

$$\begin{aligned}
\mathbb{P}\left(\left|\widehat{\sigma}_j - \sigma_j\right| > \varepsilon\right) &\leq \eta_1(n,\varepsilon),\\
\mathbb{P}\left(\left|\widetilde{R}_{jk} - \Sigma^0_{jk}\right| > \varepsilon\right) &\leq \eta_2(n,\varepsilon),
\end{aligned}$$

then we have

$$\begin{aligned}
&\mathbb{P}\left(\left|\widetilde{S}_{jk} - \Sigma_{jk}\right| > \varepsilon\right)\\
=\ &\mathbb{P}\left(\left|(\widehat{\sigma}_j\widehat{\sigma}_k - \sigma_j\sigma_k)\widetilde{R}_{jk} + \sigma_j\sigma_k\left(\widetilde{R}_{jk} - \Sigma^0_{jk}\right)\right| > \varepsilon\right)\\
\leq\ &\mathbb{P}\left(\left|(\widehat{\sigma}_j\widehat{\sigma}_k - \sigma_j\sigma_k)\widetilde{R}_{jk}\right| > \frac{\varepsilon}{2}\right) + \mathbb{P}\left(\left|\sigma_j\sigma_k\left(\widetilde{R}_{jk} - \Sigma^0_{jk}\right)\right| > \frac{\varepsilon}{2}\right)\\
\leq\ &\mathbb{P}\left(\left|\widehat{\sigma}_j\widehat{\sigma}_k - \sigma_j\sigma_k\right| > \frac{\varepsilon}{2}\right) + \mathbb{P}\left(\left|\widetilde{R}_{jk} - \Sigma^0_{jk}\right| > \frac{\varepsilon}{2\sigma^2_{\max}}\right)\\
\leq\ &\mathbb{P}\left(\left|(\widehat{\sigma}_j - \sigma_j)(\widehat{\sigma}_k - \sigma_k) + \sigma_j(\widehat{\sigma}_k - \sigma_k) + \sigma_k(\widehat{\sigma}_j - \sigma_j)\right| > \frac{\varepsilon}{2}\right) + \eta_2\left(n, \frac{\varepsilon}{2\sigma^2_{\max}}\right)\\
\leq\ &\mathbb{P}\left(\left|(\widehat{\sigma}_j - \sigma_j)(\widehat{\sigma}_k - \sigma_k)\right| > \frac{\varepsilon}{6}\right) + \mathbb{P}\left(\left|\sigma_j(\widehat{\sigma}_k - \sigma_k)\right| > \frac{\varepsilon}{6}\right)\\
&+\mathbb{P}\left(\left|\sigma_k(\widehat{\sigma}_j - \sigma_j)\right| > \frac{\varepsilon}{6}\right) + \eta_2\left(n, \frac{\varepsilon}{2\sigma^2_{max}}\right)\\
\leq\ &\mathbb{P}\left(\left|\widehat{\sigma}_j - \sigma_j\right| > \sqrt{\frac{\varepsilon}{6}}\right) + \mathbb{P}\left(|\widehat{\sigma}_k - \sigma_k| > \sqrt{\frac{\varepsilon}{6}}\right)\\
&+\mathbb{P}\left(|\widehat{\sigma}_k - \sigma_k| > \frac{\varepsilon}{6\sigma_{max}}\right) + \mathbb{P}\left(\left|\widehat{\sigma}_j - \sigma_j\right| > \frac{\varepsilon}{6\sigma_{max}}\right) + \eta_2\left(n, \frac{\varepsilon}{2\sigma^2_{max}}\right)\\
\leq\ &2\eta_1\left(n, \sqrt{\frac{\varepsilon}{6}}\right) + 2\eta_1\left(n, \frac{\varepsilon}{6\sigma_{max}}\right) + \eta_2\left(n, \frac{\varepsilon}{2\sigma^2_{max}}\right).
\end{aligned}$$

Due to Lemma 5, we have for all $t \geq 32\pi\sqrt{\frac{\log d}{n\log 2}}$

$$\mathbb{P}(|\widetilde{R}_{jk} - \Sigma^0_{jk}| > t) \leq 2\exp(-cnt^2),$$

for some generic constant c. It means that η_1 and η_2 are both of parametric exponential decay rate. we complete the proof. ■

Lemma 23 *If n_0 and n_1 are deterministic, then there exists a constant c_0 such that for any $\varepsilon \geq 32\pi\sqrt{\frac{\log d}{(n_0 \wedge n_1)\log 2}}$, we have*

$$\mathbb{P}\left(\left|\widetilde{\Sigma}_{jk} - C_{jk}\right| > \varepsilon\right) \leq 2\exp\left(-c_0 n\varepsilon^2\right), \quad \forall\, j,k = 1,\ldots,d;$$
$$\mathbb{P}(||(\widehat{\mu}_1 - \widehat{\mu}_0) - (\mu_1 - \mu_0)||_\infty > \varepsilon) \leq 2d\exp(-c_1 n\varepsilon^2).$$

Proof Using Lemma 22 and the fact that $\mathbb{P}(|n_j - \frac{n}{2}| \geq n\varepsilon) \leq 2\exp(-2n\varepsilon^2)$ for $j = 0, 1$, we have the result. ■

Remark 24 *Here n_0 and n_1 are "pretended" to be deterministic but not random variables. Later we will see that because $n_0 \wedge n_1 > \frac{n}{4}$ with an overwhelming probability, we can easily rewrite the condition $\varepsilon \geq 32\pi\sqrt{\frac{\log d}{(n_0 \wedge n_1)\log 2}}$ to be a deterministic one: $\varepsilon \geq 64\pi\sqrt{\frac{\log d}{n\log 2}}$ in the final presentation.*

Lemma 25 *Let $\lambda_{\min}(\mathbf{C}_{SS}) = \delta$. If $\delta \geq 64\pi s\sqrt{\frac{\log d}{(n_0 \wedge n_1)\log 2}}$, we have*

$$\mathbb{P}(\widetilde{\mathbf{\Sigma}}_{SS} \succ 0) \geq 1 - 2s^2\exp\left(-\frac{c_0 n\delta^2}{4s^2}\right). \tag{35}$$

Proof Let $\widehat{\mathbf{\Delta}} = \widetilde{\mathbf{\Sigma}}_{SS} - \mathbf{C}_{SS}$. Using Lemma 23, in probability $1 - 2s^2\exp(-c_0 nt^2)$, $\|\widehat{\mathbf{\Delta}}\|_{\max} \leq t$. Therefore, for any $\boldsymbol{v} \in \mathbb{R}^s$,

$$\boldsymbol{v}^T\widetilde{\mathbf{\Sigma}}_{SS}\boldsymbol{v} = \boldsymbol{v}^T\mathbf{C}_{SS}\boldsymbol{v} + \boldsymbol{v}^T\widehat{\mathbf{\Delta}}\boldsymbol{v} \geq \delta\|\boldsymbol{v}\|_2^2 + \lambda_{\min}(\widehat{\mathbf{\Delta}})\|\boldsymbol{v}\|_2^2,$$

where $\delta = \lambda_{\min}(\mathbf{C}_{SS})$. By the norm equivalence, we have

$$\|\widehat{\mathbf{\Delta}}\|_{\text{op}} \leq s\|\widehat{\mathbf{\Delta}}\|_{\max} \leq st,$$

where $||\cdot||_{\text{op}}$ is the matrix operator norm. Since

$$||-\widehat{\mathbf{\Delta}}||_{\text{op}} = \lambda_{\max}(-\widehat{\mathbf{\Delta}}) = \lambda_{\max}(\mathbf{C}_{SS} - \widetilde{\mathbf{\Sigma}}_{SS}) = -\lambda_{\min}(\widetilde{\mathbf{\Sigma}}_{SS} - \mathbf{C}_{SS}) = -\lambda_{\min}(\widehat{\mathbf{\Delta}}),$$

and

$$||-\widehat{\mathbf{\Delta}}||_{\text{op}} \leq s||-\widehat{\mathbf{\Delta}}||_{\max} = s||\widehat{\mathbf{\Delta}}||_{\max} \leq st,$$

we can further have

$$\lambda_{\min}(\widehat{\boldsymbol{\Delta}}) \geq -\|\widehat{\boldsymbol{\Delta}}\|_{\text{op}} \geq -st.$$

Therefore we have

$$\boldsymbol{v}^T \widetilde{\boldsymbol{\Sigma}}_{SS} \boldsymbol{v} \geq (\delta - st)\|\boldsymbol{v}\|_2^2, \quad \text{i.e., } \lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_{SS}) \geq \delta - st.$$

In other words, for all $t < \delta/s$, we have $\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_{SS}) > 0$. In particular, choosing $t = \delta/(2s)$, we have $\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_{SS}) = \delta/2 > 0$. This proves that Equation (35) holds with high probability. ■

Using Lemma 22, Lemma 23 and Lemma 25, Theorem 11 can be obtained using a similar proof structure of Mai et al. (2012). For concreteness and self-containedness, we provide a proof of the remaining part in the last section of the appendix.

Appendix C. Proof of Theorem 15

To prove Theorem 15, we first need to quantify the convergence rate of $\widehat{f}$ to f, or equivalently, $\widehat{f}_0 := \{\widehat{f}_{0j}\}_{j=1}^d$ and $\widehat{f}_1 := \{\widehat{f}_{1j}\}_{j=1}^d$'s convergence rates to f. By symmetry, we can focus on $\widehat{f}_0$.

Lemma 26 *Let $g_j := f_j^{-1}$ be the inverse function of f_j. We define*

$$I_n := \left[g_j\left(-\sqrt{2(1-\gamma)\log n}\right), g_j\left(\sqrt{2(1-\gamma)\log n}\right)\right],$$

then $\sup_{t \in I_n} |\widehat{f}_{0j}(t) - f_j(t)| = O_P\left(\sqrt{\frac{\log\log n}{n^\gamma}}\right)$.

Proof Using Lemma 22, a similar proof as Theorem 2 can be applied. ■

Then we can proceed to proof of Theorem 15:

Proof We define $\{j_1, \ldots, j_s\} = S$ to be the indices of the s discriminant features, that is,

$$\beta_{j_k}^* \neq 0, \quad k = 1, \ldots, s.$$

In this way, we can further define

$$T_n = \left[g_{j_1}(-\sqrt{b\log n}), g_{j_1}(\sqrt{b\log n})\right] \times \ldots, \times \left[g_{j_s}(-\sqrt{b\log n}), g_{j_s}(\sqrt{b\log n})\right],$$

for some $0 < b < 1$. Moreover, an event M_n is defined as

$$M_n := \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}_S \in T_n\}.$$

Then we have

$$\begin{aligned}
&\mathbb{P}\left(|\widehat{G}\left(\boldsymbol{X}, \frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}, \widehat{f}\right) - G(\boldsymbol{X}, \boldsymbol{\beta}^{**}, f)| > t\right) \\
&\leq \mathbb{P}\left(|\widehat{G}\left(\boldsymbol{X}, \frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}, \widehat{f}\right) - G(\boldsymbol{X}, \boldsymbol{\beta}^{**}, f)| > t \,|\, \mathcal{R}(\mathbf{X}, \boldsymbol{\beta}^*, \lambda_n), M_n\right) \\
&\quad + \mathbb{P}(M_n^c) + \mathbb{P}(\mathcal{R}(\mathbf{X}, \boldsymbol{\beta}^*, \lambda_n)^c).
\end{aligned}$$

Given $\mathcal{R}(\mathbf{X},\beta^*,\lambda_n)$ and M_n hold, we have

$$\begin{aligned}\left|\widehat{G}\left(\boldsymbol{X},\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1},\widehat{f}\right)-G(\boldsymbol{X},\boldsymbol{\beta}^{**},f)\right| \leq& \left|(f(\boldsymbol{X})-\widehat{f}(\boldsymbol{X}))^T\boldsymbol{\beta}^{**}\right|+\left|\widehat{f}(\boldsymbol{X})^T\left(\boldsymbol{\beta}^{**}-\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1}\right)\right| \\ &+\left|\widehat{\boldsymbol{\mu}}^T\left(\boldsymbol{\beta}^{**}-\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1}\right)\right|+\left|(\widehat{\boldsymbol{\mu}}-\boldsymbol{\mu}_a)^T\boldsymbol{\beta}^{**}\right| \\ \leq& \beta^{**}_{\max}||(f(\boldsymbol{X})-\widehat{f}(\boldsymbol{X}))_S||_1+||(\widehat{f}(\boldsymbol{X}))_S||_1||\boldsymbol{\beta}^{**}-\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1}||_\infty \\ &+||\widehat{\boldsymbol{\mu}}_S||_1||\boldsymbol{\beta}^{**}-\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1}||_\infty+||\boldsymbol{\beta}^{**}||_\infty||(\widehat{\boldsymbol{\mu}}-\boldsymbol{\mu}_a)_S||_1.\end{aligned}$$

Using Theorem 2,

$$\sup_{\boldsymbol{X}\in M_n}||(f(\boldsymbol{X})-\widehat{f}(\boldsymbol{X}))_S||_1=O_P\left(s\sqrt{\frac{\log\log n}{n^{1-b/2}}}\right), \tag{36}$$

and by Lemma 26,

$$\sup_{\boldsymbol{X}\in M_n}||(\widehat{f}(\boldsymbol{X}))_S||_1=O_P\left(s\sqrt{2\log n}\right).$$

Using the Gaussian tail inequality,

$$\mathbb{P}\left(f_j(X_j)\geq\sqrt{b\log n}\right)=O\left(n^{-c_2\cdot b}\right),$$

so $\mathbb{P}(M_n^c)=O\left(sn^{-c_2\cdot b}\right)$. Using Lemma 23,

$$||\widehat{\boldsymbol{\mu}}_S||_1=O_P(s\Delta_{\max})\quad\text{and}\quad||(\widehat{\boldsymbol{\mu}}-\boldsymbol{\mu}_a)_S||_1=O_P(sn^{-1/2}).$$

Using the assumption that $A+B+C\to 0$ and $B\to 0$ in the Theorem 15, we have

$$||\boldsymbol{\beta}^{**}-\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1}||_\infty=O_P\left(D_{\max}\lambda_n\right)\quad\text{and}\quad\mathbb{P}\left(\mathcal{R}(\mathbf{X},\boldsymbol{\beta}^*,\lambda_n)^c\right)=o(1). \tag{37}$$

Combining Equation (36) to (37), we have

$$\left|\widehat{G}\left(\boldsymbol{X},\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0n_1},\widehat{f}\right)-G(\boldsymbol{X},\boldsymbol{\beta}^{**},f)\right|=O_P\left(s\beta^{**}_{\max}\sqrt{\frac{\log\log n}{n^{1-b/2}}}+sD_{\max}\lambda_n(\sqrt{\log n}+\Delta_{\max})+\frac{s\beta^{**}_{\max}}{\sqrt{n}}\right).$$

This completes the proof. ■

Appendix D. Proof of Corollary 19

Proof For notation simplicity, we denote by

$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\} \quad \text{and} \quad G^* := G(\boldsymbol{X}, \boldsymbol{\beta}^{**}, f), \quad \widetilde{G} := \widehat{G}\left(\boldsymbol{X}, \frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}, \widehat{f}\right).$$

Here we note that $\text{sign}(\widetilde{G}) = \text{sign}\left(\widehat{G}\left(\boldsymbol{X}, \widehat{\boldsymbol{\beta}}, \widehat{f}\right)\right)$. Then we have

$$\begin{aligned}
\mathbb{P}\left(Y \cdot \text{sign}(\widetilde{G}) < 0 | \mathcal{D}\right) &= \mathbb{P}\left(Y \cdot \text{sign}(G^*) + Y \cdot (\text{sign}(\widetilde{G}) - \text{sign}(G^*)) < 0 \mid \mathcal{D}\right) \\
&\leq \mathbb{P}(Y \cdot \text{sign}(G^*) < 0) + \mathbb{P}\left(Y \cdot (\text{sign}(\widetilde{G}) - \text{sign}(G^*)) < 0 \mid \mathcal{D}\right) \\
&\leq \mathbb{P}(Y \cdot \text{sign}(G^*) < 0) + \mathbb{P}\left(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*) \mid \mathcal{D}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}(C(\widehat{g})) - C(g^*) &\leq \mathbb{E}\left(\mathbb{P}(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*) \mid \mathcal{D})\right) \\
&= \mathbb{E}\left(\mathbb{E}(I(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*)) \mid \mathcal{D})\right) \\
&= \mathbb{E}\left(I(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*))\right) \\
&= \mathbb{P}\left(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*))\right).
\end{aligned}$$

Given $t_{n,d,s}$ a constant depending only on (d, n, s), we have

$$\begin{aligned}
& \mathbb{P}\left(\text{sign}(\widetilde{G}) \neq \text{sign}(G^*)\right) \\
= \;& \mathbb{P}\left(\widetilde{G} \cdot G^* < 0\right) \\
= \;& \mathbb{P}\left(\widetilde{G} \cdot G^* < 0, |\widetilde{G} - G^*| < t_{n,d,s}\right) + \mathbb{P}\left(\widetilde{G} \cdot G^* < 0, |\widetilde{G} - G^*| \geq t_{n,d,s}\right) \\
\leq \;& \mathbb{P}\left(|\widetilde{G} - G^*| < t_{n,d,s}, \widetilde{G} \cdot G^* < 0\right) + \mathbb{P}\left(|\widetilde{G} - G^*| > t_{n,d,s}\right) \\
\leq \;& \mathbb{P}\left(\widetilde{G} \cdot G^* < 0 \;\mid\; |\widetilde{G} - G^*| < t_{n,d,s}\right) \mathbb{P}\left(|\widetilde{G} - G^*| < t_{n,d,s}\right) + \mathbb{P}\left(|\widetilde{G} - G^*| > t_{n,d,s}\right) \\
\leq \;& \mathbb{P}\left(|G^*| < t_{n,d,s} \;\mid\; |\widetilde{G} - G^*| < t_{n,d,s}\right) \mathbb{P}\left(|\widetilde{G} - G^*| < t_{n,d,s}\right) + \mathbb{P}\left(|\widetilde{G} - G^*| > t_{n,d,s}\right) \\
\leq \;& \mathbb{P}(|G^*| < t_{n,d,s}) + \mathbb{P}\left(|\widetilde{G} - G^*| > t_{n,d,s}\right).
\end{aligned}$$

Suppose that the conditions in Corollary 3 hold, then choosing

$$t_{n,d,s} = s^2 \log^2 n \cdot \sqrt{\frac{\log d + \log s}{n}},$$

using Corollary 17, we know that

$$\mathbb{P}\left(\left|\widehat{G}\left(\boldsymbol{X}, \frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1}, \widehat{f}\right) - G(\boldsymbol{X}, \boldsymbol{\beta}^{**}, f)\right| > t_{n,d,s}\right) = o(1).$$

And using Lemma 6, we have

$$G(\boldsymbol{X},\boldsymbol{\beta}^{**},f)=(f(\boldsymbol{X})-\boldsymbol{\mu}_a)^T\boldsymbol{\beta}^{**}\sim N\left(\frac{\tau\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}{2},\tau^2\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d\right),$$

where $\tau=\dfrac{4}{4+\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}>0$. Therefore, by simple calculation, we have

$$\mathbb{P}\left(|G(\boldsymbol{X},\boldsymbol{\beta}^{**},f)|<t_{n,d,s}\right)=\Phi\left(\frac{t_{n,d,s}-\frac{\tau\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}{2}}{\tau\sqrt{\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}}\right)-\Phi\left(\frac{-t_{n,d,s}-\frac{\tau\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}{2}}{\tau\sqrt{\boldsymbol{\mu}_d^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_d}}\right)=o(1),$$

as long as $t_{n,d,s}\to 0$ because of the continuity of Φ. This proves that $\mathbb{P}\left(\operatorname{sign}(\widetilde{G})\neq\operatorname{sign}(G^*)\right)=o(1)$ and completes the proof. The same argument can be generalized to the case where the conditions in Theorem 15 hold. ■

Appendix E. Proof of the Remaining Part of Theorem 11

we now start to prove the remaining part of Theorem 11. In the sequel, all the equalities and inequalities are element-wise. The main structure of the proof is coming from Mai et al. (2012) and we include the proof here only for the paper concreteness and self-containedness.

Lemma 27 *Given* $n_j\sim\text{Binomial}\left(n,\frac{1}{2}\right)$ *for* $j=0,1$, *we have*

$$\mathbb{P}\left(\frac{3}{16}\leq\frac{n_0n_1}{n^2}\leq\frac{1}{4}\right)\geq 1-2\exp\left(-\frac{n}{8}\right).$$

Proof Using the Hoeffding's inequality, we have

$$\mathbb{P}\left(\frac{3}{16}\leq\frac{n_0n_1}{n^2}\leq\frac{1}{4}\right)=\mathbb{P}\left(\left|n_j-\frac{n}{2}\right|\leq\frac{n}{4}\right)=\mathbb{P}\left(\left|\frac{n_j}{n}-\frac{1}{2}\right|\leq\frac{1}{4}\right)\geq 1-2\exp\left(-\frac{n}{8}\right).$$

This completes the proof. ■

Proof [Proof of the Theorem 11] We define the event

$$E_0:=\left\{\frac{3}{16}\leq\frac{n_0n_1}{n^2}\leq\frac{1}{4}\right\}.$$

Under the event E_0, we consider the optimization problem in Equation (11). We firstly consider an intermediate optimum:

$$\widetilde{\boldsymbol{\beta}}_S:=\underset{\boldsymbol{\beta}_S\in\mathbb{R}^s}{\operatorname{argmin}}\left\{\frac{1}{2n}||\boldsymbol{y}-\widetilde{\mathbf{X}}_S\boldsymbol{\beta}_S||_2^2+\lambda_n||\boldsymbol{\beta}_S||_1\right\}.$$

Reminding that $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}/n$ has been replaced by $\widetilde{\boldsymbol{\Sigma}}$ in calculating $\widehat{\boldsymbol{\beta}}$. Using Lemma 25, $\widetilde{\boldsymbol{\Sigma}}_{SS}$ is invertible with high probability. Then, under the event that $\widetilde{\boldsymbol{\Sigma}}_{SS}$ is invertible, $\widetilde{\boldsymbol{\beta}}_S$ exists and is unique, moreover

$$\widetilde{\boldsymbol{\beta}}_S=(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1}\left[\frac{n_0n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1-\widehat{\boldsymbol{\mu}}_0)_S-\lambda_n\boldsymbol{z}_S\right],$$

where z_S is the subgradient such that $z_j = \text{sign}(\widetilde{\beta}_j) \neq 0$ and $-1 \leq z_j \leq 1$ if $\widetilde{\beta}_j = 0$.

To prove that $\widehat{\boldsymbol{\beta}} = (\widetilde{\boldsymbol{\beta}}_S, 0)$ with high probability, it suffices to show that $||z_{S^c}||_\infty \leq 1$, or equivalently $\mathcal{R}_1(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)$ holds, where

$$\mathcal{R}_1(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n) := \left\{ ||\frac{n_0 n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_{S^c} - \widetilde{\boldsymbol{\Sigma}}_{S^c S}(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1}\left[\frac{n_0 n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \lambda_n z_S\right] ||_\infty \leq \lambda_n \right\}. \tag{38}$$

Then following Equation (38), with high probability, we now can write

$$\begin{aligned} &\mathbb{P}(\mathcal{R}_1(\boldsymbol{X}, \boldsymbol{\beta}^{**}, \lambda_n)^c) \\ \leq &\mathbb{P}\left(||\frac{n_0 n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_{S^c} - \widetilde{\boldsymbol{\Sigma}}_{S^c S}(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1}\left(\frac{n_0 n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \lambda_n z_S\right) ||_\infty > \lambda_n \right). \end{aligned}$$

Let $\lambda = \frac{2n^2\lambda_n}{n_0 n_1}$ and using the matrix norm equivalency, we have

$$\begin{aligned} \mathbb{P}(\mathcal{R}_1(\boldsymbol{X}, \boldsymbol{\beta}^{**}, \lambda_n)^c) &\leq \mathbb{P}\left(||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_{S^c} - \widetilde{\boldsymbol{\Sigma}}_{S^c S}(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1}\left((\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \frac{\lambda}{2} z_S\right) ||_\infty > \lambda/2 \right) \\ &\leq \mathbb{P}\Bigg(\zeta \Delta_{\max} + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_{S^c} - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_{S^c}||_\infty \\ &\qquad + (\zeta + \psi) \cdot \left(\frac{\lambda}{2} + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty \right) > \frac{\lambda}{2} \Bigg), \end{aligned}$$

where

$$\zeta := ||\widetilde{\boldsymbol{\Sigma}}_{S^c S}(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} - \mathbf{C}_{S^c S}(\mathbf{C}_{SS})^{-1}||_\infty.$$

.

The key part of the rest of proof is obtain by using the concentration inequalities for several key estimators. In Lemma 28, we give such a result:

Lemma 28 *There exist constants c_0 and c_1 such that, under the event E_0, for any $\varepsilon > 64\pi\sqrt{\frac{\log d}{n \log 2}}$, we have*

$$\begin{aligned} \mathbb{P}\left(\left| \widetilde{\Sigma}_{jk} - C_{jk} \right| > \varepsilon \right) &\leq 2\exp\left(-nc_0\varepsilon^2\right), \quad \forall\, j,k = 1,\ldots,d; && (39) \\ \mathbb{P}\left(||\widetilde{\boldsymbol{\Sigma}}_{SS} - \mathbf{C}_{SS}||_\infty > \varepsilon \right) &\leq 2s^2 \exp\left(-\frac{nc_0\varepsilon^2}{s^2} \right); && (40) \\ \mathbb{P}\left(||\widetilde{\boldsymbol{\Sigma}}_{S^c S} - \mathbf{C}_{S^c S}||_\infty > \varepsilon \right) &\leq 2(d-s)s \exp\left(-\frac{nc_0\varepsilon^2}{s^2} \right); && (41) \\ \mathbb{P}(||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)||_\infty > \varepsilon) &\leq 2d\exp(-nc_1\varepsilon^2). && (42) \end{aligned}$$

And for any $\varepsilon < 1/D_{\max}$, we have

$$\mathbb{P}\left(\zeta > \varepsilon D_{\max}(\psi + 1)(1 - D_{\max}\varepsilon)^{-1} \right) \leq 2ds \exp\left(-\frac{nc_0\varepsilon^2}{s^2} \right). \tag{43}$$

Proof [Proof of the Lemma 28] Given Lemma 23, Equation (39) and Equation (42) are correct and Equation (40) and (41) are straightforward using Equation (39). To prove that Equation (43) holds, we have the key observation from Mai et al. (2012):

$$||\widetilde{\mathbf{\Sigma}}_{S^cS}(\widetilde{\mathbf{\Sigma}}_{SS})^{-1} - \mathbf{C}_{S^cS}(\mathbf{C}_{SS})^{-1}||_\infty \leq \\ \left(\psi||\widetilde{\mathbf{\Sigma}}_{SS} - \mathbf{C}_{SS}||_\infty + ||\widetilde{\mathbf{\Sigma}}_{S^cS} - \mathbf{C}_{S^cS}||_\infty\right)\left(D_{\max} + ||(\widetilde{\mathbf{\Sigma}}_{SS})^{-1} - (\mathbf{C}_{SS})^{-1}||_\infty\right).$$

We choose

$$\varepsilon \geq \max\{||\widetilde{\mathbf{\Sigma}}_{SS} - \mathbf{C}_{SS}||_\infty, ||\widetilde{\mathbf{\Sigma}}_{S^cS} - \mathbf{C}_{S^cS}||_\infty\},$$

and substitute $||\widetilde{\mathbf{\Sigma}}_{SS} - \mathbf{C}_{SS}||_\infty$ and $||\widetilde{\mathbf{\Sigma}}_{S^cS} - \mathbf{C}_{S^cS}||_\infty$ with ε, then apply Equation (40) and Equation (41), we have Equation (43). ■

Therefore, using the condition that, under the event E_0, we have

$$\varepsilon \leq \frac{n^2\lambda_n(1-\psi)}{2D_{\max}(n^2\lambda_n + n_0n_1(1+\psi)\Delta_{\max})} = \frac{\lambda(1-\psi)}{4D_{\max}(\lambda/2 + (1+\psi)\Delta_{\max})},$$

we have

$$\Delta_{\max} \leq \frac{(1-2\varepsilon D_{\max} - \psi)\lambda}{4(1+\psi)\varepsilon D_{\max}}.$$

Noticing that if

$$\zeta \leq \frac{(\psi+1)\varepsilon D_{\max}}{1 - D_{\max}\varepsilon}, \quad ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)||_\infty \leq \frac{\lambda(1-\psi-2\varepsilon D_{\max})}{4(1+\psi)}, \\ \lambda < \Delta_{\max} \quad \text{and} \quad \Delta_{\max} \leq \frac{(1-2\varepsilon D_{\max} - \psi)\lambda}{4(1+\psi)\varepsilon D_{\max}},$$

then $\zeta\Delta_{\max} + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_{S^c} - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_{S^c}||_\infty + (\zeta+\psi)\cdot\left(\frac{\lambda}{2} + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty\right) \leq \frac{\lambda}{2}$. Accordingly, denoting by

$$E_1 = \left\{\zeta \geq \frac{(\psi+1)\varepsilon D_{\max}}{1 - D_{\max}\varepsilon}\right\}, \\ E_2 = \left\{||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)||_\infty \geq \frac{\lambda(1-\psi-2\varepsilon D_{\max})}{4(1+\psi)}\right\},$$

we have

$$\mathbb{P}(\mathcal{R}_1(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)^c) \leq \mathbb{P}(E_1) + \mathbb{P}(E_2).$$

Using Lemma 28, we have the result that

$$\mathbb{P}(\mathcal{R}_1(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)^c) \leq 2ds\cdot\exp\left(-\frac{c_0n\varepsilon^2}{s^2}\right) + 2d\cdot\exp\left(-\frac{4c_1n\lambda_n^2(1-\psi-2\varepsilon D_{\max})^2}{(1+\psi)^2}\right). \tag{44}$$

Then, to prove that $|\widetilde{\boldsymbol{\beta}}_S| > 0$ with high probability, we consider the second set:

$$\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n) = \left\{\left|\widetilde{\boldsymbol{\beta}}_S\right| > 0\right\} = \left\{\left|(\widetilde{\mathbf{\Sigma}}_{SS})^{-1}\left[\frac{n_0n_1}{n^2}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \lambda_n\boldsymbol{z}_S\right]\right| > 0\right\}.$$

Again, denoting by $\lambda = \frac{2n^2\lambda_n}{n_0 n_1}$, we have

$$\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n) = \left\{ \left| (\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} \left[(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \lambda \boldsymbol{z}_S/2 \right] \right| > 0 \right\}.$$

Denote by $\zeta_1 := ||\widetilde{\boldsymbol{\Sigma}}_{SS} - \mathbf{C}_{SS}||_\infty$ and $\zeta_2 := ||(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} - (\mathbf{C}_{SS})^{-1}||_\infty$, we have

$$\begin{aligned}(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} \left[(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - \lambda \boldsymbol{z}_S/2 \right] &= \boldsymbol{\beta}_S^{**} + (\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} [(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S] \\ &\quad + [(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} - (\mathbf{C}_{SS})^{-1}](\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S - \lambda (\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1} \boldsymbol{z}_S/2, \end{aligned} \tag{45}$$

where we remind that $\mathbf{C}_{SS}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S = \boldsymbol{\beta}_S^{**}$. Therefore

$$\begin{aligned}\mathbb{P}(\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)) \geq \mathbb{P}\Bigg(& \beta_{\min}^{**} - (\zeta_2 + D_{\max})(\lambda/2 + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty) \\ & - \zeta_2 \Delta_{\max} > 0 \Bigg). \end{aligned}$$

Moreover, when $\zeta_1 D_{\max} < 1$, we have

$$\zeta_2 \leq ||(\widetilde{\boldsymbol{\Sigma}}_{SS})^{-1}||_\infty \zeta_1 D_{\max} \leq (D_{\max} + \zeta_2)\zeta_1 D_{\max},$$

and hence $\zeta_2 < D_{\max}^2 \zeta_1 / (1 - \zeta_1 D_{\max})$. Therefore

$$\begin{aligned}&\mathbb{P}(\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)) \\ &\geq \mathbb{P}\Bigg(\omega \Delta_{\max} D_{\max} - (1 - \zeta_1 D_{\max})^{-1} (D_{\max} \lambda/2 + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty D_{\max} \\ &\quad + D_{\max}^2 \zeta_1 \Delta_{\max}) > 0 \Bigg). \end{aligned}$$

Noting that $\omega \leq 1$ because $\Delta_{\max} D_{\max} \geq ||\beta^{**}||_\infty$, we have $\lambda \leq \frac{\beta_{\min}^{**}}{2D_{\max}} \leq \frac{2\beta_{\min}^{**}}{(3+\omega)D_{\max}}$ under event E_0. Therefore, given that $\zeta_1 \leq \varepsilon$ and $||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty \leq \varepsilon$ and $\varepsilon \leq \frac{\omega}{(3+\omega)D_{\max}}$, $\varepsilon \leq \frac{\Delta_{\max}\omega}{2(\omega+3)}$, we have

$$\omega \Delta_{\max} D_{\max} - (1 - \zeta_1 D_{\max})^{-1} (D_{\max} \lambda/2 + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty D_{\max} + D_{\max}^2 \zeta_1 \Delta_{\max}) > 0.$$

Therefore

$$\begin{aligned}\mathbb{P}(\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)^c) &\leq \mathbb{P}(\zeta_1 \geq \varepsilon) + \mathbb{P}(||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty \geq \varepsilon) \\ &\leq 2s^2 \exp(-c_0 n \varepsilon^2 / s^2) + 2s \exp(-n c_1 \varepsilon^2). \end{aligned} \tag{46}$$

Combining Equation (44) and Equation (46), we have that $\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)$ holds with high probability.

Finally, using Equation (45), given that $\mathcal{R}_1(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)$ and $\mathcal{R}_2(\mathbf{X}, \boldsymbol{\beta}^{**}, \lambda_n)$ hold, we have

$$\begin{aligned}&||\frac{n^2 \widehat{\boldsymbol{\beta}}}{n_0 n_1} - \boldsymbol{\beta}^{**}||_\infty = ||\frac{n^2 \widetilde{\boldsymbol{\beta}}_S}{n_0 n_1} - \boldsymbol{\beta}_S^{**}||_\infty \\ &\quad \leq (1 - \zeta_1 D_{\max})^{-1} (D_{\max} \lambda/2 + ||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty D_{\max} + D_{\max}^2 \zeta_1 \Delta_{\max}). \end{aligned}$$

Using the fact that $\varepsilon \leq \frac{\lambda}{2D_{\max}\Delta_{\max}}$ and $\varepsilon \leq \lambda$, we have that, under the event $\zeta_1 \leq \varepsilon$ and $||(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)_S - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)_S||_\infty \leq \varepsilon$,

$$\begin{aligned}||\frac{n^2\widehat{\boldsymbol{\beta}}}{n_0 n_1} - \boldsymbol{\beta}^{**}||_\infty &\leq (1 - \zeta_1 D_{\max})^{-1}(D_{\max}\lambda/2 + \lambda D_{\max} + D_{\max}\lambda/2) \\ &\leq \frac{2D_{\max}\lambda}{1 - \frac{\lambda}{2\Delta_{\max}}} \leq 4D_{\max}\lambda.\end{aligned}$$

We finalize the proof by using Lemma 27 to show that $\mathbb{P}(E_0^c) \leq 2\exp(-n/8)$. This completes the proof. ■

References

P.J. Bickel and E. Levina. Some theory for fisher's linear discriminant function, 'naive bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10: 989–1010, 2004.

T. Cai and W. Liu. A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106:1566–1577, 2012.

T. Cai, W. Liu, and X. Luo. A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106:594–607, 2011.

E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35:2313–2351, 2007.

D. Christensen. Fast algorithms for the calculation of kendall's τ. *Computational Statistics*, 20(1): 51–62, 2005.

R.T. Clemen and R. Reilly. Correlations and copulas for decision and risk analysis. *Management Science*, 45(2):208–224, 1999.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27 (3):642–669, 1956.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

A. Eloyan, J. Muschelli, M.B. Nebel, H. Liu, F. Han, T. Zhao, A. Barber, S. Joel, J.J. Pekar, S. Mostofsky, et al. Automated diagnoses of attention deficit hyperactive disorder using magnetic resonance imaging. 2012.

J. Fan and Y. Fan. High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6):2605, 2008.

J. Fan, Y. Feng, and X. Tong. A road to classification in high dimensional space. *Arxiv preprint arXiv:1011.6095*, 2010.

J. Friedman, T. Hastie, and R. Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1, 2009.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

J.H. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2007.

C.P. Hans, D.D. Weisenburger, T.C. Greiner, R.D. Gascoyne, J. Delabie, G. Ott, H.K. Müller-Hermelink, E. Campo, R.M. Braziel, E.S. Jaffe, et al. Confirmation of the molecular classification of diffuse large b-cell lymphoma by immunohistochemistry using a tissue microarray. *Blood*, 103 (1):275–282, 2004.

T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 155–176, 1996.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag. New York, NY, 2001.

H.L. Huang and F.L. Chang. Esvm: Evolutionary support vector machine for automatic feature selection and classification of microarray data. *Biosystems*, 90(2):516–528, 2007.

H. Ji, G. Wu, X. Zhan, A. Nolan, C. Koh, A. De Marzo, H.M. Doan, J. Fan, C. Cheadle, M. Fallahi, et al. Cell-type independent myc target genes reveal a primordial signature involved in biomass accumulation. *PloS one*, 6(10):e26057, 2011.

C. A. J. Klaassen and J. A. Wellner. Efficient estimation in the bivariate normal copula model: Normal margins are least-favorable. *Bernoulli*, 3(1):55–77, 1997.

W.H. Kruskal. Ordinal measures of association. *Journal of the American Statistical Association*, 53 No. 284.:814–861, 1958.

J.T. Leek and J.D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.

Y. Lin and Y. Jeon. Discriminant analysis through a semiparametric model. *Biometrika*, 90(2): 379–392, 2003.

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, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High dimensional semiparametric gaussian copula graphical models. *Annals of Statistics*, 2012.

H. Lovec, A. Grzeschiczek, M.B. Kowalski, and T. Möröy. Cyclin d1/bcl-1 cooperates with myc genes in the generation of b-cell lymphoma in transgenic mice. *The EMBO journal*, 13(15):3487, 1994.

Q. Mai, H. Zou, and M. Yuan. A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 2012.

P. Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, pages 1269–1283, 1990.

M.N. McCall, B.M. Bolstad, and R.A. Irizarry. Frozen robust multiarray analysis (frma). *Biostatistics*, 11(2):242–253, 2010.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3), 2006.

J. Nocedal and S.J. Wright. *Numerical optimization (2nd ed.)*. Springer verlag, 2006.

J.D. Power, A.L. Cohen, S.M. Nelson, G.S. Wig, K.A. Barnes, J.A. Church, A.C. Vogel, T.O. Laumann, F.M. Miezin, B.L. Schlaggar, et al. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

P. Ravikumar, M. Wainwright, G. Raskutti, and B. Yu. Model selection in Gaussian graphical models: High-dimensional consistency of ℓ_1-regularized MLE. In *Advances in Neural Information Processing Systems 22*, Cambridge, MA, 2009. MIT Press.

A.J. Rothman, P.J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, 2:494–515, 2008.

K. Scheinberg, S. Ma, , and D. Glodfarb. Sparse inverse covariance selection via alternating linearization methods. In *Advances in Neural Information Processing Systems (NIPS), 23,*, 2010.

J. Shao, Y. Wang, X. Deng, and S. Wang. Sparse linear discriminant analysis by thresholding for high dimensional data. *Arxiv preprint arXiv:1105.3561*, 2011.

G.R. Shorack and J.A. Wellner. *Empirical Processes With Applications to Statistics*. Wiley, 1986.

J.B. Smith and E. Wickstrom. Antisense c-myc and immunostimulatory oligonucleotide inhibition of tumorigenesis in a murine b-cell lymphoma transplant model. *Journal of the National Cancer Institute*, 90(15):1146–1154, 1998.

G. Strang. *Introduction to Linear Algebra*. Wellesley Cambridge Pr, 2003.

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99 (10):6567, 2002.

A.B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

S van de Geer. *Empirical Processes in M-estimation*, volume 105. Cambridge university press Cambridge, UK, 2000.

M. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5): 2183–2202, May 2009.

L. Wang, J. Zhu, and H. Zou. Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics*, 24(3):412–419, 2008.

S. Wang and J. Zhu. Improved centroids estimation for the nearest shrunken centroid classifier. *Bioinformatics*, 23(8):972, 2007.

D. Witten and R. Tibshirani. Penalized classification using fishers linear discriminant. *Journal of the Royal Statistical Society, Series B*, 2011.

L. Xue and H. Zou. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *Annals of Statistics*, 2012.

M. Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286, 2010.

P. Zhao and B. Yu. On model selection consistency of lasso. *J. of Mach. Learn. Res.*, 7:2541–2567, 2007.

T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman. The huge package for high-dimensional undirected graph estimation in r. *The Journal of Machine Learning Research*, 98888:1059–1062, 2012.

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 14 (2013) 673-701 Submitted 12/11; Revised 9/12; Published 2/13

Bayesian Nonparametric Hidden Semi-Markov Models

Matthew J. Johnson MATTJJ@CSAIL.MIT.EDU
Alan S. Willsky WILLSKY@MIT.EDU
Laboratory for Information and Decision Systems
Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139-4307, USA

Editor: David Blei

Abstract

There is much interest in the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) as a natural Bayesian nonparametric extension of the ubiquitous Hidden Markov Model for learning from sequential and time-series data. However, in many settings the HDP-HMM's strict Markovian constraints are undesirable, particularly if we wish to learn or encode non-geometric state durations. We can extend the HDP-HMM to capture such structure by drawing upon explicit-duration semi-Markov modeling, which has been developed mainly in the parametric non-Bayesian setting, to allow construction of highly interpretable models that admit natural prior information on state durations.

In this paper we introduce the explicit-duration Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) and develop sampling algorithms for efficient posterior inference. The methods we introduce also provide new methods for sampling inference in the finite Bayesian HSMM. Our modular Gibbs sampling methods can be embedded in samplers for larger hierarchical Bayesian models, adding semi-Markov chain modeling as another tool in the Bayesian inference toolbox. We demonstrate the utility of the HDP-HSMM and our inference methods on both synthetic and real experiments.

Keywords: Bayesian nonparametrics, time series, semi-Markov, sampling algorithms, Hierarchical Dirichlet Process Hidden Markov Model

1. Introduction

Given a set of sequential data in an unsupervised setting, we often aim to infer meaningful states, or "topics," present in the data along with characteristics that describe and distinguish those states. For example, in a speaker diarization (or who-spoke-when) problem, we are given a single audio recording of a meeting and wish to infer the number of speakers present, when they speak, and some characteristics governing their speech patterns (Tranter and Reynolds, 2006; Fox et al., 2008). Or in separating a home power signal into the power signals of individual devices, we would be able to perform the task much better if we were able to exploit our prior knowledge about the levels and durations of each device's power modes (Kolter and Johnson, 2011). Such learning problems for sequential data are pervasive, and so we would like to build general models that are both flexible enough to be applicable to many domains and expressive enough to encode the appropriate information.

Hidden Markov Models (HMMs) have proven to be excellent general models for approaching learning problems in sequential data, but they have two significant disadvantages: (1) state duration distributions are necessarily restricted to a geometric form that is not appropriate for many real-world data, and (2) the number of hidden states must be set a priori so that model complexity is not inferred from data in a Bayesian way.

Recent work in Bayesian nonparametrics has addressed the latter issue. In particular, the Hierarchical Dirichlet Process HMM (HDP-HMM) has provided a powerful framework for inferring arbitrarily large state complexity from data (Teh et al., 2006; Beal et al., 2002). However, the HDP-HMM does not address the issue of non-Markovianity in real data. The Markovian disadvantage is even compounded in the nonparametric setting, since non-Markovian behavior in data can lead to the creation of unnecessary extra states and unrealistically rapid switching dynamics (Fox et al., 2008).

One approach to avoiding the rapid-switching problem is the Sticky HDP-HMM (Fox et al., 2008), which introduces a learned global self-transition bias to discourage rapid switching. Indeed, the Sticky model has demonstrated significant performance improvements over the HDP-HMM for several applications. However, it shares the HDP-HMM's restriction to geometric state durations, thus limiting the model's expressiveness regarding duration structure. Moreover, its global self-transition bias is shared among all states, and so it does not allow for learning state-specific duration information. The infinite Hierarchical HMM (Heller et al., 2009) induces non-Markovian state durations at the coarser levels of its state hierarchy, but even the coarser levels are constrained to have a sum-of-geometrics form, and hence it can be difficult to incorporate prior information. Furthermore, constructing posterior samples from any of these models can be computationally expensive, and finding efficient algorithms to exploit problem structure is an important area of research.

These potential limitations and needed improvements to the HDP-HMM motivate this investigation into explicit-duration semi-Markov modeling, which has a history of success in the parametric (and usually non-Bayesian) setting. We combine semi-Markovian ideas with the HDP-HMM to construct a general class of models that allow for both Bayesian nonparametric inference of state complexity as well as general duration distributions. In addition, the sampling techniques we develop for the Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) provide new approaches to inference in HDP-HMMs that can avoid some of the difficulties which result in slow mixing rates. We demonstrate the applicability of our models and algorithms on both synthetic and real data sets.

The remainder of this paper is organized as follows. In Section 2, we describe explicit-duration HSMMs and existing HSMM message-passing algorithms, which we use to build efficient Bayesian inference algorithms. We also provide a brief treatment of the Bayesian nonparametric HDP-HMM and sampling inference algorithms. In Section 3 we develop the HDP-HSMM and related models. In Section 4 we develop extensions of the weak-limit and direct assignment samplers (Teh et al., 2006) for the HDP-HMM to our models and describe some techniques for improving the computational efficiency in some settings.

Section 5 demonstrates the effectiveness of the HDP-HSMM on both synthetic and real data. In synthetic experiments, we demonstrate that our sampler mixes very quickly on data generated by both HMMs and HSMMs and accurately learns parameter values and state cardinality. We also show that while an HDP-HMM is unable to capture the statistics of an HSMM-generated sequence, we can build HDP-HSMMs that efficiently learn whether data were generated by an HMM or HSMM. As a real-data experiment, we apply the HDP-HSMM to a problem in power signal disaggregation.

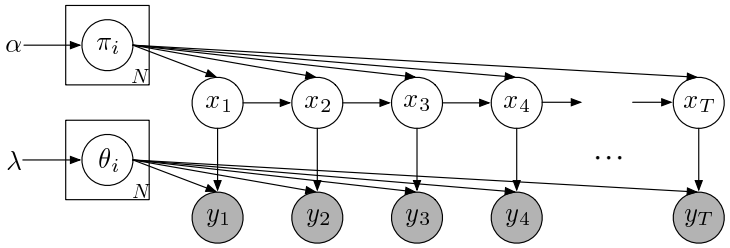


Figure 1: Basic graphical model for the Bayesian HMM. Parameters for the transition, emission, and initial state distributions are random variables. The symbol α represents the hyper-parameter for the prior distributions on state-transition parameters. The shaded nodes indicate observations on which we condition to form the posterior distribution over the unshaded latent components.

2. Background and Notation

In this section, we outline three main background topics: our notation for Bayesian HMMs, conventions for explicit-duration HSMMs, and the Bayesian nonparametric HDP-HMM.

2.1 HMMs

The core of the HMM consists of two layers: a layer of hidden *state* variables and a layer of *observation* or *emission* variables, as shown in Figure 1. The hidden state sequence, $x = (x_t)_{t=1}^T$, is a sequence of random variables on a finite alphabet, $x_t \in \{1, 2, \ldots, N\}$, that form a Markov chain. In this paper, we focus on time-homogeneous models, in which the transition distribution does not depend on t. The transition parameters are collected into a row-stochastic transition matrix $\pi = (\pi_{ij})_{i,j=1}^N$ where $\pi_{ij} = p(x_{t+1} = j | x_t = i)$. We also use $\{\pi_i\}$ to refer to the set of rows of the transition matrix. We use $p(y_t | x_t, \{\theta_i\})$ to denote the emission distribution, where $\{\theta_i\}$ represents parameters.

The Bayesian approach allows us to model uncertainty over the parameters and perform model averaging (for example, forming a prediction of an observation y_{T+1} by integrating out all possible parameters and state sequences), generally at the expense of somewhat more expensive algorithms. This paper is concerned with the Bayesian approach and so the model parameters are treated as random variables, with their priors denoted $p(\pi|\alpha)$ and $p(\{\theta_i\}|H)$.

2.2 HSMMs

There are several approaches to hidden semi-Markov models (Murphy, 2002; Yu, 2010). We focus on *explicit duration* semi-Markov modeling; that is, we are interested in the setting where each state's duration is given an explicit distribution. Such HSMMs are generally treated from a non-Bayesian perspective in the literature, where parameters are estimated and fixed via an approximate maximum-likelihood procedure (particularly the natural Expectation-Maximization algorithm, which constitutes a local search).

The basic idea underlying this HSMM formalism is to augment the generative process of a standard HMM with a random state duration time, drawn from some state-specific distribution when

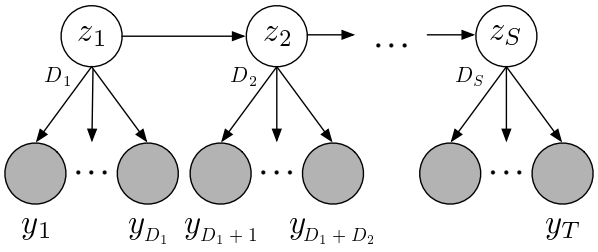


Figure 2: HSMM interpreted as a Markov chain on a set of super-states, $(z_s)_{s=1}^S$. The number of shaded nodes associated with each z_s, denoted by D_s, is drawn from a state-specific duration distribution.

the state is entered. The state remains constant until the duration expires, at which point there is a Markov transition to a new state. We use the random variable D_t to denote the duration of a state that is entered at time t, and we write the probability mass function for the random variable as $p(d_t|x_t = i)$.

A graphical model for the explicit-duration HSMM is shown in Figure 2 (from Murphy, 2002), though the number of nodes in the graphical model is itself random. In this picture, we see there is a Markov chain (without self-transitions) on S "super-state" nodes, $(z_s)_{s=1}^S$, and these super-states in turn emit random-length segments of observations, of which we observe the first T. Here, the symbol D_s is used to denote the random length of the observation segment of super-state s for $s = 1, \ldots, S$. The "super-state" picture separates the Markovian transitions from the segment durations.

When defining an HSMM model, one must also choose whether the observation sequence ends exactly on a segment boundary or whether the observations are *censored* at the end, so that the final segment may possibly be cut off in the observations. We focus on the right-censored formulation in this paper, but our models and algorithms can easily be modified to the uncensored or left-censored cases. For a further discussion, see Guédon (2007).

It is possible to perform efficient message-passing inference along an HSMM state chain (conditioned on parameters and observations) in a way similar to the standard alpha-beta dynamic programming algorithm for standard HMMs. The "backwards" messages are crucial in the development of efficient sampling inference in Section 4 because the message values can be used to efficiently compute the posterior information necessary to block-sample the hidden state sequence (x_t), and so we briefly describe the relevant part of the existing HSMM message-passing algorithm. As derived in Murphy (2002), we can define and compute the backwards messages[1] B and B^* as follows:

1. In Murphy (2002) and others, the symbols β and β^* are used for the messages, but to avoid confusion with our HDP parameter β, we use the symbols B and B^* for messages.

$$
\begin{aligned}
B_t(i) &:= p(y_{t+1:T}|x_t = i, F_t = 1) \\
&= \sum_j B_t^*(j) p(x_{t+1} = j|x_t = i), \\
B_t^*(i) &:= p(y_{t+1:T}|x_{t+1} = i, F_t = 1) \\
&= \sum_{d=1}^{T-t} B_{t+d}(i) \underbrace{p(D_{t+1} = d|x_{t+1} = i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1} = i, D_{t+1} = d)}_{\text{likelihood term}} \\
&\quad + \underbrace{p(D_{t+1} > T - t|x_{t+1} = i) p(y_{t+1:T}|x_{t+1} = i, D_{t+1} > T - t)}_{\text{censoring term}}, \\
B_T(i) &:= 1,
\end{aligned}
$$

where we have split the messages into B and B^* components for convenience and used $y_{k_1:k_2}$ to denote $(y_{k_1}, \ldots, y_{k_2})$. D_{t+1} represents the duration of the segment beginning at time $t+1$. The conditioning on the parameters of the distributions, namely the observation, duration, and transition parameters, is suppressed from the notation.

We write $F_t = 1$ to indicate a new segment begins at $t+1$ (Murphy, 2002), and so to compute the message from $t+1$ to t we sum over all possible lengths d for the segment beginning at $t+1$, using the backwards message at $t+d$ to provide aggregate future information given a boundary just after $t+d$. The final additive term in the expression for $B_t^*(i)$ is described in Guédon (2007); it constitutes the contribution of state segments that run off the end of the provided observations, as per the censoring assumption, and depends on the survival function of the duration distribution.

Though a very similar message-passing subroutine is used in HMM Gibbs samplers, there are significant differences in computational cost between the HMM and HSMM message computations. The greater expressive power of the HSMM model necessarily increases the computational cost: the above message passing requires $\mathcal{O}(T^2N + TN^2)$ basic operations for a chain of length T and state cardinality N, while the corresponding HMM message passing algorithm requires only $\mathcal{O}(TN^2)$. However, if the support of the duration distribution is limited, or if we truncate possible segment lengths included in the inference messages to some maximum d_{max}, we can instead express the asymptotic message passing cost as $\mathcal{O}(Td_{\text{max}}N + TN^2)$. Such truncations are often natural as the duration prior often causes the message contributions to decay rapidly with sufficiently large d. Though the increased complexity of message-passing over an HMM significantly increases the cost per iteration of sampling inference for a global model, the cost is offset because HSMM samplers need far fewer total iterations to converge. See the experiments in Section 5.

2.3 The HDP-HMM and Sticky HDP-HMM

The HDP-HMM (Teh et al., 2006) provides a natural Bayesian nonparametric treatment of the classical Hidden Markov Model. The model employs an HDP prior over an infinite state space, which enables both inference of state complexity and Bayesian mixing over models of varying complexity. We provide a brief overview of the HDP-HMM model and relevant inference algorithms, which we extend to develop the HDP-HSMM. A much more thorough treatment of the HDP-HMM can be found in, for example, Fox (2009).

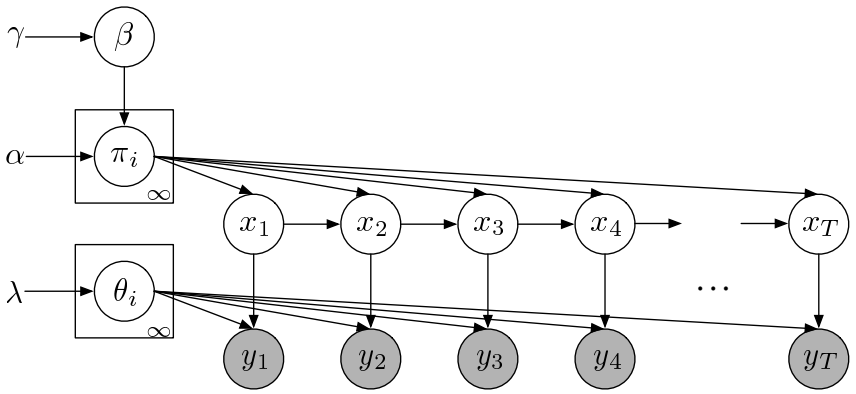


Figure 3: Graphical model for the HDP-HMM.

The generative process HDP-HMM(γ, α, H) given concentration parameters $\gamma, \alpha > 0$ and base measure (observation prior) H can be summarized as:

$$\begin{aligned}
&\beta \sim \text{GEM}(\gamma), \\
&\pi_i \overset{\text{iid}}{\sim} \text{DP}(\alpha, \beta) \qquad\qquad \theta_i \overset{\text{iid}}{\sim} H \qquad\qquad i = 1, 2, \ldots, \\
&x_t \sim \pi_{x_{t-1}}, \\
&y_t \sim f(\theta_{x_t}) \qquad\qquad t = 1, 2, \ldots, T,
\end{aligned}$$

where GEM denotes a stick breaking process (Sethuraman, 1994) and f denotes an observation distribution parameterized by draws from H. We set $x_1 := 1$. We have also suppressed explicit conditioning from the notation. See Figure 3 for a graphical model.

The HDP plays the role of a prior over infinite transition matrices: each π_j is a DP draw and is interpreted as the transition distribution from state j. The π_j are linked by being DP draws parameterized by the same discrete measure β, thus $\mathbb{E}[\pi_j] = \beta$ and the transition distributions tend to have their mass concentrated around a typical set of states, providing the desired bias towards re-entering and re-using a consistent set of states.

The Chinese Restaurant Franchise and direct-assignment collapsed sampling methods described in Teh et al. (2006); Fox (2009) are approximate inference algorithms for the full infinite dimensional HDP, but they have a particular weakness in the sequential-data context of the HDP-HMM: each state transition must be re-sampled individually, and strong correlations within the state sequence significantly reduce mixing rates (Fox, 2009). As a result, finite approximations to the HDP have been studied for the purpose of providing faster mixing. Of particular note is the popular weak limit approximation, used in Fox et al. (2008), which has been shown to reduce mixing times for HDP-HMM inference while sacrificing little of the "tail" of the infinite transition matrix. In this paper, we describe how the HDP-HSMM with geometric durations can provide an HDP-HMM sampling inference algorithm that maintains the "full" infinite-dimensional sampling process while mitigating the detrimental mixing effects due to the strong correlations in the state sequence, thus providing a new alternative to existing HDP-HMM sampling methods.

The Sticky HDP-HMM augments the HDP-HMM with an extra parameter $\kappa > 0$ that biases the process towards self-transitions and thus provides a method to encourage longer state durations.

The Sticky-HDP-HMM$(\gamma, \alpha, \kappa, H)$ generative process can be written

$$\begin{aligned}
&\beta \sim \text{GEM}(\gamma), \\
&\pi_i \overset{\text{iid}}{\sim} \text{DP}(\alpha+\kappa, \beta+\kappa\delta_j) \qquad\qquad \theta_i \overset{\text{iid}}{\sim} H \qquad\qquad i = 1, 2, \ldots, \\
&x_t \sim \pi_{x_{t-1}}, \\
&y_t \sim f(\theta_{x_t}) \qquad\qquad t = 1, 2, \ldots, T,
\end{aligned}$$

where δ_j denotes an indicator function that takes value 1 at index j and 0 elsewhere. While the Sticky HDP-HMM allows some control over duration statistics, the state duration distributions remain geometric; the goal of this work is to provide a model in which any duration distributions may be used.

3. Models

In this section, we introduce the explicit-duration HSMM-based models that we use in the remainder of the paper. We define the finite Bayesian HSMM and the HDP-HSMM and show how they can be used as components in more complex models, such as in a factorial structure. We describe generative processes that do not allow self-transitions in the state sequence, but we emphasize that we can also allow self-transitions and still employ the inference algorithms we describe; in fact, allowing self-transitions simplifies inference in the HDP-HSMM, since complications arise as a result of the hierarchical prior and an elimination of self-transitions. However, there is a clear modeling gain by eliminating self-transitions: when self-transitions are allowed, the "explicit duration distributions" do not model the state duration statistics directly. To allow direct modeling of state durations, we must consider the case where self-transitions do not occurr.

We do not investigate here the problem of selecting particular observation and duration distribution classes; model selection is a fundamental challenge in generative modeling, and models must be chosen to capture structure in any particular data. Instead, we provide the HDP-HSMM and related models as tools in which modeling choices (such as the selection of observation and duration distribution classes to fit particular data) can be made flexibly and naturally.

3.1 Finite Bayesian HSMM

The finite Bayesian HSMM is a combination of the Bayesian HMM approach with semi-Markov state durations and is the model we generalize to the HDP-HSMM. Some forms of finite Bayesian HSMMs have been described previously, such as in Hashimoto et al. (2009) which treats observation parameters as Bayesian latent variables, but to the best of our knowledge the first fully Bayesian treatment of all latent components of the HSMM was given in Johnson and Willsky (2010) and later independently in Dewar et al. (2012), which allows self-transitions.

It is instructive to compare this construction with that of the finite model used in the weak-limit HDP-HSMM sampler that will be described in Section 4.2, since in that case the hierarchical ties between rows of the transition matrix requires particular care.

The generative process for a Bayesian HSMM with N states and observation and duration parameter prior distributions of H and G, respectively, can be summarized as

$$\begin{aligned}
\pi_i &\overset{\text{iid}}{\sim} \text{Dir}(\alpha(1-\delta_i)) \qquad (\theta_i, \omega_i) \overset{\text{iid}}{\sim} H \times G && i = 1, 2, \ldots, N, \\
z_s &\sim \pi_{z_{s-1}}, && \\
D_s &\sim g(\omega_{z_s}), && s = 1, 2, \ldots, \\
x_{t_s^1:t_s^2} &= z_s, && \\
y_{t_s^1:t_s^2} &\overset{\text{iid}}{\sim} f(\theta_{z_s}) \qquad t_s^1 = \sum_{\bar{s}<s} D_{\bar{s}} \qquad t_s^2 = t_s^1 + D_s - 1,
\end{aligned}$$

where f and g denote observation and duration distributions parameterized by draws from H and G, respectively. The indices t_s^1 and t_s^2 denote the first and last index of segment s, respectively, and $x_{t_s^1:t_s^2} := (x_{t_s^1}, x_{t_s^1+1}, \ldots, x_{t_s^2})$. We use $\text{Dir}(\alpha(1-\delta_i))$ to denote a symmetric Dirichlet distribution with parameter α except with the ith component of the hyperparameter vector set to zero, hence fixing $\pi_{ii} = 0$ and ensuring there will be no self-transitions sampled in the *super-state sequence* (z_s). We also define the *label sequence* (x_t) for convenience; the pair (z_s, D_s) is the run-length encoding of (x_t). The process as written generates an infinite sequence of observations; we observe a finite prefix of size T.

Note, crucially, that in this definition the π_i are not tied across various i. In the HDP-HSMM, as well as the weak limit model used for approximate inference in the HDP-HSMM, the π_i will be tied through the hierarchical prior (specifically via β), and that connection is necessary to penalize the total number of states and encourage a small, consistent set of states to be visited in the state sequence. However, the interaction between the hierarchical prior and the elimination of self-transitions presents an inference challenge.

3.2 HDP-HSMM

The generative process of the HDP-HSMM is similar to that of the HDP-HMM (as described in, for example, Fox et al. (2008)), with some extra work to include duration distributions. The process HDP-HSMM(γ, α, H, G), illustrated in Figure 4, can be written

$$\begin{aligned}
\beta &\sim \text{GEM}(\gamma), && \\
\pi_i &\overset{\text{iid}}{\sim} \text{DP}(\alpha, \beta) \qquad (\theta_i, \omega_i) \overset{\text{iid}}{\sim} H \times G && i = 1, 2, \ldots, \\
z_s &\sim \bar{\pi}_{z_{s-1}}, && \\
D_s &\sim g(\omega_{z_s}) && s = 1, 2, \ldots, \\
x_{t_s^1:t_s^2} &= z_s, && \\
y_{t_s^1:t_s^2} &\overset{\text{iid}}{\sim} f(\theta_{x_t}) \qquad t_s^1 = \sum_{\bar{s}<s} D_{\bar{s}} \qquad t_s^2 = t_s^1 + D_s - 1,
\end{aligned}$$

where we use $\bar{\pi}_i := \frac{\pi_{ij}}{1-\pi_{ii}}(1-\delta_{ij})$ to eliminate self-transitions in the *super-state sequence* (z_s). As with the finite HSMM, we define the *label sequence* (x_t) for convenience. We observe a finite prefix of size T of the observation sequence.

Note that the atoms we edit to eliminate self-transitions are the same atoms that are affected by the global sticky bias in the Sticky HDP-HMM.

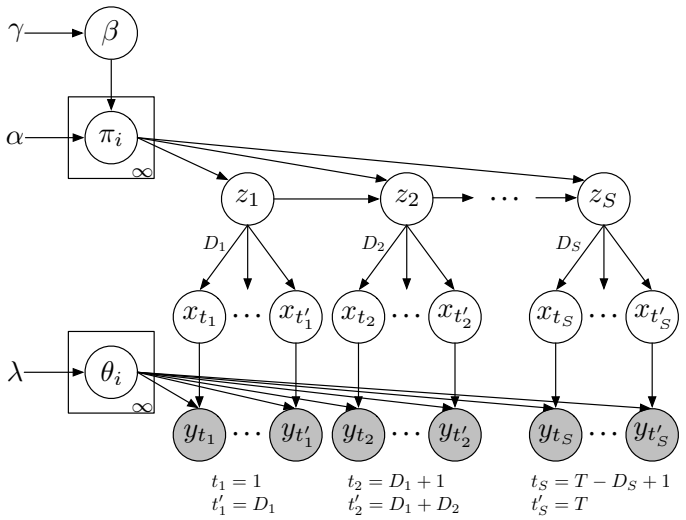


Figure 4: A graphical model for the HDP-HSMM in which the number of segments S, and hence the number of nodes, is random.

3.3 Factorial Structure

We can easily compose our sequential models into other common model structures, such as the factorial structure of the factorial HMM (Ghahramani and Jordan, 1997). Factorial models are very useful for source separation problems, and when combined with the rich class of sequential models provided by the HSMM, one can use prior duration information about each source to greatly improve performance (as demonstrated in Section 5). Here, we briefly outline the factorial model and its uses.

If we use $y \sim \text{HDP-HSMM}(\alpha, \gamma, H, G)$ to denote an observation sequence generated by the process defined in Sections 3.2 to 3.2, then the generative process for a factorial HDP-HSMM with K component sequences can be written

$$\begin{aligned} y^{(k)} &\sim \text{HDP-HSMM}(\alpha_k, \gamma_k, H_k, G_k) && k = 1, 2, \ldots, K, \\ \bar{y}_t &:= \sum_{k=1}^{K} y_t^{(k)} + w_t && t = 1, 2, \ldots, T, \end{aligned}$$

where w_t is a noise process independent of the other components of the model states.

A graphical model for a factorial HMM can be seen in Figure 5, and a factorial HSMM or factorial HDP-HSMM simply replaces the hidden state chains with semi-Markov chains. Each chain, indexed by superscripts, evolves with independent dynamics and produces independent emissions, but the observations are combinations of the independent emissions. Note that each component HSMM is not restricted to any fixed number of states.

Such factorial models are natural ways to frame source separation or disaggregation problems, which require identifying component emissions and component states. With the Bayesian framework, we also model uncertainty and ambiguity in such a separation. In Section 5.2 we demonstrate the use of a factorial HDP-HSMM for the task of disaggregating home power signals.

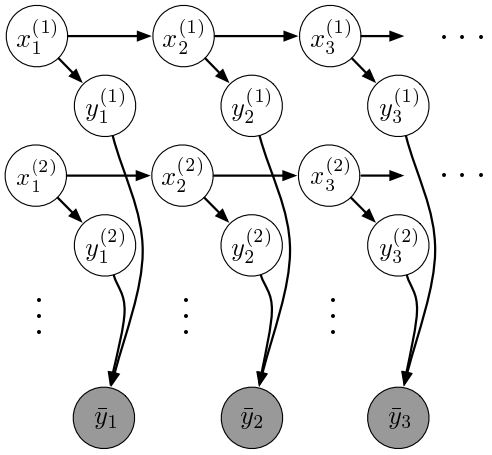


Figure 5: A graphical model for the factorial HMM, which can naturally be extended to factorial structures involving the HSMM or HDP-HSMM.

Problems in source separation or disaggregation are often ill-conditioned, and so one relies on prior information in addition to the source independence structure to solve the separation problem. Furthermore, representation of uncertainty is often important, since there may be several good explanations for the data. These considerations motivate Bayesian inference as well as direct modeling of state duration statistics.

4. Inference Algorithms

We describe three Gibbs sampling inference algorithms, beginning with a sampling algorithm for the finite Bayesian HSMM, which is built upon in developing algorithms for the HDP-HSMM in the sequel. Next, we develop a weak-limit Gibbs sampling algorithm for the HDP-HSMM, which parallels the popular weak-limit sampler for the HDP-HMM and its sticky extension. Finally, we introduce a collapsed sampler which parallels the direct assignment sampler of Teh et al. (2006). For all both of the HDP-HSMM samplers there is a loss of conjugacy with the HDP prior due to the fact that self-transitions in the super-state sequence are not permitted (see Section 4.2.1). We develop auxiliary variables to form an augmented representation that effectively recovers conjugacy and hence enables fast Gibbs steps.

In comparing the weak limit and direct assignment sampler, the most important trade-offs are that the direct assignment sampler works with the infinite model by integrating out the transition matrix π while simplifying bookkeeping by maintaining part of β; it also collapses the observation and duration parameters. However, the variables in the label sequence (x_t) are coupled by the integration, and hence each element of the label sequence must be resampled sequentially. In contrast, the weak limit sampler represents all latent components of the model (up to an adjustable finite approximation for the HDP) and thus allows block resampling of the label sequence by exploiting HSMM message passing.

We end the section with a discussion of leveraging changepoint side-information to greatly accelerate inference.

4.1 A Gibbs Sampler for the Finite Bayesian HSMM

In this section, we describe a blocked Gibbs sampler for the finite HSMM using standard priors.

4.1.1 OUTLINE OF GIBBS SAMPLER

To perform posterior inference in a finite Bayesian HSMM, we construct a Gibbs sampler resembling that for finite HMMs. Our goal is to construct samples from the posterior

$$p((x_t), \{\theta_i\}, \{\pi_i\}, \{\omega_i\} | (y_t), H, G, \alpha)$$

by drawing samples from the distribution, where G represents the prior over duration parameters. We can construct these samples by following a Gibbs sampling algorithm in which we iteratively sample from the appropriate conditional distributions of (x_t), $\{\pi_i\}$, $\{\omega_i\}$, and $\{\theta_i\}$.

Sampling $\{\theta_i\}$ or $\{\omega_i\}$ from their respective conditional distributions can be easily reduced to standard problems depending on the particular priors chosen. Sampling the transition matrix rows $\{\pi_i\}$ is straightforward if the prior on each row is Dirichlet over the off-diagonal entries and so we do not discuss it in this section, but we note that when the rows are tied together hierarchically (as in the weak-limit approximation to the HDP-HSMM), resampling the $\{\pi_i\}$ correctly requires particular care (see Section 4.2.1).

Sampling $(x_t)|\{\theta_i\}, \{\pi_i\}, (y_t)$ in a finite Bayesian Hidden semi-Markov Model was first introduced in Johnson and Willsky (2010) and, in independent work, later in Dewar et al. (2012). In the following section we develop the algorithm for block-sampling the state sequence (x_t) from its conditional distribution by employing the HSMM message-passing scheme.

4.1.2 BLOCKED CONDITIONAL SAMPLING OF (x_t) WITH MESSAGE PASSING

To block sample $(x_t)|\{\theta_i\}, \{\pi_i\}, \{\omega_i\}, (y_t)$ in an HSMM we can extend the standard block state sampling scheme for an HMM. The key challenge is that to block sample the states in an HSMM we must also be able to sample the posterior duration variables.

If we compute the backwards messages B and B^* described in Section 2.2, then we can easily draw a posterior sample for the first state according to

$$\begin{aligned} p(x_1 = k | y_{1:T}) &\propto p(x_1 = k) p(y_{1:T} | x_1 = k, F_0 = 1) \\ &= p(x_1 = k) B_0^*(k), \end{aligned}$$

where we have used the assumption that the observation sequence begins on a segment boundary ($F_0 = 1$) and suppressed notation for conditioning on parameters.

We can also use the messages to efficiently draw a sample from the posterior duration distribution for the sampled initial state. Conditioning on the initial state draw, $\bar{x}_1$, the posterior duration of the first state is:

$$\begin{aligned} p(D_1 = d | y_{1:T}, x_1 = \bar{x}_1, F_0 = 1) &= \frac{p(D_1 = d, y_{1:T} | x_1 = \bar{x}_1, F_0)}{p(y_{1:T} | x_1 = \bar{x}_1, F_0)} \\ &= \frac{p(D_1 = d | x_1 = \bar{x}_1, F_0) p(y_{1:d} | D_1 = d, x_1 = \bar{x}_1, F_0) p(y_{d+1:T} | D_1 = d, x_1 = \bar{x}_1, F_0)}{p(y_{1:T} | x_1 = \bar{x}_1, F_0)} \\ &= \frac{p(D_1 = d) p(y_{1:d} | D_1 = d, x_1 = \bar{x}_1, F_0 = 1) B_d(\bar{x}_1)}{B_0^*(\bar{x}_1)}. \end{aligned}$$

We repeat the process by using x_{D_1+1} as our new initial state with initial distribution $p(x_{D_1+1} = i|x_1 = \bar{x}_1)$, and thus draw a block sample for the entire label sequence.

4.2 A Weak-Limit Gibbs Sampler for the HDP-HSMM

The weak-limit sampler for an HDP-HMM (Fox et al., 2008) constructs a finite approximation to the HDP transitions prior with finite L-dimensional Dirichlet distributions, motivated by the fact that the infinite limit of such a construction converges in distribution to a true HDP:

$$\begin{aligned} \beta|\gamma &\sim \text{Dir}(\gamma/L, \ldots, \gamma/L), \\ \pi_i|\alpha, \beta &\sim \text{Dir}(\alpha\beta_1, \ldots, \alpha\beta_L) \qquad\qquad i = 1, \ldots, L, \end{aligned}$$

where we again interpret π_i as the transition distribution for state i and β as the distribution which ties state distributions together and encourages shared sparsity. Practically, the weak limit approximation enables the complete representation of the transition matrix in a finite form, and thus, when we also represent all parameters, allows block sampling of the entire label sequence at once, resulting in greatly accelerated mixing in many circumstances. The parameter L gives us control over the approximation, with the guarantee that the approximation will become exact as L grows; see Ishwaran and Zarepour (2000), especially Theorem 1, for a discussion of theoretical guarantees. Note that the weak limit approximation is more convenient for us than the truncated stick-breaking approximation because it directly models the state transition probabilities, while stick lengths in the HDP do not directly represent state transition probabilities because multiple sticks in constructing π_i can be sampled at the same atom of β.

We can employ the weak limit approximation to create a finite HSMM that approximates inference in the HDP-HSMM. This approximation technique often results in greatly accelerated mixing, and hence it is the technique we employ for the experiments in the sequel. However, the inference algorithm of Section 4.1 must be modified to incorporate the fact that the $\{\pi_i\}$ are no longer mutually independent and are instead tied through the shared β. This dependence between the transition rows introduces potential conjugacy issues with the hierarchical Dirichlet prior; the following section explains the difficulty as well as a clean solution via auxiliary variables.

The beam sampling technique (Van Gael et al., 2008) can be applied here with little modification, as in Dewar et al. (2012), to sample over the approximation parameter L, thus avoiding the need to set L a priori while still allowing instantiation of the transition matrix and block sampling of the state sequence. This technique is especially useful if the number of states could be very large and is difficult to bound a priori. We do not explore beam sampling here.

4.2.1 CONDITIONAL SAMPLING OF $\{\pi_i\}$ WITH DATA AUGMENTATION

To construct our overall Gibbs sampler, we need to be able to easily resample the transition matrix π given the other components of the model. However, by ruling out self-transitions while maintaining a hierarchical link between the transition rows, the model is no longer fully conjugate, and hence resampling is not necessarily easy. To observe the loss of conjugacy using the hierarchical prior required in the weak-limit approximation, note that we can summarize the relevant portion of the

generative model as

$$\begin{aligned} \beta|\gamma &\sim \text{Dir}(\gamma,\ldots,\gamma), \\ \pi_j|\beta &\sim \text{Dir}(\alpha\beta_1,\ldots,\alpha\beta_L) && j=1,\ldots,L, \\ x_t|\{\pi_j\},x_{t-1} &\sim \bar{\pi}_{x_{t-1}} && t=2,\ldots,T, \end{aligned}$$

where $\bar{\pi}_j$ represents π_j with the jth component removed and renormalized appropriately:

$$\bar{\pi}_{ji} = \frac{\pi_{ji}(1-\delta_{ij})}{1-\pi_{jj}}$$

with $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ otherwise. The deterministic transformation from π_j to $\bar{\pi}_j$ eliminates self-transitions. Note that we have suppressed the observation parameter set, duration parameter set, and observation sequence sampling for simplicity.

Consider the distribution of $\pi_1|(x_t),\beta$, the first row of the transition matrix:

$$\begin{aligned} p(\pi_1|(x_t),\beta) &\propto p(\pi_1|\beta)p((x_t)|\pi_1) \\ &\propto \pi_{11}^{\alpha\beta_1-1}\pi_{12}^{\alpha\beta_2-1}\cdots\pi_{1L}^{\alpha\beta_L-1}\left(\frac{\pi_{12}}{1-\pi_{11}}\right)^{n_{12}}\left(\frac{\pi_{13}}{1-\pi_{11}}\right)^{n_{13}}\cdots\left(\frac{\pi_{1L}}{1-\pi_{11}}\right)^{n_{1L}}, \end{aligned}$$

where n_{ij} are the number of transitions from state i to state j in the state sequence (x_t). Essentially, because of the extra $\frac{1}{1-\pi_{11}}$ terms from the likelihood without self-transitions, we cannot reduce this expression to the Dirichlet form over the components of π_1, and therefore we cannot proceed with sampling m and resampling β and π as in Teh et al. (2006).

However, we can introduce auxiliary variables to recover conjugacy, following the general data augmentation technique described in Van Dyk and Meng (2001). We define an extended generative model with extra random variables, and then show through simple manipulations that conditional distributions simplify with the additional variables, hence allowing us to cycle simple Gibbs updates to produce a sampler.

For simplicity, we focus on the first row of the transition matrix, namely π_1, and the draws that depend on it; the reasoning easily extends to the other rows. We also drop the parameter α for convenience. First, we write the relevant portion of the generative process as

$$\begin{aligned} &\pi_1|\beta \sim \text{Dir}(\beta), \\ &z_i|\bar{\pi}_1 \sim \bar{\pi}_1 \quad i=1,\ldots,n, \\ &y_i|z_i \sim f(z_i) \quad i=1,\ldots,n. \end{aligned}$$

Here, n counts the total number of transitions out of state 1 and the $\{z_i\}$ represent the transitions out of state 1 to a specific state: sampling $z_i=k$ represents a transition from state 1 to state k. The $\{y_i\}$ represent the observations on which we condition; in particular, if we have $z_i=k$ then the y_i corresponds to an emission from state k in the HSMM. See the graphical model in Figure 6(a) for a depiction of the relationship between the variables.

We can introduce auxiliary variables $\{\rho_i\}_{i=1}^n$, where each ρ_i is independently drawn from a geometric distribution supported on $\{0,1,\ldots\}$ with success parameter $1-\pi_{11}$: $\rho_i|\pi_{11} \sim \text{Geo}(1-$

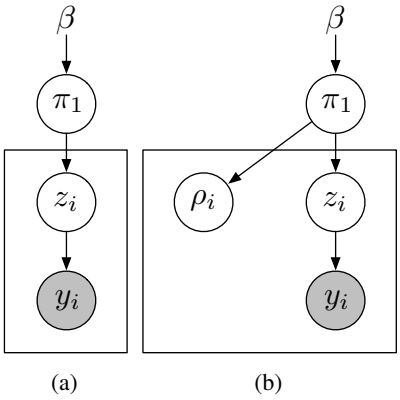


Figure 6: Simplified depiction of the relationship between the auxiliary variables and the rest of the model; 6(a) depicts the nonconjugate setting and 6(b) shows the introduced auxiliary variables $\{\rho_i\}$.

π_{11}) (See Figure 6(b)). Thus our posterior becomes:

$$\begin{aligned}
p(\pi_1|\{z_i\},\{\rho_i\}) &\propto p(\pi_1)p(\{z_i\}|\pi_1)p(\{\rho_i\}|\{\pi_{1i}\}) \\
&\propto \pi_{11}^{\beta_1-1}\pi_{12}^{\beta_2-1}\cdots\pi_{1L}^{\beta_L-1}\left(\frac{\pi_{12}}{1-\pi_{11}}\right)^{n_2}\cdots\left(\frac{\pi_{1L}}{1-\pi_{11}}\right)^{n_L}\left(\prod_{i=1}^{n}\pi_{11}^{\rho_i}(1-\pi_{11})\right) \\
&= \pi_{11}^{\beta_1+\sum_i\rho_i-1}\pi_{12}^{\beta_2+n_2-1}\cdots\pi_{1L}^{\beta_L+n_L-1} \\
&\propto \text{Dir}\left(\beta_1+\sum_i\rho_i,\beta_2+n_2,\ldots,\beta_L+n_L\right).
\end{aligned}$$

Noting that $n=\sum_i n_i$, we recover conjugacy and hence can iterate simple Gibbs steps.

We can compare the numerical performance of the auxiliary variable sampler to a Metropolis-Hastings sampler in the simplified model. For a detailed evaluation, see Johnson and Willsky (2012); in deference to space considerations, we only reproduce two figures from that report here. Figure 7 shows the sample chain autocorrelations for the first component of π in both samplers. Figure 8 compares the Multivariate Scale Reduction Factors of Brooks and Gelman (1998) for the two samplers, where good mixing is indicated by achieving the statistic's asymptotic value of unity.

We can easily extend the data augmentation to the full HSMM, and once we have augmented the data with the auxiliary variables $\{\rho_s\}$ we are once again in the conjugate setting. A graphical model for the weak-limit approximation to the HDP-HSMM including the auxiliary variables is shown in Figure 9.

For a more detailed derivation as well as further numerical experiments, see Johnson and Willsky (2012).

4.3 A Direct Assignment Sampler for the HDP-HSMM

Though all the experiments in this paper are performed with the weak-limit sampler, we provide a direct assignment (DA) sampler as well for theoretical completeness and because it may be useful

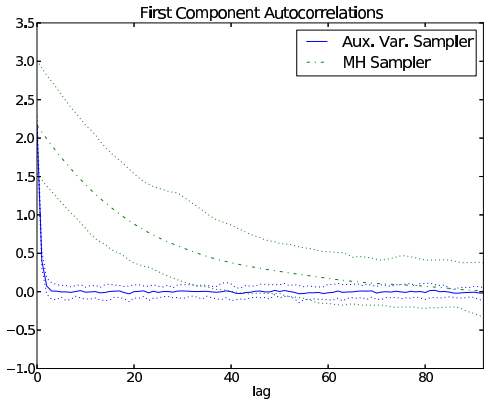
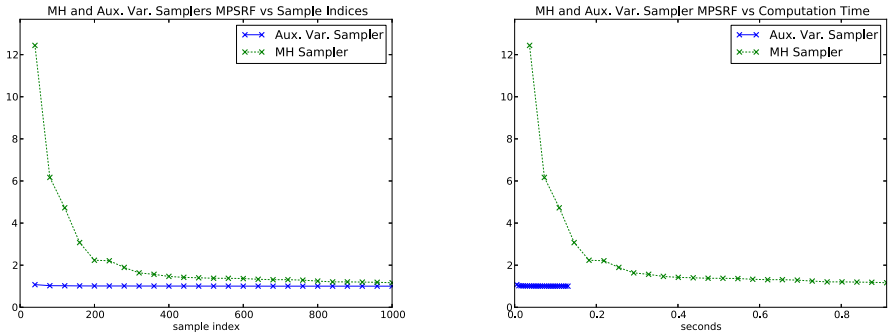


Figure 7: Empirical sample chain autocorrelation for the first component of π for both the proposed auxiliary variable sampler and a Metropolis-Hastings sampler. The rapidly diminishing autocorrelation for the auxiliary variable sampler is indicative of fast mixing.



(a) (b)

Figure 8: Multivariate Potential Scale Reduction Factors for both the proposed auxiliary variable sampler and a Metropolis-Hastings sampler. The auxiliary variable sampler rapidly achieves the statistic's asymptotic value of unity. Note that the auxiliary variable sampler is also much more efficient to execute, as shown in 8(b).

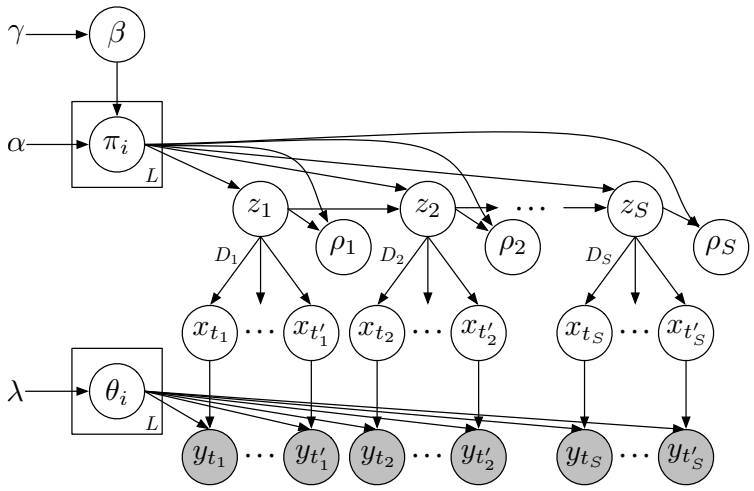


Figure 9: Graphical model for the weak-limit approximation including auxiliary variables.

in cases where there is insufficient data to inform some latent parameters so that marginalization is necessary for mixing or estimating marginal likelihoods (such as in some topic models). As mentioned previously, in the direct assignment sampler for the HDP-HMM the infinite transition matrix π is analytically marginalized out along with the observation parameters (if conjugate priors are used). The sampler represents explicit instantiations of the state sequence (x_t) and the "used" prefix of the infinite vector β: $\beta_{1:K}$ where $K = \#\{x_t : t = 1, \ldots, T\}$. There are also auxiliary variables m used to resample β, but for simplicity we do not discuss them here; see Teh et al. (2006) for details.

Our DA sampler additionally represents the auxiliary variables necessary to recover HDP conjugacy (as introduced in the previous section). Note that the requirement for, and correctness of, the auxiliary variables described in the finite setting in Section 4.2.1 immediately extends to the infinite setting as a consequence of the Dirichlet Process's definition in terms of the finite Dirichlet distribution and the Kolmogorov extension theorem (Çinlar, 2010, Chapter 4); for a detailed discussion, see Orbanz (2009). The connection to the finite case can also be seen in the sampling steps of the direct assignment sampler for the HDP-HMM, in which the global weights β over K instantiated components are resampled according to $(\beta_{1:K}, \beta_{\text{rest}})|\alpha, (x_t) \sim \text{Dir}(\alpha + n_1, \ldots, \alpha + n_K, \alpha)$ where n_i is the number of transitions into state i and Dir is the finite Dirichlet distribution.

4.3.1 Resampling (x_t)

As described in Fox (2009), the basic HDP-HMM DA sampling step for each element x_t of the label sequence is to sample a new label k with probability proportional (over k) to

$$p(x_t = k|(x_{\setminus t}), \beta) \propto \underbrace{\frac{\alpha\beta_k + n_{x_{t-1},k}}{\alpha + n_{x_{t-1},\cdot}}}_{\text{left-transition}} \cdot \underbrace{\frac{\alpha\beta_{x_{t+1}} + n_{k,x_{t+1}} + \mathbf{1}[x_{t-1} = k = x_{t+1}]}{\alpha + n_{k,\cdot} + \mathbf{1}[x_{t-1} = k]}}_{\text{right-transition}} \cdot \underbrace{f_{\text{obs}}(y_t|x_t = k)}_{\text{observation}}$$

for $k = 1, \ldots, K+1$ where $K = \#\{x_t : t = 1, \ldots, T\}$ and where $\mathbf{1}$ is an indicator function taking value 1 if its argument condition is true and 0 otherwise.[2] The variables n_{ij} are transition counts in the portion of the state sequence we are conditioning on; that is, $n_{ij} = \#\{x_\tau = i, x_{\tau+1} = j : \tau \in \{1, \ldots, T-1\} \setminus \{t-1, t\}\}$. The function f_{obs} is a predictive likelihood:

$$\begin{aligned} f_{\text{obs}}(y_t|k) &:= p(y_t|x_t = k, \{y_\tau : x_\tau = k\}, H) \\ &= \int_{\theta_k} \underbrace{p(y_t|x_t = k, \theta_k)}_{\text{likelihood}} \underbrace{\prod_{\tau: x_\tau = k} p(y_\tau|x_\tau = k, \theta_k)}_{\text{likelihood of data with same label}} \underbrace{p(\theta_k|H)}_{\text{observation parameter prior}} \mathrm{d}\theta_k. \end{aligned}$$

We can derive this step by writing the complete joint probability $p((x_t), (y_t)|\beta, H)$ leveraging exchangeability; this joint probability value is proportional to the desired posterior probability $p(x_t|(x_{\setminus t}), (y_t), \beta, H)$. When we consider each possible assignment $x_t = k$, we can cancel all the terms that are invariant over k, namely all the transition probabilities other than those to and from x_t and all data likelihoods other than that for y_t. However, this cancellation process relies on the fact that for the HDP-HMM there is no distinction between self-transitions and new transitions: the term for each t in the complete posterior simply involves transition scores no matter the labels of x_{t+1} and x_{t-1}. In the HDP-HSMM case, we must consider segments and their durations separately from transitions.

To derive an expression for resampling x_t in the case of the HDP-HSMM, we can similarly consider writing out an expression for the joint probability $p((x_t), (y_t)|\beta, H, G)$, but we notice that as we vary our assignment of x_t over k, the terms in the expression must change: if $x_{t-1} = \bar{k}$ or $x_{t+1} = \bar{k}$, the probability expression includes a segment term for entire contiguous run of label $\bar{k}$. Hence, since we can only cancel terms that are invariant over k, our score expression must include terms for the adjacent segments into which x_t may merge. See Figure 10 for an illustration.

The final expression for the probability of sampling the new value of x_t to be k then consists of between 1 and 3 segment score terms, depending on merges with adjacent segments, each of which has the form

$$\begin{aligned} p(x_t = k|(x_{\setminus t}), \beta, H, G) \propto \underbrace{\frac{\alpha\beta_k + n_{x_{\text{prev}},k}}{\alpha(1-\beta_{x_{\text{prev}}}) + n_{x_{\text{prev}},\cdot}}}_{\text{left-transition}} \cdot \underbrace{\frac{\alpha\beta_{x_{\text{next}}} + n_{k,x_{\text{next}}}}{\alpha(1-\beta_k) + n_{k,\cdot}}}_{\text{right-transition}} \\ \cdot \underbrace{f_{\text{dur}}(t^2 - t^1 + 1)}_{\text{duration}} \cdot \underbrace{f_{\text{obs}}(y_{t^1:t^2}|k)}_{\text{observation}}, \end{aligned}$$

where we have used t^1 and t^2 to denote the first and last indices of the segment, respectively. Transition scores at the start and end of the chain are not included.

The function $f_{\text{dur}}(d|k)$ is the corresponding duration predictive likelihood evaluated on a duration d, which depends on the durations of other segments with label k and any duration hyperparameters. The function f_{obs} now represents a *block* or *joint* predictive likelihood over all the data in a segment (see, for example, Murphy (2007) for a thorough discussion of the Gaussian case). Note that the denominators in the transition terms are affected by the elimination of self-transitions by a rescaling of the "total mass." The resulting chain is ergodic if the duration predictive score f_{dur} has a support that includes $\{1, 2, \ldots, d_{\text{max}}\}$, so that segments can be split and merged in any combination.

2. The indicator variables are present because the two transition probabilities are not independent but rather exchangeable.

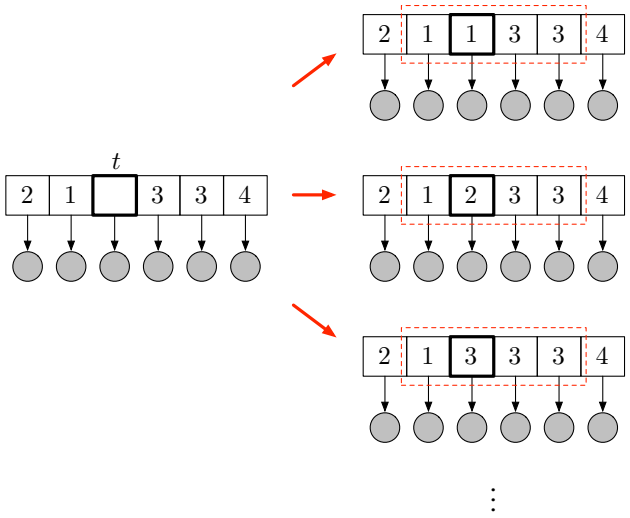


Figure 10: Illustration of the Gibbs step to resample x_t for the DA sampler for the HDP-HSMM. The red dashed boxes indicate the elements of the label sequence that contribute to the score computation for $k = 1, 2, 3$ which produce two, three, and two segment terms, respectively. The label sequence element being resample is emphasized in bold.

4.3.2 Resampling β and Auxiliary Variables ρ

To allow conjugate resampling of β, auxiliary variables must be introduced to deal with the conjugacy issue raised in Section 4.2. In the direct assignment samplers, the auxiliary variables are not used to resample diagonal entries of the transition matrix π, which is marginalized out, but rather to directly resample β. In particular, with each segment s we associate an auxiliary count ρ_s which is independent of the data and only serves to preserve conjugacy in the HDP. We periodically re-sample via

$$\pi_{ii}|\alpha, \beta \sim \text{Beta}(\alpha\beta_i, \alpha(1-\beta_i)),$$
$$\rho_s|\pi_{ii}, z_s \sim \text{Geo}(1-\pi_{z_s,z_s}).$$

The count $n_{i,i}$, which is used in resampling the auxiliary variables m of Teh et al. (2006) which in turn are then used to resample β, is the total of the auxiliary variables for other segments with the same label: $n_{i,i} = \sum_{\bar{s}\neq s, z_{\bar{s}}=z_s} \rho_{\bar{s}}$. This formula can be interpreted as simply sampling the number of self-transitions we may have seen at segment s given β and the counts of self- and non-self transitions in the super-state sequence. Note π_{ii} is independent of the data given (z_s); as before, this auxiliary variable procedure is a convenient way to integrate out numerically the diagonal entries of the transition matrix.

By using the total auxiliary as the statistics for $n_{i,i}$, we can resample $\beta|(x_t), \alpha, \gamma$ according to the procedure for the HDP-HMM as described in Teh et al. (2006).

4.4 Exploiting Changepoint Side-Information

In many circumstances, we may not need to consider all time indices as possible changepoints at which the super-state may switch; it may be easy to rule out many non-changepoints from consid-

eration. For example, in the power disaggregation application in Section 5, we can run inexpensive changepoint detection on the observations to get a list of *possible* changepoints, ruling out many obvious non-changepoints. The possible changepoints divide the label sequence into state *blocks*, where within each block the label sequence must be constant, though sequential blocks may have the same label. By only allowing super-state switching to occur at these detected changepoints, we can greatly reduce the computation of all the samplers considered.

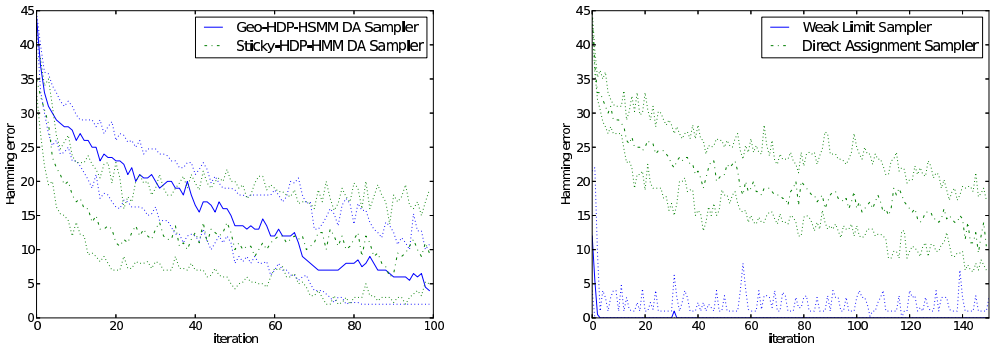
In the case of the weak-limit sampler, the complexity of the bottleneck message-passing step is reduced to a function of the number of possible changepoints (instead of total sequence length): the asymptotic complexity becomes $\mathcal{O}(T_{\text{change}}^2 N + N^2 T_{\text{change}})$, where T_{change}, the number of possible changepoints, may be dramatically smaller than the sequence length T. We simply modify the backwards message-passing procedure to sum only over the possible durations:

$$\begin{aligned} B_t^*(i) &:= p(y_{t+1:T}|x_{t+1}=i, F_t=1) \\ &= \sum_{d\in\mathbb{D}} B_{t+d}(i) \underbrace{\tilde{p}(D_{t+1}=d|x_{t+1}=i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1}=i, D_{t+1}=d)}_{\text{likelihood term}} \\ &\quad + \underbrace{\tilde{p}(D_{t+1}>T-t|x_{t+1}=i) p(y_{t+1:T}|x_{t+1}=i, D_{t+1}>T-t)}_{\text{censoring term}}, \end{aligned}$$

where $\tilde{p}$ represents the duration distribution restricted to the set of possible durations $\mathbb{D} \subset \mathbb{N}^+$ and re-normalized. We similarly modify the forward-sampling procedure to only consider possible durations. It is also clear how to adapt the DA sampler: instead of re-sampling each element of the label sequence (x_t) we simply consider the block label sequence, resampling each block's label (allowing merging with adjacent blocks).

5. Experiments

In this section, we evaluate the proposed HDP-HSMM sampling algorithms on both synthetic and real data. First, we compare the HDP-HSMM direct assignment sampler to the weak limit sampler as well as the Sticky HDP-HMM direct assignment sampler, showing that the HDP-HSMM direct assignment sampler has similar performance to that for the Sticky HDP-HMM and that the weak limit sampler is much faster. Next, we evaluate the HDP-HSMM weak limit sampler on synthetic data generated from finite HSMMs and HMMs. We show that the HDP-HSMM applied to HSMM data can efficiently learn the correct model, including the correct number of states and state labels, while the HDP-HMM is unable to capture non-geometric duration statistics. We also apply the HDP-HSMM to data generated by an HMM and demonstrate that, when equipped with a duration distribution class that includes geometric durations, the HDP-HSMM can also efficiently learn an HMM model when appropriate with little loss in efficiency. Next, we use the HDP-HSMM in a factorial (Ghahramani and Jordan, 1997) structure for the purpose of disaggregating a whole-home power signal into the power draws of individual devices. We show that encoding simple duration prior information when modeling individual devices can greatly improve performance, and further that a Bayesian treatment of the parameters is advantageous. We also demonstrate how changepoint side-information can be leveraged to significantly speed up computation. The Python code used to perform these experiments as well as Matlab code is available online at `http://github.com/mattjj/pyhsmm`.



(a) (b)

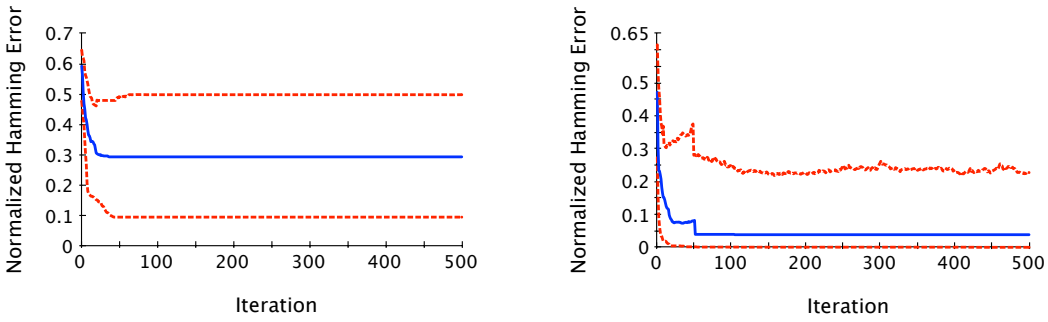
Figure 11: 11(a) compares the Geometric-HDP-HSMM direct assignment sampler with that of the Sticky HDP-HMM, both applied to HMM data. The sticky parameter κ was chosen to maximize mixing. 11(b) compares the HDP-HSMM direct assignment sampler with the weak limit sampler. In all plots, solid lines are the median error at each time over 25 independent chains; dashed lines are 25th and 75th percentile errors.

5.1 Synthetic Data

Figure 11 compares the HDP-HSMM direct assignment sampler to that of the Sticky HDP-HMM as well as the HDP-HSMM weak limit sampler. Figure 11(a) shows that the direct assignment sampler for a Geometric-HDP-HSMM performs similarly to the Sticky HDP-HSMM direct assignment sampler when applied to data generated by an HMM with scalar Gaussian emissions. Figures 11(b) shows that the weak limit sampler mixes much more quickly than the direct assignment sampler. Each iteration of the weak limit sampler is also much faster to execute (approximately 50x faster in our implementations in Python). Due to its much greater efficiency, we focus on the weak limit sampler for the rest of this section; we believe it is a superior inference algorithm whenever an adequately large approximation parameter L can be chosen a priori.

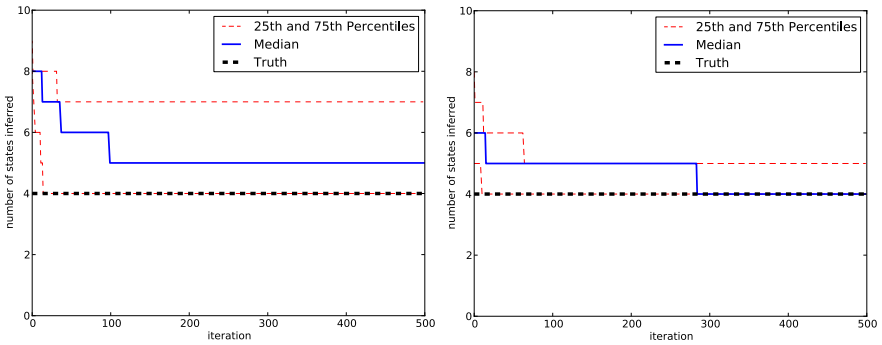
Figure 12 summarizes the results of applying both a Poisson-HDP-HSMM and an HDP-HMM to data generated from an HSMM with four states, Poisson durations, and 2-dimensional mixture-of-Gaussian emissions. In the 25 Gibbs sampling runs for each model, we applied 5 chains to each of 5 generated observation sequences. The HDP-HMM is unable to capture the non-Markovian duration statistics and so its state sampling error remains high, while the HDP-HSMM equipped with Poisson duration distributions is able to effectively learn the correct temporal model, including duration, transition, and emission parameters, and thus effectively separate the states and significantly reduce posterior uncertainty. The HDP-HMM also frequently fails to identify the true number of states, while the posterior samples for the HDP-HSMM concentrate on the true number; see Figure 13.

By setting the class of duration distributions to be a superclass of the class of geometric distributions, we can allow an HDP-HSMM model to learn an HMM from data when appropriate. One such distribution class is the class of negative binomial distributions, denoted $\text{NegBin}(r,p)$, the discrete analog of the Gamma distribution, which covers the class of geometric distributions when $r = 1$. By placing a (non-conjugate) prior over r that includes $r = 1$ in its support, we allow the



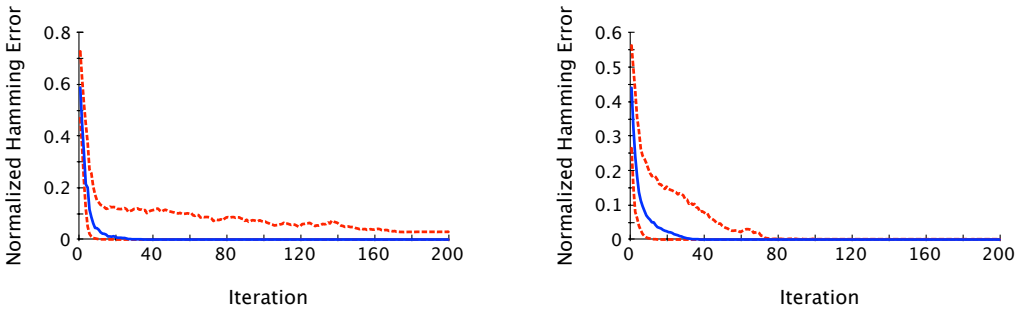
(a) HDP-HMM (b) HDP-HSMM

Figure 12: State-sequence Hamming error of the HDP-HMM and Poisson-HDP-HSMM applied to data from a Poisson-HSMM. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed lines indicate the chains with the 10th and 90th percentile errors at each iteration. The jumps in the plot correspond to a change in the ranking of the 25 chains.



(a) HDP-HMM (b) HDP-HSMM

Figure 13: Number of states inferred by the HDP-HMM and Poisson-HDP-HSMM applied to data from a four-state Poisson-HSMM. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed lines indicate the chains with the 10th and 90th percentile errors at each iteration.



(a) HDP-HMM (b) HDP-HSMM

Figure 14: The HDP-HSMM and HDP-HMM applied to data from an HMM. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed line indicates the chains with the 10th and 90th percentile error at each iteration.

model to learn geometric durations as well as significantly non-geometric distributions with modes away from zero. Figure 14 shows a negative binomial HDP-HSMM learning an HMM model from data generated from an HMM with four states. The observation distribution for each state is a 10-dimensional Gaussian, again with parameters sampled i.i.d. from a NIW prior. The prior over r was set to be uniform on $\{1, 2, \ldots, 6\}$, and all other priors were chosen to be similarly non-informative. The sampler chains quickly concentrated at $r = 1$ for all state duration distributions. There is only a slight loss in mixing time for the HDP-HSMM compared to the HDP-HMM. This experiment demonstrates that with the appropriate choice of duration distribution the HDP-HSMM can effectively learn an HMM model.

5.2 Power Disaggregation

In this section we show an application of the HDP-HSMM factorial structure to an unsupervised power signal disaggregation problem. The task is to estimate the power draw from individual devices, such as refrigerators and microwaves, given an aggregated whole-home power consumption signal. This disaggregation problem is important for energy efficiency: providing consumers with detailed power use information at the device level has been shown to improve efficiency significantly, and by solving the disaggregation problem one can provide that feedback without instrumenting every individual device with monitoring equipment. This application demonstrates the utility of including duration information in priors as well as the significant speedup achieved with changepoint-based inference.

The power disaggregation problem has a rich history (Zeifman and Roth, 2011) with many proposed approaches for a variety of problem specifications. Some recent work (Kim et al., 2010) has considered applying factorial HSMMs to the disaggregation problem using an EM algorithm; our work here is distinct in that (1) we do not use training data to learn device models but instead rely on simple prior information and learn the model details during inference, (2) our states are not restricted to binary values and can model multiple different power modes per device, and (3) we use Gibbs sampling to learn all levels of the model. The work in Kim et al. (2010) also explores many

other aspects of the problem, such as additional data features, and builds a very compelling complete solution to the disaggregation problem, while we focus on the factorial time series modeling itself.

For our experiments, we used the REDD data set (Kolter and Johnson, 2011), which monitors many homes at high frequency and for extended periods of time. We chose the top 5 power-drawing devices (refrigerator, lighting, dishwasher, microwave, furnace) across several houses and identified 18 24-hour segments across 4 houses for which many (but not always all) of the devices switched on at least once. We applied a 20-second median filter to the data, and each sequence is approximately 5000 samples long.

We constructed simple priors that set the rough power draw levels and duration statistics of the modes for several devices. For example, the power draw from home lighting changes infrequently and can have many different levels, so an HDP-HSMM with a bias towards longer negative-binomial durations is appropriate. On the other hand, a refrigerator's power draw cycle is very regular and usually exhibits only three modes, so our priors biased the refrigerator HDP-HSMM to have fewer modes and set the power levels accordingly. For details on our prior specification, see Appendix A. We did not truncate the duration distributions during inference, and we set the weak limit approximation parameter L to be twice the number of expected modes for each device; for example, for the refrigerator device we set $L = 6$ and for lighting we set $L = 20$. We performed sampling inference independently on each observation sequence.

As a baseline for comparison, we also constructed a factorial sticky HDP-HMM (Fox et al., 2008) with the same observation priors and with duration biases that induced the same average mode durations as the corresponding HDP-HSMM priors. We also compare to the factorial HMM performance presented in Kolter and Johnson (2011), which fit device models using an EM algorithm on training data. For the Bayesian models, we performed inference separately on each aggregate data signal.

The set of possible changepoints is easily identifiable in these data, and a primary task of the model is to organize the jumps observed in the observations into an explanation in terms of the individual device models. By simply computing first differences and thresholding, we are able to reduce the number of potential changepoints we need to consider from 5000 to 100-200, and hence we are able to speed up state sequence resampling by orders of magnitude. See Figure 15 for an illustration.

To measure performance, we used the error metric of Kolter and Johnson (2011):

$$\text{Acc.} = 1 - \frac{\sum_{t=1}^{T} \sum_{i=1}^{K} \left| \hat{y}_t^{(i)} - y_t^{(i)} \right|}{2 \sum_{t=1}^{T} \bar{y}_t}$$

where $\bar{y}_t$ refers to the observed total power consumption at time t, $y_t^{(i)}$ is the true power consumed at time t by device i, and $\hat{y}_t^{(i)}$ is the estimated power consumption. We produced 20 posterior samples for each model and report the median accuracy of the component emission means compared to the ground truth provided in REDD. We ran our experiments on standard desktop machines (Intel Core i7-920 CPUs, released Q4 2008), and a sequence with about 200 detected changepoints would resample each component chain in 0.1 seconds, including block sampling the state sequence and resampling all observation, duration, and transition parameters. We collected samples after every 50 such iterations.

Our overall results are summarized in Figure 16 and Table 1. Both Bayesian approaches improved upon the EM-based approach because they allowed flexibility in the device models that

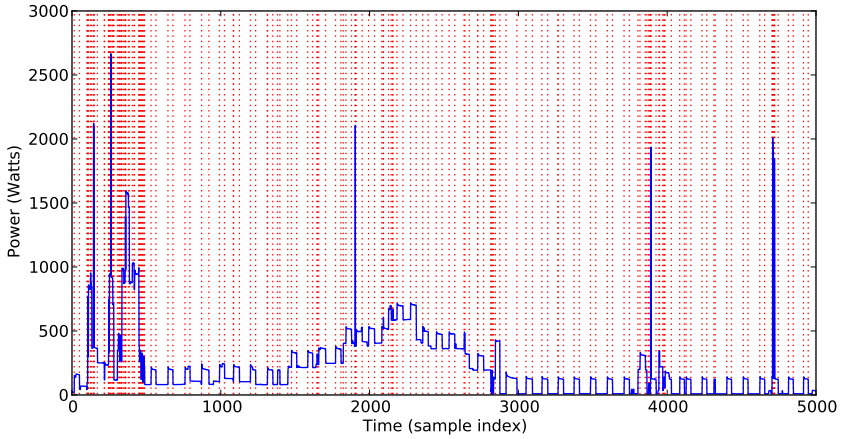


Figure 15: An total power observation sequence from the power disaggregation data set. Vertical dotted red lines indicate changepoints detected with a simple first-differences. By using the changepoint-based algorithms described in Section 4.4 we can greatly accelerate inference speed for this application.

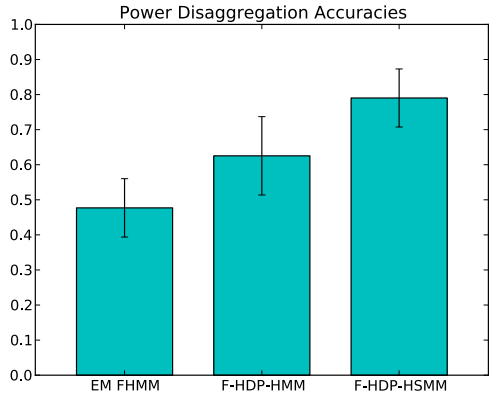


Figure 16: Overall accuracy comparison between the EM-trained FHMM of Kolter and Johnson (2011), the factorial sticky HDP-HMM, and the factorial HDP-HSMM.

could be fit during inference, while the EM-based approach fixed device model parameters that may not be consistent across homes. Furthermore, the incorporation of duration structure and prior information provided a significant performance increase for the HDP-HSMM approach. Detailed performance comparisons between the HDP-HMM and HDP-HSMM approaches can be seen in Figure 17. Finally, Figures 18 and 19 shows total power consumption estimates for the two models on two selected data sequences.

We note that the nonparametric prior was very important for modeling the power consumption due to lighting. Power modes arise from combinations of lights switched on in the user's home, and hence the number of levels that are observed is highly uncertain a priori. For the other devices the number of power modes (and hence states) is not so uncertain, but duration statistics can provide a

House	EM FHMM	F-HDP-HMM	F-HDP-HSMM
1	46.6%	69.0%	82.1%
2	50.8%	70.7%	84.8%
3	33.3%	67.3%	81.5%
6	55.7%	61.8%	77.7%
Mean	47.7%	67.2%	81.5%

Table 1: Performance comparison broken down by house.

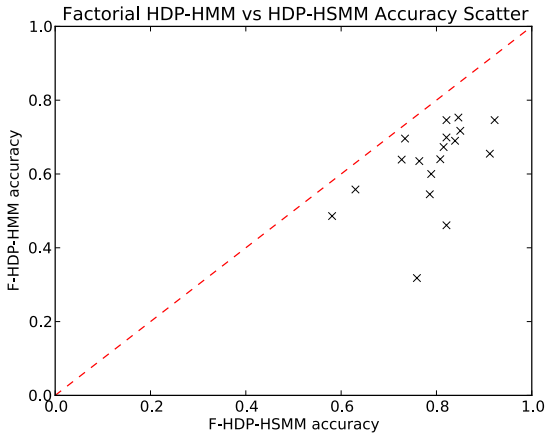
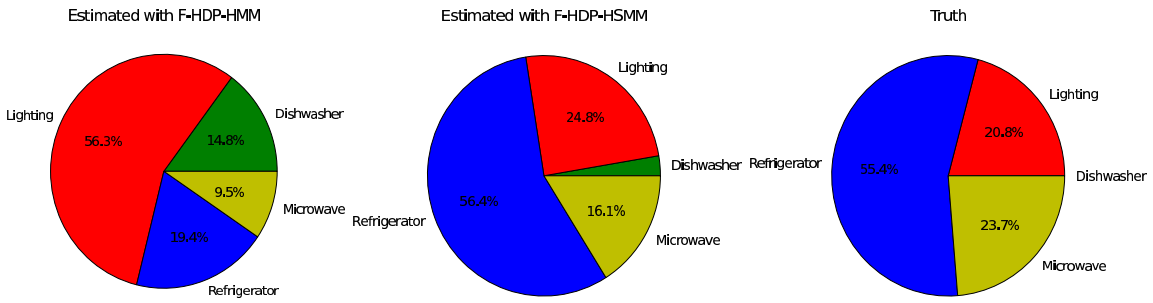
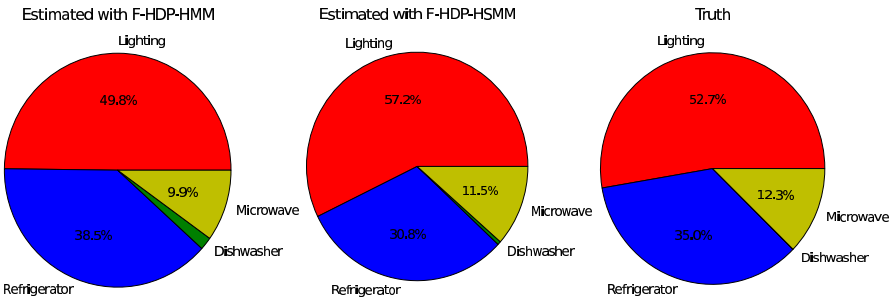


Figure 17: Performance comparison between the HDP-HMM and HDP-HSMM approaches broken down by data sequence.



(a) (b) (c)

Figure 18: Estimated total power consumption for a data sequence where the HDP-HSMM significantly outperformed the HDP-HMM due to its modeling of duration regularities.



(a) (b) (c)

Figure 19: Estimated total power consumption for a data sequence where both the HDP-HMM and HDP-HSMM approaches performed well.

strong clue for disaggregation; for these, the main advantage of our model is in providing Bayesian inference and duration modeling.

6. Conclusion

We have developed the HDP-HSMM and two Gibbs sampling inference algorithms, the weak limit and direct assignment samplers, uniting explicit-duration semi-Markov modeling with new Bayesian nonparametric techniques. These models and algorithms not only allow learning from complex sequential data with non-Markov duration statistics in supervised and unsupervised settings, but also can be used as tools in constructing and performing infernece in larger hierarchical models. We have demonstrated the utility of the HDP-HSMM and the effectiveness of our inference algorithms with real and synthetic experiments, and we believe these methods can be built upon to provide new tools for many sequential learning problems.

Acknowledgments

The authors thank J. Zico Kolter, Emily Fox, and Ruslan Salakhutdinov for invaluable discussions and advice. We also thank the anonymous reviewers for helpful fixes and suggestions. This work was supported in part by a MURI through ARO Grant W911NF-06-1-0076, in part through a MURI through AFOSR Grant FA9550-06-1-303, and in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374.

Appendix A. Power Disaggregation Priors

We used simple hand-set priors for the power disaggregation experiments in Section 5.2, where each prior had two free parameters that were set to encode rough means and variances for each device mode's durations and emissions. To put priors on multiple modes we used atomic mixture models in the priors. For example, refrigerators tend to exhibit an "off" mode near zero Watts, an "on" mode near 100-140 Watts, and a "high" mode near 300-400 Watts; we include each of these regimes in the prior by specifying three sets of hyperparameters, and a state samples observation parameters

by first sampling one of the three sets of hyperparameters uniformly at random and then sampling observation parameters using those hyperparameters

A comprehensive summary of our prior settings for the Factorial HDP-HSMM are in Table 2. Observation distributions were all Gaussian with state-specific latent means and fixed variances. We use Gauss$(\mu_0, \sigma_0^2; \sigma^2)$ to denote a Gaussian observation distribution prior with a fixed variance of σ^2 and a prior over its mean parameter that is Gaussian distributed with mean μ_0 and variance σ_0^2; that is, it denotes that a state's mean parameter μ is sampled according to $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and an observation from that state is sampled from $\mathcal{N}(\mu, \sigma^2)$. Similarly, we use NegBin$(\alpha, \beta; r)$ to denote Negative Binomial duration distribution priors where a latent state-specific "success" parameter p is drawn from $p \sim$ Beta(α, β) and the parameter r is fixed, so that state durations for that state are then drawn from NegBin(p, r). (Note choosing $r = 1$ sets a geometric duration class.)

We set the priors for the Factorial Sticky HDP-HMM by using the same set of observation prior parameters as for the HDP-HSMM and setting state-specific sticky bias parameters so as to match the expected durations encoded in the HDP-HSMM duration priors. For an example of real data observation sequences, see Figure 20.

A natural extension of this model would be a more elaborate hierarchical model which learns the hyperparameter mixtures automatically from training data. As our experiment is meant to emphasize the merits of the HDP-HSMM and sampling inference, we leave this extension to future work.

References

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. *Advances in Neural Information Processing Systems*, 14:577–584, 2002.

S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, pages 434–455, 1998.

E. Çinlar. *Probability and Stochastics*. Springer Verlag, 2010.

M. Dewar, C. Wiggins, and F. Wood. Inference in hidden markov models with explicit state duration distributions. *Signal Processing Letters, IEEE*, (99):1–1, 2012.

E. B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. Ph.D. thesis, MIT, Cambridge, MA, 2009.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the International Conference on Machine Learning*, July 2008.

Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2): 245–273, 1997.

Y. Guédon. Exploring the state sequence space for hidden markov and semi-markov chains. *Computational Statistics and Data Analysis*, 51(5):2379–2409, 2007. ISSN 0167-9473. doi: http://dx.doi.org/10.1016/j.csda.2006.03.015.

K. Hashimoto, Y. Nankaku, and K. Tokuda. A bayesian approach to hidden semi-markov model based speech synthesis. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

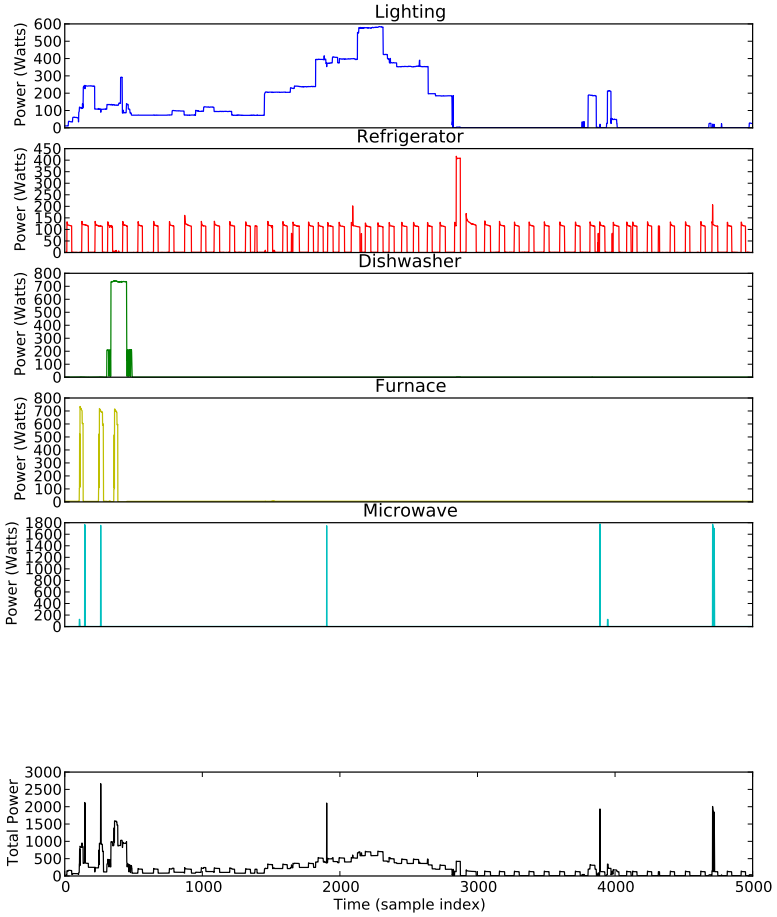


Figure 20: Example real data observation sequences for the power disaggregation experiments.

Device	Base Measures		Specific States	
	Observations	Durations	Observations	Durations
Lighting	$\text{Gauss}(300, 200^2; 5^2)$	$\text{NegBin}(5, 220; 12)$	$\text{Gauss}(0, 1; 5^2)$	$\text{NegBin}(5, 220; 12)$
Refrigerator	$\text{Gauss}(110, 50^2; 10^2)$	$\text{NegBin}(100, 600; 10)$	$\text{Gauss}(0, 1; 5^2)$ $\text{Gauss}(115, 10^2; 10^2)$ $\text{Gauss}(425, 30^2; 10^2)$	$\text{NegBin}(100, 600; 10)$ $\text{NegBin}(100, 600; 10)$ $\text{NegBin}(100, 600; 10)$
Dishwasher	$\text{Gauss}(225, 25^2; 10^2)$	$\text{NegBin}(100, 200; 10)$	$\text{Gauss}(0, 1; 5^2)$ $\text{Gauss}(225, 25^2; 10^2)$ $\text{Gauss}(900, 200^2; 10^2)$	$\text{NegBin}(1, 2000; 1)$ $\text{NegBin}(100, 200; 10)$ $\text{NegBin}(40, 500; 10)$
Furnace	$\text{Gauss}(600, 100^2; 20^2)$	$\text{NegBin}(40, 40; 10)$	$\text{Gauss}(0, 1; 5^2)$	$\text{NegBin}(1, 50; 1)$
Microwave	$\text{Gauss}(1700, 200^2; 50^2)$	$\text{NegBin}(200, 1; 50)$	$\text{Gauss}(0, 1; 5^2)$	$\text{NegBin}(1, 1000; 1)$

Table 2: Power disaggregation prior parameters for each device. Observation priors encode rough power levels that are expected from devices. Duration priors encode duration statistics that are expected from devices.

K. A. Heller, Y. W. Teh, and D. Görür. Infinite hierarchical hidden Markov models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009.

H. Ishwaran and M. Zarepour. Markov chain monte carlo in approximate dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390, 2000.

M. J. Johnson and A. S. Willsky. The Hierarchical Dirichlet Process Hidden Semi-Markov Model. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon, USA, 2010. AUAI Press.

M. J. Johnson and A. S. Willsky. Dirichlet posterior sampling with truncated multinomial likelihoods. 2012. arXiv:1208.6537v2.

H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han. Unsupervised disaggregation of low frequency power measurements. Technical report, HP Labs Tech. Report, 2010.

J. Z. Kolter and M. J. Johnson. REDD: A Public Data Set for Energy Disaggregation Research. In *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.

K. Murphy. Hidden semi-markov models (segment models). *Technical Report*, November 2002. URL `http://www.cs.ubc.ca/~murphyk/Papers/segment.pdf`.

K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, 2007.

P. Orbanz. Construction of nonparametric bayesian models from parametric bayes equations. *Advances in Neural Information Processing Systems*, 2009.

J. Sethuraman. A constructive definition of dirichlet priors. In *Statistica Sinica*, volume 4, pages 639–650, 1994.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

S. E. Tranter and D. A. Reynolds. An overview of automatic speaker diarization systems. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1557–1565, 2006.

D. A. Van Dyk and X. L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1088–1095. ACM, 2008.

S. Z. Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *Consumer Electronics, IEEE Transactions on*, 57(1):76–84, 2011.

Journal of Machine Learning Research 14 (2013) 703-727 Submitted 3/12; Revised 10/12; Published 3/13

Differential Privacy for Functions and Functional Data

Rob Hall RJHALL@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15289, USA

Alessandro Rinaldo ARINALDO@STAT.CMU.EDU
Larry Wasserman LARRY@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15289, USA

Editor: Charles Elkan

Abstract

Differential privacy is a rigorous cryptographically-motivated characterization of data privacy which may be applied when releasing summaries of a database. Previous work has focused mainly on methods for which the output is a finite dimensional vector, or an element of some discrete set. We develop methods for releasing functions while preserving differential privacy. Specifically, we show that adding an appropriate Gaussian process to the function of interest yields differential privacy. When the functions lie in the reproducing kernel Hilbert space (RKHS) generated by the covariance kernel of the Gaussian process, then the correct noise level is established by measuring the "sensitivity" of the function in the RKHS norm. As examples we consider kernel density estimation, kernel support vector machines, and functions in RKHSs.

Keywords: differential privacy, density estimation, Gaussian processes, reproducing kernel Hilbert space

1. Introduction

Suppose we have database D which consists of measurements of a set of individuals. We want to release a summary of D without compromising the privacy of those individuals in the database. One framework for defining privacy rigorously in such problems is *differential privacy* (Dwork et al., 2006b; Dwork, 2006). The basic idea is to produce an output via random noise addition. An algorithm which does this may be thought of as inducing a distribution P_D on the output space (where the randomness is due to internal "coin flips" of the algorithm), for every input data set D. Differential privacy, defined in Section 2, requires that P_D not depend too strongly on any single element of the database D.

The literature on differential privacy is vast. Algorithms that preserve differential privacy have been developed for boosting, parameter estimation, clustering, logistic regression, SVM learning and many other learning tasks. See, for example, Dwork et al. (2010), Chaudhuri and Monteleoni (2008), Smith (2011), Chaudhuri and Monteleoni (2011), Nissim et al. (2007), Kasiviswanathan et al. (2008), Barak et al. (2007), and references therein. In all these cases, the data (both the input and output) are assumed to be real numbers or vectors. In this paper we are concerned with a setting in which the output, and possibly the input data set, consist of functions.

A concept that has been important in differential privacy is the "sensitivity" of the output (Dwork et al., 2006b). In the case of vector valued output the sensitivity is typically measured in the Euclidean norm or the ℓ_1-norm. We find that when the output is a function the sensitivity may be measured in terms of an RKHS norm. To establish privacy a Gaussian process may be added to the function with noise level calibrated to the "RKHS sensitivity" of the output.

The motivation for considering function valued data is two-fold. First, in some problems the data are naturally function valued, that is, each data point is a function. For example, growth curves, temperature profiles, and economic indicators are often of this form. This has given rise to a subfield of statistics known as functional data analysis (see, for instance, Ramsay and Silverman, 1997). Second, even if the data are not functions, we may want to release a data summary that is a function. For example, if the data $d_1, \ldots, d_n \in \mathbb{R}^d$ are a sample from a distribution with density f then we can estimate the density with the kernel density estimator

$$\widehat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} W\left(\frac{||x-d_i||}{h}\right), \quad x \in \mathbb{R}^d,$$

where W is a kernel (see, for instance, Wasserman, 2006) and $h > 0$ is the bandwidth parameter. The density estimator is useful for may tasks such as clustering and classification. We may then want to release a "version" of the density estimator $\widehat{f}$ in a way which fulfills the criteria of differential privacy. The utility of such a procedure goes beyond merely estimating the underlying density. In fact, suppose the goal is to release a privatized database. With a differentially private density estimator in hand, a large sample of data may be drawn from that density. The release of such a sample would inherit the differential privacy properties of the density estimator: see, in particular, Wasserman and Zhou (2010) and Machanavajjhala et al. (2008). This is a very attractive proposition, since a differentially private sample of data could be used as the basis for any number of statistical analyses which may have been brought to bear against the original data (for instance, exploratory data analysis, model fitting, etc).

Histograms are an example of a density estimator that has been "privatized" in previous literature (Wasserman and Zhou, 2010; Chawla et al., 2005). However, as density estimators, histograms are suboptimal because they are not smooth. Specifically, they do converge at the minimax rate under the assumption that the true density is smooth. The preferred method for density estimation in statistics is kernel density estimation. The methods developed in this paper lead to a private kernel density estimator.

In addition to kernel density estimation, there are a myriad of other scenarios in which the result of a statistical analysis is a function. For example, the regression function or classification function from a supervised learning task. We demonstrate how the theory we develop may be applied in these contexts as well.

Since a function over a real valued domain is characterized by an infinite number of points it is not feasible to output the function directly. We give methods which in essence permit the user to request the evaluation of the private function at arbitrarily many input points. These points may be specified a-priori, or after the construction of the function, or even adaptively based on the outputs given, it makes no difference to the privacy guarantee. However we note that in the case when the points are specified a-priori (for example, a grid over the domain) that the release of the function values corresponds to the release of a finite dimensional vector. In this case we may regard this work as providing a conceptually clean way to determine the sensitivity of this vector so that standard finite dimensional privacy techniques may be applied.

Outline. After putting our contribution in the context of some related work, we introduce some notation and review the definition of differential privacy in Section 2. We also give a demonstration of a technique to achieve the differential privacy for a vector valued output. The theory for demonstrating differentially privacy of functions is established in Section 3. In Section 4 we apply the theory to the problems of kernel density estimation and kernel SVM learning. We also demonstrate how the theory may apply to a broad class of functions (a Sobolev space). Section 5 discusses possible algorithms for outputting functions.

1.1 Related Work

There are a few lines of research in the differential privacy literature that are related to this work. In addition to the foundational papers mentioned above there has already been interest in the differentially private release of functions such as support vector machines. Independently Rubinstein et al. (2010) and Chaudhuri and Monteleoni (2011) demonstrated that a private approximation to a non-private kernel support vector machine could be made by considering a certain finite dimensional projection of the original function. In essence they construct a finite dimensional feature space so that the classification function may be characterized by a finite dimensional vector of coefficients, at which point standard techniques may be brought to bear to ensure privacy. Here we give an alternate method for the release of the classification function (or regression function) which avoids this approximation, although we do so by employing a weaker type of differential privacy. Our method in essence results in an infinite dimensional private function which is not necessarily characterized by any finite dimensional vector. Rather we can permit the user to query the value of the function at any arbitrary number of input points.

There has also been research in the literature regarding the generation of synthetic data sets. For example, Barak et al. (2007) and more recently Hardt et al. (2010) give techniques which output a differentially private contingency table. A recent summary of related methods is in Charest (2012). These contingency tables may be subjected to whatever statistical analysis is required, while still maintaining privacy. In the case that the original data is not categorical (for example, containing real valued measurements) there are two immediate options, the first is to divide the range of the variables up into bins and to essentially make the data categorical then to apply these techniques. This is conceptually simple however the resulting density estimator fails to achieve the correct convergence rate under the usual regularity conditions Wasserman (2006) (namely the histogram density estimator achieves the rate of $n^{-2/(2+d)}$ whereas the kernel density estimate achieves the rate $n^{-4/(4+d)}$ in d dimensions). The second approach is to perform some other kind of density estimation on the input data in a way which admits the differential privacy, and then to sample that density to generate synthetic data. So far differentially private density estimation is considered in Smith (2011) for parametric models and Wasserman and Zhou (2010) for non-parametric estimation. In this work we give a technique which is useful for the differentially private estimation of a density in arbitrary dimension, and which achieves the same convergence rate (up to constants) as the non-private estimator.

It is worth noting that the availability of a differentially private synthetic data set allows the recipient to compute whatever function he wishes on the data. For example he may compute his own support vector machine using that data. This may seem to obviate the need for methods to compute private kernel machines and other functions. However we note that the techniques mentioned above for releasing a data set involved computing a density estimator as a first step (be it discrete or

otherwise) and so the quality of the released data (which will ultimately dictate the quality of the learned function) depends on the convergence rate of these estimators—which are typically slow in high dimensions. On the other hand a kernel support vector machine learned on the original data may converge to a good classifier with a relatively small number of samples, and so building a private version of that directly may lead to better classification performance. Therefore although private synthetic data may be seen as a panacea for differential privacy, it is important to remember that it in essence taints all analyses with the curse of dimensionality.

2. Differential Privacy

Here we recall the definition of differential privacy and introduce some notation. Let $D = (d_1, \ldots, d_n) \in \mathcal{D}$ be an input database in which d_i represents a row or an individual, and where $\mathcal{D}$ is the space of all such databases of n elements. For two databases D, D', we say they are "adjacent" or "neighboring" and write $D \sim D'$ whenever both have the same number of elements, but differ in one element. In other words, there exists a permutation of D having Hamming distance of 2 to D'. In some other works databases are called "adjacent" whenever one database contains the other together with exactly one additional element.

We may characterize a non-private algorithm in terms of the function it outputs, for example by, $\theta : \mathcal{D} \to \mathbb{R}^d$. Thus we write $\theta_D = \theta(D)$ to mean the output when the input database is D. Thus, a computer program which outputs a vector may be characterized as a family of vectors $\{\theta_D : D \in \mathcal{D}\}$, one for every possible input database. Likewise a randomized computer program may be characterized by the distributions $\{P_D : D \in \mathcal{D}\}$ it induces on the output space (for example, $\mathbb{R}^d$) when the input is D. In the literature such a set of distributions is sometimes referred to as a "mechanism." We consider randomized algorithms where the input is a database in $\mathcal{D}$ and the output takes values in a measurable space Ω endowed with the σ-field $\mathcal{A}$. Thus, to each such algorithm there correspond the set of distributions $\{P_D : D \in \mathcal{D}\}$ on $(\Omega, \mathcal{A})$ indexed by databases. We phrase the definition of differential privacy using this characterization of randomized algorithms.

Definition 1 (Differential Privacy) *A set of distributions $\{P_D : D \in \mathcal{D}\}$ is called (α, β)-differentially private, or said to "achieve (α, β)-DP" whenever for all $D \sim D' \in \mathcal{D}$ we have:*

$$P_D(A) \leq e^{\alpha} P_{D'}(A) + \beta, \quad \forall A \in \mathcal{A}, \tag{1}$$

where $\alpha, \beta \geq 0$ are parameters, and $\mathcal{A}$ is the finest σ-field on which all P_D are defined.

Typically the above definition is called "approximate differential privacy" whenever $\beta > 0$, and "$(\alpha, 0)$-differential privacy" is shortened to "α-differential privacy." It is important to note that the relation $D \sim D'$ is symmetric, and so the inequality (1) is required to hold when D and D' are swapped. Throughout this paper we take $\alpha \leq 1$, since this simplifies some proofs. We note that an alternate notion of adjacency for databases also appears in some of the differential privacy literature. There databases are called adjacent whenever one is a strict subset of the other and contains exactly one less entry. We remark that the techniques we present can be reformulated under this definition, but we use the above definition of adjacency since it leads to slightly simpler forms for the sensitivity.

The σ-field $\mathcal{A}$ is rarely mentioned in the literature on differential privacy but is actually quite important. For example if we were to take $\mathcal{A} = \{\Omega, \emptyset\}$ then the condition (1) is trivially satisfied by

any randomized algorithm. To make the definition as strong as possible we insist that $\mathcal{A}$ be the finest available σ-field on which the P_D are defined. Therefore when Ω is discrete the typical σ-field is $\mathcal{A} = 2^{\Omega}$ (the class of all subsets of Ω), and when Ω is a space with a topology it is typical to use the completion of the Borel σ-field (the smallest σ-field containing all open sets). We raise this point since when Ω is a space of functions, the choice of σ-field is more delicate.

2.1 Differential Privacy of Finite Dimensional Vectors

Dwork et al. (2006a) give a technique to achieve approximate differential privacy for general vector valued outputs in which the "sensitivity" may be bounded. We review this below, since the result is important in the demonstration of the privacy of our methods which output functions. What follows in this section is a mild alteration to the technique developed by Dwork et al. (2006a) and McSherry and Mironov (2009), in that the "sensitivity" of the class of vectors is measured in the Mahalanobis distance rather than the usual Euclidean distance.

In demonstrating the differential privacy, we make use of the following lemma which is simply an explicit statement of an argument used in a proof by Dwork et al. (2006a).

Lemma 2 *Suppose that, for all $D \sim D'$, there exists a set $A^{\star}_{D,D'} \in \mathcal{A}$ such that, for all $S \in \mathcal{A}$,*

$$S \subseteq A^{\star}_{D,D'} \Rightarrow P_D(S) \leq e^{\alpha} P_{D'}(S) \tag{2}$$

and

$$P_D(A^{\star}_{D,D'}) \geq 1 - \beta. \tag{3}$$

Then the family $\{P_D\}$ achieves the (α,β)-DP.

Proof Let $S \in \mathcal{A}$. Then,

$$\begin{aligned} P_D(S) &= P_D(S \cap A^{\star}) + P_D(S \cap A^{\star C}) \leq P_D(S \cap A^{\star}) + \beta \\ &\leq e^{\alpha} P_{D'}(S \cap A^{\star}) + \beta \leq e^{\alpha} P_{D'}(S) + \beta. \end{aligned}$$

The first inequality is due to (3), the second is due to (2) and the third is due to the subadditivity of measures. ■

The above result shows that, so long as there is a large enough (in terms of the measure P_D) set on which the $(\alpha,0)$-DP condition holds, then the approximate (α,β)-DP is achieved.

Remark 1 *If $(\Omega,\mathcal{A})$ has a σ-finite dominating measure λ, then for (2) to hold a sufficient condition is that the ratio of the densities be bounded on some set $A^{\star}_{D,D'}$:*

$$\forall a \in A^{\star}_{D,D'} : \frac{dP_D}{d\lambda}(a) \leq e^{\alpha} \frac{dP_{D'}}{d\lambda}(a). \tag{4}$$

This follows from the inequality

$$P_D(S) = \int_S \frac{dP_D}{d\lambda}(a)\, d\lambda(a) \leq \int_S e^{\alpha} \frac{dP_{D'}}{d\lambda}(a)\, d\lambda(a) = e^{\alpha} P_{D'}(S).$$

In our next result we show that approximate differential privacy is achieved via (4) when the output is a real vector, say $v_D = v(D) \in \mathbb{R}^d$, whose dimension does not depend on the database D. An example is when the database elements $d_i \in \mathbb{R}^d$ and the output is the mean vector $v(D) = n^{-1}\sum_{i=1}^n d_i$. We note that this is basically a re-statement of the well-known fact that the addition of appropriate Gaussian noise to a vector valued output will lead to approximate differential privacy.

Proposition 3 *Suppose that, for a positive definite symmetric matrix $M \in \mathbb{R}^{d\times d}$, the family of vectors $\{v_D : D \in \mathcal{D}\} \subset \mathbb{R}^d$ satisfies*

$$\sup_{D\sim D'} \|M^{-1/2}(v_D - v_{D'})\|_2 \leq \Delta. \tag{5}$$

Then the randomized algorithm which, for input database D outputs

$$\widetilde{v}_D = v_D + \frac{c(\beta)\Delta}{\alpha} Z, \quad Z \sim \mathcal{N}_d(0, M)$$

achieves (α,β)-DP whenever

$$c(\beta) \geq \sqrt{2\log\frac{2}{\beta}}. \tag{6}$$

Proof Since the Gaussian measure on $\mathbb{R}^d$ admits the Lebesgue measure λ as a σ-finite dominating measure we consider the ratio of the densities

$$\frac{dP_D(x)/d\lambda}{dP_{D'}(x)/d\lambda} = \exp\left\{\frac{\alpha^2}{2c(\beta)^2\Delta^2}\left[(x-v_{D'})M^{-1}(x-v_{D'}) - (x-v_D)^T M^{-1}(x-v_D)\right]\right\}.$$

This ratio exceeds e^α only when

$$2x^T M^{-1}(v_D - v_{D'}) + v_{D'}^T M^{-1} v_{D'} - v_D^T M^{-1} v_D \geq 2\frac{c(\beta)^2\Delta^2}{\alpha}.$$

We consider the probability of this set under P_D, in which case we have $x = v_D + \frac{c(\beta)\Delta}{\alpha} M^{1/2} z$, where z is an isotropic normal with unit variance. We have

$$\frac{c(\beta)\Delta}{\alpha} z^T M^{-1/2}(v_D - v_{D'}) \geq \frac{c(\beta)^2\Delta^2}{\alpha^2} - \frac{1}{2}(v_D - v_{D'})^T M^{-1}(v_D - v_{D'}).$$

Multiplying by $\frac{\alpha}{c(\beta)\Delta}$ and using (5) gives

$$z^T M^{-1/2}(v_D - v_{D'}) \geq \frac{c(\beta)\Delta}{\alpha} - \frac{\alpha\Delta}{2c(\beta)}.$$

Note that the left side is a normal random variable with mean zero and variance smaller than Δ^2. The probability of this set is increasing with the variance of said variable, and so we examine the probability when the variance equals Δ^2. We also restrict to $\alpha \leq 1$, and let $y \sim \mathcal{N}(0,1)$, yielding

$$\begin{aligned} P\left(z^T M^{-1/2}(v_D - v_{D'}) \geq \frac{c(\beta)\Delta}{\alpha} - \frac{\alpha\Delta}{2c(\beta)}\right) &\leq P\left(\Delta y \geq \frac{c(\beta)\Delta}{\alpha} - \frac{\alpha\Delta}{2c(\beta)}\right) \\ &\leq P\left(y \geq c(\beta) - \frac{1}{2c(\beta)}\right) \\ &\leq \beta, \end{aligned}$$

where $c(\beta)$ is as defined in (6) and the final inequality is proved in Dwork (2006). Thus lemma 2 gives the differential privacy. ■

Remark 2 *The quantity (5) is a mild modification of the usual notion of "sensitivity" or "global sensitivity" (Dwork et al., 2006b). It is nothing more than the sensitivity measured in the Mahalanobis distance corresponding to the matrix M. The case $M = I$ corresponds to the usual Euclidean distance, a setting that has been studied previously by McSherry and Mironov (2009), among others. The use of this matrix allows for smaller noise magnitudes in the case when the difference of vectors on neighboring data sets are elements of some ellipsoid rather than a sphere. A simple case when this is advantageous is when the released function is an affine transformation of a sample mean of input vectors.*

2.2 The Implications of Approximate Differential Privacy

The above definitions provide a strong privacy guarantee in the sense that they aim to protect against an adversary having almost complete knowledge of the private database. Specifically, an adversary knowing all but one of the data elements and having observed the output of a private procedure, will remain unable to determine the identity of the data element which is unknown to him. To see this, we provide an analog of theorem 2.4 of Wasserman and Zhou (2010), who consider the case of α-differential privacy.

Let the adversary's database be denoted by $D_A = (d_1, \ldots, d_{n-1})$, and the private database by $D = (d_1, \ldots, d_n)$. First note that before observing the output of the private algorithm, the adversary could determine that the private database D lay in the set $\{(d_1, \ldots, d_{n-1}, d) \in \mathcal{D}\}$. Thus, the private database comprises his data with one more element. Since all other databases may be excluded from consideration by the adversary we concentrate on those in the above set. In particular, we obtain the following analog of theorem 2.4 of Wasserman and Zhou (2010).

Proposition 4 *Let $X \sim P_D$ where the family P_D achieves the (α, β)-approximate DP. Any level γ test of: $H : D = D_0$ vs $V : D \neq D_0$ has power bounded above by $\gamma e^{\alpha} + \beta$.*

The above result follows immediately from noting that the rejection region of the test is a measurable set in the space and so obeys the constraint of the differential privacy. The implication of the above proposition is that the power of the test will be bounded close to its size. When α, β are small, this means that the test is close to being "trivial" in the sense that it is no more likely to correctly reject a false hypothesis than it is to incorrectly reject the true one.

3. Approximate Differential Privacy for Functions

The goal of the release a function raises a number of questions. First what does it mean for a computer program to output a function? Second, how can the differential privacy be demonstrated? In this section we continue to treat randomized algorithms as measures, however now they are measures over function spaces. In section 5 we demonstrate concrete algorithms, which in essence output the function on any arbitrary countable set of points.

We cannot expect the techniques for finite dimensional vectors to apply directly when dealing with functions. The reason is that σ-finite dominating measures of the space of functions do not

exist, and, therefore, neither do densities. However, there exist probability measures on the spaces of functions. Below, we demonstrate the approximate differential privacy of measures on function spaces, by considering random variables which correspond to evaluating the random function on a finite set of points.

We consider the family of functions over $T = \mathbb{R}^d$ (where appropriate we may restrict to a compact subset such as the unit cube in d-dimensions):

$$\{f_D : D \in \mathcal{D}\} \subset \mathbb{R}^T.$$

A before, we consider randomized algorithms which on input D, output some $\widetilde{f}_D \sim P_D$ where P_D is a measure on $\mathbb{R}^T$ corresponding to D. The nature of the σ-field on this space will be described below.

3.1 Differential Privacy on the Field of Cylinders

We define the "cylinder sets" of functions (see Billingsley, 1995) for all finite subsets $S = (x_1, \ldots, x_n)$ of T, and Borel sets B of $\mathbb{R}^n$

$$C_{S,B} = \left\{ f \in \mathbb{R}^T : (f(x_1), \ldots, f(x_n)) \in B \right\}.$$

These are just those functions which take values in prescribed sets, at those points in S. The family of sets: $\mathcal{C}_S = \{C_{S,B} : B \in \mathcal{B}(\mathbb{R}^n)\}$ forms a σ-field for each fixed S, since it is the preimage of $\mathcal{B}(\mathbb{R}^n)$ under the operation of evaluation on the fixed finite set S. Taking the union over all finite sets S yields the collection

$$\mathcal{F}_0 = \bigcup_{S:|S|<\infty} \mathcal{C}_S.$$

This is a field (see Billingsley, 1995 page 508) although not a σ-field, since it does not have the requisite closure under countable intersections (namely it does not contain cylinder sets for which S is countably infinite). We focus on the creation of algorithms for which the differential privacy holds over the field of cylinder sets, in the sense that, for all $D \sim D' \in \mathcal{D}$,

$$P(\widetilde{f}_D \in A) \leq e^{\alpha} P(\widetilde{f}_{D'} \in A) + \beta, \quad \forall A \in \mathcal{F}_0. \tag{7}$$

This statement appears to be prima facie unlike the definition (1), since $\mathcal{F}_0$ is not a σ-field on $\mathbb{R}^T$. However, we give a limiting argument which demonstrates that to satisfy (7) is to achieve the approximate (α, β)-DP throughout the generated σ-field. First we note that satisfying (7) implies that the release of any finite evaluation of the function achieves the differential privacy. Since for any finite $S \subset T$, we have that $\mathcal{C}_S \subset \mathcal{F}_0$, we readily obtain the following result.

Proposition 5 *Let $x_1, \ldots, x_n$ be any finite set of points in T chosen a-priori. Then whenever (7) holds, the release of the vector*

$$\left(\widetilde{f}_D(x_1), \ldots, \widetilde{f}_D(x_n) \right)$$

satisfies the (α, β)-DP.

Proof We have that

$$P_D\left(\left(\widetilde{f}(x_1), \ldots, \widetilde{f}(x_n) \right) \in A \right) = P_D(\widetilde{f} \in C_{\{x_1,\ldots,x_n\},A}).$$

The claimed privacy guarantee follows from (7). ■

We now give a limiting argument to extend (7) to the generated σ-field (or, equivalently, the σ-field generated by the cylinders of dimension 1)

$$\mathcal{F} \stackrel{\text{def}}{=} \sigma(\mathcal{F}_0) = \bigcup_S \mathcal{C}_S$$

where the union extends over all the countable subsets S of T. The second equality above is due to Billingsley (1995) theorem 36.3 part ii.

Note that, for countable S, the cylinder sets take the form

$$C_{S,B} = \left\{ f \in \mathbb{R}^T : f(x_i) \in B_i,\ i = 1,2,\ldots \right\} = \bigcap_{i=1}^{\infty} C_{\{x_i\},B_i},$$

where B_i's are Borel sets of $\mathbb{R}$.

Proposition 6 *Let (7) hold. Then, the family $\{P_D : D \in \mathcal{D}\}$ on $(\mathbb{R}^T, \mathcal{F})$ satisfies for all $D \sim D' \in \mathcal{D}$:*

$$P_D(A) \leq e^{\alpha} P_{D'}(A) + \beta, \quad \forall A \in \mathcal{F}.$$

Proof Define $C_{S,B,n} = \bigcap_{i=1}^{n} C_{\{t_i\},B_i}$. Then, the sets $C_{S,B,n}$ form a sequence of sets which decreases towards $C_{S,B}$ and $C_{S,B} = \lim_{n\to\infty} C_{S,B,n}$. Since the sequence of sets is decreasing and the measure in question is a probability (hence bounded above by 1), we have

$$P_D(C_{S,B}) = P_D(\lim_{n\to\infty} C_{S,B,n}) = \lim_{n\to\infty} P_D(C_{S,B,n}).$$

Therefore, for each pair $D \sim D'$ and for every $\varepsilon > 0$, there exists an n_0 so that for all $n \geq n_0$

$$|P_D(C_{S,B}) - P_D(C_{S,B,n})| \leq \varepsilon, \quad |P_{D'}(C_{S,B}) - P_{D'}(C_{S,B,n})| \leq \varepsilon.$$

The number n_0 depends on whichever is the slowest sequence to converge. Finally we obtain

$$\begin{aligned} P_D(C_{S,B}) &\leq P_D(C_{S,B,n_0}) + \varepsilon \\ &\leq e^{\alpha} P_{D'}(C_{S,B,n_0}) + \beta + \varepsilon \\ &\leq e^{\alpha} P_{D'}(C_{S,B}) + \beta + (1 + e^{\alpha})\varepsilon \\ &\leq e^{\alpha} P_{D'}(C_{S,B}) + \beta + 3\varepsilon. \end{aligned}$$

Since this holds for all $\varepsilon > 0$ we conclude that $P_D(C_{S,B}) \leq e^{\alpha} P_{D'}(C_{S,B}) + \beta$. ■

In principle, if it were possible for a computer to release a complete description of the function $\widetilde{f}_D$ then this result would demonstrate the privacy guarantee achieved by our algorithm. In practice a computer algorithm which runs in a finite amount of time may only output a finite set of points, hence this result is mainly of theoretical interest. However, in the case in which the functions to be output are continuous, and the restriction is made that P_D are measures over $C[0,1]$ (the continuous functions on the unit interval), another description of the σ-field becomes available. Namely, the

restriction of $\mathcal{F}$ to the elements of $C[0,1]$ corresponds to the Borel σ-field over $C[0,1]$ with the topology induced by the uniform norm ($\|f\|_\infty = \sup_t |f(t)|$). Therefore in the case of continuous functions, differential privacy over $\mathcal{F}_0$ hence leads to differential privacy throughout the Borel σ-field.

In summary, we find that if every finite dimensional projection of the released function satisfies differential privacy, then so does every countable-dimensional projection. We now explore techniques which achieve the differential privacy over these σ-fields.

3.2 Differential Privacy via the Exponential Mechanism

A straightforward means to output a function in a way which achieves the differential privacy is to make use of the so-called "exponential mechanism" of McSherry and Talwar (2007). This approach entails the construction of a suitable finite set of functions $G = \{g_i, \ldots, g_m\} \in \mathbb{R}^T$, in which every f_D has a reasonable approximation, under some distance function d. Then, when the input is D, a function is chosen to output by sampling the set of G with probabilities given by

$$P_D(g_i) \propto \exp\left\{\frac{-\alpha}{2s} d(g_i, f_D)\right\}, \quad s \stackrel{\text{def}}{=} \sup_{D \sim D'} d(f_D, f_{D'}).$$

McSherry and Talwar (2007) demonstrate that such a technique achieves the α-differential privacy, which is strictly stronger than the (α,β)-differential privacy we consider here. Although this technique is conceptually appealing for its simplicity, it remains challenging to use in practice since the set of functions G may need to be very large in order to ensure the utility of the released function (in the sense of expected error). Since the algorithm which outputs from P_D must obtain the normalization constant to the distribution above, it must evidently compute the probabilities for each g_i, which may be extremely time consuming. Note that techniques such as importance sampling are also difficult to bring to bear against this problem when it is important to maintain utility.

The technique given above can be interpreted as outputting a discrete random variable, and fulfilling privacy definition with respect to the σ-field consisting of the powerset of G. This implies the privacy with respect to the cylinder sets, since the restriction of each cylinder set to the elements of G corresponds some subset of G.

We note that the exponential mechanism above essentially corresponded to a discretization of the function space $\mathbb{R}^T$. An alternative is to discretize the input space T, and to approximate the function by a piecewise constant function where the pieces correspond to the discretization of T. Thereupon the approximation may be regarded as a real valued vector, with one entry for the value of each piece of the function. This is conceptually appealing but it remains to be seen whether the sensitivity of such a vector valued output could be bounded. In the next section we describe a method which may be regarded as similar to the above, and which has the nice property that the choice of discretization is immaterial to the method and to the determination of sensitivity.

3.3 Differential Privacy via Gaussian Process Noise

We propose to use measures P_D over functions, which are Gaussian processes. The reason is that there is a strong connection between these measures over the infinite dimensional function space, and the Gaussian measures over finite dimensional vector spaces such as those used in Proposition 3. Therefore, with some additional technical machinery which we will illustrate next, it is possible to move from differentially private measures over vectors to those over functions.

A Gaussian process indexed by T is a collection of random variables $\{X_t : t \in T\}$, for which each finite subset is distributed as a multivariate Gaussian (see, for instance, Adler, 1990; Adler and Taylor, 2007). A sample from a Gaussian process may be considered as a function $T \to \mathbb{R}$, by examining the so-called "sample path" $t \to X_t$. The Gaussian process is determined by the mean and covariance functions, defined on T and T^2 respectively, as

$$m(t) = \mathbb{E}X_t, \quad K(s,t) = \mathrm{Cov}(X_s, X_t).$$

For any finite subset $S \subset T$, the random vector $\{X_t : t \in S\}$ has a normal distribution with the means, variances, and covariances given by the above functions. Such a "finite dimensional distribution" may be regarded as a projection of the Gaussian process. Below we propose particular mean and covariance functions for which Proposition 3 will hold for all finite dimensional distributions. These will require some smoothness properties of the family of functions $\{f_D\}$. We first demonstrate the technical machinery which allows us to move from finite dimensional distributions to distributions on the function space, and then we give differentially private measures on function spaces of one dimension. Finally, we extend our results to multiple dimensions.

Proposition 7 *Let G be a sample path of a Gaussian process having mean zero and covariance function K. Let M denote the Gram matrix*

$$M(x_1,\ldots,x_n) = \begin{pmatrix} K(x_1,x_1) & \cdots & K(x_1,x_n) \\ \vdots & \ddots & \vdots \\ K(x_n,x_1) & \cdots & K(x_n,x_n) \end{pmatrix}.$$

Let $\{f_D : D \in \mathcal{D}\}$ be a family of functions indexed by databases. Then the release of

$$\widetilde{f}_D = f_D + \frac{\Delta c(\beta)}{\alpha} G$$

is (α,β)-differentially private (with respect to the cylinder σ-field $\mathcal{F}$) whenever

$$\sup_{D \sim D'} \sup_{n<\infty} \sup_{(x_1,\ldots,x_n)\in T^n} \left\| M^{-1/2}(x_1,\ldots,x_n) \begin{pmatrix} f_D(x_1) - f_{D'}(x_1) \\ \vdots \\ f_D(x_n) - f_{D'}(x_n) \end{pmatrix} \right\|_2 \leq \Delta. \tag{8}$$

Proof For any finite set $(x_1,\ldots,x_n) \in T^n$, the vector $(G(x_1),\ldots,G(x_n))$ follows a multivariate normal distribution having mean zero and covariance matrix specified by $\mathrm{Cov}(G(x_i),G(x_j)) = K(x_i,x_j)$. Thus for the vector obtained by evaluation of $\widetilde{f}$ at those points, differential privacy is demonstrated by Proposition 3 since (8) implies the sensitivity bound (5). Thus, for any $n < \infty$ and any $(x_1,\ldots,x_n) \in T^n$ we have $B \in \mathcal{B}(\mathbb{R}^n)$

$$P_D\left(\left(\widetilde{f}(x_1),\ldots,\widetilde{f}(x_n)\right) \in B\right) \leq e^{\alpha} P_{D'}\left(\left(\widetilde{f}(x_1),\ldots,\widetilde{f}(x_n)\right) \in B\right) + \beta$$

Finally note that for any $A \in \mathcal{F}_0$, we may write $A = C_{X_n,B}$ for some finite n, some vector $X_n = (x_1,\ldots,x_n) \in T^n$ and some Borel set B. Then

$$P_D(\widetilde{f} \in A) = P_D\left(\left(\widetilde{f}(x_1),\ldots,\widetilde{f}(x_n)\right) \in B\right).$$

Combining this with the above gives the requisite privacy statement for all $A \in \mathcal{F}_0$. Proposition 6 carries this to $\mathcal{F}$. ■

3.4 Functions in a Reproducing Kernel Hilbert Space

When the family of functions lies in the reproducing kernel Hilbert space (RKHS) which corresponds to the covariance kernel of the Gaussian process, then establishing upper bounds of the form (8) is simple. Below, we give some basic definitions for RKHSs, and refer the reader to Bertinet and Agnan (2004) for a more detailed account. We first recall that the RKHS is generated from the closure of those functions which can be represented as finite linear combinations of the kernel,

$$\mathcal{H}_0 = \left\{ \sum_{i=1}^{n} \xi_i K_{x_i} \right\}$$

for some finite n and sequence $\xi_i \in \mathbb{R}$, $x_i \in T$, and where $K_x = K(x,\cdot)$. For two functions $f = \sum_{i=1}^{n} \theta_i K_{x_i}$ and $g = \sum_{j=1}^{m} \xi_j K_{y_j}$ the inner product is given by

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \sum_{j=1^m} \theta_i \xi_j K(x_i, y_j),$$

and the corresponding norm of f is $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$. This gives rise to the "reproducing" nature of the Hilbert space, namely, $\langle K_x, K_y \rangle_{\mathcal{H}} = K(x,y)$. Furthermore, the functionals $\langle K_x, \cdot \rangle_{\mathcal{H}}$ correspond to point evaluation,

$$\langle K_x, f \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \theta_i K(x_i, x) = f(x).$$

The RKHS $\mathcal{H}$ is then the closure of $\mathcal{H}_0$ with respect to the RKHS norm. We now present the main theorem which suggests an upper bound of the form required in Proposition 7.

Proposition 8 *For $f \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS corresponding to the kernel K, and for any finite sequence $x_1, \ldots, x_n$ of distinct points in T, we have:*

$$\left\| \begin{pmatrix} K(x_1,x_1) & \cdots & K(x_1,x_n) \\ \vdots & \ddots & \vdots \\ K(x_n,x_1) & \cdots & K(x_n,x_n) \end{pmatrix}^{-1/2} \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \right\|_2 \leq \|f\|_{\mathcal{H}}.$$

The proof is in the appendix. Together with Proposition 7, this result implies the following.

Corollary 9 *For $\{f_D : D \in \mathcal{D}\} \subseteq \mathcal{H}$, the release of*

$$\widetilde{f}_D = f_D + \frac{\Delta c(\beta)}{\alpha} G$$

is (α, β)-differentially private (with respect to the cylinder σ-field) whenever

$$\Delta \geq \sup_{D \sim D'} \|f_D - f_{D'}\|_{\mathcal{H}}.$$

and when G is the sample path of a Gaussian process having mean zero and covariance function K, given by the reproducing kernel of $\mathcal{H}$.

4. Examples

We now give some examples in which the above technique may be used to construct private versions of functions in an RKHS.

4.1 Kernel Density Estimation

Let f_D be the kernel density estimator, where D is regarded as a sequence of points $x_i \in T$ as $i = 1,\ldots,n$ drawn from a distribution with density f. Let h denote the bandwidth. Assuming a Gaussian kernel, the estimator is

$$f_D(x) = \frac{1}{n(2\pi h^2)^{d/2}} \sum_{i=1}^{n} \exp\left\{\frac{-\|x-x_i\|_2^2}{2h^2}\right\}, \quad x \in T.$$

Let $D \sim D'$ so that $D' = x_1,\ldots,x_{n-1},x_n'$ (no loss of generality is incurred by demanding that the data sequences differ in their last element). Then,

$$(f_D - f_{D'})(x) = \frac{1}{n(2\pi h^2)^{d/2}} \left(\exp\left\{-\frac{\|x-x_n\|_2^2}{2h^2}\right\} - \exp\left\{-\frac{\|x-x_n'\|_2^2}{2h^2}\right\}\right).$$

If we use the Gaussian kernel as the covariance function for the Gaussian process then upper bounding the RKHS norm of this function is trivial. Thus, let $K(x,y) = \exp\left\{-\frac{\|x-y\|_2^2}{2h^2}\right\}$. Then $f_D - f_{D'} = \frac{1}{n(2\pi h^2)^{d/2}}\left(K_{x_n} - K_{x_n'}\right)$ and

$$\begin{aligned}\|f_D - f_{D'}\|_{\mathcal{H}}^2 &= \left(\frac{1}{n(2\pi h^2)^{d/2}}\right)^2 \left(K(x_n,x_n) + K(x_n',x_n') - 2K(x_n,x_n')\right) \\ &\leq 2\left(\frac{1}{n(2\pi h^2)^{d/2}}\right)^2.\end{aligned}$$

If we release

$$\widetilde{f}_D = f_D + \frac{c(\beta)\sqrt{2}}{\alpha n(2\pi h^2)^{d/2}} G$$

where G is a sample path of a Gaussian process having mean zero and covariance K, then differential privacy is demonstrated by corollary 9. We may compare the utility of the released estimator to that of the non-private version. Under standard smoothness assumptions on f, it is well-known (see Wasserman, 2006) that the risk is

$$R = \mathbb{E}\int (f_D(x) - f(x))^2 dx = c_1 h^4 + \frac{c_2}{nh^d},$$

for some constants c_1 and c_2. The optimal bandwidth is $h \asymp (1/n)^{1/(4+d)}$ in which case $R = O(n^{-\frac{4}{4+d}})$.

For the differentially private function it is easy to see that

$$\mathbb{E}\int (\widetilde{f}_D(x) - f(x))^2 dx = O\left(h^4 + \frac{c_2}{nh^d}\right).$$

Therefore, at least in terms of rates, no accuracy has been lost.

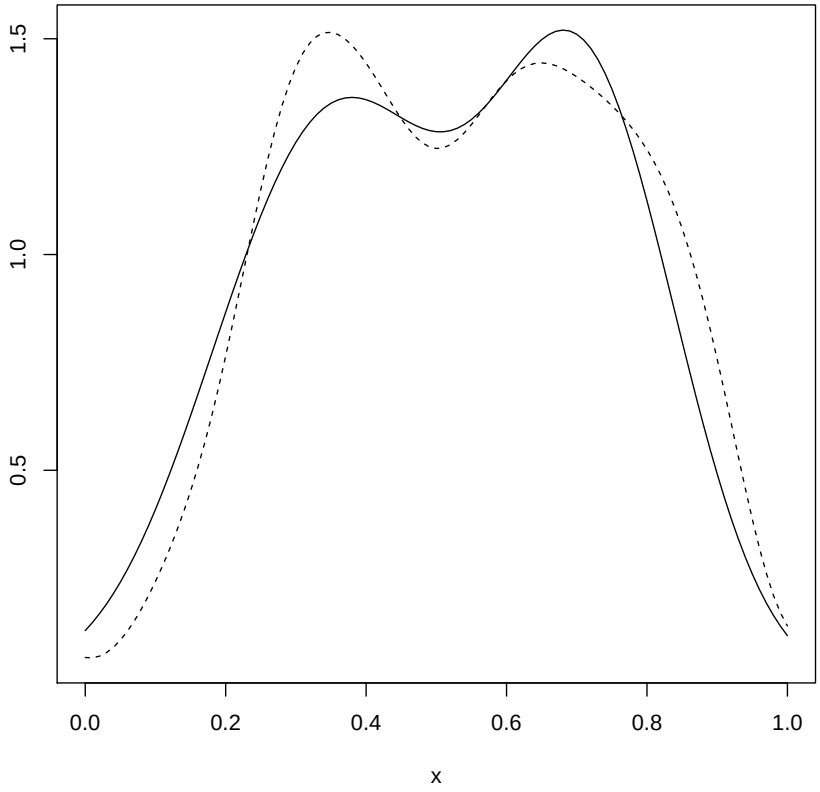


Figure 1: An example of a kernel density estimator (the solid curve) and the released version (the dashed curve). This uses the method developed in Section 4.1. Here we sampled $n = 100$ points from a mixture of two normals centered at 0.3 and 0.7 respectively. We use $h = 0.1$ and have $\alpha = 1$ and $\beta = 0.1$. The Gaussian Process is evaluated on an evenly spaced grid of 1000 points between 0 and 1. Note that gross features of the original kernel density estimator remain, namely the two peaks.

4.1.1 NON-ISOTROPIC KERNELS

The above demonstration of privacy also holds when the kernel is replaced by a non-isotropic Gaussian kernel. In this case the kernel density estimate may take the form

$$f_D(x) = \frac{1}{n(2\pi)^{d/2}|H|^{1/2}} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2}(x-x_i)^T H^{-1}(x-x_i) \right\}, \quad x \in T,$$

where H is a positive definite matrix and $|H|$ is the determinant. For example it may be required to employ a different choice of bandwidth for each coordinate of the space, in which case H would be

a diagonal matrix having non-equal entries on the diagonal. So long as H is fixed a-priori, privacy may be established by adding a Gaussian process having mean zero and covariance given by

$$K(x,y) = \exp\left\{-\frac{1}{2}(x-y)^T H^{-1}(x-y)\right\}.$$

As above, the sensitivity is upper bounded, as

$$\|f_D - f_{D'}\|_{\mathcal{H}_f}^2 \leq 2\left(\frac{1}{n(2\pi)^{d/2}|H|^{1/2}}\right)^2.$$

Therefore it satisfies the (α,β)-DP to release

$$\widetilde{f}_D = f_D + \frac{c(\beta)\sqrt{2}}{\alpha n(2\pi)^{d/2}|H|^{1/2}}G,$$

where G is a sample path of a Gaussian process having mean zero and covariance K.

4.1.2 PRIVATE CHOICE OF BANDWIDTH

Note that the above assumed that h (or H) was fixed a-priori by the user. In usual statistical settings h is a parameter that is tuned depending on the data (not simply set to the correct order of growth as a function of n). Thus rather than fixed h the user would use $\widehat{h}$ which depends on the data itself. In order to do this it is necessary to find a differentially private version of $\widehat{h}$ and then to employ the composition property of differential privacy.

The typical way that the bandwidth is selected is by employing the leave-one-out cross validation. This consists of choosing a grid of candidate values for h, evaluating the leave one out log likelihood for each value, and then choosing whichever is the maximizer. This technique may be amenable to private analysis via the "exponential mechanism," however it would evidently require that T be a compact set which is known a-priori. An alternative is to use a "rule of thumb" (see Scott, 1992) for determining the bandwidth which is given by

$$\widehat{h}_j = \left(\frac{4}{(d+1)n}\right)^{\frac{1}{d+4}} \frac{IQR_j}{1.34}$$

In which IQR_j is the observed interquartile range of the data along the j^{th} coordinate. Thus this method gives a diagonal matrix H as in the above section. To make a private version $\widetilde{h}_j$ we may use the technique of Dwork and Lei (2009) in which a differentially private algorithm for the interquartile range was developed.

4.2 Functions in a Sobolev Space

The above technique worked easily since we chose a particular RKHS in which we knew the kernel density estimator to live. What's more, since the functions themselves lay in the generating set of functions for that space, the determination of the norm of the difference $f_D - f_{D'}$ was extremely simple. In general we may not be so lucky that the family of functions is amenable to such analysis. In this section we demonstrate a more broadly applicable technique which may be used whenever the functions are sufficiently smooth. Consider the Sobolev space

$$H^1[0,1] = \left\{f \in C[0,1] : \int_0^1 (\partial f(x))^2 \, d\lambda(x) < \infty\right\}.$$

This is a RKHS with the kernel $K(x,y) = \exp\{-\gamma\,|x-y|\}$ for positive constant γ. The norm in this space is given by

$$\|f\|^2_{\mathcal{H}} = \frac{1}{2}\left(f(0)^2 + f(1))^2\right) + \frac{1}{2\gamma}\int_0^1 (\partial f(x))^2 + \gamma^2 f(t)^2\, d\lambda(t). \tag{9}$$

See Bertinet and Agnan (2004) (p. 316) and Parzen (1961) for details. Thus for a family of functions in one dimension which lay in the Sobolev space H^1, we may determine a noise level necessary to achieve the differential privacy by bounding the above quantity for the difference of two functions. For functions over higher dimensional domains (as $[0,1]^d$ for some $d > 1$) we may construct an RKHS by taking the d-fold tensor product of the above RKHS (see, in particular, Parzen, 1963; Aronszajn, 1950, for details of the construction). The resulting space has the reproducing kernel

$$K(x,y) = \exp\{-\gamma\|x-y\|_1\},$$

and is the completion of the set of functions

$$\mathcal{G}_0 = \left\{f : [0,1]^d \to \mathbb{R} : f(x_1,\ldots,x_d) = f_1(x_1)\cdots f_d(x_d), f_i \in H^1[0,1]\right\}.$$

The norm over this set of functions is given by:

$$\|f\|^2_{\mathcal{G}_0} = \prod_{j=1}^d \|f_i\|^2_{\mathcal{H}}. \tag{10}$$

The norm over the completed space agrees with the above on $\mathcal{G}_0$. The explicit form is obtained by substituting (9) into the right hand side of (10) and replacing all instances of $\prod_{j=1}^d f_j(x_j)$ with $f(x_1,\ldots,x_j)$. Thus the norm in the completed space is defined for all f possessing all first partial derivatives which are all in $\mathcal{L}_2$.

We revisit the example of a kernel density estimator (with an isotropic Gaussian kernel). We note that this isotropic kernel function is in the set G_0 defined above, as

$$\phi_{\mu,h}(x) = \frac{1}{(2\pi h^2)^{d/2}}\exp\left\{-\frac{\|x-\mu\|_2^2}{2h^2}\right\} = \prod_{j=1}^d \frac{1}{\sqrt{2\pi h}}\exp\left\{-\frac{(x_j-\mu_j)^2}{2h^2}\right\} = \prod_{j=1}^d \phi_{\mu_j,h}(x_j).$$

Where $\phi_{\mu,h}$ is the isotropic Gaussian kernel on $\mathbb{R}^d$ with mean vector μ and $\phi_{\mu_j,h}$ is the Gaussian kernel in one dimension with mean μ_j. We obtain the norm of the latter one dimensional function by bounding the elements of the sum in (9) ad follows:

$$\int_0^1 (\partial\phi_{\mu_j,h}(x))^2\, d\lambda(x) \;\leq\; \int_{-\infty}^{\infty}\left(\partial\frac{1}{\sqrt{2\pi}h}e^{-(x-\mu_j)^2/2h^2}\right)^2 d\lambda(x) = \frac{1}{4\sqrt{\pi}h^3},$$

and

$$\int_0^1 \phi_{\mu_j,h}(x)^2\, d\lambda(x) \;\leq\; \int_{-\infty}^{\infty}\frac{1}{2\pi h^2}e^{-(x-\mu_j)^2/h^2}\, d\lambda(x) = \frac{1}{2\sqrt{2\pi}h},$$

where we have used the fact that

$$\phi_{\mu_j,h}(x)^2 \leq \frac{1}{2\pi h}, \quad \forall x \in \mathbb{R}^d.$$

Therefore, choosing $\gamma = 1/h$ leads to

$$\|\phi_{\mu_j,h}\|_{\mathcal{H}}^2 \leq \frac{1}{2\pi h^2} + \frac{1}{8\sqrt{\pi}h^2} + \frac{1}{4\sqrt{2\pi}h^2} \leq \frac{1}{\sqrt{2\pi}h^2},$$

and

$$\|\phi_{\mu,h}\|_{\mathcal{H}}^2 \leq \frac{1}{(2\pi)^{d/2}h^{2d}}.$$

Finally,

$$\|f_D - f_{D'}\|_{\mathcal{H}} = n^{-1}\|\phi_{x_n,h} - \phi_{x_n',h}\|_{\mathcal{H}} \leq \frac{2}{(2\pi)^{d/4}nh^d}$$

Therefore, we observe a technique which attains higher generality than the ad-hoc analysis of the preceding section. However this is at the expense of the noise level, which grows at a higher rate as d increases. An example of the technique applied to the same kernel density estimation problem as above is given in Figure 2.

4.3 Minimizers of Regularized Functionals in an RKHS

The construction of the following section is due to Bousquet and Elisseeff (2002), who were interested in determining the sensitivity of certain kernel machines (among other algorithms) with the aim of bounding the generalization error of the output classifiers. Rubinstein et al. (2010) noted that these bounds are useful for establishing the noise level required for differential privacy of support vector machines. They are also useful for our approach to privacy in a function space.

We consider classification and regression schemes in which the data sets $D = \{z_1, \ldots, z_n\}$ with $z_i = (x_i, y_i)$, where $x_i \in [0,1]^d$ are some covariates, and y_i is some kind of label, either taking values on $\{-1, +1\}$ in the case of classification or some taking values in some interval when the goal is regression. Thus the output functions are from $[0,1]^d$ to a subset of $\mathbb{R}$. The functions we are interested in take the form

$$f_D = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{z_i \in D} \ell(g, z_i) + \lambda \|g\|_{\mathcal{H}}^2 \tag{11}$$

where $\mathcal{H}$ is some RKHS to be determined, and ℓ is the so-called "loss function." We now recall a definition from Bousquet and Elisseeff (2002) (using M in place of their σ to prevent confusion):

Definition 10 (M-admissible loss function: see Bousquet and Elisseeff, 2002) *A loss function: $\ell(g,z) = c(g(x),y)$ is called M-admissible whenever c it is convex in its first argument and Lipschitz with constant M in its first argument.*

We will now demonstrate that for (11), whenever the loss function is admissible, the minimizers on adjacent data sets may be bounded close together in RKHS norm. Denote the part of the optimization due to the loss function:

$$L_D(f) = \frac{1}{n} \sum_{z_i \in D} \ell(f, z_i).$$

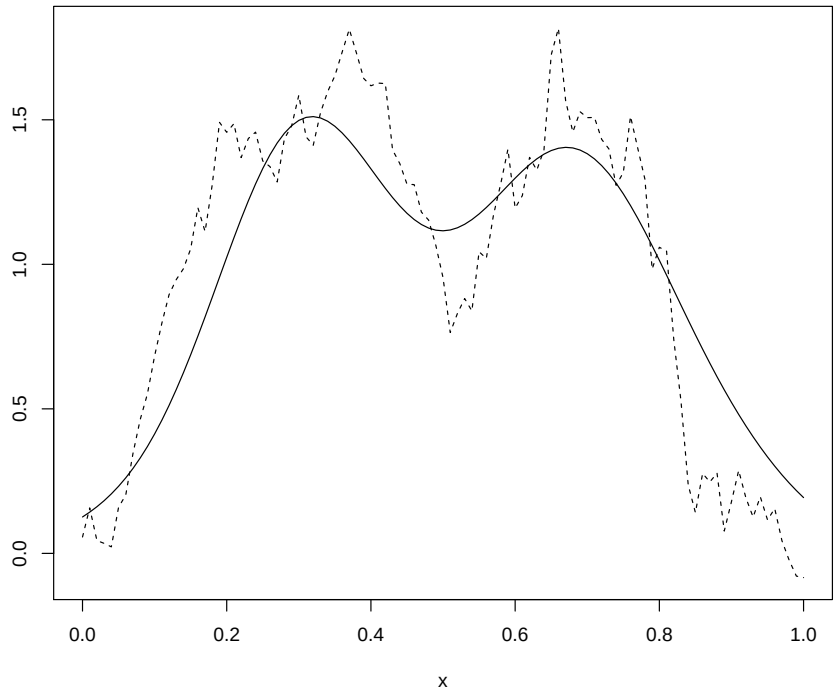


Figure 2: An example of a kernel density estimator (the solid curve) and the released version (the dashed curve). The setup is the same as in Figure 1, but the privacy mechanism developed in Section 4.2 was used instead. Note that the released function does not have the desirable smoothness of released function from Figure 1.

Using the technique from the proof of lemma 20 of Bousquet and Elisseeff (2002) we find that since ℓ is convex in its first argument we have

$$L_D(f_D + \eta\delta_{D',D}) - L_D(f_D) \leq \eta(L_D(f_{D'}) - L_D(f_D)),$$

where $\eta \in [0,1]$ and we use $\delta_{D',D} = f_{D'} - f_D$. This also holds when f_D and $f_{D'}$ swap places. Summing the resulting inequality with the above and rearranging yields

$$L_D(f_{D'} - \eta\delta_{D',D}) - L_D(f_{D'}) \leq L_D(f_D) - L_D(f_D + \eta\delta_{D',D}).$$

Due to the definition of $f_D, f_{D'}$ as the minimizers of their respective functionals we have

$$L_D(f_D) + \lambda\|f_D\|^2_{\mathcal{H}} \leq L_D(f_D + \eta\delta_{D',D}) + \lambda\|f_D + \eta\delta_{D',D}\|^2_{\mathcal{H}}$$
$$L_{D'}(f_{D'}) + \lambda\|f_{D'}\|^2_{\mathcal{H}} \leq L_{D'}(f_{D'} - \eta\delta_{D',D}) + \lambda\|f_{D'} - \eta\delta_{D',D}\|^2_{\mathcal{H}}.$$

This leads to the inequalities

$$\begin{aligned}
0 &\geq \lambda\left(\|f_D\|_{\mathcal{H}}^2 - \|f_D+\eta\delta_{D',D}\|_{\mathcal{H}}^2 + \|f_{D'}\|_{\mathcal{H}}^2 - \|f_{D'}-\eta\delta_{D',D}\|_{\mathcal{H}}^2\right)\\
&\quad + L_D(f_D) - L_D(f_D+\eta\delta_{D',D} + L_{D'}(f_{D'}) - L_{D'}(f_{D'}-\eta\delta_{D',D})\\
&\geq 2\lambda\|\eta\delta_{D',D}\|_{\mathcal{H}}^2 - L_D(f_{D'}) + L_D(f_{D'}-\eta\delta_{D',D}) + L_{D'}(f_{D'}) - L_{D'}(f_{D'}-\eta\delta_{D',D})\\
&= 2\lambda\|\eta\delta_{D',D}\|_{\mathcal{H}}^2 + \frac{1}{n}\left(\ell(z,f_{D'}) - \ell(z,f_{D'}-\eta\delta_{D',D}) + \ell(z',f_{D'}) - \ell(z',f_{D'}-\eta\delta_{D',D})\right).
\end{aligned}$$

Moving the loss function term to the other side and using the Lipschitz property we finally obtain that

$$\|f_D - f_{D'}\|_{\mathcal{H}}^2 \leq \frac{M}{\lambda n}\|f_D - f_{D'}\|_\infty.$$

What's more, the reproducing property together with Cauchy-Schwarz inequality yields

$$|f_D(x) - f_{D'}(x)| = |\langle f_D - f_{D'}, K_x\rangle_{\mathcal{H}}| \leq \|f_D - f_{D'}\|_{\mathcal{H}}\sqrt{K(x,x)}.$$

Combining with the previous result gives

$$\|f_D - f_{D'}\|_{\mathcal{H}}^2 \leq \frac{M}{\lambda n}\|f_D - f_{D'}\|_{\mathcal{H}}\sqrt{\sup_x K(x,x)},$$

which, in turn, leads to

$$\|f_D - f_{D'}\|_{\mathcal{H}} \leq \frac{M}{\lambda n}\sqrt{\sup_x K(x,x)}.$$

For a soft-margin kernel SVM we have the loss function: $\ell(g,z) = (1-yg(x))_+$, which means the positive part of the term in parentheses. Since the label y takes on either plus or minus one, we find this to be 1-admissible. An example of a kernel SVM in $T = \mathbb{R}^2$ is shown in Figure 3.

5. Algorithms

There are two main modes in which functions f_D could be released by the holder of the data D to the outside parties. The first is a "batch" setting in which the parties designate some finite collection of points $x_1 \ldots, x_n \in T$. The database owner computes $\widetilde{f}_D(x_i)$ for each i and return the vector of results. At this point the entire transaction would end with only the collection of pairs $(x_i, \widetilde{f}_D(x_i))$ being known to the outsiders. An alternative is the "online" setting in which outside users repeatedly specify points in $x_i \in T$, the database owner replies with $\widetilde{f}_D(x_i)$, but unlike the former setting he remains available to respond to more requests for function evaluations. We name these settings "batch" and "online" for their resemblance of the batch and online settings typically considered in machine learning algorithms.

The batch method is nothing more than sampling a multivariate Gaussian, since the set $x_1, \ldots, x_n \in T$ specifies the finite dimensional distribution of the Gaussian process from which to sample. The released vector is simply

$$\begin{pmatrix} \widetilde{f}_D(x_1) \\ \vdots \\ \widetilde{f}_D(x_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} f_D(x_1) \\ \vdots \\ f_D(x_n) \end{pmatrix}, \frac{c(\beta)\Delta}{\alpha}\begin{pmatrix} K(x_1,x_1) & \cdots & K(x_1,x_n) \\ \vdots & \ddots & \vdots \\ K(x_n,x_1) & \cdots & K(x_n,x_n) \end{pmatrix}\right).$$

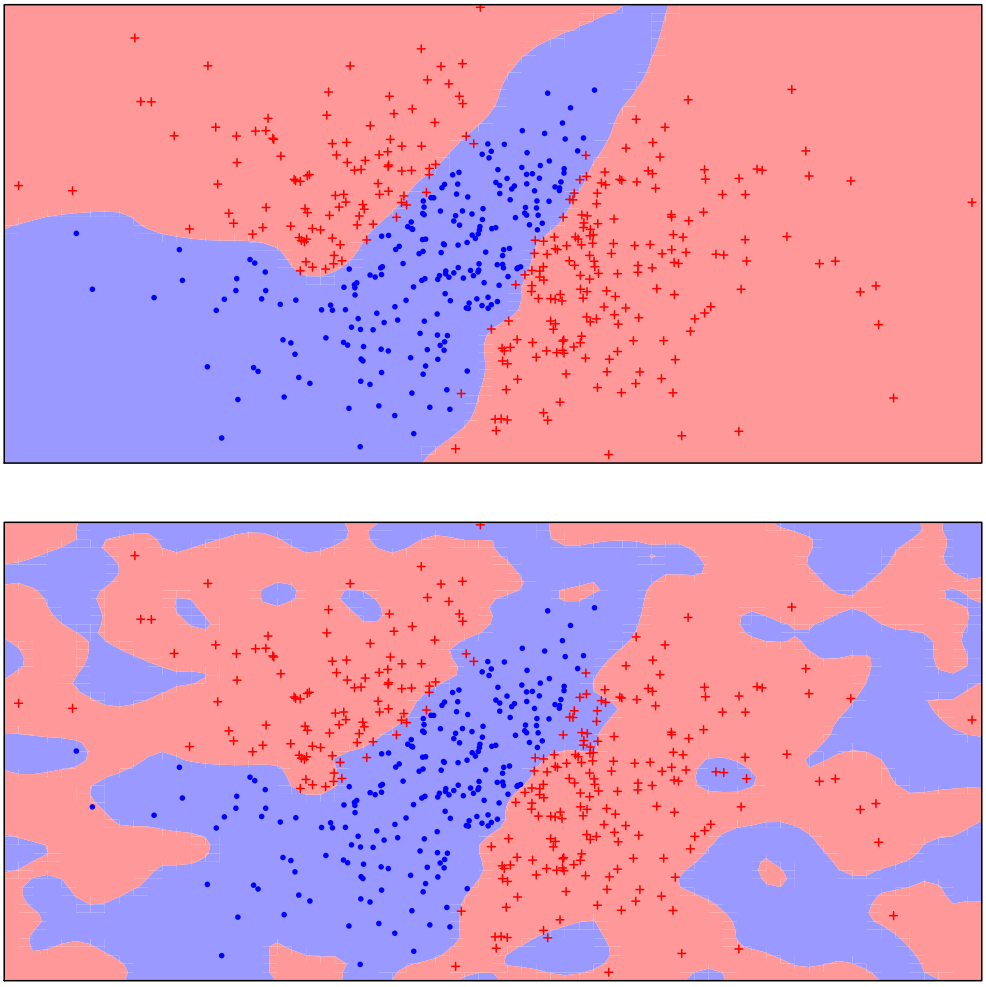


Figure 3: An example of a kernel support vector machine. In the top image are the data points, with the colors representing the two class labels (also points are used for one class and crosses for the other). The background color corresponds to the class predicted by the learned kernel svm. In the bottom image are the same data points, with the predictions of the private kernel svm. This example uses the Gaussian kernel for classification.

In the online setting, the data owner upon receiving a request for evaluation at x_i would sample the Gaussian process conditioned on the samples already produced at $x_1, \ldots, x_{i-1}$. Let

$$C_i = \begin{pmatrix} K(x_1,x_1) & \cdots & K(x_1,x_{i-1}) \\ \vdots & \ddots & \vdots \\ K(x_{i-1},x_1) & \cdots & K(x_{i-1},x_{i-1}) \end{pmatrix},$$

$$G_i = \begin{pmatrix} \widetilde{f}_D(x_1) \\ \vdots \\ \widetilde{f}_D(x_{i-1}) \end{pmatrix}, \quad V_i = \begin{pmatrix} K(x_1, x_i) \\ \vdots \\ K(x_{i-1}, x_i) \end{pmatrix}.$$

Then,

$$\widetilde{f}_D(x_i) \sim \mathcal{N}\left(V_i^T C_i^{-1} G_i,\ K(x_i,x_i) - V_i^T C_i^{-1} V_i\right).$$

The database owner may track the inverse matrix C_i^{-1} and after each request update it into C_{i+1}^{-1} by making use of Schurs Complements combined with the matrix inversion lemma. Nevertheless we note that as i increases the computational complexity of answering the request will in general grow. In the very least, the construction of V_i takes time proportional to i. This may make this approach problematic to implement in practice. However we note that when using the covariance kernel

$$K(x,y) = \exp\{-\gamma\,|x-y|_1\}$$

that a more efficient algorithm presents itself. This is the kernel considered in section 4.2. Due to the above form of K, we find that for $x < y < z$ we have: $K(x,z) = K(x,y)K(y,z)$. Therefore in using the above algorithm we would find that V_i is always contained in the span of at most two rows of C_i. This is most evident when, for instance, $x_i < \min_{j<i} x_j$. In this case let $m = \arg\min_{j<i} x_j$ $V_i = K(x_i, x_m)C_i(m)$, in which $C_i(m)$ means the m^{th} row of C_i. Therefore $C_i^{-1}V_i$ will be a sparse vector with exactly one non-zero entry (taking value $K(x,x_m)$) in the m^{th} position. Similar algebra applies whenever x_i falls between two previous points, in which case V_i lays in the span of the two rows corresponding to the closest point on the left and the closest on the right. Using the above kernel with some choice of γ let

$$\rho(x,y) = e^{\gamma|x-y|} - e^{-\gamma|x-y|}.$$

Let $\xi(x_i) = \widetilde{f}_D(x_i) - f_D(x_i)$ represent the noise process. We find that the conditional distribution of $\xi(x_i)$ to be Normal with mean and variance given by:

$$\mathbb{E}\xi(x_i) = \begin{cases} K(x_i, x_{(1)})\xi(x_{(1)}) & x_i < x_{(1)} \\ K(x_i, x_{(i-1)})\xi(x_{(i-1)}) & x_i > x_{(i-1)} \\ \frac{\rho(x_{(j+1)},x_i)}{\rho(x_{(j)},x_{(j+1)})}\xi(x_{(j)}) + \frac{\rho(x_{(j)},x_i)}{\rho(x_{(j)},x_{(j+1)})}\xi(x_{(j+1)}) & x_{(j)} < x_i < x_{(j+1)}, \end{cases}$$

and

$$\mathrm{Var}[\widetilde{f}_D(x_i)] = \begin{cases} 1 - K(x, x_{(1)})^2 & x_i < x_{(1)} \\ 1 - K(x, x_{(i-1)})^2 & x_i > x_{(i-1)} \\ 1 - K(x, x_{(j)})\frac{\rho(x_{(j+1)},x_i)}{\rho(x_{(j)},x_{(j+1)})} - K(x, x_{(j+1)})\frac{\rho(x_{(j)},x_i)}{\rho(x_{(j)},x_{(j+1)})} & x_{(j)} < x_i < x_{(j+1)}, \end{cases}$$

where $x_{(1)} < x_{(2)} < \cdots < x_{(i-1)}$ are the points $x_1, \ldots, x_{i-1}$ after being sorted into increasing order. In using the above algorithm it is only necessary for the data owner to store the values x_i and $\widetilde{f}_D(x_i)$. When using the proper data structures, for example a sorted doubly linked list for the x_i it is possible to determine the mean and variance using the above technique in time proportional to $\log(i)$ which is a significant improvement over the general linear time scheme above (note that the linked list is suggested since then it is possible to update the list in constant time).

6. Conclusion

We have shown how to add random noise to a function in such a way that differential privacy is preserved. It would be interesting to study this method in the many applications of functional data analysis (Ramsay and Silverman, 1997).

Interesting future work will be to address the issue of lower bounds for private functions. Specifically, we can ask: Given that we want to release a differentially private function, what is the least amount of noise that must necessarily be added in order to preserve differential privacy? This question has been addressed in detail for real-valued, count-valued and vector-valued data (see, for example, Hardt and Talwar, 2010). However, those techniques apply to the case of $\beta = 0$ whereupon the family $\{P_D\}$ are all mutually absolutely continuous. In the case of $\beta > 0$ which we consider this no longer applies and so the determination of lower bounds is complicated (for example, since quantities such as the KL divergence are no longer bounded). Some work in this direction is in McGregor et al. (2010) and Chaudhuri and Hsu (2012).

Acknowledgments

We thank the anonymous reviewers for constructive comments. This research was partially supported by Army contract DAAD19-02-1-3-0389 to Cylab, and NSF Grants BCS0941518 and SES1130706 to the Department of Statistics, both at Carnegie Mellon University.

Appendix A.

Proof of Proposition 8. Note that invertibility of the matrix is safely assumed due to Mercer's theorem. Denote the matrix by M^{-1}. Denote by P the operator $\mathcal{H} \to \mathcal{H}$ defined by

$$P = \sum_{i=1}^{n} K_{x_i} \sum_{j=1}^{n} (M^{-1})_{i,j} \left\langle K_{x_j}, \cdot \right\rangle_{\mathcal{H}}$$

We find this operator to be idempotent in the sense that $P = P^2$:

$$\begin{aligned}
P^2 &= \sum_{i=1}^{n} K_{x_i} \sum_{j=1}^{n} (M^{-1})_{i,j} \left\langle K_{x_j}, \sum_{k=1}^{n} K_{x_k} \sum_{l=1}^{n} (M^{-1})_{k,l} \left\langle K_{x_l}, \cdot \right\rangle_{\mathcal{H}} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} K_{x_i} \sum_{j=1}^{n} (M^{-1})_{i,j} \sum_{k=1}^{n} \left\langle K_{x_j}, K_{x_k} \right\rangle_{\mathcal{H}} \sum_{l=1}^{n} (M^{-1})_{k,l} \left\langle K_{x_l}, \cdot \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} K_{x_i} \sum_{j=1}^{n} (M^{-1})_{i,j} \sum_{k=1}^{n} M_{j,k} \sum_{l=1}^{n} (M^{-1})_{k,l} \left\langle K_{x_l}, \cdot \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} K_{x_i} \sum_{l=1}^{n} (M^{-1})_{i,l} \left\langle K_{x_l}, \cdot \right\rangle_{\mathcal{H}} \\
&= P.
\end{aligned}$$

P is also self-adjoint due to the symmetry of M,

$$\begin{aligned}
\langle Pf, g\rangle_{\mathcal{H}} &= \left\langle \sum_{i=1}^{n} K_{x_i} \sum_{j=1}^{n} (M^{-1})_{i,j} \left\langle K_{x_j}, f\right\rangle_{\mathcal{H}}, g \right\rangle_{\mathcal{H}} \\
&= \left\langle \sum_{i=1}^{n} \langle K_{x_i}, g\rangle_{\mathcal{H}} \sum_{j=1}^{n} (M^{-1})_{i,j} K_{x_j}, f \right\rangle_{\mathcal{H}} \\
&= \left\langle \sum_{j=1}^{n} K_{x_j} \sum_{i=1}^{n} (M^{-1})_{i,j} \langle K_{x_i}, g\rangle_{\mathcal{H}}, f \right\rangle_{\mathcal{H}} \\
&= \left\langle \sum_{j=1}^{n} K_{x_j} \sum_{i=1}^{n} (M^{-1})_{j,i} \langle K_{x_i}, g\rangle_{\mathcal{H}}, f \right\rangle_{\mathcal{H}} \\
&= \langle Pg, f\rangle_{\mathcal{H}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|f\|_{\mathcal{H}}^2 &= \langle f, f\rangle_{\mathcal{H}} \\
&= \langle Pf + (f - Pf), Pf + (f - Pf)\rangle_{\mathcal{H}} \\
&= \langle Pf, Pf\rangle_{\mathcal{H}} + 2\langle Pf, f - Pf\rangle_{\mathcal{H}} + \langle f - Pf, f - Pf\rangle_{\mathcal{H}} \\
&= \langle Pf, Pf\rangle_{\mathcal{H}} + 2\left\langle f, Pf - P^2 f\right\rangle_{\mathcal{H}} + \langle f - Pf, f - Pf\rangle_{\mathcal{H}} \\
&= \langle Pf, Pf\rangle_{\mathcal{H}} + \langle f - Pf, f - Pf\rangle_{\mathcal{H}} \\
&\geq \langle Pf, Pf\rangle_{\mathcal{H}} \\
&= \langle f, Pf\rangle_{\mathcal{H}}.
\end{aligned}$$

The latter term is nothing more than the left hand side in the statement. In summary the quantity in the statement of the theorem is just the square RKHS norm in the restriction of $\mathcal{H}$ to the subspace spanned by the functions K_{x_i}. □

References

R.J. Adler. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes*, volume 12 of *Lecture Notes–Monograph series*. Institute of Mathematical Statistics, 1990.

R.J. Adler and J.E. Taylor. *Random Fields and Geometry (Springer Monographs in Mathematics)*. Springer, 1 edition, June 2007. ISBN 0387481125.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.

B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 273–282, 2007.

A. Bertinet and Thomas C. Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

P. Billingsley. *Probability and Measure*. Wiley-Interscience, 3 edition, 1995.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

A. Charest. *Creation and Analysis of Differentially-Private Synthetic Datasets*. PhD thesis, Carnegie Mellon University, 2012.

K. Chaudhuri and D. Hsu. Convergence rates for differentially private statistical estimation. In *ICML'12*, 2012.

K. Chaudhuri and C. Monteleoni. Privacy preserving logistic regression. *NIPS 2008*, 2008.

K. Chaudhuri and C. Monteleoni. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.

S. Chawla, C. Dwork, F. McSherry, and K. Talwar. On the utility of privacy-preserving histograms. *UAI*, 2005.

C. Dwork. Differential privacy. *33rd International Colloquium on Automata, Languages and Programming*, pages 1–12, 2006.

C. Dwork and J. Lei. Differential privacy and robust statistics. *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 371–380, May–June 2009.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *EUROCRYPT*, pages 486–503, 2006a.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006b.

C. Dwork, G.N. Rothblum, and S. Vadhan. Boosting and differential privacy. *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium*, pages 51–60, 2010.

M. Hardt and K. Talwar. On the geometry of differential privacy. *STOC '10 Proceedings of the 42nd ACM Symposium on Theory of computing*, pages 705–714, 2010.

M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *Technical Report*, 2010.

S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008.

A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets Practice on the Map. *Proceedings of the 24th International Conference on Data Engineering*, pages 277–286, 2008.

A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan. The limits of two-party differential privacy. In *FOCS'10*, pages 81–90, 2010.

F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 627–636, New York, NY, USA, 2009. ACM.

F. McSherry and K. Talwar. Mechanism design via differential privacy. *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. *Proceedings of the 39th Annual ACM Symposium on the Theory of Computing*, pages 75–84, 2007.

E. Parzen. An approach to time series analysis. *The Annals of Mathematical Statistics*, 32(4): 951–989, 1961.

E. Parzen. Probability density functionals and reproducing kernel hilbert spaces. *Proceedings of the Symposium on Time Series Analysis*, 196:155–169, 1963.

J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer, 1997.

B.I.P. Rubinstein, P.L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 2010.

D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*. Wiley, 1992.

A. Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the 43rd Annual ACM Symposium on the Theory of Computing*, STOC '11, pages 813–822, New York, NY, USA, 2011. ACM.

L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

L. Wasserman and S. Zhou. A statistical framework for differential privacy. *The Journal of the American Statistical Association*, 105:375–389, 2010.

Journal of Machine Learning Research 14 (2013) 729-769 Submitted 3/12; Revised 11/12; Published 3/13

Sparsity Regret Bounds for Individual Sequences in Online Linear Regression*

Sébastien Gerchinovitz SEBASTIEN.GERCHINOVITZ@ENS.FR
École Normale Supérieure†
45 rue d'Ulm
Paris, FRANCE

Editor: Nicolò Cesa-Bianchi

Abstract

We consider the problem of online linear regression on arbitrary deterministic sequences when the ambient dimension d can be much larger than the number of time rounds T. We introduce the notion of *sparsity regret bound*, which is a deterministic online counterpart of recent risk bounds derived in the stochastic setting under a sparsity scenario. We prove such regret bounds for an online-learning algorithm called SeqSEW and based on exponential weighting and data-driven truncation. In a second part we apply a parameter-free version of this algorithm to the stochastic setting (regression model with random design). This yields risk bounds of the same flavor as in Dalalyan and Tsybakov (2012a) but which solve two questions left open therein. In particular our risk bounds are adaptive (up to a logarithmic factor) to the unknown variance of the noise if the latter is Gaussian. We also address the regression model with fixed design.

Keywords: sparsity, online linear regression, individual sequences, adaptive regret bounds

1. Introduction

Sparsity has been extensively studied in the stochastic setting over the past decade. This notion is key to address statistical problems that are high-dimensional, that is, where the number of unknown parameters is of the same order or even much larger than the number of observations. This is the case in many contemporary applications such as computational biology (e.g., analysis of DNA sequences), collaborative filtering (e.g., Netflix, Amazon), satellite and hyperspectral imaging, and high-dimensional econometrics (e.g., cross-country growth regression problems).

A key message about sparsity is that, although high-dimensional statistical inference is impossible in general (i.e., without further assumptions), it becomes statistically feasible if among the many unknown parameters, only few of them are non-zero. Such a situation is called a *sparsity scenario* and has been the focus of many theoretical, computational, and practical works over the past decade in the stochastic setting. On the theoretical side, most sparsity-related risk bounds take the form of the so-called *sparsity oracle inequalities*, that is, risk bounds expressed in terms of the number of non-zero coordinates of the oracle vector. As of now, such theoretical guarantees have only been proved under stochastic assumptions.[1]

*. A shorter version appeared in the proceedings of COLT 2011 (see Gerchinovitz 2011).

†. This research was carried out within the INRIA project CLASSIC hosted by École Normale Supérieure and CNRS.

1. One could object that most high-probability risk bounds derived for ℓ^1-regularization methods are in fact deterministic inequalities that hold true whenever the noise vector ε belong to some set S (see, e.g., Bickel et al. 2009). However,

In this paper we address the prediction possibilities under a sparsity scenario in both deterministic and stochastic settings. We first prove that theoretical guarantees similar to sparsity oracle inequalities can be obtained in a deterministic online setting, namely, online linear regression on individual sequences. The newly obtained deterministic prediction guarantees are called *sparsity regret bounds*. We prove such bounds for an online-learning algorithm which, in its most sophisticated version, is fully automatic in the sense that no preliminary knowledge is needed for the choice of its tuning parameters. In the second part of this paper, we apply our sparsity regret bounds—of deterministic nature—to the stochastic setting (regression model with random design). One of our key results is that, thanks to our online tuning techniques, these deterministic bounds imply sparsity oracle inequalities that are adaptive to the unknown variance of the noise (up to logarithmic factors) when the latter is Gaussian. In particular, this solves an open question raised by Dalalyan and Tsybakov (2012a).

In the next paragraphs, we introduce our main setting and motivate the notion of sparsity regret bound from an online-learning viewpoint. We then detail our main contributions with respect to the statistical literature and the machine-learning literature.

1.1 Introduction of a Deterministic Counterpart of Sparsity Oracle Inequalities

We consider the problem of online linear regression on arbitrary deterministic sequences. A forecaster has to predict in a sequential fashion the values $y_t \in \mathbb{R}$ of an unknown sequence of observations given some input data $x_t \in \mathcal{X}$ and some base forecasters $\varphi_j : \mathcal{X} \to \mathbb{R}$, $1 \leqslant j \leqslant d$, on the basis of which he outputs a prediction $\widehat{y}_t \in \mathbb{R}$. The quality of the predictions is assessed by the square loss. The goal of the forecaster is to predict almost as well as the best linear forecaster $u \cdot \varphi \triangleq \sum_{j=1}^{d} u_j \varphi_j$, where $u \in \mathbb{R}^d$, that is, to satisfy, uniformly over all individual sequences $(x_t, y_t)_{1 \leqslant t \leqslant T}$, a regret bound of the form

$$\sum_{t=1}^{T} \big(y_t - \widehat{y}_t\big)^2 \leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 + \Delta_{T,d}(u) \right\}$$

for some regret term $\Delta_{T,d}(u)$ that should be as small as possible and, in particular, sublinear in T. (For the sake of introduction, we omit the dependencies of $\Delta_{T,d}(u)$ on the amplitudes $\max_{1 \leqslant t \leqslant T} |y_t|$ and $\max_{1 \leqslant t \leqslant T} \max_{1 \leqslant j \leqslant d} |\varphi_j(x_t)|$.)

In this setting the version of the sequential ridge regression forecaster studied by Azoury and Warmuth (2001) and Vovk (2001) can be tuned to have a regret $\Delta_{T,d}(u)$ of order at most $d \ln\big(T \|u\|_2^2\big)$. When the ambient dimension d is much larger than the number of time rounds T, the latter regret bound may unfortunately be larger than T and is thus somehow trivial. Since the regret bound $d \ln T$ is optimal in a certain sense (see, e.g., the lower bound of Vovk 2001, Theorem 2), additional assumptions are needed to get interesting theoretical guarantees.

A natural assumption, which has already been extensively studied in the stochastic setting, is that there is a sparse vector u^* (i.e., with $s \ll T/(\ln T)$ non-zero coefficients) such that the linear combination $u^* \cdot \varphi$ has a small cumulative square loss. If the forecaster knew in advance the support $J(u^*) \triangleq \{j : u_j^* \neq 0\}$ of u^*, he could apply the same forecaster as above but only to the s-dimensional linear subspace $\big\{u \in \mathbb{R}^d : \forall j \notin J(u^*), u_j = 0\big\}$. The regret bound of this "oracle" would be roughly of order $s \ln T$ and thus sublinear in T. Under this sparsity scenario, a sublinear regret thus seems

the fact that $\varepsilon \in S$ with high-probability is only guaranteed via concentration arguments, so it is a consequence of the underlying statistical assumptions.

possible, though, of course, the aforementioned regret bound $s \ln T$ can only be used as an ideal benchmark (since the support of u^* is unknown).

In this paper, we prove that a regret bound proportional to s is achievable (up to logarithmic factors). In Corollary 2 and its refinements (Corollary 7 and Theorem 10), we indeed derive regret bounds of the form

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 + \big(\|u\|_0 + 1\big)\, g_{T,d}\big(\|u\|_1, \|\varphi\|_\infty\big) \right\} , \tag{1}$$

where $\|u\|_0$ denotes the number of non-zero coordinates of u and where g grows at most logarithmically in T, d, $\|u\|_1 \triangleq \sum_{j=1}^{d} |u_j|$, and $\|\varphi\|_\infty \triangleq \sup_{x \in \mathcal{X}} \max_{1 \leqslant j \leqslant d} |\varphi_j(x)|$. We call regret bounds of the above form *sparsity regret bounds*.

This work is in connection with several papers that belong either to the statistical or to the machine-learning literature. Next we discuss these papers and some related references.

1.2 Related Works in the Stochastic Setting

The above regret bound (1) can be seen as a deterministic online counterpart of the so-called *sparsity oracle inequalities* introduced in the stochastic setting in the past decade. The latter are risk bounds expressed in terms of the number of non-zero coordinates of the oracle vector—see (2) below. More formally, consider the regression model with random of fixed design. The forecaster observes independent random pairs $(X_1, Y_1), \ldots, (X_T, Y_T) \in \mathcal{X} \times \mathbb{R}$ given by

$$Y_t = f(X_t) + \varepsilon_t \ , \quad 1 \leqslant t \leqslant T \ ,$$

where the $X_t \in \mathcal{X}$ are either i.i.d random variables (random design) or fixed elements (fixed design), denoted in both cases by capital letters in this paragraph, and where the ε_t are i.i.d. square-integrable real random variables with zero mean (conditionally on the X_t if the design is random). The goal of the forecaster is to construct an estimator $\widehat{f}_T : \mathcal{X} \to \mathbb{R}$ of the unknown regression function $f : \mathcal{X} \to \mathbb{R}$ based on the sample $(X_t, Y_t)_{1 \leqslant t \leqslant T}$. Depending on the nature of the design, the performance of $\widehat{f}_T$ is measured through its risk $R\big(\widehat{f}_T\big)$:

$$R\big(\widehat{f}_T\big) \triangleq \begin{cases} \displaystyle\int_{\mathcal{X}} \Big(f(x) - \widehat{f}_T(x)\Big)^2 P^X(\mathrm{d}x) & \text{(random design)} \\ \displaystyle\frac{1}{T} \sum_{t=1}^{T} \big(f(X_t) - \widehat{f}_T(X_t)\big)^2 & \text{(fixed design),} \end{cases}$$

where P^X denotes the common distribution of the X_t if the design is random. With the above notations, and given a dictionary $\varphi = (\varphi_1, \cdots, \varphi_d)$ of base forecasters $\varphi_j : \mathcal{X} \to \mathbb{R}$ as previously, typical examples of *sparsity oracle inequalities* take approximately the form

$$R\big(\widehat{f}_T\big) \leqslant C \inf_{u \in \mathbb{R}^d} \left\{ R\big(u \cdot \varphi\big) + \frac{\|u\|_0 \ln d + 1}{T} \right\} \tag{2}$$

in expectation or with high probability, for some constant $C \geqslant 1$. Thus, sparsity oracle inequalities are risk bounds involving a trade-off between the risk $R(u \cdot \varphi)$ and the number of non-zero coordinates $\|u\|_0$ of any comparison vector $u \in \mathbb{R}^d$. In particular, they indicate that $\widehat{f}_T$ has a small risk

under a sparsity scenario, that is, if f is well approximated by a sparse linear combination $u^* \cdot \varphi$ of the base forecasters φ_j, $1 \leqslant j \leqslant d$.

Sparsity oracle inequalities were first derived by Birgé and Massart (2001) via ℓ^0-regularization methods (through model-selection arguments). Later works in this direction include, among many other papers, those of Birgé and Massart (2007), Abramovich et al. (2006), and Bunea et al. (2007a) in the regression model with fixed design and that of Bunea et al. (2004) in the random design case.

More recently, a large body of research has been dedicated to the analysis of ℓ^1-regularization methods, which are convex and thus computationally tractable variants of ℓ^0-regularization methods. A celebrated example is the Lasso estimator introduced by Tibshirani (1996) and Donoho and Johnstone (1994). Under some assumptions on the design matrix,[2] such methods have been proved to satisfy sparsity oracle inequalities of the form (2) (with $C = 1$ in the recent paper by Koltchinskii et al. 2011). A list of few references—but far from being comprehensive—includes the works of Bunea et al. (2007b), Candes and Tao (2007), van de Geer (2008), Bickel et al. (2009), Koltchinskii (2009a), Koltchinskii (2009b),Hebiri and van de Geer (2011), Koltchinskii et al. (2011) and Lounici et al. (2011). We refer the reader to the monograph by Bühlmann and van de Geer (2011) for a detailed account on ℓ^1-regularization.

A third line of research recently focused on procedures based on exponential weighting. Such methods were proved to satisfy sharp sparsity oracle inequalities (i.e., with leading constant $C = 1$), either in the regression model with fixed design (Dalalyan and Tsybakov, 2007, 2008; Rigollet and Tsybakov, 2011; Alquier and Lounici, 2011) or in the regression model with random design (Dalalyan and Tsybakov, 2012a; Alquier and Lounici, 2011). These papers show that a trade-off can be reached between strong theoretical guarantees (as with ℓ^0-regularization) and computational efficiency (as with ℓ^1-regularization). They indeed propose aggregation algorithms which satisfy sparsity oracle inequalities under almost no assumption on the base forecasters $(\varphi_j)_j$, and which can be approximated numerically at a reasonable computational cost for large values of the ambient dimension d.

Our online-learning algorithm SeqSEW is inspired from a statistical method of Dalalyan and Tsybakov (2008, 2012a). Following the same lines as in Dalalyan and Tsybakov (2012b), it is possible to slightly adapt the statement of our algorithm to make it computationally tractable by means of Langevin Monte-Carlo approximation—without affecting its statistical properties. The technical details are however omitted in this paper, which only focuses on the theoretical guarantees of the algorithm SeqSEW.

1.3 Previous Works on Sparsity in the Framework of Individual Sequences

To the best of our knowledge, Corollary 2 and its refinements (Corollary 7 and Theorem 10) provide the first examples of sparsity regret bounds in the sense of (1). To comment on the optimality of such regret bounds and compare them to related results in the framework of individual sequences, note that (1) can be rewritten in the equivalent form:

2. Despite their computational efficiency, the aforementioned ℓ^1-regularized methods still suffer from a drawback: their ℓ^0-oracle properties hold under rather restrictive assumptions on the design; namely, that the φ_j should be nearly orthogonal (see the detailed discussion in van de Geer and Bühlmann 2009).

For all $s \in \mathbb{N}$ and all $U > 0$,

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 - \inf_{\substack{\|u\|_0 \leqslant s \\ \|u\|_1 \leqslant U}} \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 \leqslant \big(s+1\big)\, g_{T,d}\big(U, \|\varphi\|_\infty\big) \;,$$

where g grows at most logarithmically in T, d, U, and $\|\varphi\|_\infty$. When $s \ll T$, this upper bound matches (up to logarithmic factors) the lower bound of order $s \ln T$ that follows in a straightforward manner from Theorem 2 of Vovk (2001). Indeed, if $s \ll T$, $\mathcal{X} = \mathbb{R}^d$, and $\varphi_j(x) = x_j$, then for any forecaster, there is an individual sequence $(x_t, y_t)_{1 \leqslant t \leqslant T}$ such that the regret of this forecaster on $\big\{u \in \mathbb{R}^d : \|u\|_0 \leqslant s \text{ and } \|u\|_1 \leqslant d\big\}$ is bounded from below by a quantity of order $s \ln T$. Therefore, up to logarithmic factors, any algorithm satisfying a sparsity regret bound of the form (1) is minimax optimal on intersections of ℓ^0-balls (of radii $s \ll T$) and ℓ^1-balls. This is in particular the case for our algorithm SeqSEW, but this contrasts with related works discussed below.

Recent works in the field of online convex optimization addressed the sparsity issue in the online deterministic setting, but from a quite different angle. They focus on algorithms which output sparse linear combinations, while we are interested in algorithms whose regret is small under a sparsity scenario, that is, on ℓ^0-balls of small radii. See, for example, the papers by Langford et al. (2009), Shalev-Shwartz and Tewari (2011), Xiao (2010), Duchi et al. (2010) and the references therein. All these articles focus on convex regularization. In the particular case of ℓ^1-regularization under the square loss, the aforementioned works propose algorithms which predict as a sparse linear combination $\widehat{y}_t = \widehat{u}_t \cdot \varphi(x_t)$ of the base forecasts (i.e., $\|\widehat{u}_t\|_0$ is small), while no such guarantee can be proved for our algorithm SeqSEW. However they prove bounds on the ℓ^1-regularized regret of the form

$$\sum_{t=1}^{T} \Big((y_t - \widehat{u}_t \cdot x_t)^2 + \lambda \|\widehat{u}_t\|_1 \Big) \leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^{T} \Big((y_t - u \cdot x_t)^2 + \lambda \|u\|_1 \Big) + \widetilde{\Delta}_{T,d}(u) \right\} \;, \tag{3}$$

for some regret term $\widetilde{\Delta}_{T,d}(u)$ which is suboptimal on intersections of ℓ^0- and ℓ^1-balls as explained below. The truncated gradient algorithm of Langford et al. (2009, Corollary 4.1) satisfies such a regret bound[3] with $\widetilde{\Delta}_{T,d}(u)$ at least of order $\|\varphi\|_\infty \sqrt{dT}$ when the base forecasts $\varphi_j(x_t)$ are dense in the sense that $\max_{1 \leqslant t \leqslant T} \sum_{j=1}^{d} \varphi_j^2(x_t) \approx d \, \|\varphi\|_\infty^2$. This regret bound grows as a power of and not logarithmically in d as is expected for sparsity regret bounds (recall that we are interested in the case when $d \gg T$).

The three other papers mentioned above do prove (some) regret bounds with a logarithmic dependence in d, but these bounds do not have the dependence in $\|u\|_1$ and T we are looking for. For $p - 1 \approx 1/(\ln d)$, the p-norm RDA method of Xiao (2010) and the algorithm SMIDAS of Shalev-Shwartz and Tewari (2011)—the latter being a particular case of the algorithm COMID of Duchi et al. (2010) specialized to the p-norm divergence—satisfy regret bounds of the above form (3) with

3. The bound stated in Langford et al. (2009, Corollary 4.1) differs from (3) in that the constant before the infimum is equal to $C = 1/(1 - 2c_d^2 \eta)$, where $c_d^2 \approx \max_{1 \leqslant t \leqslant T} \sum_{j=1}^{d} \varphi_j^2(x_t) \leqslant d \, \|\varphi\|_\infty^2$, and where a reasonable choice for η can easily be seen to be $\eta \approx 1/\sqrt{2c_d^2 T}$. If the base forecasts $\varphi_j(x_t)$ are dense in the sense that $c_d^2 \approx d \, \|\varphi\|_\infty^2$, then we have $C \approx 1 + \sqrt{2c_d^2/T}$, which yields a regret bound with leading constant 1 as in (3) and with $\widetilde{\Delta}_{T,d}(u)$ at least of order $\sqrt{c_d^2 T} \approx \|\varphi\|_\infty \sqrt{dT}$.

$\widetilde{\Delta}_{T,d}(u) \approx \mu \|u\|_1 \sqrt{T \ln d}$, for some gradient-based constant μ. Therefore, in all three cases, the function $\widetilde{\Delta}$ grows at least linearly in $\|u\|_1$ and as $\sqrt{T}$. This is in contrast with the logarithmic dependence in $\|u\|_1$ and the fast rate $O(\ln T)$ we are looking for and prove, for example, in Corollary 2.

Note that the suboptimality of the aforementioned algorithms is specific to the goal we are pursuing, that is, prediction on ℓ^0-balls (intersected with ℓ^1-balls). On the contrary the rate $\|u\|_1 \sqrt{T \ln d}$ is more suited and actually nearly optimal for learning on ℓ^1-balls (see Gerchinovitz and Yu 2011). Moreover, the predictions output by our algorithm SeqSEW are not necessarily sparse linear combinations of the base forecasts. A question left open is thus whether it is possible to design an algorithm which both ouputs sparse linear combinations (which is statistically useful and sometimes essential for computational issues) and satisfies a sparsity regret bound of the form (1).

1.4 PAC-Bayesian Analysis in the Framework of Individual Sequences

To derive our sparsity regret bounds, we follow a PAC-Bayesian approach combined with the choice of a sparsity-favoring prior. We do not have the space to review the PAC-Bayesian literature in the stochastic setting and only refer the reader to Catoni (2004) for a thorough introduction to the subject. As for the online deterministic setting, PAC-Bayesian-type inequalities were proved in the framework of prediction with expert advice, for example, by Freund et al. (1997) and Kivinen and Warmuth (1999), or in the same setting as ours with a Gaussian prior by Vovk (2001). More recently, Audibert (2009) proved a PAC-Bayesian result on individual sequences for general losses and prediction sets. The latter result relies on a unifying assumption called the online variance inequality, which holds true, for example, when the loss function is exp-concave. In the present paper, we only focus on the particular case of the square loss. We first use Theorem 4.6 of Audibert (2009) to derive a non-adaptive sparsity regret bound. We then provide an adaptive online PAC-Bayesian inequality to automatically adapt to the unknown range of the observations $\max_{1 \leqslant t \leqslant T} |y_t|$.

1.5 Application to the Stochastic Setting When the Noise Level Is Unknown

In Section 4.1 we apply an automatically-tuned version of our algorithm SeqSEW on i.i.d. data. Thanks to the standard online-to-batch conversion, our sparsity regret bounds—of deterministic nature—imply a sparsity oracle inequality of the same flavor as a result of Dalalyan and Tsybakov (2012a). However, our risk bound holds on the whole $\mathbb{R}^d$ space instead of ℓ^1-balls of finite radii, which solves one question left open by Dalalyan and Tsybakov (2012a, Section 4.2). Besides, and more importantly, our algorithm does not need the a priori knowledge of the variance of the noise when the latter is Gaussian. Since the noise level is unknown in practice, adapting to it is important. This solves a second question raised by Dalalyan and Tsybakov (2012a, Section 5.1, Remark 6).

1.6 Outline of the Paper

This paper is organized as follows. In Section 2 we describe our main (deterministic) setting as well as our main notations. In Section 3 we prove the aforementioned sparsity regret bounds for our algorithm SeqSEW, first when the forecaster has access to some a priori knowledge on the observations (Sections 3.1 and 3.2), and then when no a priori information is available (Section 3.3), which yields a fully automatic algorithm. In Section 4 we apply the algorithm SeqSEW to two stochastic settings: the regression model with random design (Section 4.1) and the regression model with fixed design (Section 4.2). Finally the appendix contains some proofs and several useful inequalities.

2. Setting and Notations

The main setting considered in this paper is an instance of the game of prediction with expert advice called *prediction with side information (under the square loss)* or, more simply, *online linear regression* (see Cesa-Bianchi and Lugosi 2006, Chapter 11 for an introduction to this setting). The data sequence $(x_t, y_t)_{t\geqslant 1}$ at hand is deterministic and arbitrary and we look for theoretical guarantees that hold for every *individual* sequence. We give in Figure 1 a detailed description of our online protocol.

Parameters: input data set $\mathcal{X}$, base forecasters $\varphi = (\varphi_1, \ldots, \varphi_d)$ with $\varphi_j : \mathcal{X} \to \mathbb{R}$, $1 \leqslant j \leqslant d$.

Initial step: the environment chooses a sequence of observations $(y_t)_{t\geqslant 1}$ in $\mathbb{R}$ and a sequence of input data $(x_t)_{t\geqslant 1}$ in $\mathcal{X}$ but the forecaster has not access to them.

At each time round $t \in \mathbb{N}^* \triangleq \{1, 2, \ldots\}$,

1. The environment reveals the input data $x_t \in \mathcal{X}$.
2. The forecaster chooses a prediction $\widehat{y}_t \in \mathbb{R}$
   (possibly as a linear combination of the $\varphi_j(x_t)$, but this is not necessary).
3. The environment reveals the observation $y_t \in \mathbb{R}$.
4. Each linear forecaster $u \cdot \varphi \triangleq \sum_{j=1}^d u_j \varphi_j$, $u \in \mathbb{R}^d$, incurs the loss $\big(y_t - u \cdot \varphi(x_t)\big)^2$ and the forecaster incurs the loss $(y_t - \widehat{y}_t)^2$.

Figure 1: The online linear regression setting.

Note that our online protocol is described as if the environment were oblivious to the forecaster's predictions. Actually, since we only consider deterministic forecasters, all regret bounds of this paper also hold when $(x_t)_{t\geqslant 1}$ and $(y_t)_{t\geqslant 1}$ are chosen by an adversarial environment.

Two stochastic batch settings are also considered later in this paper. See Section 4.1 for the regression model with random design, and Section 4.2 for the regression model with fixed design.

2.1 Some Notations

We now define some notations. We write $\mathbb{N} \triangleq \{0, 1, \ldots\}$ and $\mathrm{e} \triangleq \exp(1)$. Vectors in $\mathbb{R}^d$ will be denoted by bold letters. For all $u, v \in \mathbb{R}^d$, the standard inner product in $\mathbb{R}^d$ between $u = (u_1, \ldots, u_d)$ and $v = (v_1, \ldots, v_d)$ will be denoted by $u \cdot v = \sum_{i=j}^d u_j v_j$; the ℓ^0-, ℓ^1-, and ℓ^2-norms of $u = (u_1, \ldots, u_d)$ are respectively defined by

$$\|u\|_0 \triangleq \sum_{j=1}^d \mathbb{I}_{\{u_j \neq 0\}} = \big|\{j : u_j \neq 0\}\big| , \qquad \|u\|_1 \triangleq \sum_{j=1}^d |u_j| , \qquad \text{and} \quad \|u\|_2 \triangleq \left(\sum_{j=1}^d u_j^2\right)^{1/2} .$$

The set of all probability distributions on a set Θ (endowed with some σ-algebra, for example, the Borel σ-algebra when $\Theta = \mathbb{R}^d$) will be denoted by $\mathcal{M}_1^+(\Theta)$. For all $\rho, \pi \in \mathcal{M}_1^+(\Theta)$, the Kullback-Leibler divergence between ρ and π is defined by

$$\mathcal{K}(\rho,\pi) \triangleq \begin{cases} \displaystyle\int_{\mathbb{R}^d} \ln\left(\frac{\mathrm{d}\rho}{\mathrm{d}\pi}\right) \mathrm{d}\rho & \text{if } \rho \text{ is absolutely continuous with respect to } \pi; \\ +\infty & \text{otherwise,} \end{cases}$$

where $\frac{\mathrm{d}\rho}{\mathrm{d}\pi}$ denotes the Radon-Nikodym derivative of ρ with respect to π.

For all $x \in \mathbb{R}$ and $B > 0$, we denote by $\lceil x \rceil$ the smallest integer larger than or equal to x, and by $[x]_B$ its thresholded (or clipped) value:

$$[x]_B \triangleq \begin{cases} -B & \text{if } x < -B; \\ x & \text{if } -B \leqslant x \leqslant B; \\ B & \text{if } x > B. \end{cases}$$

Finally, we will use the (natural) conventions $1/0 = +\infty$, $(+\infty) \times 0 = 0$, and $0 \ln(1 + U/0) = 0$ for all $U \geqslant 0$. Any sum $\sum_{s=1}^{0} a_s$ indexed from 1 up to 0 is by convention equal to 0.

3. Sparsity Regret Bounds for Individual Sequences

In this section we prove sparsity regret bounds for different variants of our algorithm SeqSEW. We first assume in Section 3.1 that the forecaster has access in advance to a bound B_y on the observations $|y_t|$ and a bound B_Φ on the trace of the empirical Gram matrix. We then remove these requirements one by one in Sections 3.2 and 3.3.

3.1 Known Bounds B_y on the Observations and B_Φ on the Trace of the Empirical Gram Matrix

To simplify the analysis, we first assume that, at the beginning of the game, the number of rounds T is known to the forecaster and that he has access to a bound B_y on all the observations $y_1, \ldots, y_T$ and to a bound B_Φ on the trace of the empirical Gram matrix, that is,

$$y_1, \ldots, y_T \in [-B_y, B_y] \qquad \text{and} \qquad \sum_{j=1}^{d} \sum_{t=1}^{T} \varphi_j^2(x_t) \leqslant B_\Phi \ .$$

The first version of the algorithm studied in this paper is defined in Figure 2 (adaptive variants will be introduced later). We name it *SeqSEW* for it is a variant of the Sparse Exponential Weighting algorithm introduced in the stochastic setting by Dalalyan and Tsybakov (2007, 2008) which is tailored for the prediction of individual sequences.

The choice of the heavy-tailed prior π_τ is due to Dalalyan and Tsybakov (2007). The role of heavy-tailed priors to tackle the sparsity issue was already pointed out earlier; see, for example, the discussion by Seeger (2008, Section 2.1). In high dimension, such heavy-tailed priors favor sparsity: sampling from these prior distributions (or posterior distributions based on them) typically results in approximately sparse vectors, that is, vectors having most coordinates almost equal to zero and the few remaining ones with quite large values.

Parameters: threshold $B > 0$, inverse temperature $\eta > 0$, and prior scale $\tau > 0$ with which we associate the *sparsity prior* $\pi_\tau \in \mathcal{M}_1^+(\mathbb{R}^d)$ defined by

$$\pi_\tau(\mathrm{d}u) \triangleq \prod_{j=1}^{d} \frac{(3/\tau)\,\mathrm{d}u_j}{2\big(1+|u_j|/\tau\big)^4} \ .$$

Initialization: $p_1 \triangleq \pi_\tau$.

At each time round $t \geqslant 1$,

1. Get the input data x_t and predict[a] as $\widehat{y}_t \triangleq \displaystyle\int_{\mathbb{R}^d} \big[u \cdot \varphi(x_t)\big]_B \, p_t(\mathrm{d}u)$;

2. Get the observation y_t and compute the posterior distribution $p_{t+1} \in \mathcal{M}_1^+(\mathbb{R}^d)$ as

$$p_{t+1}(\mathrm{d}u) \triangleq \frac{\exp\left(-\eta \displaystyle\sum_{s=1}^{t} \Big(y_s - \big[u \cdot \varphi(x_s)\big]_B\Big)^2\right)}{W_{t+1}} \pi_\tau(\mathrm{d}u) \ ,$$

where

$$W_{t+1} \triangleq \int_{\mathbb{R}^d} \exp\left(-\eta \sum_{s=1}^{t} \Big(y_s - \big[v \cdot \varphi(x_s)\big]_B\Big)^2\right) \pi_\tau(\mathrm{d}v) \ .$$

a. The clipping operator $[\cdot]_B$ is defined in Section 2.

Figure 2: The algorithm $\mathrm{SeqSEW}_\tau^{B,\eta}$.

Proposition 1 *Assume that, for a known constant $B_y > 0$, the $(x_1,y_1),\dots,(x_T,y_T)$ are such that $y_1,\dots,y_T \in [-B_y,B_y]$. Then, for all $B \geqslant B_y$, all $\eta \leqslant 1/(8B^2)$, and all $\tau > 0$, the algorithm* $\mathrm{SeqSEW}_\tau^{B,\eta}$ *satisfies*

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{u\in\mathbb{R}^d} \left\{ \sum_{t=1}^{T} \big(y_t - u\cdot\varphi(x_t)\big)^2 + \frac{4}{\eta}\,\|u\|_0 \ln\!\left(1 + \frac{\|u\|_1}{\|u\|_0\,\tau}\right) \right\} + \tau^2 \sum_{j=1}^{d}\sum_{t=1}^{T} \varphi_j^2(x_t) \ . \tag{4}$$

Corollary 2 *Assume that, for some known constants $B_y > 0$ and $B_\Phi > 0$, the $(x_1,y_1),\dots,(x_T,y_T)$ are such that $y_1,\dots,y_T \in [-B_y,B_y]$ and $\sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) \leqslant B_\Phi$.*

Then, when used with $B = B_y$, $\eta = \dfrac{1}{8B_y^2}$, and $\tau = \sqrt{\dfrac{16B_y^2}{B_\Phi}}$, the algorithm $\mathrm{SeqSEW}_\tau^{B,\eta}$ *satisfies*

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{u\in\mathbb{R}^d} \left\{ \sum_{t=1}^{T} \big(y_t - u\cdot\varphi(x_t)\big)^2 + 32\,B_y^2\,\|u\|_0 \ln\left(1 + \frac{\sqrt{B_\Phi}\,\|u\|_1}{4\,B_y\,\|u\|_0}\right) \right\} + 16B_y^2 \ . \tag{5}$$

Note that, if $\|\varphi\|_\infty \triangleq \sup_{x\in\mathcal{X}} \max_{1\leqslant j\leqslant d} |\varphi_j(x)|$ is finite, then the last corollary provides a *sparsity regret bound* in the sense of (1). Indeed, in this case, we can take $B_\Phi = d\,T\,\|\varphi\|_\infty^2$, which yields a regret bound proportional to $\|u\|_0$ and that grows logarithmically in d, T, $\|u\|_1$, and $\|\varphi\|_\infty$.

To prove Proposition 1, we first need the following deterministic PAC-Bayesian inequality which is at the core of our analysis. It is a straightforward consequence of Theorem 4.6 of Audibert (2009) when applied to the square loss. An adaptive variant of this inequality will be provided in Section 3.2.

Lemma 3 *Assume that for some known constant $B_y > 0$, we have $y_1, \dots, y_T \in [-B_y, B_y]$.*

For all $\tau > 0$, if the algorithm $\mathrm{SeqSEW}_\tau^{B,\eta}$ is used with $B \geqslant B_y$ and $\eta \leqslant 1/(8B^2)$, then

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^{T} \Big(y_t - \big[u \cdot \varphi(x_t)\big]_B \Big)^2 \rho(\mathrm{d}u) + \frac{\mathcal{K}(\rho, \pi_\tau)}{\eta} \right\} \tag{6}$$

$$\leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t) \big)^2 \rho(\mathrm{d}u) + \frac{\mathcal{K}(\rho, \pi_\tau)}{\eta} \right\} . \tag{7}$$

Proof (of Lemma 3) Inequality (6) is a straightforward consequence of Theorem 4.6 of Audibert (2009) when applied to the square loss, the set of prediction functions $\mathcal{G} \triangleq \big\{ x \mapsto \big[u \cdot \varphi(x)\big]_B : u \in \mathbb{R}^d \big\}$, and the prior[4] $\widetilde{\pi}_\tau$ on $\mathcal{G}$ induced by the prior π_τ on $\mathbb{R}^d$ via the mapping $u \in \mathbb{R}^d \mapsto \big[u \cdot \varphi(\cdot)\big]_B \in \mathcal{G}$.

To apply the aforementioned theorem, recall from Cesa-Bianchi and Lugosi (2006, Section 3.3) that the square loss is $1/(8B^2)$-exp-concave on $[-B, B]$ and thus η-exp-concave,[5] since $\eta \leqslant 1/(8B^2)$ by assumption. Therefore, by Theorem 4.6 of Audibert (2009) with the variance function $\delta_\eta \equiv 0$ (see the comments following Remark 4.1 therein), we get

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{\mu \in \mathcal{M}_1^+(\mathcal{G})} \left\{ \int_{\mathcal{G}} \sum_{t=1}^{T} \big(y_t - g(x_t) \big)^2 \mu(\mathrm{d}g) + \frac{\mathcal{K}(\mu, \widetilde{\pi}_\tau)}{\eta} \right\}$$

$$\leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^{T} \Big(y_t - \big[u \cdot \varphi(x_t)\big]_B \Big)^2 \rho(\mathrm{d}u) + \frac{\mathcal{K}(\widetilde{\rho}, \widetilde{\pi}_\tau)}{\eta} \right\} ,$$

where the last inequality follows by restricting the infimum over $\mathcal{M}_1^+\big(\mathcal{G}\big)$ to the subset $\big\{\widetilde{\rho} : \rho \in \mathcal{M}_1^+(\mathbb{R}^d)\big\} \subset \mathcal{M}_1^+\big(\mathcal{G}\big)$, where $\widetilde{\rho} \in \mathcal{M}_1^+\big(\mathcal{G}\big)$ denotes the probability distribution induced by $\rho \in \mathcal{M}_1^+(\mathbb{R}^d)$ via the mapping $u \in \mathbb{R}^d \mapsto \big[u \cdot \varphi(\cdot)\big]_B \in \mathcal{G}$. Inequality (6) then follows from the fact that for all $\rho \in \mathcal{M}_1^+(\mathbb{R}^d)$, we have $\mathcal{K}(\widetilde{\rho}, \widetilde{\pi}_\tau) \leqslant \mathcal{K}(\rho, \pi_\tau)$ by joint convexity of $\mathcal{K}(\cdot, \cdot)$.

As for Inequality (7), it follows from (6) by noting that

$$\forall y \in [-B, B], \quad \forall x \in \mathbb{R}, \qquad \big| y - [x]_B \big| \leqslant |y - x| .$$

Therefore, truncation to $[-B, B]$ can only improve prediction under the square loss if the observations are $[-B, B]$-valued, which is the case here since by assumption $y_t \in [-B_y, B_y] \subset [-B, B]$ for all $t = 1, \dots, T$. ■

Remark 4 *As can be seen from the previous proof, if the prior π_τ used to define the algorithm SeqSEW was replaced with any prior $\pi \in \mathcal{M}_1^+(\mathbb{R}^d)$, then Lemma 3 would still hold true with π instead*

4. The set $\mathcal{G}$ is endowed with the σ-algebra generated by all the coordinate mappings $g \in \mathcal{G} \mapsto g(x) \in \mathbb{R}$, $x \in \mathcal{X}$ (where $\mathbb{R}$ is endowed with its Borel σ-algebra).

5. This means that for all $y \in [-B, B]$, the function $x \mapsto \exp\big(-\eta (y - x)^2\big)$ is concave on $[-B, B]$.

of π_τ. This fact is natural from a PAC-Bayesian perspective (see, e.g., Catoni, 2004; Dalalyan and Tsybakov, 2008). We only—but crucially—use the particular shape of the sparsity-favoring prior π_τ to derive Proposition 1 from the PAC-Bayesian bound (7).

Proof (of Proposition 1) Our proof mimics the proof of Theorem 5 by Dalalyan and Tsybakov (2008). We thus only write the outline of the proof and stress the minor changes that are needed to derive Inequality (4). The key technical tools provided by Dalalyan and Tsybakov (2008) are reproduced in Appendix B.2 for the convenience of the reader.

Let $u^* \in \mathbb{R}^d$. Since $B \geqslant B_y$ and $\eta \leqslant 1/(8B^2)$, we can apply Lemma 3 and get

$$\begin{aligned}\sum_{t=1}^T (y_t - \widehat{y}_t)^2 &\leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho(\mathrm{d}u) + \frac{\mathcal{K}(\rho, \pi_\tau)}{\eta} \right\} \\ &\leqslant \underbrace{\int_{\mathbb{R}^d} \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u)}_{(1)} + \underbrace{\frac{\mathcal{K}(\rho_{u^*,\tau}, \pi_\tau)}{\eta}}_{(2)} . \end{aligned} \tag{8}$$

In the last inequality, $\rho_{u^*,\tau}$ is taken as the translated of π_τ at u^*, namely,

$$\rho_{u^*,\tau}(\mathrm{d}u) \triangleq \frac{\mathrm{d}\pi_\tau}{\mathrm{d}u}(u - u^*)\,\mathrm{d}u = \prod_{j=1}^d \frac{(3/\tau)\,\mathrm{d}u_j}{2\big(1 + |u_j - u_j^*|/\tau\big)^4} .$$

The two terms (1) and (2) can be upper bounded as in the proof of Theorem 5 by Dalalyan and Tsybakov (2008). By a symmetry argument recalled in Lemma 22 (Appendix B.2), the first term (1) can be rewritten as

$$\int_{\mathbb{R}^d} \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u) = \sum_{t=1}^T \big(y_t - u^* \cdot \varphi(x_t)\big)^2 + \tau^2 \sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) . \tag{9}$$

As for the term (2), we have, as is recalled in Lemma 23,

$$\frac{\mathcal{K}(\rho_{u^*,\tau}, \pi_\tau)}{\eta} \leqslant \frac{4}{\eta} \|u^*\|_0 \ln\left(1 + \frac{\|u^*\|_1}{\|u^*\|_0 \tau}\right) . \tag{10}$$

Combining (8), (9), and (10), which all hold for all $u^* \in \mathbb{R}^d$, we get Inequality (4). ■

Proof (of Corollary 2) Applying Proposition 1, we have, since $B \geqslant B_y$ and $\eta \leqslant 1/(8B^2)$,

$$\begin{aligned}\sum_{t=1}^T (y_t - \widehat{y}_t)^2 &\leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 + \frac{4}{\eta} \|u\|_0 \ln\left(1 + \frac{\|u\|_1}{\|u\|_0 \tau}\right) \right\} + \tau^2 \sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) \\ &\leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 + \frac{4}{\eta} \|u\|_0 \ln\left(1 + \frac{\|u\|_1}{\|u\|_0 \tau}\right) \right\} + \tau^2 B_\Phi , \end{aligned}$$

since $\sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) \leqslant B_\Phi$ by assumption. The particular (and nearly optimal) choices of η and τ given in the statement of the corollary then yield the desired inequality (5). ■

We end this subsection with a natural question about approximate sparsity: Proposition 1 ensures a low regret with respect to sparse linear combinations $u \cdot \varphi$, but what can be said for approximately sparse linear combinations, that is, predictors of the form $u \cdot \varphi$ where $u \in \mathbb{R}^d$ is very close to a sparse vector? As can be seen from the proof of Lemma 23 in Appendix B.2, the sparsity-related term

$$\frac{4}{\eta} \|u\|_0 \ln\left(1 + \frac{\|u\|_1}{\|u\|_0 \tau}\right)$$

in the regret bound of Proposition 1 can actually be replaced with the smaller (and continous) term

$$\frac{4}{\eta} \sum_{j=1}^{d} \ln\left(1 + |u_j|/\tau\right) \ .$$

The last term is always smaller than the former and guarantees that the regret is small with respect to any approximately sparse vector $u \in \mathbb{R}^d$.

3.2 Unknown Bound B_y on the Observations but Known Bound B_Φ on the Trace of the Empirical Gram Matrix

In the previous section, to prove the upper bounds stated in Lemma 3 and Proposition 1, we assumed that the forecaster had access to a bound B_y on the observations $|y_t|$ and to a bound B_Φ on the trace of the empirical Gram matrix. In this section, we remove the first requirement and prove a sparsity regret bound for a variant of the algorithm $\mathrm{SeqSEW}_\tau^{B,\eta}$ which is adaptive to the unknown bound $B_y = \max_{1 \leqslant t \leqslant T} |y_t|$; see Proposition 5 and Remark 6 below.

For this purpose we consider the algorithm of Figure 3, which we call SeqSEW_τ^* thereafter. It differs from $\mathrm{SeqSEW}_\tau^{B,\eta}$ defined in the previous section in that the threshold B and the inverse temperature η are now allowed to vary over time and are chosen at each time round as a function of the data available to the forecaster.

The idea of truncating the base forecasts was used many times in the past; see, for example, the work of Vovk (2001) in the online linear regression setting, that of Györfi et al. (2002, Chapter 10) for the regression problem with random design, and the papers of Györfi and Ottucsák (2007) and Biau et al. (2010) for sequential prediction of unbounded time series under the square loss. A key ingredient in the present paper is to perform truncation with respect to a data-driven threshold.

Proposition 5 *For all $\tau > 0$, the algorithm SeqSEW_τ^* satisfies*

$$\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 + 32 B_{T+1}^2 \|u\|_0 \ln\left(1 + \frac{\|u\|_1}{\|u\|_0 \tau}\right) \right\} + \tau^2 \sum_{j=1}^{d} \sum_{t=1}^{T} \varphi_j^2(x_t) + 5 B_{T+1}^2 \ ,$$

where $B_{T+1}^2 \triangleq \max_{1 \leqslant t \leqslant T} y_t^2$.

Remark 6 In view of Proposition 1, the algorithm SeqSEW_τ^* satisfies a sparsity regret bound which is adaptive to the unknown bound $B_y = \max_{1 \leqslant t \leqslant T} |y_t|$. The price for the automatic tuning with respect to B_y consists only of the additive term $5B_{T+1}^2 = 5B_y^2$.

Parameter: prior scale $\tau > 0$ with which we associate the *sparsity prior* $\pi_\tau \in \mathcal{M}_1^+(\mathbb{R}^d)$ defined by

$$\pi_\tau(\mathrm{d}u) \triangleq \prod_{j=1}^{d} \frac{(3/\tau)\,\mathrm{d}u_j}{2\big(1+|u_j|/\tau\big)^4} \ .$$

Initialization: $B_1 \triangleq 0$, $\eta_1 \triangleq +\infty$, and $p_1 \triangleq \pi_\tau$.

At each time round $t \geqslant 1$,

1. Get the input data x_t and predict[a] as $\widehat{y}_t \triangleq \displaystyle\int_{\mathbb{R}^d} \big[u \cdot \varphi(x_t)\big]_{B_t} p_t(\mathrm{d}u)$;
2. Get the observation y_t and update:
   - the threshold $B_{t+1} \triangleq \max_{1 \leqslant s \leqslant t} |y_s|$,
   - the inverse temperature $\eta_{t+1} \triangleq 1/\big(8B_{t+1}^2\big)$,
   - and the posterior distribution $p_{t+1} \in \mathcal{M}_1^+(\mathbb{R}^d)$ as

$$p_{t+1}(\mathrm{d}u) \triangleq \frac{\exp\left(-\eta_{t+1}\displaystyle\sum_{s=1}^{t}\Big(y_s - \big[u\cdot\varphi(x_s)\big]_{B_s}\Big)^2\right)}{W_{t+1}}\,\pi_\tau(\mathrm{d}u) \ ,$$

where $$W_{t+1} \triangleq \int_{\mathbb{R}^d} \exp\left(-\eta_{t+1}\sum_{s=1}^{t}\Big(y_s - \big[v\cdot\varphi(x_s)\big]_{B_s}\Big)^2\right)\pi_\tau(\mathrm{d}v) \ .$$

a. The clipping operator $[\cdot]_B$ is defined in Section 2.

Figure 3: The algorithm SeqSEW_τ^*.

As in the previous section, several corollaries can be derived from Proposition 5. If the forecaster has access beforehand to a quantity $B_\Phi > 0$ such that $\sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) \leqslant B_\Phi$, then a suboptimal but reasonable choice of τ is given by $\tau = 1/\sqrt{B_\Phi}$; see Corollary 7 below. The simpler tuning $\tau = 1/\sqrt{dT}$ of Corollary 8 will be useful in the stochastic batch setting (cf., Section 4).[6] The proofs of the next corollaries are immediate.

Corollary 7 *Assume that, for a known constant $B_\Phi > 0$, the $(x_1,y_1),\dots,(x_T,y_T)$ are such that $\sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) \leqslant B_\Phi$. Then, when used with $\tau = 1/\sqrt{B_\Phi}$, the algorithm* SeqSEW_τ^* *satisfies*

$$\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2 \leqslant \inf_{u\in\mathbb{R}^d}\left\{\sum_{t=1}^{T}\big(y_t-u\cdot\varphi(x_t)\big)^2 + 32B_{T+1}^2\,\|u\|_0 \ln\left(1+\frac{\sqrt{B_\Phi}\,\|u\|_1}{\|u\|_0}\right)\right\} + 5B_{T+1}^2 + 1 \ ,$$

6. The tuning $\tau = 1/\sqrt{dT}$ only uses the knowledge of T, which is known by the forecaster in the stochastic batch setting. In that framework, another simple and easy-to-analyse tuning is given by $\tau = 1/(\|\varphi\|_\infty\sqrt{dT})$—which corresponds to $B_\Phi = dT\,\|\varphi\|_\infty^2$—but it requires that $\|\varphi\|_\infty \triangleq \sup_{x\in\mathcal{X}} \max_{1\leqslant j\leqslant d} |\varphi_j(x)|$ be finite. Note that the last tuning satisfies the scale-invariant property pointed out by Dalalyan and Tsybakov (2012a, Remark 4).

where $B_{T+1}^2 \triangleq \max_{1 \leqslant t \leqslant T} y_t^2$.

Corollary 8 *Assume that T is known to the forecaster at the beginning of the prediction game. Then, when used with $\tau = 1/\sqrt{dT}$, the algorithm* SeqSEW$^*_\tau$ *satisfies*

$$\sum_{t=1}^T (y_t - \widehat{y}_t)^2 \leqslant \inf_{u \in \mathbb{R}^d} \left\{ \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 + 32 B_{T+1}^2 \, \|u\|_0 \ln\left(1 + \frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right) \right\} + \frac{1}{dT} \sum_{j=1}^d \sum_{t=1}^T \varphi_j^2(x_t) + 5 B_{T+1}^2 \;,$$

where $B_{T+1}^2 \triangleq \max_{1 \leqslant t \leqslant T} y_t^2$.

As in the previous section, to prove Proposition 5, we first need a key PAC-Bayesian inequality. The next lemma is an adaptive variant of Lemma 3.

Lemma 9 *For all $\tau > 0$, the algorithm* SeqSEW$^*_\tau$ *satisfies*

$$\sum_{t=1}^T (y_t - \widehat{y}_t)^2 \leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^T \Big(y_t - \big[u \cdot \varphi(x_t)\big]_{B_t}\Big)^2 \rho(\mathrm{d}u) + 8 B_{T+1}^2 \, \mathcal{K}(\rho, \pi_\tau) \right\} + 4 B_{T+1}^2 \tag{11}$$

$$\leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho(\mathrm{d}u) + 8 B_{T+1}^2 \, \mathcal{K}(\rho, \pi_\tau) \right\} + 5 B_{T+1}^2 \;, \tag{12}$$

where $B_{T+1}^2 \triangleq \max_{1 \leqslant t \leqslant T} y_t^2$.

Proof (of Lemma 9) The proof is based on arguments that are similar to those underlying Lemma 3, except that we now need to deal with B and η changing over time. In the same spirit as in Auer et al. (2002), Cesa-Bianchi et al. (2007) and Györfi and Ottucsák (2007), our analysis relies on the control of $(\ln W_{t+1})/\eta_{t+1} - (\ln W_t)/\eta_t$ where $W_1 \triangleq 1$ and, for all $t \geqslant 2$,

$$W_t \triangleq \int_{\mathbb{R}^d} \exp\left(-\eta_t \sum_{s=1}^{t-1} \Big(y_s - \big[u \cdot \varphi(x_s)\big]_{B_s}\Big)^2 \right) \pi_\tau(\mathrm{d}u) \;.$$

Before controlling $(\ln W_{t+1})/\eta_{t+1} - (\ln W_t)/\eta_t$, we first need a little comment. Note that all η_t's such that $\eta_t = +\infty$ (i.e., $B_t = 0$) can be replaced with any finite value without changing the predictions of the algorithm (since the sum $\sum_{s=1}^{t-1}$ above equals zero). Therefore, we assume in the sequel that $(\eta_t)_{t \geqslant 1}$ is a non-decreasing sequence of *finite* positive real numbers.

First step: On the one hand, we have

$$\frac{\ln W_{T+1}}{\eta_{T+1}} - \frac{\ln W_1}{\eta_1} = \frac{1}{\eta_{T+1}} \ln \int_{\mathbb{R}^d} \exp\left(-\eta_{T+1} \sum_{t=1}^T \Big(y_t - \big[u \cdot \varphi(x_t)\big]_{B_t}\Big)^2 \right) \pi_\tau(\mathrm{d}u) - \frac{1}{\eta_1} \ln 1$$

$$= - \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)} \left\{ \int_{\mathbb{R}^d} \sum_{t=1}^T \Big(y_t - \big[u \cdot \varphi(x_t)\big]_{B_t}\Big)^2 \rho(\mathrm{d}u) + \frac{\mathcal{K}(\rho, \pi_\tau)}{\eta_{T+1}} \right\} \;, \tag{13}$$

where the last equality follows from a convex duality argument for the Kullback-Leibler divergence (cf., e.g., Catoni 2004, p. 159) which we recall in Proposition 21 in Appendix B.1.

Second step: On the other hand, we can rewrite $(\ln W_{T+1})/\eta_{T+1} - (\ln W_1)/\eta_1$ as a telescopic sum and get

$$\frac{\ln W_{T+1}}{\eta_{T+1}} - \frac{\ln W_1}{\eta_1} = \sum_{t=1}^{T} \left(\frac{\ln W_{t+1}}{\eta_{t+1}} - \frac{\ln W_t}{\eta_t} \right) = \sum_{t=1}^{T} \Bigg(\underbrace{\frac{\ln W_{t+1}}{\eta_{t+1}} - \frac{\ln W'_{t+1}}{\eta_t}}_{(1)} + \underbrace{\frac{1}{\eta_t} \ln \frac{W'_{t+1}}{W_t}}_{(2)} \Bigg) , \tag{14}$$

where W'_{t+1} is obtained from W_{t+1} by replacing η_{t+1} with η_t; namely,

$$W'_{t+1} \triangleq \int_{\mathbb{R}^d} \exp\left(-\eta_t \sum_{s=1}^{t} \Big(y_s - \big[u \cdot \varphi(x_s)\big]_{B_s} \Big)^2 \right) \pi_\tau(\mathrm{d}u) \, .$$

Let $t \in \{1, \dots, T\}$. The first term (1) is non-positive by Jensen's inequality (note that $x \mapsto x^{\eta_{t+1}/\eta_t}$ is concave on $\mathbb{R}_+^*$ since $\eta_{t+1} \leqslant \eta_t$ by construction). As for the second term (2), by definition of W'_{t+1},

$$\begin{aligned}
&\frac{1}{\eta_t} \ln \frac{W'_{t+1}}{W_t} \\
&\quad = \frac{1}{\eta_t} \ln \int_{\mathbb{R}^d} \frac{\exp\left(-\eta_t \Big(y_t - \big[u \cdot \varphi(x_t)\big]_{B_t} \Big)^2 \right) \exp\left(-\eta_t \sum_{s=1}^{t-1} \Big(y_s - \big[u \cdot \varphi(x_s)\big]_{B_s} \Big)^2 \right)}{W_t} \pi_\tau(\mathrm{d}u) \\
&\quad = \frac{1}{\eta_t} \ln \int_{\mathbb{R}^d} \exp\left(-\eta_t \Big(y_t - \big[u \cdot \varphi(x_t)\big]_{B_t} \Big)^2 \right) p_t(\mathrm{d}u) \, .
\end{aligned} \tag{15}$$

where (15) follows by definition of p_t. The next paragraphs are dedicated to upper bounding the last integral above. First note that this is straightforward in the particular case where $y_t \in [-B_t, B_t]$. Indeed, by definition of $\eta_t \triangleq 1/(8B_t^2)$ and by the fact that the square loss is $1/(8B_t^2)$-exp-concave on $[-B_t, B_t]$ (as in Lemma 3),[7] we get from Jensen's inequality that

$$\int_{\mathbb{R}^d} e^{-\eta_t \left(y_t - \left[u \cdot \varphi(x_t)\right]_{B_t} \right)^2} p_t(\mathrm{d}u) \leqslant \exp\left(-\eta_t \left(y_t - \int_{\mathbb{R}^d} \big[u \cdot \varphi(x_t)\big]_{B_t} p_t(\mathrm{d}u) \right)^2 \right) = e^{-\eta_t (y_t - \widehat{y}_t)^2} ,$$

where the last equality follows by definition of $\widehat{y}_t$. Taking the logarithms of both sides of the last inequality and dividing by η_t, we can see that the quantity on the right-hand side of (15) is bounded from above by $-\big(y_t - \widehat{y}_t\big)^2$.

In the general case, we cannot assume that $y_t \in [-B_t, B_t]$, since it may happen that $|y_t| > \max_{1 \leqslant s \leqslant t-1} |y_s| \triangleq B_t$. As shown below, we can still use the exp-concavity of the square loss if we replace y_t with its clipped version $[y_t]_{B_t}$. More precisely, setting $\widehat{y}_{t,u} \triangleq [u \cdot \varphi(x_t)]_{B_t}$ for all $u \in \mathbb{R}^d$, the square loss appearing in the right-hand side of (15) equals

$$\begin{aligned}
\big(y_t - \widehat{y}_{t,u}\big)^2 &= \big([y_t]_{B_t} - \widehat{y}_{t,u}\big)^2 + \big(y_t - [y_t]_{B_t}\big)^2 + 2\big(y_t - [y_t]_{B_t}\big)\big([y_t]_{B_t} - \widehat{y}_{t,u}\big) \\
&= \big([y_t]_{B_t} - \widehat{y}_{t,u}\big)^2 + \big(y_t - [y_t]_{B_t}\big)^2 + 2\big(y_t - [y_t]_{B_t}\big)\big([y_t]_{B_t} - \widehat{y}_t\big) + c_{t,u} \, ,
\end{aligned} \tag{16}$$

7. To be more exact, we assigned some arbitrary finite value to η_t when $B_t = 0$. However, in this case, the square loss is of course η_t-exp-concave on $[-B_t, B_t] = \{0\}$ whatever the value of η_t.

where we set

$$\begin{aligned} c_{t,u} &\triangleq 2\big(y_t - [y_t]_{B_t}\big)\big(\widehat{y}_t - \widehat{y}_{t,u}\big) \\ &\geqslant -4B_t\big|y_t - [y_t]_{B_t}\big| \geqslant -4B_t(B_{t+1} - B_t)\ , \end{aligned} \tag{17}$$

where the last two inequalities follow from the property $\widehat{y}_t, \widehat{y}_{t,u} \in [-B_t, B_t]$ (by construction) and from the elementary[8] yet useful upper bound $\big|y_t - [y_t]_{B_t}\big| \leqslant B_{t+1} - B_t$.

Combining (16) with the lower bound (17) yields that, for all $u \in \mathbb{R}^d$,

$$\big(y_t - \widehat{y}_{t,u}\big)^2 \geqslant \big([y_t]_{B_t} - \widehat{y}_{t,u}\big)^2 + C_t\ , \tag{18}$$

where we set $C_t \triangleq \big(y_t - [y_t]_{B_t}\big)^2 + 2\big(y_t - [y_t]_{B_t}\big)\big([y_t]_{B_t} - \widehat{y}_t\big) - 4B_t(B_{t+1} - B_t)$.

We can now continue the upper bounding of $(1/\eta_t)\ln(W'_{t+1}/W_t)$. Indeed, substituting the lower bound (18) into (15), we get that

$$\begin{aligned} \frac{1}{\eta_t}\ln\frac{W'_{t+1}}{W_t} &\leqslant \frac{1}{\eta_t}\ln\left[\int_{\mathbb{R}^d}\exp\Big(-\eta_t\big([y_t]_{B_t} - \widehat{y}_{t,u}\big)^2\Big)\,p_t(\mathrm{d}u)\right] - C_t \\ &\leqslant \frac{1}{\eta_t}\ln\left[\exp\left(-\eta_t\left([y_t]_{B_t} - \int_{\mathbb{R}^d}\widehat{y}_{t,u}\,p_t(\mathrm{d}u)\right)^2\right)\right] - C_t \qquad (19) \\ &= -\big([y_t]_{B_t} - \widehat{y}_t\big)^2 - C_t \qquad (20) \\ &= -\big(y_t - \widehat{y}_t\big)^2 + 4B_t(B_{t+1} - B_t)\ , \qquad (21) \end{aligned}$$

where (19) follows by Jensen's inequality (recall that $\eta_t \triangleq 1/(8B_t^2)$ and that the square loss is $1/(8B_t^2)$-exp-concave on $[-B_t, B_t]$),[9] where (20) is entailed by definition of $\widehat{y}_{t,u}$ and $\widehat{y}_t$, and where (21) follows by definition of C_t above and by elementary calculations.

Summing (21) over $t = 1, \ldots, T$ and using the upper bound $B_t(B_{t+1} - B_t) \leqslant B_{t+1}^2 - B_t^2$, Equation (14) yields

$$\begin{aligned} \frac{\ln W_{T+1}}{\eta_{T+1}} - \frac{\ln W_1}{\eta_1} &\leqslant -\sum_{t=1}^T (y_t - \widehat{y}_t)^2 + 4\sum_{t=1}^T\big(B_{t+1}^2 - B_t^2\big) \\ &= -\sum_{t=1}^T (y_t - \widehat{y}_t)^2 + 4B_{T+1}^2\ . \end{aligned} \tag{22}$$

Third step: Putting (13) and (22) together, we get the PAC-Bayesian inequality

$$\sum_{t=1}^T (y_t - \widehat{y}_t)^2 \leqslant \inf_{\rho \in \mathcal{M}_1^+(\mathbb{R}^d)}\left\{\int_{\mathbb{R}^d}\sum_{t=1}^T\Big(y_t - \big[u\cdot\varphi(x_t)\big]_{B_t}\Big)^2\rho(\mathrm{d}u) + \frac{\mathcal{K}(\rho,\pi_\tau)}{\eta_{T+1}}\right\} + 4B_{T+1}^2\ ,$$

which yields (11) since $\eta_{T+1} \triangleq 1/(8B_{T+1}^2)$ by definition.[10] The other PAC-Bayesian inequality (12), which is stated for non-truncated base forecasts, is a direct consequence of (11) and of the following two arguments: for all $u \in \mathbb{R}^d$ and all $t = 1, \ldots, T$,

$$\big(y_t - [u\cdot\varphi(x_t)]_{B_t}\big)^2 \leqslant \big(y_t - u\cdot\varphi(x_t)\big)^2 + (B_{t+1} - B_t)^2 \tag{23}$$

8. To see why this is true, it suffices to rewrite $[y_t]_{B_t}$ in the three cases $y_t < -B_t$, $|y_t| \leqslant B_t$, or $y_t > B_t$.
9. Same remark as in Footnote 7.
10. If $B_{T+1} = 0$, then $y_t = \widehat{y}_t = 0$ for all $1 \leqslant t \leqslant T$, which immediately yields (11).

and

$$\sum_{t=1}^{T}(B_{t+1}-B_t)^2 \leqslant B_{T+1}^2 \ . \tag{24}$$

Complement: proof of (23) and (24).
To see why (23) is true, we can distinguish between several cases. First note that this inequality is straightforward when $|y_t| \leqslant B_t$ (indeed, in this case, clipping $u\cdot\varphi(x_t)$ to $[-B_t,B_t]$ can only improve prediction). We can thus assume that $|y_t| > B_t$, or just[11] that $y_t > B_t$. In this case, we can distinguish between three sub-cases:

- if $u\cdot\varphi(x_t) < -B_t$, then clipping improves prediction since $y_t > B_t$;
- if $-B_t \leqslant u\cdot\varphi(x_t) \leqslant B_t$, then the clipping operator $[\cdot]_B$ has no effect on $u\cdot\varphi(x_t)$;
- if $u\cdot\varphi(x_t) > B_t$, then $[u\cdot\varphi(x_t)]_{B_t} = B_t$ so that $(y_t - [u\cdot\varphi(x_t)]_{B_t})^2 = (B_{t+1}-B_t)^2$ since $B_{t+1} = y_t$.

Therefore, in all three sub-cases described above, we have

$$(y_t - [u\cdot\varphi(x_t)]_{B_t})^2 \leqslant \max\left\{(y_t - u\cdot\varphi(x_t))^2, (B_{t+1}-B_t)^2\right\} \ ,$$

which concludes the proof of (23). As for (24), it follows from the inequality

$$\sum_{t=1}^{T}(B_{t+1}-B_t)^2 \leqslant \sup_{\substack{\Delta_1,\dots,\Delta_T\geqslant 0\\ \sum_{t=1}^T\Delta_t = B_{T+1}}}\left\{\sum_{t=1}^{T}\Delta_t^2\right\} = B_{T+1}^2 \ ,$$

where the last equality is entailed by convexity of the function $(\Delta_1,\dots,\Delta_T)\mapsto\sum_{t=1}^T\Delta_t^2$ on the polytope $\left\{(\Delta_1,\dots,\Delta_T)\in\mathbb{R}_+^T : \sum_{t=1}^T\Delta_t = B_{T+1}\right\}$. This concludes the proof. ■

Proof (of Proposition 5) The proof follows exactly the same lines as in Proposition 1 except that we apply Lemma 9 instead of Lemma 3. Indeed, using Lemma 9 and restricting the infimum to the $\rho_{u^*,\tau}$, $u^*\in\mathbb{R}^d$ (cf., (40)), we get that

$$\begin{aligned}\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2 &\leqslant \inf_{u^*\in\mathbb{R}^d}\left\{\int_{\mathbb{R}^d}\sum_{t=1}^{T}\big(y_t - u\cdot\varphi(x_t)\big)^2\rho_{u^*,\tau}(\mathrm{d}u) + 8B_{T+1}^2\,\mathcal{K}(\rho_{u^*,\tau},\pi_\tau)\right\} + 5B_{T+1}^2\\ &\leqslant \inf_{u^*\in\mathbb{R}^d}\left\{\sum_{t=1}^{T}\big(y_t - u^*\cdot\varphi(x_t)\big)^2 + 32B_{T+1}^2\,\|u^*\|_0\ln\left(1+\frac{\|u^*\|_1}{\|u^*\|_0\,\tau}\right)\right\}\\ &\quad + \tau^2\sum_{j=1}^{d}\sum_{t=1}^{T}\varphi_j^2(x_t) + 5B_{T+1}^2 \ ,\end{aligned}$$

where the last inequality follows from Lemmas 22 and 23. ■

11. If $y_t < -B_t$, it suffices to apply (23) with $-y_t$ and $-u$.

3.3 A Fully Automatic Algorithm

In the previous section, we proved that adaptation to B_y was possible. If we also no longer assume that a bound B_Φ on the trace of the empirical Gram matrix is available to the forecaster, then we can use a doubling trick on the nondecreasing quantity

$$\gamma_t \triangleq \ln\left(1+\sqrt{\sum_{s=1}^{t}\sum_{j=1}^{d}\varphi_j^2(x_s)}\right)$$

and repeatedly run the algorithm SeqSEW$^*_\tau$ of the previous section for rapidly-decreasing values of τ. This yields a sparsity regret bound with extra logarithmic multiplicative factors as compared to Proposition 5, but which holds for a fully automatic algorithm; see Theorem 10 below.

More formally, our algorithm SeqSEW$^*_*$ is defined as follows. The set of all time rounds $t = 1,2,\ldots$ is partitioned into regimes $r = 0,1,\ldots$ whose final time instances t_r are data-driven. Let $t_{-1} \triangleq 0$ by convention. We call *regime* r, $r = 0,1,\ldots$, the sequence of time rounds $(t_{r-1}+1,\ldots,t_r)$ where t_r is the first date $t \geqslant t_{r-1}+1$ such that $\gamma_t > 2^r$. At the beginning of regime r, we restart the algorithm SeqSEW$^*_\tau$ defined in Figure 3 with the parameter τ set to $\tau_r \triangleq 1/\big(\exp(2^r)-1\big)$.

In particular, on each regime r, the current instance of the algorithm SeqSEW$^*_{\tau_r}$ only uses the past observations y_s, $s \in \{t_{r-1}+1,\ldots,t-1\}$, to perform the online trunction and to tune the inverse temperature parameter. Therefore, the algorithm SeqSEW$^*_*$ is fully automatic.

Theorem 10 *Without requiring any preliminary knowledge at the beginning of the prediction game,* SeqSEW$^*_*$ *satisfies, for all* $T \geqslant 1$ *and all* $(x_1,y_1),\ldots,(x_T,y_T) \in \mathcal{X}\times\mathbb{R}$,

$$\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2 \leqslant \inf_{u\in\mathbb{R}^d}\left\{\sum_{t=1}^{T}\big(y_t-u\cdot\varphi(x_t)\big)^2 + 128\Big(\max_{1\leqslant t\leqslant T}y_t^2\Big)\|u\|_0\ln\left(e+\sqrt{\sum_{t=1}^{T}\sum_{j=1}^{d}\varphi_j^2(x_t)}\right)\right.$$
$$\left. + 32\Big(\max_{1\leqslant t\leqslant T}y_t^2\Big)A_T\|u\|_0\ln\left(1+\frac{\|u\|_1}{\|u\|_0}\right)\right\} + \Big(1+9\max_{1\leqslant t\leqslant T}y_t^2\Big)A_T\ ,$$

where $A_T \triangleq 2+\log_2\ln\Big(e+\sqrt{\sum_{t=1}^{T}\sum_{j=1}^{d}\varphi_j^2(x_t)}\Big)$.

Though the algorithm SeqSEW$^*_*$ is fully automatic, two possible improvements could be addressed in the future. From a theoretical viewpoint, can we contruct a fully automatic algorithm with a bound similar to Theorem 10 but without the extra logarithmic factor A_T? From a practical viewpoint, is it possible to perform the adaptation to B_Φ without restarting the algorithm repeatedly (just like we did for B_y)? A smoother time-varying tuning $(\tau_t)_{t\geqslant 2}$ might enable to answer both questions. This would be very probably at the price of a more involved analysis (e.g., if we adapt the PAC-Bayesian bound of Lemma 9, then a third approximation term would appear in (14) since π_{τ_t} changes over time).

Proof sketch (of Theorem 10) The proof relies on the use of Corollary 7 on all regimes r visited up to time T. More precisely, note that $\gamma_{t_r-1} \leqslant 2^r$ by definition of t_r (except maybe in the trivial case when $t_r = t_{r-1}+1$), which entails that

$$\sum_{t=t_{r-1}+1}^{t_r-1}\sum_{j=1}^{d}\varphi_j^2(x_t) \leqslant \Big(e^{2^r}-1\Big)^2 \triangleq B_{\Phi,r}\ .$$

Since we tuned the instance of the algorithm SeqSEW$^*_\tau$ on regime r with $\tau = \tau_r \triangleq 1/\sqrt{B_{\Phi,r}}$, we can apply Corollary 7 on regime r for all r. Summing the corresponding regret bounds over r then yields the desired result. See Appendix A.1 for a detailed proof. ■

Theorem 10 yields the following corollary. It upper bounds the regret of the algorithm SeqSEW$^*_*$ uniformly over all $u \in \mathbb{R}^d$ such that $\|u\|_0 \leqslant s$ and $\|u\|_1 \leqslant U$, where the sparsity level $s \in \mathbb{N}$ and the ℓ^1-diameter $U > 0$ are both unknown to the forecaster. The proof is postponed to Appendix A.1.

Corollary 11 *Fix $s \in \mathbb{N}$ and $U > 0$. Then, for all $T \geqslant 1$ and all $(x_1,y_1),\dots,(x_T,y_T) \in \mathcal{X} \times \mathbb{R}$, the regret of the algorithm* SeqSEW$^*_*$ *on $\big\{u \in \mathbb{R}^d : \|u\|_0 \leqslant s \text{ and } \|u\|_1 \leqslant U\big\}$ is bounded by*

$$\sum_{t=1}^T (y_t - \widehat{y}_t)^2 - \inf_{\substack{\|u\|_0 \leqslant s \\ \|u\|_1 \leqslant U}} \sum_{t=1}^T \big(y_t - u \cdot \varphi(x_t)\big)^2$$
$$\leqslant 128 \Big(\max_{1 \leqslant t \leqslant T} y_t^2\Big)\, s \ln\left(e + \sqrt{\sum_{t=1}^T \sum_{j=1}^d \varphi_j^2(x_t)}\right) + 32\Big(\max_{1 \leqslant t \leqslant T} y_t^2\Big) A_T\, s \ln\left(1 + \frac{U}{s}\right)$$
$$+ \Big(1 + 9 \max_{1 \leqslant t \leqslant T} y_t^2\Big) A_T \ ,$$

where $A_T \triangleq 2 + \log_2 \ln\Big(e + \sqrt{\sum_{t=1}^T \sum_{j=1}^d \varphi_j^2(x_t)}\Big)$.

4. Adaptivity to the Unknown Variance in the Stochastic Setting

In this section, we apply the online algorithm SeqSEW$^*_\tau$ of Section 3.2 to two related stochastic settings: the regression model with random design (Section 4.1) and the regression model with fixed design (Section 4.2). The sparsity regret bounds proved for this algorithm on individual sequences imply in both settings sparsity oracle inequalities with leading constant 1. These risk bounds are of the same flavor as in Dalalyan and Tsybakov (2008, 2012a) but they are adaptive (up to a logarithmic factor) to the unknown variance σ^2 of the noise if the latter is Gaussian. In particular, we solve two questions left open by Dalalyan and Tsybakov (2012a) in the random design case.

In the sequel, just like in the online deterministic setting, we assume that the forecaster has access to a dictionary $\varphi = (\varphi_1,\dots,\varphi_d)$ of measurable base forecasters $\varphi_j : \mathcal{X} \to \mathbb{R}$, $j = 1,\dots,d$.

4.1 Regression Model With Random Design

In this section we apply the algorithm SeqSEW$^*_\tau$ to the regression model with random design. In this batch setting the forecaster is given at the beginning of the game T independent random copies $(X_1,Y_1),\dots,(X_T,Y_T)$ of $(X,Y) \in \mathcal{X} \times \mathbb{R}$ whose common distribution is unknown. We assume thereafter that $\mathbb{E}[Y^2] < \infty$; the goal of the forecaster is to estimate the regression function $f : \mathcal{X} \to \mathbb{R}$ defined by $f(x) \triangleq \mathbb{E}[Y|X = x]$ for all $x \in \mathcal{X}$. Setting $\varepsilon_t \triangleq Y_t - f(X_t)$ for all $t = 1,\dots,T$, note that

$$Y_t = f(X_t) + \varepsilon_t \ , \quad 1 \leqslant t \leqslant T \ ,$$

and that the pairs $(X_1,\varepsilon_1),\dots,(X_T,\varepsilon_T)$ are i.i.d. and such that $\mathbb{E}[\varepsilon_1^2] < \infty$ and $\mathbb{E}[\varepsilon_1|X_1] = 0$ almost surely. In the sequel, we denote the distribution of X by P^X and we set, for all measurable functions

$h : \mathcal{X} \to \mathbb{R}$,

$$\|h\|_{L^2} \triangleq \left(\int_{\mathcal{X}} h(x)^2 P^X(\mathrm{d}x) \right)^{1/2} = \left(\mathbb{E}\big[h(X)^2\big] \right)^{1/2} .$$

Next we construct an estimator $\widehat{f}_T : \mathcal{X} \to \mathbb{R}$ based on the sample $(X_1,Y_1),\dots,(X_T,Y_T)$ that satisfies a sparsity oracle inequality, that is, its expected L^2-risk $\mathbb{E}\big[\big\| f - \widehat{f}_T \big\|_{L^2}^2 \big]$ is almost as small as the smallest L^2-risk $\|f - u\cdot\varphi\|_{L^2}^2$, $u \in \mathbb{R}^d$, up to some additive term proportional to $\|u\|_0$.

4.1.1 Algorithm and Main Result

Even if the whole sample $(X_1,Y_1),\dots,(X_T,Y_T)$ is available at the beginning of the prediction game, we treat it in a sequential fashion. We run the algorithm SeqSEW$^*_\tau$ of Section 3.2 from time 1 to time T with $\tau = 1/\sqrt{dT}$ (note that T is known in this setting). Using the standard online-to-batch conversion (see, e.g., Littlestone 1989; Cesa-Bianchi et al. 2004), we define our estimator $\widehat{f}_T : \mathcal{X} \to \mathbb{R}$ as the uniform average

$$\widehat{f}_T \triangleq \frac{1}{T} \sum_{t=1}^{T} \widetilde{f}_t \tag{25}$$

of the estimators $\widetilde{f}_t : \mathcal{X} \to \mathbb{R}$ sequentially built by the algorithm SeqSEW$^*_\tau$ as

$$\widetilde{f}_t(x) \triangleq \int_{\mathbb{R}^d} \big[u\cdot\varphi(x)\big]_{B_t} \, p_t(\mathrm{d}u) \, . \tag{26}$$

Note that, contrary to much prior work from the statistics community such as those of Catoni (2004), Bunea and Nobel (2008) and Dalalyan and Tsybakov (2012a), the estimators $\widetilde{f}_t : \mathcal{X} \to \mathbb{R}$ are tuned online. Therefore, $\widehat{f}_T$ does not depend on any prior knowledge on the unknown distribution of the (X_t,Y_t), $1 \leqslant t \leqslant T$, such as the unknown variance $\mathbb{E}\big[(Y - f(X))^2\big]$ of the noise, the norms $\big\|\varphi_j\big\|_\infty$, or the norms $\big\| f - \varphi_j \big\|_\infty$ (actually, the functions φ_j and $f - \varphi_j$ do not even need to be bounded in ℓ^∞-norm).

In this respect, this work improves on that of Bunea and Nobel (2008) who tune their online forecasters as a function of $\|f\|_\infty$ and $\sup_{u\in\mathcal{U}} \|u\cdot\varphi\|_\infty$, where $\mathcal{U} \subset \mathbb{R}^d$ is a bounded comparison set.[12] Their technique is not appropriate when $\|f\|_\infty$ is unknown and it cannot be extended to the case where $\mathcal{U} = \mathbb{R}^d$ (since $\sup_{u\in\mathbb{R}^d} \|u\cdot\varphi\|_\infty = +\infty$ if $\varphi \neq \mathbf{0}$). The major technical difference is that we truncate the base forecasts $u\cdot\varphi(X_t)$ instead of truncating the observations Y_t. In particular, this enables us to aggregate the base forecasters $u\cdot\varphi$ for all $u \in \mathbb{R}^d$, that is, over the whole $\mathbb{R}^d$ space.

The next sparsity oracle inequality is the main result of this section. It follows from the deterministic regret bound of Corollory 8 and from Jensen's inequality. Two corollaries are to be derived later.

Theorem 12 *Assume that $(X_1,Y_1),\dots,(X_T,Y_T) \in \mathcal{X}\times\mathbb{R}$ are independent random copies of $(X,Y) \in \mathcal{X}\times\mathbb{R}$, where $\mathbb{E}[Y^2] < +\infty$ and $\big\|\varphi_j\big\|_{L^2}^2 \triangleq \mathbb{E}[\varphi_j(X)^2] < +\infty$ for all $j = 1,\dots,d$. Then, the estimator*

12. Bunea and Nobel (2008) study the case where $\mathcal{U}$ is the (scaled) simplex in $\mathbb{R}^d$ or the set of its vertices.

$\widehat{f}_T$ defined in (25)-(26) *satisfies*

$$\mathbb{E}\left[\left\|f-\widehat{f}_T\right\|_{L^2}^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\|f-u\cdot\varphi\|_{L^2}^2 + 32\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,\|u\|_0 \ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT}\sum_{j=1}^d \left\|\varphi_j\right\|_{L^2}^2 + 5\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,.$$

Proof sketch (of Theorem 12) By Corollary 8 and by definition of $\widetilde{f}_t$ above and $\widehat{y}_t \triangleq \widetilde{f}_t(X_t)$ in Figure 3, we have, *almost surely*,

$$\sum_{t=1}^T (Y_t-\widetilde{f}_t(X_t))^2 \leqslant \inf_{u\in\mathbb{R}^d}\left\{\sum_{t=1}^T \big(Y_t-u\cdot\varphi(X_t)\big)^2 + 32\left(\max_{1\leqslant t\leqslant T} Y_t^2\right)\|u\|_0\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT}\sum_{j=1}^d\sum_{t=1}^T \varphi_j^2(X_t) + 5\max_{1\leqslant t\leqslant T} Y_t^2\,.$$

Taking the expectations of both sides and applying Jensen's inequality yields the desired result. For a detailed proof, see Appendix A.2. ■

Theorem 12 above can be used under several assumptions on the distribution of the output Y. In all cases, it suffices to upper bound the amplitude $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]$. We present below a general corollary and explain later why our fully automatic procedure $\widehat{f}_T$ solves two questions left open by Dalalyan and Tsybakov (2012a) (see Corollary 14 below).

4.1.2 A GENERAL COROLLARY

The next sparsity oracle inequality follows from Theorem 12 and from the upper bounds on $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]$ entailed by Lemmas 24–26 in Appendix B. The proof is postponed to Appendix A.2.

Corollary 13 *Assume that $(X_1,Y_1),\dots,(X_T,Y_T)\in\mathcal{X}\times\mathbb{R}$ are independent random copies of $(X,Y)\in\mathcal{X}\times\mathbb{R}$, that $\sup_{1\leqslant j\leqslant d}\left\|\varphi_j\right\|_{L^2}^2 < +\infty$, that $\mathbb{E}|Y|<+\infty$, and that one of the following assumptions holds on the distribution of $\Delta Y \triangleq Y-\mathbb{E}[Y]$.*

- $\big(\mathrm{BD}(B)\big)$: *$|\Delta Y|\leqslant B$ almost surely for a given constant $B>0$;*
- $\big(\mathrm{SG}(\sigma^2)\big)$: *$\Delta Y$ is subgaussian with variance factor $\sigma^2>0$, that is, $\mathbb{E}\left[e^{\lambda\Delta Y}\right]\leqslant e^{\lambda^2\sigma^2/2}$ for all $\lambda\in\mathbb{R}$;*
- $\big(\mathrm{BEM}(\alpha,M)\big)$: *$\Delta Y$ has a bounded exponential moment, that is, $\mathbb{E}\left[e^{\alpha|\Delta Y|}\right]\leqslant M$ for some given constants $\alpha>0$ and $M>0$;*
- $\big(\mathrm{BM}(\alpha,M)\big)$: *$\Delta Y$ has a bounded moment, that is, $\mathbb{E}\big[|\Delta Y|^\alpha\big]\leqslant M$ for some given constants $\alpha>2$ and $M>0$.*

Then, the estimator $\widehat{f}_T$ defined above satisfies

$$\mathbb{E}\left[\left\|f-\widehat{f}_T\right\|_{L^2}^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\|f-u\cdot\varphi\|_{L^2}^2 + 64\left(\frac{\mathbb{E}[Y]^2}{T}+\psi_T\right)\|u\|_0 \ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT}\sum_{j=1}^d \left\|\varphi_j\right\|_{L^2}^2 + 10\left(\frac{\mathbb{E}[Y]^2}{T}+\psi_T\right),$$

where

$$\psi_T \triangleq \frac{1}{T}\mathbb{E}\left[\max_{1\leqslant t\leqslant T}\big(Y_t-\mathbb{E}[Y_t]\big)^2\right] \leqslant \begin{cases} \dfrac{B^2}{T} & \text{under Assumption } \big(\mathrm{BD}(\mathrm{B})\big), \\ \dfrac{2\sigma^2\ln(2eT)}{T} & \text{under Assumption } \big(\mathrm{SG}(\sigma^2)\big), \\ \dfrac{\ln^2\big((M+e)T\big)}{\alpha^2 T} & \text{under Assumption } \big(\mathrm{BEM}(\alpha,\mathrm{M})\big), \\ \dfrac{M^{2/\alpha}}{T^{(\alpha-2)/\alpha}} & \text{under Assumption } \big(\mathrm{BM}(\alpha,\mathrm{M})\big). \end{cases}$$

Several comments can be made about Corollary 13. We first stress that, if $T \geqslant 2$, then the two "bias" terms $\mathbb{E}[Y]^2/T$ above can be avoided, at least at the price of a multiplicative factor of $2T/(T-1) \leqslant 4$. This can be achieved via a slightly more sophisticated online clipping—see Remark 19 in Appendix A.2.

Second, under the assumptions $\big(\mathrm{BD}(\mathrm{B})\big)$, $\big(\mathrm{SG}(\sigma^2)\big)$, or $\big(\mathrm{BEM}(\alpha,\mathrm{M})\big)$, the key quantity ψ_T is respectively of the order of $1/T$, $\ln(T)/T$ and $\ln^2(T)/T$. Up to a logarithmic factor, this corresponds to the classical fast rate of convergence $1/T$ obtained in the random design setting for different aggregation problems (see, e.g., Catoni 1999; Juditsky et al. 2008; Audibert 2009 for model-selection-type aggregation and Dalalyan and Tsybakov 2012a for linear aggregation). We were able to get similar rates—with, however, a fully automatic procedure—since our online algorithm SeqSEW^*_τ is well suited for bounded individual sequences with an unknown bound. More precisely, the finite i.i.d. sequence $Y_1,\dots,Y_T$ is almost surely uniformly bounded by the random bound $\max_{1\leqslant t\leqslant T}|Y_t|$. Our individual sequence techniques adapt sequentially to this random bound, yielding a regret bound that scales as $\max_{1\leqslant t\leqslant T} Y_t^2$. As a result, the risk bounds obtained after the online-to-batch conversion scale as $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]/T$. If the distribution of the output Y is sufficiently lightly-tailed—which includes the quite general bounded-exponential-moment assumption—then we can recover the fast rate of convergence $1/T$ up to a logarithmic factor.

We note that there is still a question left open for heavy-tailed output distributions. For example, under the bounded moment assumption $\big(\mathrm{BM}(\alpha,\mathrm{M})\big)$, the rate $T^{-(\alpha-2)/\alpha}$ that we proved does not match the faster rate $T^{-\alpha/(\alpha+2)}$ obtained by Juditsky et al. (2008) and Audibert (2009) under a similar assumption. Their methods use some preliminary knowledge on the output distribution (such as the exponent α). Thus, obtaining the same rate with a procedure tuned in an automatic fashion—just like our method $\widehat{f}_T$—is a challenging task. For this purpose, a different tuning of η_t or a more sophisticated online truncation might be necessary.

Third, several variations on the assumptions are possible. First note that several classical assumptions on Y expressed in terms of $f(X)$ and $\varepsilon \triangleq Y - f(X)$ are either particular cases of the above corollary or can be treated similarly. Indeed, each of the four assumptions above on

$\Delta Y \triangleq Y - \mathbb{E}[Y] = f(X) - \mathbb{E}[f(X)] + \varepsilon$ is satisfied as soon as both the distribution of $f(X) - \mathbb{E}[f(X)]$ and the conditional distribution of ε (conditionally on X) satisfy the same type of assumption. For example, if $f(X) - \mathbb{E}[f(X)]$ is subgaussian with variance factor σ_X^2 and if ε is subgaussian conditionally on X with a variance factor uniformly bounded by a constant σ_ε^2, then ΔY is subgaussian with variance factor $\sigma_X^2 + \sigma_\varepsilon^2$ (see also Remark 20 in Appendix A.2 to avoid conditioning).

The assumptions on $f(X) - \mathbb{E}[f(X)]$ and ε can also be mixed together. For instance, as explained in Remark 20 in Appendix A.2, under the classical assumptions

$$\|f\|_\infty < +\infty \qquad \text{and} \qquad \mathbb{E}\Big[e^{\alpha|\varepsilon|} \,\Big|\, X\Big] \leqslant M \quad \text{a.s.} \tag{27}$$

or

$$\|f\|_\infty < +\infty \qquad \text{and} \qquad \mathbb{E}\Big[e^{\lambda\varepsilon} \,\Big|\, X\Big] \leqslant e^{\lambda^2\sigma^2/2} \quad \text{a.s.}, \quad \forall \lambda \in \mathbb{R}\,, \tag{28}$$

the key quantity ψ_T in the corollary can be bounded from above by

$$\psi_T \leqslant \begin{cases} \dfrac{8\,\|f\|_\infty^2}{T} + \dfrac{2\ln^2\big((M+e)T\big)}{\alpha^2 T} & \text{under the set of assumptions (27),} \\ \dfrac{8\,\|f\|_\infty^2}{T} + \dfrac{4\sigma^2\ln(2eT)}{T} & \text{under the set of assumptions (28).} \end{cases}$$

In particular, under the set of assumptions (28), our procedure $\widehat{f}_T$ solves two questions left open by Dalalyan and Tsybakov (2012a). We discuss below our contributions in this particular case.

4.1.3 QUESTIONS LEFT OPEN BY DALALYAN AND TSYBAKOV

In this subsection we focus on the case when the set of assumptions (28) holds true. Namely, the regression function f is bounded (by an unknown constant) and the noise $\varepsilon \triangleq Y - f(X)$ is subgaussian conditionally on X with an unknown variance factor $\sigma^2 > 0$. An important particular case is when $\|f\|_\infty < +\infty$ and when the noise ε is independent of X and normally distributed $\mathcal{N}(0,\sigma^2)$.

Under the set of assumptions (28), the two terms $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]$ of Theorem 12 can be upper bounded in a simpler and slightly tighter way as compared to the proof of Corollary 13 (we only use the inequality $(x+y)^2 \leqslant 2x^2 + 2y^2$ once, instead of twice). It yields the following sparsity oracle inequality.

Corollary 14 *Assume that $(X_1,Y_1),\dots,(X_T,Y_T) \in \mathcal{X}\times\mathbb{R}$ are independent random copies of $(X,Y) \in \mathcal{X}\times\mathbb{R}$ such that the set of assumptions* (28) *above holds true. Then, the estimator $\widehat{f}_T$ defined in* (25)-(26) *satisfies*

$$\begin{aligned} &\mathbb{E}\left[\left\|f - \widehat{f}_T\right\|_{L^2}^2\right] \\ &\quad\leqslant \inf_{u\in\mathbb{R}^d}\left\{\|f - u\cdot\varphi\|_{L^2}^2 + 64\Big(\|f\|_\infty^2 + 2\sigma^2\ln(2eT)\Big)\frac{\|u\|_0}{T}\ln\left(1 + \frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} \\ &\qquad + \frac{1}{dT}\sum_{j=1}^d \left\|\varphi_j\right\|_{L^2}^2 + \frac{10}{T}\Big(\|f\|_\infty^2 + 2\sigma^2\ln(2eT)\Big)\,. \end{aligned}$$

Proof We apply Theorem 12 and bound $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]$ from above. By the elementary inequality $(x+y)^2 \leqslant 2x^2+2y^2$ for all $x,y\in\mathbb{R}$, we get

$$\begin{aligned}\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right] &= \mathbb{E}\left[\max_{1\leqslant t\leqslant T} \big(f(X_t)+\varepsilon_t\big)^2\right] \leqslant 2\left(\|f\|_\infty^2 + \mathbb{E}\left[\max_{1\leqslant t\leqslant T}\varepsilon_t^2\right]\right)\\ &\leqslant 2\left(\|f\|_\infty^2 + 2\sigma^2\ln(2eT)\right)\,,\end{aligned}$$

where the last inequality follows from Lemma 24 in Appendix B and from the fact that, for all $1\leqslant t\leqslant T$ and all $\lambda\in\mathbb{R}$, we have $\mathbb{E}\big[e^{\lambda\varepsilon_t}\big] = \mathbb{E}\big[e^{\lambda\varepsilon}\big] = \mathbb{E}\big[\mathbb{E}\big[e^{\lambda\varepsilon}\,\big|\,X\big]\big] \leqslant e^{\lambda^2\sigma^2/2}$ by (28). (Note that the assumption of conditional subgaussianity in (28) is stronger than what we need, that is, subgaussianity without conditioning.) This concludes the proof. ■

The above bound is of the same order (up to a $\ln T$ factor) as the sparsity oracle inequality proved in Proposition 1 of Dalalyan and Tsybakov (2012a). For the sake of comparison we state below with our notations (e.g., β therein corresponds to $1/\eta$ in this paper) a straightforward consequence of this proposition, which follows by Jensen's inequality and the particular[13] choice $\tau = \min\big\{1/\sqrt{dT}, R/(4d)\big\}$.

Proposition 15 (A consequence of Prop. 1 of Dalalyan and Tsybakov 2012a)
Assume that $\sup_{1\leqslant j\leqslant d}\big\|\varphi_j\big\|_\infty < \infty$ *and that the set of assumptions (28) above holds true. Then, for all* $R>0$ *and all* $\eta \leqslant \bar\eta(R) \triangleq \big(2\sigma^2 + 2\sup_{\|u\|_1\leqslant R}\|u\cdot\varphi - f\|_\infty^2\big)^{-1}$*, the mirror averaging aggregate* $\widehat{f}_T : \mathcal{X}\to\mathbb{R}$ *defined by Dalalyan and Tsybakov (2012a, Equations (1) and (3)) with* $\tau = \min\big\{1/\sqrt{dT}, R/(4d)\big\}$ *satisfies*

$$\begin{aligned}\mathbb{E}\left[\left\|f-\widehat{f}_T\right\|_{L^2}^2\right] \leqslant \inf_{\|u\|_1\leqslant R/2}&\left\{\|f-u\cdot\varphi\|_{L^2}^2 + \frac{4}{\eta}\,\frac{\|u\|_0}{T+1}\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1+2d}{\|u\|_0}\right)\right\}\\ &+\frac{4}{dT}\sum_{j=1}^d\big\|\varphi_j\big\|_{L^2}^2 + \frac{1}{(T+1)\eta}\,.\end{aligned}$$

We can now discuss the two questions left open by Dalalyan and Tsybakov (2012a).
Risk bound on the whole $\mathbb{R}^d$ *space.* Despite the similarity of the two bounds, the sparsity oracle inequality stated in Proposition 15 above only holds for vectors u within an ℓ^1-ball of finite radius $R/2$, while our bound holds over the whole $\mathbb{R}^d$ space. Moreover, the parameter R above has to be chosen in advance, but it cannot be chosen too large since $1/\eta \geqslant 1/\bar\eta(R)$, which grows as R^2 when $R\to+\infty$ (if $\varphi\neq\mathbf{0}$). Dalalyan and Tsybakov (2012a, Section 4.2) thus asked whether it was possible to get a bound with $1/\eta < +\infty$ such that the infimum in Proposition 15 extends to the whole $\mathbb{R}^d$ space. Our results show that, thanks to data-driven truncation, the answer is positive.

Note that it is still possible to transform the bound of Proposition 15 into a bound over the whole $\mathbb{R}^d$ space if the parameter R is chosen (illegally) as $R = 2\|u^*\|_1$ (or as a tight upper bound of the last

13. Proposition 1 of Dalalyan and Tsybakov (2012a) may seem more general than Corollary 14 at first sight since it holds for all $\tau>0$, but this is actually also the case for Corollary 14. The proof of the latter would indeed have remained true had we replaced $\tau = 1/\sqrt{dT}$ with any value of $\tau>0$ (see Proposition 5). We however chose the reasonable value $\tau = 1/\sqrt{dT}$ to make our algorithm parameter-free. As noted earlier, if $\|\varphi\|_\infty \triangleq \sup_{x\in\mathcal{X}}\max_{1\leqslant j\leqslant d}|\varphi_j(x)|$ is finite and known by the forecaster, another simple and easy-to-analyse tuning is given by $\tau = 1/(\|\varphi\|_\infty\sqrt{dT})$.

quantity), where $u^* \in \mathbb{R}^d$ minimizes over $\mathbb{R}^d$ the regularized risk

$$\|f - u\cdot\varphi\|_{L^2}^2 + \frac{4}{\bar\eta(2\|u\|_1)}\,\frac{\|u\|_0}{T+1}\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1+2d}{\|u\|_0}\right) + \frac{4}{dT}\sum_{j=1}^d \|\varphi_j\|_{L^2}^2 + \frac{1}{(T+1)\,\bar\eta(2\|u\|_1)}\,.$$

For instance, choosing $R = 2\|u^*\|_1$ and $\eta = \bar\eta(R)$, we get from Proposition 15 that the expected L^2-risk $\mathbb{E}\big[\|f-\widehat{f}_T\|_{L^2}^2\big]$ of the corresponding procedure is upper bounded by the infimum of the above regularized risk over all $u \in \mathbb{R}^d$. However, this parameter tuning is illegal since $\|u^*\|_1$ is not known in practice. On the contrary, thanks to data-driven truncation, the prior knowledge of $\|u^*\|_1$ is not required by our procedure.

Adaptivity to the unknown variance of the noise. The second open question, which was raised by Dalalyan and Tsybakov (2012a, Section 5.1, Remark 6), deals with the prior knowledge of the variance factor σ^2 of the noise. The latter is indeed required by their algorithm for the choice of the inverse temperature parameter η. Since the noise level σ^2 is unknown in practice, the authors asked the important question whether adaptivity to σ^2 was possible. Up to a $\ln T$ factor, Corollary 14 above provides a positive answer.

4.2 Regression Model With Fixed Design

In this section, we consider the regression model with fixed design. In this batch setting the forecaster is given at the beginning of the game a T-sample $(x_1,Y_1),\dots,(x_T,Y_T) \in \mathcal{X}\times\mathbb{R}$, where the x_t are deterministic elements in $\mathcal{X}$ and where

$$Y_t = f(x_t) + \varepsilon_t\,, \quad 1 \leqslant t \leqslant T, \tag{29}$$

for some i.i.d. sequence $\varepsilon_1,\dots,\varepsilon_T \in \mathbb{R}$ (with unknown distribution) and some unknown function $f : \mathcal{X} \to \mathbb{R}$. Next we construct an estimator $\widehat{f}_T : \mathcal{X} \to \mathbb{R}$ of f based on the sample $(x_1,Y_1),\dots,(x_T,Y_T)$ that satisfies a sparsity oracle inequality, that is, its expected mean squared error $\mathbb{E}\big[\frac{1}{T}\sum_{t=1}^T (f(x_t) - \widehat{f}_T(x_t))^2\big]$ is almost as small as the smallest mean squared error $\frac{1}{T}\sum_{t=1}^T (f(x_t) - u\cdot\varphi(x_t))^2$, $u \in \mathbb{R}^d$, up to some additive term proportional to $\|u\|_0$.

In this setting, just like in Section 4.1, our algorithm and the corresponding analysis are a straightforward consequence of the general results on individual sequences developed in Section 3. As in the random design setting, the sample $(x_1,Y_1),\dots,(x_T,Y_T)$ is treated in a sequential fashion. We run the algorithm SeqSEW$^*_\tau$ defined in Figure 3 from time 1 to time T with the particular choice of $\tau = 1/\sqrt{dT}$. We then define our estimator $\widehat{f}_T : \mathcal{X} \to \mathbb{R}$ by

$$\widehat{f}_T(x) \triangleq \begin{cases} \dfrac{1}{n_x}\displaystyle\sum_{\substack{1\leqslant t\leqslant T\\ t:x_t=x}} \widetilde{f}_t(x) & \text{if } x \in \{x_1,\dots,x_T\}\,, \\ 0 & \text{if } x \notin \{x_1,\dots,x_T\}\,, \end{cases} \tag{30}$$

where $n_x \triangleq \big|\big\{t : x_t = x\big\}\big| = \sum_{t=1}^T \mathbb{I}_{\{x_t=x\}}$, and where the estimators $\widetilde{f}_t : \mathcal{X} \to \mathbb{R}$ sequentially built by the algorithm SeqSEW$^*_\tau$ are defined by

$$\widetilde{f}_t(x) \triangleq \int_{\mathbb{R}^d} \big[u\cdot\varphi(x)\big]_{B_t}\, p_t(\mathrm{d}u)\,. \tag{31}$$

In the particular case when the x_t are all distinct, $\widehat{f}_T$ is simply defined by $\widehat{f}_T(x_t) \triangleq \widetilde{f}_t(x_t)$ for all $t \in \{1,\dots,T\}$ and by $\widehat{f}_T(x) = 0$ otherwise. Therefore, in this case, $\widehat{f}_T$ only uses the observations $y_1,\dots,y_{t-1}$ to estimate $f(x_t)$ (in particular, $\widehat{f}_T(x_1)$ is deterministic).

The next theorem is the main result of this subsection. It follows as in the random design setting from the deterministic regret bound of Corollory 8 and from Jensen's inequality. The proof is postponed to Appendix A.3.

Theorem 16 *Consider the regression model with fixed design described in* (29)*. Then, the estimator* $\widehat{f}_T$ *defined in* (30)–(31) *satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t)-\widehat{f}_T(x_t)\big)^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t)-u\cdot\varphi(x_t)\big)^2 + 32\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,\|u\|_0\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT^2}\sum_{j=1}^{d}\sum_{t=1}^{T}\varphi_j^2(x_t) + 5\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,.$$

As in Section 4.1, the amplitude $\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]$ can be upper bounded under various assumptions. The proof of the following corollary is postponed to Appendix A.3.

Corollary 17 *Consider the regression model with fixed design described in* (29)*. Assume that one of the following assumptions holds on the distribution of* ε_1.

- $\big(\mathrm{BD}(B)\big)$: $|\varepsilon_1| \leqslant B$ *almost surely for a given constant* $B > 0$*;*
- $\big(\mathrm{SG}(\sigma^2)\big)$: ε_1 *is subgaussian with variance factor* $\sigma^2 > 0$*, that is,* $\mathbb{E}\left[e^{\lambda\varepsilon_1}\right] \leqslant e^{\lambda^2\sigma^2/2}$ *for all* $\lambda \in \mathbb{R}$*;*
- $\big(\mathrm{BEM}(\alpha,M)\big)$: ε *has a bounded exponential moment, that is,* $\mathbb{E}\left[e^{\alpha|\varepsilon|}\right] \leqslant M$ *for some given constants* $\alpha > 0$ *and* $M > 0$*;*
- $\big(\mathrm{BM}(\alpha,M)\big)$: ε *has a bounded moment, that is,* $\mathbb{E}\big[|\varepsilon|^\alpha\big] \leqslant M$ *for some given constants* $\alpha > 2$ *and* $M > 0$*.*

Then, the estimator $\widehat{f}_T$ *defined in* (30)–(31) *satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t)-\widehat{f}_T(x_t)\big)^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t)-u\cdot\varphi(x_t)\big)^2 + 64\left(\frac{\max_{1\leqslant t\leqslant T} f^2(x_t)}{T}+\psi_T\right)\|u\|_0\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT^2}\sum_{j=1}^{d}\sum_{t=1}^{T}\varphi_j^2(x_t) + 10\left(\frac{\max_{1\leqslant t\leqslant T} f^2(x_t)}{T}+\psi_T\right)\,,$$

where

$$\psi_T \triangleq \frac{1}{T}\mathbb{E}\left[\max_{1\leqslant t\leqslant T}\varepsilon_t^2\right] \leqslant \begin{cases} \dfrac{B^2}{T} & \text{if Assumption } \big(\mathrm{BD}(\mathrm{B})\big) \text{ holds,} \\ \dfrac{2\sigma^2\ln(2eT)}{T} & \text{if Assumption } \big(\mathrm{SG}(\sigma^2)\big) \text{ holds,} \\ \dfrac{\ln^2\left((M+e)T\right)}{\alpha^2 T} & \text{if Assumption } \big(\mathrm{BEM}(\alpha,\mathrm{M})\big) \text{ holds,} \\ \dfrac{M^{2/\alpha}}{T^{(\alpha-2)/\alpha}} & \text{if Assumption } \big(\mathrm{BM}(\alpha,\mathrm{M})\big) \text{ holds.} \end{cases}$$

The above bound is of the same flavor as that of Dalalyan and Tsybakov (2008, Theorem 5). It has one advantage and one drawback. On the one hand, we note two additional "bias" terms $\big(\max_{1\leqslant t\leqslant T} f^2(x_t)\big)/T$ as compared to the bound of Dalalyan and Tsybakov (2008, Theorem 5). As of now, we have not been able to remove them using ideas similar to what we did in the random design case (see Remark 19 in Appendix A.2). On the other hand, under Assumption $\big(\mathrm{SG}(\sigma^2)\big)$, contrary to Dalalyan and Tsybakov (2008), our algorithm does not require the prior knowledge of the variance factor σ^2 of the noise.

Acknowledgments

The author would like to thank Arnak Dalalyan, Gilles Stoltz, and Pascal Massart for their helpful feedback and suggestions, as well as two anonymous reviewers for their insightful comments, one of which helped us simplify the online tuning carried out in Section 3.2. The author acknowledges the support of the French Agence Nationale de la Recherche (ANR), under grant PARCIMONIE (`http://www.proba.jussieu.fr/ANR/Parcimonie`), and of the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886.

Appendix A. Proofs

In this appendix we provide the proofs of some results stated above.

A.1 Proofs of Theorem 10 and Corollary 11

Before proving Theorem 10, we first need the following comment. Since the algorithm SeqSEW$^*_\tau$ is restarted at the beginning of each regime, the threshold values B_t used on regime r by the algorithm SeqSEW$^*_\tau$ are not computed on the basis of all past observations $y_1,\dots,y_{t-1}$ but only on the basis of the past observations y_t, $t\in\{t_{r-1}+1,\dots,t-1\}$. To avoid any ambiguity, we set $B_{r,t_{r-1}+1}\triangleq 0$ and

$$B_{r,t}\triangleq \max_{t_{r-1}+1\leqslant s\leqslant t-1}|y_s|\,,\quad t\in\{t_{r-1}+2,\dots,t_r\}\,.$$

Proof (of Theorem 10) We denote by $R\triangleq\min\{r\in\mathbb{N}: T\leqslant t_r\}$ the index of the last regime. For notational convenience, we re-define $t_R\triangleq T$ (even if $\gamma_T\leqslant 2^R$).

We upper bound the regret of the algorithm SeqSEW$^*_*$ on $\{1,\dots,T\}$ by the sum of its regrets on each time interval. To do so, first note that[14]

$$\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2=\sum_{r=0}^{R}\sum_{t=t_{r-1}+1}^{t_r}(y_t-\widehat{y}_t)^2=\sum_{r=0}^{R}\left((y_{t_r}-\widehat{y}_{t_r})^2+\sum_{t=t_{r-1}+1}^{t_r-1}(y_t-\widehat{y}_t)^2\right)$$
$$\leqslant\sum_{r=0}^{R}\left(2(y_{t_r}^2+B_{r,t_r}^2)+\sum_{t=t_{r-1}+1}^{t_r-1}(y_t-\widehat{y}_t)^2\right) \tag{32}$$
$$\leqslant\sum_{r=0}^{R}\left(\sum_{t=t_{r-1}+1}^{t_r-1}(y_t-\widehat{y}_t)^2\right)+4(R+1){y_T^*}^2\,, \tag{33}$$

where we set $y_T^*\triangleq\max_{1\leqslant t\leqslant T}|y_t|$, where (32) follows from the upper bound $(y_{t_r}-\widehat{y}_{t_r})^2\leqslant 2(y_{t_r}^2+\widehat{y}_{t_r}^2)\leqslant 2(y_{t_r}^2+B_{r,t_r}^2)$ (since $|\widehat{y}_{t_r}|\leqslant B_{r,t_r}$ by construction), and where (33) follows from the inequalities $y_{t_r}^2\leqslant {y_T^*}^2$ and

$$B_{r,t_r}^2\triangleq\max_{t_{r-1}+1\leqslant t\leqslant t_r-1}y_t^2\leqslant {y_T^*}^2\,.$$

But, for every $r=0,\dots,R$, the trace of the empirical Gram matrix on $\{t_{r-1}+1,\dots,t_r-1\}$ is upper bounded by

$$\sum_{t=t_{r-1}+1}^{t_r-1}\sum_{j=1}^{d}\varphi_j^2(x_t)\leqslant\sum_{t=1}^{t_r-1}\sum_{j=1}^{d}\varphi_j^2(x_t)\leqslant(e^{2^r}-1)^2\,,$$

where the last inequality follows from the fact that $\gamma_{t_r-1}\leqslant 2^r$ (by definition of t_r). Since in addition $\tau_r\triangleq 1/\sqrt{(e^{2^r}-1)^2}$, we can apply Corollory 7 on each period $\{t_{r-1}+1,\dots,t_r-1\}$, $r=0,\dots,R$, with $B_\Phi=(e^{2^r}-1)^2$ and get from (33) the upper bound

$$\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2\leqslant\sum_{r=0}^{R}\inf_{u\in\mathbb{R}^d}\left\{\sum_{t=t_{r-1}+1}^{t_r-1}\big(y_t-u\cdot\varphi(x_t)\big)^2+\Delta_r(u)\right\}+4(R+1){y_T^*}^2\,, \tag{34}$$

where

$$\Delta_r(u)\triangleq 32B_{r,t_r}^2\|u\|_0\ln\left(1+\frac{\left(e^{2^r}-1\right)\|u\|_1}{\|u\|_0}\right)+5B_{r,t_r}^2+1\,. \tag{35}$$

Since the infimum is superadditive and since $\big(y_{t_r}-u\cdot\varphi(x_{t_r})\big)^2\geqslant 0$ for all $r=0,\dots,R$, we get from (34) that

$$\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2\leqslant\inf_{u\in\mathbb{R}^d}\sum_{r=0}^{R}\left(\sum_{t=t_{r-1}+1}^{t_r}\big(y_t-u\cdot\varphi(x_t)\big)^2+\Delta_r(u)\right)+4(R+1){y_T^*}^2$$
$$=\inf_{u\in\mathbb{R}^d}\left\{\sum_{t=1}^{T}\big(y_t-u\cdot\varphi(x_t)\big)^2+\sum_{r=0}^{R}\Delta_r(u)\right\}+4(R+1){y_T^*}^2\,. \tag{36}$$

Let $u\in\mathbb{R}^d$. Next we bound $\sum_{r=0}^{R}\Delta_r(u)$ and $4(R+1){y_T^*}^2$ from above. First note that, by the upper bound $B_{r,t_r}^2\leqslant {y_T^*}^2$ and by the elementary inequality $\ln(1+xy)\leqslant\ln\big((1+x)(1+y)\big)=\ln(1+$

14. In the trivial cases where $t_r=t_{r-1}+1$ for some r, the sum $\sum_{t=t_{r-1}+1}^{t_r-1}(y_t-\widehat{y}_t)^2$ equals 0 by convention.

$x)+\ln(1+y)$ with $x=e^{2^r}-1$ and $y=\|u\|_1/\|u\|_0$, (35) yields

$$\Delta_r(u) \leqslant 32\,{y_T^*}^2\,\|u\|_0\,2^r + 32\,{y_T^*}^2\,\|u\|_0 \ln\left(1+\frac{\|u\|_1}{\|u\|_0}\right) + 5{y_T^*}^2 + 1\;.$$

Summing over $r=0,\dots,R$, we get

$$\sum_{r=0}^{R}\Delta_r(u) \leqslant 32\left(2^{R+1}-1\right){y_T^*}^2\,\|u\|_0 + (R+1)\left(32\,{y_T^*}^2\,\|u\|_0\ln\left(1+\frac{\|u\|_1}{\|u\|_0}\right)+5{y_T^*}^2+1\right)\;. \qquad (37)$$

First case: $R=0$

Substituting (37) in (36), we conclude the proof by noting that $A_T \geqslant 2+\log_2 1 \geqslant 1$ and that $\ln\left(e+\sqrt{\sum_{t=1}^T\sum_{j=1}^d \varphi_j^2(x_t)}\right)\geqslant 1$.

Second case: $R\geqslant 1$

Since $R\geqslant 1$, we have, by definition of t_{R-1},

$$2^{R-1} < \gamma_{t_{R-1}} \triangleq \ln\left(1+\sqrt{\sum_{t=1}^{t_{R-1}}\sum_{j=1}^d \varphi_j^2(x_t)}\right) \leqslant \ln\left(e+\sqrt{\sum_{t=1}^{T}\sum_{j=1}^d \varphi_j^2(x_t)}\right)\;.$$

The last inequality entails that $2^{R+1}-1\leqslant 4\cdot 2^{R-1}\leqslant 4\ln\left(e+\sqrt{\sum_{t=1}^T\sum_{j=1}^d\varphi_j^2(x_t)}\right)$ and that $R+1\leqslant 2+\log_2\ln\left(e+\sqrt{\sum_{t=1}^T\sum_{j=1}^d\varphi_j^2(x_t)}\right)\triangleq A_T$. Therefore, one the one hand, via (37),

$$\begin{aligned}\sum_{r=0}^{R}\Delta_r(u) \leqslant{}& 128\,{y_T^*}^2\,\|u\|_0\ln\left(e+\sqrt{\sum_{t=1}^T\sum_{j=1}^d\varphi_j^2(x_t)}\right) + 32\,{y_T^*}^2 A_T\,\|u\|_0\ln\left(1+\frac{\|u\|_1}{\|u\|_0}\right)\\ &+A_T\left(5{y_T^*}^2+1\right)\;,\end{aligned}$$

and, on the other hand,

$$4(R+1){y_T^*}^2 \leqslant 4A_T\,{y_T^*}^2\;.$$

Substituting the last two inequalities in (36) and noting that ${y_T^*}^2=\max_{1\leqslant t\leqslant T}y_t^2$ concludes the proof. ■

Proof (of Corollary 11) The proof is straightforward. In view of Theorem 10, we just need to check that the quantity (continuously extended in $s=0$)

$$128\Big(\max_{1\leqslant t\leqslant T}y_t^2\Big)\,s\ln\left(e+\sqrt{\sum_{t=1}^T\sum_{j=1}^d\varphi_j^2(x_t)}\right)+32\Big(\max_{1\leqslant t\leqslant T}y_t^2\Big)A_T\,s\ln\left(1+\frac{U}{s}\right)$$

is non-decreasing in $s\in\mathbb{R}_+$ and in $U\in\mathbb{R}_+$.

This is clear for U. The fact that it also non-decreasing in s comes from the following remark. For all $U\geqslant 0$, the function $s\in(0,+\infty)\mapsto s\ln(1+U/s)$ has a derivative equal to

$$\ln\left(1+\frac{U}{s}\right)-\frac{U/s}{1+U/s}\qquad\text{for all}\quad s>0\;.$$

From the elementary inequality

$$\ln(1+u) = -\ln\left(\frac{1}{1+u}\right) \geqslant -\left(\frac{1}{1+u} - 1\right) = \frac{u}{1+u}\,,$$

which holds for all $u \in (-1,+\infty)$, the above derivative is nonnegative for all $s > 0$ so that the continuous extension $s \in \mathbb{R}_+ \mapsto s\ln(1+U/s)$ is non-decreasing. ■

A.2 Proofs of Theorem 12 and Corollary 13

In this subsection, we set $\varepsilon \triangleq Y - f(X)$, so that the pairs $(X_1,\varepsilon_1),\ldots,(X_T,\varepsilon_T)$ are independent copies of $(X,\varepsilon) \in \mathcal{X}\times\mathbb{R}$. We also define $\sigma \geqslant 0$ by

$$\sigma^2 \triangleq \mathbb{E}\big[\varepsilon^2\big] = \mathbb{E}\big[(Y - f(X))^2\big]\,.$$

Proof (of Theorem 12) By Corollory 8 and the definitions of $\widetilde{f}_t$ in (26) and $\widehat{y}_t \triangleq \widetilde{f}_t(X_t)$ in Figure 3, we have, *almost surely*,

$$\begin{aligned}\sum_{t=1}^T (Y_t - \widetilde{f}_t(X_t))^2 \leqslant \inf_{u\in\mathbb{R}^d}\Bigg\{\sum_{t=1}^T \big(Y_t - u\cdot\varphi(X_t)\big)^2 + 32\left(\max_{1\leqslant t\leqslant T} Y_t^2\right)\|u\|_0 \ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\Bigg\} \\ + \frac{1}{dT}\sum_{j=1}^d\sum_{t=1}^T \varphi_j^2(X_t) + 5\max_{1\leqslant t\leqslant T} Y_t^2\,.\end{aligned}$$

It remains to take the expectations of both sides with respect to $\big((X_1,Y_1),\ldots,(X_T,Y_T)\big)$. First note that for all $t = 1,\ldots,T$, since $\varepsilon_t \triangleq Y_t - f(X_t)$, we have

$$\begin{aligned}\mathbb{E}\left[\big(Y_t - \widetilde{f}_t(X_t)\big)^2\right] &= \mathbb{E}\left[\big(\varepsilon_t + f(X_t) - \widetilde{f}_t(X_t)\big)^2\right] \\ &= \sigma^2 + \mathbb{E}\left[\big(f(X_t) - \widetilde{f}_t(X_t)\big)^2\right]\,,\end{aligned}$$

since $\mathbb{E}\big[\varepsilon_t^2\big] = \mathbb{E}\big[\varepsilon^2\big] \triangleq \sigma^2$ on the one hand, and, on the other hand, $\widetilde{f}_t$ is a built on $(X_s,Y_s)_{1\leqslant s\leqslant t-1}$ and $\mathbb{E}\big[\varepsilon_t\big|(X_s,Y_s)_{1\leqslant s\leqslant t-1},X_t\big] = \mathbb{E}\big[\varepsilon_t\big|X_t\big] = 0$ (from the independence of $(X_s,Y_s)_{1\leqslant s\leqslant t-1}$ and (X_t,Y_t) and by definition of f).

In the same way,

$$\mathbb{E}\left[\big(Y_t - u\cdot\varphi(X_t)\big)^2\right] = \sigma^2 + \mathbb{E}\left[\big(f(X_t) - u\cdot\varphi(X_t)\big)^2\right]\,.$$

Therefore, by Jensen's inequality and the concavity of the infimum, the last inequality becomes, after taking the expectations of both sides,

$$\begin{aligned}T\sigma^2 + \sum_{t=1}^T \mathbb{E}\left[\big(f(X_t) - \widetilde{f}_t(X_t)\big)^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\Bigg\{T\sigma^2 + \sum_{t=1}^T \mathbb{E}\left[\big(f(X_t) - u\cdot\varphi(X_t)\big)^2\right] \\ + 32\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]\|u\|_0 \ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\Bigg\} \\ + \frac{1}{dT}\sum_{j=1}^d\sum_{t=1}^T \mathbb{E}\left[\varphi_j^2(X_t)\right] + 5\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]\,.\end{aligned}$$

Noting that the $T\sigma^2$ cancel out, dividing the two sides by T, and using the fact that $X_t \sim X$ in the right-hand side, we get

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left(f(X_t)-\widetilde{f}_t(X_t)\right)^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\|f-u\cdot\varphi\|_{L^2}^2 + 32\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,\|u\|_0\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT}\sum_{j=1}^{d}\|\varphi_j\|_{L^2}^2 + 5\,\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T}\,.$$

The right-hand side of the last inequality is exactly the upper bound stated in Theorem 12. To conclude the proof, we thus only need to check that $\mathbb{E}\big[\|f-\widehat{f}_T\|_{L^2}^2\big]$ is bounded from above by the left-hand side. But by definition of $\widehat{f}_T$ and by convexity of the square loss,

$$\mathbb{E}\left[\left\|f-\widehat{f}_T\right\|_{L^2}^2\right] \triangleq \mathbb{E}\left[\left(f(X)-\frac{1}{T}\sum_{t=1}^{T}\widetilde{f}_t(X)\right)^2\right] \leqslant \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left(f(X)-\widetilde{f}_t(X)\right)^2\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left(f(X_t)-\widetilde{f}_t(X_t)\right)^2\right]\,.$$

The last equality follows classically from the fact that, for all $t=1,\dots,T$, $(X_s,Y_s)_{1\leqslant s\leqslant t-1}$ (on which $\widetilde{f}_t$ is constructed) is independent from both X_t and X and the fact that $X_t\sim X$. ■

Remark 18 *The fact that the inequality stated in Corollary 8 has a leading constant equal to* 1 *on individual sequences is crucial to derive in the stochastic setting an oracle inequality in terms of the (excess) risks* $\mathbb{E}\left[\|f-\widehat{f}_T\|_{L^2}^2\right]$ *and* $\|f-u\cdot\varphi\|_{L^2}^2$. *Indeed, if the constant appearing in front of the infimum was equal to* $C>1$, *then the* $T\sigma^2$ *would not cancel out in the previous proof, so that the resulting expected inequality would contain a non-vanishing additive term* $(C-1)\sigma^2$.

Proof (of Corollary 13) We can apply Theorem 12. Then, to prove the upper bound on $\mathbb{E}\left[\|f-\widehat{f}_T\|_{L^2}^2\right]$, it suffices to show that

$$\frac{\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right]}{T} \leqslant 2\left(\frac{\mathbb{E}[Y]^2}{T}+\psi_T\right)\,. \tag{38}$$

Recall that

$$\psi_T \triangleq \frac{1}{T}\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T}\left(Y_t-\mathbb{E}[Y_t]\right)^2\right] = \frac{1}{T}\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T}(\Delta Y)_t^2\right]\,,$$

where we defined $(\Delta Y)_t \triangleq Y_t-\mathbb{E}[Y_t] = Y_t-\mathbb{E}[Y]$ for all $t=1,\dots,T$.

From the elementary inequality $(x+y)^2\leqslant 2x^2+2y^2$ for all $x,y\in\mathbb{R}$, we have

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right] \triangleq \mathbb{E}\left[\max_{1\leqslant t\leqslant T}\left(\mathbb{E}[Y]+(\Delta Y)_t\right)^2\right] \leqslant 2\,\mathbb{E}[Y]^2+2\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T}(\Delta Y)_t^2\right]\,.$$

Dividing both sides by T, we get (38).

As for the upper bound on ψ_T, since the $(\Delta Y)_t$, $1 \leqslant t \leqslant T$, are distributed as ΔY, we can apply Lemmas 24, 25, and 26 in Appendix B.3 to bound ψ_T from above under the assumptions $\big(\mathrm{SG}(\sigma^2)\big)$, $\big(\mathrm{BEM}(\alpha,\mathrm{M})\big)$, and $\big(\mathrm{BM}(\alpha,\mathrm{M})\big)$ respectively (the upper bound under $\big(\mathrm{BD}(\mathrm{B})\big)$ is straightforward):

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T}(\Delta Y)_t^2\right] \leqslant \begin{cases} B^2 & \text{if Assumption } \big(\mathrm{BD}(\mathrm{B})\big) \text{holds,} \\ \sigma^2 + 2\sigma^2\ln(2eT) & \text{if Assumption } \big(\mathrm{SG}(\sigma^2)\big) \text{holds,} \\ \frac{\ln^2\big((M+e)T\big)}{\alpha^2} & \text{if Assumption } \big(\mathrm{BEM}(\alpha,\mathrm{M})\big) \text{holds,} \\ (MT)^{2/\alpha} & \text{if Assumption } \big(\mathrm{BM}(\alpha,\mathrm{M})\big) \text{holds .} \end{cases}$$

■

Remark 19 *If $T \geqslant 2$, then the two "bias" terms $\mathbb{E}[Y]^2/T$ appearing in Corollary 13 can be avoided, at least at the price of a multiplicative factor of $2T/(T-1) \leqslant 4$. It suffices to use a slightly more sophisticated online clipping defined as follows. The first round $t=1$ is only used to observe Y_1. Then, the algorithm* SeqSEW^*_τ *is run with $\tau = 1/\sqrt{dT}$ from round 2 up to round T with the following important modification: instead of truncating the predictions to $[-B_t, B_t]$, which is best suited to the case $\mathbb{E}[Y]=0$, we truncate them to the interval*

$$\big[Y_1 - B'_t, Y_1 + B'_t\big] \;, \quad \text{where} \quad B'_t \triangleq \max_{1\leqslant s\leqslant t-1} |Y_s - Y_1| \;.$$

If η_t is changed accordingly, that is, if $\eta_t = 1/(8B_t'^2)$, then it easy to see that the resulting procedure $\widehat{f}_T \triangleq \frac{1}{T-1}\sum_{s=2}^T \widetilde{f}_s$ (where $\widetilde{f}_2, \ldots, \widetilde{f}_T$ are the estimators output by SeqSEW^*_τ*) satisfies*

$$\mathbb{E}\left[\left\| f - \widehat{f}_T \right\|_{L^2}^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{ \|f - u\cdot\varphi\|_{L^2}^2 + 64\left(\frac{Var[Y]}{T-1} + \psi_{T-1}\right)\|u\|_0 \ln\left(1 + \frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\}$$
$$+ \frac{1}{dT}\sum_{j=1}^d \left\|\varphi_j\right\|_{L^2}^2 + 10\left(\frac{Var[Y]}{T-1} + \psi_{T-1}\right) \;,$$

where $Var[Y] \triangleq \mathbb{E}\big[(Y - \mathbb{E}[Y])^2\big]$. Comparing the last bound to that of Corollary 13, we note that the two terms $\mathbb{E}[Y]^2/T$ are absent, and that we loose a multiplicative factor at most of 4 since $Var[Y] \leqslant \mathbb{E}\big[\max_{2\leqslant t\leqslant T}(Y_t - \mathbb{E}[Y_t])^2\big] \triangleq (T-1)\psi_{T-1}$ so that

$$\frac{Var[Y]}{T-1} + \psi_{T-1} \leqslant 2\psi_{T-1} \leqslant 2\left(\frac{T}{T-1}\right)\psi_T \leqslant 4\psi_T \;.$$

Remark 20 *We mentioned after Corollary 13 that each of the four assumptions on ΔY is fulfilled as soon as both the distribution of $f(X) - \mathbb{E}[f(X)]$ and the conditional distribution of ε (conditionally on X) satisfy the same type of assumption. It actually extends to the more general case when the conditional distribution of ε given X is replaced with the distribution of ε itself (without conditioning). This relies on the elementary upper bound*

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T}(\Delta Y)_t^2\right] = \mathbb{E}\left[\max_{1\leqslant t\leqslant T}\big(f(X_t) - \mathbb{E}[f(X)] + \varepsilon_t\big)^2\right]$$
$$\leqslant 2\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T}\big(f(X_t) - \mathbb{E}[f(X)]\big)^2\right] + 2\,\mathbb{E}\left[\max_{1\leqslant t\leqslant T}\varepsilon_t^2\right] \;.$$

From the last inequality, we can also see that assumptions of different nature can be made on $f(X) - \mathbb{E}[f(X)]$ *and* ε*, such as the assumptions given in* (27) *or in* (28)*.*

A.3 Proofs of Theorem 16 and Corollary 17

Proof (of Theorem 16) The proof follows the sames lines as in the proof of Theorem 12. We thus only sketch the main arguments. In the sequel, we set $\sigma^2 \triangleq \mathbb{E}\big[\varepsilon_1^2\big]$.

Applying Corollory 8 we have, *almost surely*,

$$\sum_{t=1}^{T}\big(Y_t - \widetilde{f}_t(x_t)\big)^2 \leqslant \inf_{u\in\mathbb{R}^d}\left\{\sum_{t=1}^{T}\big(Y_t - u\cdot\varphi(x_t)\big)^2 + 32\left(\max_{1\leqslant t\leqslant T} Y_t^2\right)\|u\|_0 \ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT}\sum_{j=1}^{d}\sum_{t=1}^{T}\varphi_j^2(x_t) + 5\max_{1\leqslant t\leqslant T} Y_t^2\ .$$

Taking the expectations of both sides, expanding the squares $\big(Y_t - \widetilde{f}_t(x_t)\big)^2$ and $\big(Y_t - u\cdot\varphi(x_t)\big)^2$, noting that two terms $T\sigma^2$ cancel out,[15] and then dividing both sides by T, we get

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t) - \widetilde{f}_t(x_t)\big)^2\right] \leqslant \inf_{u\in\mathbb{R}^d}\left\{\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t) - u\cdot\varphi(x_t)\big)^2 + 32\,\frac{\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]}{T}\,\|u\|_0\ln\left(1+\frac{\sqrt{dT}\,\|u\|_1}{\|u\|_0}\right)\right\} + \frac{1}{dT^2}\sum_{j=1}^{d}\sum_{t=1}^{T}\varphi_j^2(x_t) + 5\,\frac{\mathbb{E}\big[\max_{1\leqslant t\leqslant T} Y_t^2\big]}{T}\ .$$

The right-hand side is exactly the upper bound stated in Theorem 16. We thus only need to check that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t) - \widehat{f}_T(x_t)\big)^2\right] \leqslant \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\big(f(x_t) - \widetilde{f}_t(x_t)\big)^2\right]\ . \tag{39}$$

This is an equality if the x_t are all distinct. In general we get an inequality which follows from the convexity of the square loss. Indeed, by definition of n_x, we have, almost surely,

$$\begin{aligned}\sum_{t=1}^{T}\big(f(x_t) - \widehat{f}_T(x_t)\big)^2 &= \sum_{x\in\{x_1,\dots,x_T\}}\ \sum_{\substack{1\leqslant t\leqslant T\\ t:x_t=x}}\big(f(x_t) - \widehat{f}_T(x_t)\big)^2 = \sum_{x\in\{x_1,\dots,x_T\}} n_x\big(f(x) - \widehat{f}_T(x)\big)^2\\ &= \sum_{x\in\{x_1,\dots,x_T\}} n_x\left(f(x) - \frac{1}{n_x}\sum_{\substack{1\leqslant t\leqslant T\\ t:x_t=x}}\widetilde{f}_t(x)\right)^2\\ &\leqslant \sum_{x\in\{x_1,\dots,x_T\}} n_x\,\frac{1}{n_x}\sum_{\substack{1\leqslant t\leqslant T\\ t:x_t=x}}\big(f(x) - \widetilde{f}_t(x)\big)^2 = \sum_{t=1}^{T}\big(f(x_t) - \widetilde{f}_t(x_t)\big)^2\ ,\end{aligned}$$

15. Note that $\mathbb{E}\big[(f(x_t) - \widetilde{f}(x_t))\varepsilon_t\big] = 0$ since $\widetilde{f}_t(x_t)$ and ε_t are independent. This is due to the fact that $\widetilde{f}_t$ is built from the past data only. In particular, truncating the predictions to $B = \max_{1\leqslant t\leqslant T}|Y_t|$ might not work. A similar comment could be made in the random design case (Section 4.1).

where the second line is by definition of $\widehat{f}_T$ and where the last line follows from Jensen's inequality. Dividing both sides by T and taking their expectations, we get (39), which concludes the proof. ■

Proof (of Corollary 17) First note that

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Y_t^2\right] \triangleq \mathbb{E}\left[\max_{1\leqslant t\leqslant T} \big(f(x_t)+\varepsilon_t\big)^2\right] \leqslant 2\left(\max_{1\leqslant t\leqslant T} f^2(x_t) + \mathbb{E}\left[\max_{1\leqslant t\leqslant T} \varepsilon_t^2\right]\right) .$$

The proof then follows exactly the same lines as for Corollary 13 with the sequence (ε_t) instead of the sequence $\big((\Delta Y)_t\big)$. ■

Appendix B. Tools

Next we provide several (in)equalities that prove to be useful throughout the paper.

B.1 A Duality Formula for the Kullback-Leibler Divergence

We recall below a key duality formula satisfied by the Kullback-Leibler divergence and whose proof can be found, for example, in the monograph by Catoni (2004, pp. 159–160). We use the notations of Section 2.

Proposition 21 *For any measurable space $(\Theta,\mathcal{B})$, any probability distribution π on $(\Theta,\mathcal{B})$, and any measurable function $h:\Theta\to[a,+\infty)$ bounded from below (by some $a\in\mathbb{R}$), we have*

$$-\ln\int_\Theta e^{-h}\mathrm{d}\pi = \inf_{\rho\in\mathcal{M}_1^+(\Theta)}\left\{\int_\Theta h\,\mathrm{d}\rho + \mathcal{K}(\rho,\pi)\right\} ,$$

where $\mathcal{M}_1^+(\Theta)$ denotes the set of all probability distributions on $(\Theta,\mathcal{B})$, and where the expectations $\int_\Theta h\,\mathrm{d}\rho\in[a,+\infty]$ are always well defined since h is bounded from below.

B.2 Some Tools to Exploit Our PAC-Bayesian Inequalities

In this subsection we recall two results needed for the derivation of Proposition 1 and Proposition 5 from the PAC-Bayesian inequalities (7) and (12). The proofs are due to Dalalyan and Tsybakov (2007, 2008) and we only reproduce them for the convenience of the reader.[16]

For any $u^*\in\mathbb{R}^d$ and $\tau>0$, define $\rho_{u^*,\tau}$ as the translated of π_τ at u^*, namely,

$$\rho_{u^*,\tau} \triangleq \frac{\mathrm{d}\pi_\tau}{\mathrm{d}u}(u-u^*)\,\mathrm{d}u = \prod_{j=1}^d \frac{(3/\tau)\,\mathrm{d}u_j}{2\big(1+|u_j-u_j^*|/\tau\big)^4} . \tag{40}$$

16. The notations are however slightly modified because of the change in the statistical setting and goal. The target predictions $(f(x_1),\dots,f(x_T))$ are indeed replaced with the observations $(y_1,\dots,y_T)$ and the prediction loss $\|f-f_u\|_n^2$ is replaced with the cumulative loss $\sum_{t=1}^T\big(y_t-u\cdot\varphi(x_t)\big)^2$. Moreover, the analysis of the present proof is slightly simpler since we just need to consider the case $L_0=+\infty$ according to the notations of Theorem 5 by Dalalyan and Tsybakov (2008).

Lemma 22 *For all $u^* \in \mathbb{R}^d$ and $\tau > 0$, the probability distribution $\rho_{u^*,\tau}$ satisfies*

$$\int_{\mathbb{R}^d} \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u) = \sum_{t=1}^{T} \big(y_t - u^* \cdot \varphi(x_t)\big)^2 + \tau^2 \sum_{j=1}^{d} \sum_{t=1}^{T} \varphi_j^2(x_t) \ .$$

Lemma 23 *For all $u^* \in \mathbb{R}^d$ and $\tau > 0$, the probability distribution $\rho_{u^*,\tau}$ satisfies*

$$\mathcal{K}(\rho_{u^*,\tau}, \pi_\tau) \leqslant 4 \|u^*\|_0 \ln\left(1 + \frac{\|u^*\|_1}{\|u^*\|_0 \tau}\right) \ .$$

Proof (of Lemma 22) For all $t \in \{1, \dots, T\}$ we expand the square $\big(y_t - u \cdot \varphi(x_t)\big)^2 = \big(y_t - u^* \cdot \varphi(x_t) + (u^* - u) \cdot \varphi(x_t)\big)^2$ and use the linearity of the integral to get

$$\begin{aligned}
&\int_{\mathbb{R}^d} \sum_{t=1}^{T} \big(y_t - u \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u) \qquad (41)\\
&\quad = \sum_{t=1}^{T} \big(y_t - u^* \cdot \varphi(x_t)\big)^2 + \sum_{t=1}^{T} \int_{\mathbb{R}^d} \big((u^* - u) \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u)\\
&\quad\quad + \underbrace{\sum_{t=1}^{T} 2\big(y_t - u^* \cdot \varphi(x_t)\big) \int_{\mathbb{R}^d} (u^* - u) \cdot \varphi(x_t)\, \rho_{u^*,\tau}(\mathrm{d}u)}_{=0}
\end{aligned}$$

The last sum equals zero by symmetry of $\rho_{u^*,\tau}$ around u^*, which yields $\int_{\mathbb{R}} u \, \rho_{u^*,\tau}(\mathrm{d}u) = u^*$. As for the second sum of the right-hand side, it can be bounded from above similarly. Indeed, expanding the inner product and then the square $\big((u^* - u) \cdot \varphi(x_t)\big)^2$ we have, for all $t = 1, \dots, T$,

$$\big((u^* - u) \cdot \varphi(x_t)\big)^2 = \sum_{j=1}^{d} (u_j^* - u_j)^2 \varphi_j^2(x_t) + \sum_{1 \leqslant j \neq k \leqslant d} (u_j^* - u_j)(u_k^* - u_k)\, \varphi_j(x_t)\, \varphi_k(x_t) \ .$$

By symmetry of $\rho_{u^*,\tau}$ around u^* and the fact that $\rho_{u^*,\tau}$ is a product-distribution, we get

$$\begin{aligned}
\sum_{t=1}^{T} \int_{\mathbb{R}^d} \big((u^* - u) \cdot \varphi(x_t)\big)^2 \rho_{u^*,\tau}(\mathrm{d}u) &= \sum_{t=1}^{T} \sum_{j=1}^{d} \varphi_j^2(x_t) \int_{\mathbb{R}^d} (u_j^* - u_j)^2 \rho_{u^*,\tau}(\mathrm{d}u) + 0\\
&= \sum_{t=1}^{T} \sum_{j=1}^{d} \varphi_j^2(x_t) \int_{\mathbb{R}} (u_j^* - u_j)^2 \frac{(3/\tau)\,\mathrm{d}u_j}{2\big(1 + |u_j - u_j^*|/\tau\big)^4} \qquad (42)\\
&= \tau^2 \sum_{t=1}^{T} \sum_{j=1}^{d} \varphi_j^2(x_t) \int_{\mathbb{R}} \frac{3t^2 \mathrm{d}t}{2(1 + |t|)^4} \qquad (43)\\
&= \tau^2 \sum_{t=1}^{T} \sum_{j=1}^{d} \varphi_j^2(x_t) \ , \qquad (44)
\end{aligned}$$

where (42) follows by definition of $\rho_{u^*,\tau}$, where (43) is obtained by the change of variables $t = (u_j - u_j^*)/\tau$, and where (44) follows from the equality $\int_{\mathbb{R}} \frac{3t^2 \mathrm{d}t}{2\big(1 + |t|\big)^4} = 1$ that can be proved by integrating by parts. Substituting (44) into (41) concludes the proof. ∎

Proof (of Lemma 23) By definition of $\rho_{u^*,\tau}$ and π_τ, we have

$$\mathcal{K}(\rho_{u^*,\tau},\pi_\tau) \triangleq \int_{\mathbb{R}^d}\left(\ln\frac{\mathrm{d}\rho_{u^*,\tau}}{\mathrm{d}\pi_\tau}(u)\right)\rho_{u^*,\tau}(\mathrm{d}u) = \int_{\mathbb{R}^d}\left(\ln\prod_{j=1}^d \frac{\left(1+|u_j|/\tau\right)^4}{\left(1+|u_j-u_j^*|/\tau\right)^4}\right)\rho_{u^*,\tau}(\mathrm{d}u)$$
$$= 4\int_{\mathbb{R}^d}\left(\sum_{j=1}^d \ln\frac{1+|u_j|/\tau}{1+|u_j-u_j^*|/\tau}\right)\rho_{u^*,\tau}(\mathrm{d}u)\ . \tag{45}$$

But, for all $u\in\mathbb{R}^d$, by the triangle inequality,

$$1+|u_j|/\tau \leqslant 1+|u_j^*|/\tau+|u_j-u_j^*|/\tau \leqslant \left(1+|u_j^*|/\tau\right)\left(1+|u_j-u_j^*|/\tau\right)\ ,$$

so that Equation (45) yields the upper bound

$$\mathcal{K}(\rho_{u^*,\tau},\pi_\tau) \leqslant 4\sum_{j=1}^d \ln\left(1+|u_j^*|/\tau\right) = 4\sum_{j:u_j^*\neq 0}\ln\left(1+|u_j^*|/\tau\right)\ .$$

We now recall that $\|u^*\|_0 \triangleq \left|\{j : u_j^*\neq 0\}\right|$ and apply Jensen's inequality to the concave function $x\in(-1,+\infty)\longmapsto\ln(1+x)$ to get

$$\sum_{j:u_j^*\neq 0}\ln\left(1+|u_j^*|/\tau\right) = \|u^*\|_0\,\frac{1}{\|u^*\|_0}\sum_{j:u_j^*\neq 0}\ln\left(1+|u_j^*|/\tau\right) \leqslant \|u^*\|_0\ln\left(1+\frac{\sum_{j:u_j^*\neq 0}|u_j^*|}{\|u^*\|_0\,\tau}\right)$$
$$\leqslant \|u^*\|_0\ln\left(1+\frac{\|u^*\|_1}{\|u^*\|_0\,\tau}\right)\ .$$

This concludes the proof. ■

B.3 Some Maximal Inequalities

Next we prove three maximal inequalities needed for the derivation of Corollaries 13 and 17 from Theorems 12 and 16 respectively. Their proofs are quite standard but we provide them for the convenience of the reader.

Lemma 24 *Let $Z_1,\dots,Z_T$ be $T\geqslant 1$ (centered) real random variables such that, for a given constant $\nu\geqslant 0$, we have*

$$\forall t\in\{1,\dots,T\},\quad \forall\lambda\in\mathbb{R},\quad \mathbb{E}\left[e^{\lambda Z_t}\right]\leqslant e^{\lambda^2\nu/2}\ . \tag{46}$$

Then,

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right]\leqslant 2\nu\ln(2eT)\ .$$

Lemma 25 *Let $Z_1,\dots,Z_T$ be $T\geqslant 1$ real random variables such that, for some given constants $\alpha>0$ and $M>0$, we have*

$$\forall t\in\{1,\dots,T\},\quad \mathbb{E}\left[e^{\alpha|Z_t|}\right]\leqslant M\ .$$

Then,

$$\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right]\leqslant \frac{\ln^2\left((M+e)T\right)}{\alpha^2}\ .$$

Lemma 26 *Let $Z_1, \dots, Z_T$ be $T \geqslant 1$ real random variables such that, for some given constants $\alpha > 2$ and $M > 0$, we have*

$$\forall t \in \{1, \dots, T\}, \quad \mathbb{E}\big[|Z_t|^\alpha\big] \leqslant M \,.$$

Then,

$$\mathbb{E}\left[\max_{1 \leqslant t \leqslant T} Z_t^2\right] \leqslant (MT)^{2/\alpha} \,.$$

Proof (of Lemma 24) Let $t \in \{1, \dots, T\}$. From the subgaussian assumption (46) it is well known (see, e.g., Massart 2007, Chapter 2) that for all $x \geqslant 0$, we have

$$\forall t \in \{1, \dots, T\} \,, \quad \mathbb{P}\big(|Z_t| > x\big) \leqslant 2e^{-x^2/(2\nu)} \,.$$

Let $\delta \in (0,1)$. By the change of variables $x = \sqrt{2\nu \ln(2T/\delta)}$, the last inequality entails that, for all $t = 1, \dots, T$, we have $|Z_t| \leqslant \sqrt{2\nu \ln(2T/\delta)}$ with probability at least $1 - \delta/T$. Therefore, by a union bound, we get, with probability at least $1 - \delta$,

$$\forall t \in \{1, \dots, T\} \,, \quad |Z_t| \leqslant \sqrt{2\nu \ln(2T/\delta)} \,.$$

As a consequence, with probability at least $1 - \delta$,

$$\max_{1 \leqslant t \leqslant T} Z_t^2 \leqslant 2\nu \ln(2T/\delta) \leqslant 2\nu \ln(1/\delta) + 2\nu \ln(2T) \,.$$

It now just remains to integrate the last inequality over $\delta \in (0,1)$ as is made precise below. By the change of variables $\delta = e^{-z}$, the latter inequality yields

$$\forall z > 0 \,, \quad \mathbb{P}\left[\left(\frac{\max_{1 \leqslant t \leqslant T} Z_t^2 - 2\nu \ln(2T)}{2\nu}\right)_+ > z\right] \leqslant e^{-z} \,, \tag{47}$$

where for all $x \in \mathbb{R}$, $x_+ \triangleq \max\{x, 0\}$ denotes the positive part of x. Using the well-known fact that $\mathbb{E}[\xi] = \int_0^{+\infty} \mathbb{P}(\xi > z) \mathrm{d}z$ for all nonnegative real random variable ξ, we get

$$\begin{aligned}
\mathbb{E}\left[\frac{\max_{1 \leqslant t \leqslant T} Z_t^2 - 2\nu \ln(2T)}{2\nu}\right] &\leqslant \mathbb{E}\left[\left(\frac{\max_{1 \leqslant t \leqslant T} Z_t^2 - 2\nu \ln(2T)}{2\nu}\right)_+\right] \\
&= \int_0^{+\infty} \mathbb{P}\left[\left(\frac{\max_{1 \leqslant t \leqslant T} Z_t^2 - 2\nu \ln(2T)}{2\nu}\right)_+ > z\right] \mathrm{d}z \\
&\leqslant \int_0^{+\infty} e^{-z} \mathrm{d}z \;= 1 \,,
\end{aligned}$$

where the last line follows from (47) above. Rearranging terms, we get $\mathbb{E}\big[\max_{1 \leqslant t \leqslant T} Z_t^2\big] \leqslant 2\nu + 2\nu \ln(2T)$, which concludes the proof. ∎

Proof (of Lemma 25) We first need the following definitions. Let $\psi_\alpha : \mathbb{R}_+ \to \mathbb{R}$ be a convex majorant of $x \mapsto e^{\alpha\sqrt{x}}$ on $\mathbb{R}_+$ defined by

$$\psi_\alpha(x) \triangleq \begin{cases} e & \text{if } x < 1/\alpha^2 \,, \\ e^{\alpha\sqrt{x}} & \text{if } x \geqslant 1/\alpha^2 \,. \end{cases}$$

We associate with ψ_α its generalized inverse $\psi_\alpha^{-1} : \mathbb{R} \to \mathbb{R}_+$ defined by

$$\psi_\alpha^{-1}(y) = \begin{cases} 1/\alpha^2 & \text{if } y < e \,, \\ (\ln y)^2/\alpha^2 & \text{if } y \geqslant e \,. \end{cases}$$

Elementary manipulations show that:

- ψ_α is nondecreasing and convex on $\mathbb{R}_+$;
- ψ_α^{-1} is nondecreasing on $\mathbb{R}$;
- $x \leqslant \psi_\alpha^{-1}\big(\psi_\alpha(x)\big)$ for all $x \in \mathbb{R}_+$.

The proof is based on a Pisier-type argument as is done, for example, by Massart (2007, Lemma 2.3) to prove the maximal inequality $\mathbb{E}[\max_{1\leqslant t\leqslant T} \xi_t] \leqslant \sqrt{2\nu \ln T}$ for all subgaussian real random variables ξ_t, $1 \leqslant t \leqslant T$, with common variance factor $\nu \geqslant 0$.

From the inequality $x \leqslant \psi_\alpha^{-1}\big(\psi_\alpha(x)\big)$ for all $x \in \mathbb{R}_+$ we have

$$\begin{aligned} \mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right] &\leqslant \psi_\alpha^{-1}\left(\psi_\alpha\left(\mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right]\right)\right) \\ &\leqslant \psi_\alpha^{-1}\left(\mathbb{E}\left[\psi_\alpha\Big(\max_{1\leqslant t\leqslant T} Z_t^2\Big)\right]\right) = \psi_\alpha^{-1}\left(\mathbb{E}\left[\max_{1\leqslant t\leqslant T} \psi_\alpha\big(Z_t^2\big)\right]\right) \,, \end{aligned}$$

where the last two inequalities follow by Jensen's inequality (since ψ_α is convex) and the fact that both ψ_α^{-1} and ψ_α are nondecreasing.

Since $\psi_\alpha \geqslant 0$ and ψ_α^{-1} is nondecreasing we get

$$\begin{aligned} \mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right] &\leqslant \psi_\alpha^{-1}\left(\mathbb{E}\left[\sum_{t=1}^T \psi_\alpha\big(Z_t^2\big)\right]\right) = \psi_\alpha^{-1}\left(\sum_{t=1}^T \mathbb{E}\Big[\psi_\alpha\big(Z_t^2\big)\Big]\right) \\ &\leqslant \psi_\alpha^{-1}\left(\sum_{t=1}^T \mathbb{E}\Big[e^{\alpha|Z_t|} + e\Big]\right) \\ &\leqslant \psi_\alpha^{-1}\big(MT + eT\big) = \frac{\ln^2\big(MT + eT\big)}{\alpha^2} \,, \end{aligned}$$

where the second line follows from the inequality $\psi_\alpha(x) \leqslant e + e^{\alpha\sqrt{x}}$ for all $x \in \mathbb{R}_+$, and where the last line follows from the bounded exponential moment assumption and the definition of ψ_α^{-1}. It concludes the proof. ■

Proof (of Lemma 26) As in the previous proof, we have, by Jensen's inequality and the fact that $x \mapsto x^{\alpha/2}$ is convex and nondecreasing on $\mathbb{R}_+$ (since $\alpha > 2$),

$$\begin{aligned} \mathbb{E}\left[\max_{1\leqslant t\leqslant T} Z_t^2\right] &\leqslant \mathbb{E}\left[\left(\max_{1\leqslant t\leqslant T} Z_t^2\right)^{\alpha/2}\right]^{2/\alpha} = \mathbb{E}\left[\max_{1\leqslant t\leqslant T} \big|Z_t\big|^\alpha\right]^{2/\alpha} \\ &\leqslant \mathbb{E}\left[\sum_{t=1}^T \big|Z_t\big|^\alpha\right]^{2/\alpha} \leqslant (MT)^{2/\alpha} \end{aligned}$$

by the bounded-moment assumption, which concludes the proof. ■

References

F. Abramovich, Y. Benjamini, D.L. Donoho, and I.M. Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *Ann. Statist.*, 34(2):584–653, 2006.

P. Alquier and K. Lounici. PAC-Bayesian bounds for sparse regression estimation with exponential weights. *Electron. J. Stat.*, 5:127–145, 2011.

J.-Y. Audibert. Fast learning rates in statistical inference through aggregation. *Ann. Statist.*, 37(4): 1591–1646, 2009. ISSN 0090-5364.

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *J. Comp. Sys. Sci.*, 64:48–75, 2002.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Mach. Learn.*, 43(3):211–246, 2001. ISSN 0885-6125.

G. Biau, K. Bleakley, L. Györfi, and G. Ottucsák. Nonparametric sequential prediction of time series. *J. Nonparametr. Stat.*, 22(3–4):297–317, 2010.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009. ISSN 0090-5364.

L. Birgé and P. Massart. Gaussian model selection. *J. Eur. Math. Soc.*, 3:203–268, 2001.

L. Birgé and P. Massart. Minimal penalties for Gaussian model selection. *Probab. Theory Relat. Fields*, 138:33–73, 2007.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer, Heidelberg, 2011.

F. Bunea and A. Nobel. Sequential procedures for aggregating arbitrary estimators of a conditional mean. *IEEE Trans. Inform. Theory*, 54(4):1725–1735, 2008. ISSN 0018-9448.

F. Bunea, A. B. Tsybakov, and M. H. Wegkamp. Aggregation for regression learning. Technical report, 2004. Available at `http://arxiv.org/abs/math/0410214`.

F. Bunea, A. B. Tsybakov, and M. H. Wegkamp. Aggregation for Gaussian regression. *Ann. Statist.*, 35(4):1674–1697, 2007a. ISSN 0090-5364.

F. Bunea, A. B. Tsybakov, and M. H. Wegkamp. Sparsity oracle inequalities for the Lasso. *Electron. J. Stat.*, 1:169–194, 2007b. ISSN 1935-7524.

E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Ann. Statist.*, 35(6):2313–2351, 2007.

O. Catoni. Universal aggregation rules with exact bias bounds. Technical Report PMA-510, Laboratoire de Probabilités et Modèles Aléatoires, CNRS, Paris, 1999.

O. Catoni. *Statistical Learning Theory and Stochastic Optimization*. Springer, New York, 2004.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inform. Theory*, 50(9):2050–2057, 2004. ISSN 0018-9448.

N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2/3):321–352, 2007.

A. Dalalyan and A. B. Tsybakov. Aggregation by exponential weighting and sharp oracle inequalities. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07)*, pages 97–111, 2007. ISBN 978-3-540-72925-9.

A. Dalalyan and A. B. Tsybakov. Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity. *Mach. Learn.*, 72(1-2):39–61, 2008. ISSN 0885-6125.

A. Dalalyan and A. B. Tsybakov. Mirror averaging with sparsity priors. *Bernoulli*, 18(3):914–944, 2012a.

A. Dalalyan and A. B. Tsybakov. Sparse regression learning by aggregation and Langevin Monte-Carlo. *J. Comput. System Sci.*, 78:1423–1443, 2012b.

D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81 (3):425–455, 1994. ISSN 0006-3444.

J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT'10)*, pages 14–26, 2010.

Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC'97)*, pages 334–343, 1997.

S. Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *JMLR Workshop and Conference Proceedings*, 19 (COLT 2011 Proceedings):377–396, 2011.

S. Gerchinovitz and J.Y. Yu. Adaptive and optimal online linear regression on ℓ^1-balls. In J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, editors, *Algorithmic Learning Theory*, volume 6925 of *Lecture Notes in Computer Science*, pages 99–113. Springer Berlin/Heidelberg, 2011.

L. Györfi and G. Ottucsák. Sequential prediction of unbounded stationary time series. *IEEE Trans. Inform. Theory*, 53(5):1866–1872, 2007.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer-Verlag, New York, 2002. ISBN 0-387-95441-4.

M. Hebiri and S. van de Geer. The Smooth-Lasso and other $\ell^1 + \ell^2$-penalized methods. *Electron. J. Stat.*, 5:1184–1226, 2011.

A. Juditsky, P. Rigollet, and A. B. Tsybakov. Learning by mirror averaging. *Ann. Statist.*, 36(5): 2183–2206, 2008. ISSN 0090-5364.

J. Kivinen and M. K. Warmuth. Averaging expert predictions. In *Proceedings of the 4th European Conference on Computational Learning Theory (EuroCOLT'99)*, pages 153–167, 1999.

V. Koltchinskii. Sparse recovery in convex hulls via entropy penalization. *Ann. Statist.*, 37(3): 1332–1359, 2009a. ISSN 0090-5364.

V. Koltchinskii. Sparsity in penalized empirical risk minimization. *Ann. Inst. Henri Poincaré Probab. Stat.*, 45(1):7–57, 2009b. ISSN 0246-0203.

V. Koltchinskii, K. Lounici, and A.B. Tsybakov. Nuclear norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Statist.*, 39(5):2302–2329, 2011.

J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801, 2009. ISSN 1532–4435.

N. Littlestone. From on-line to batch learning. In *Proceedings of the 2nd Annual Conference on Learning Theory (COLT'89)*, pages 269–284, 1989.

K. Lounici, M. Pontil, S. van de Geer, and A.B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *Ann. Statist.*, 39(4):2164–2204, 2011.

P. Massart. *Concentration Inequalities and Model Selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

P. Rigollet and A. B. Tsybakov. Exponential Screening and optimal rates of sparse estimation. *Ann. Statist.*, 39(2):731–771, 2011.

M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *J. Mach. Learn. Res.*, 9:759–813, 2008.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for ℓ^1-regularized loss minimization. *J. Mach. Learn. Res.*, 12:1865–1892, 2011.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1): 267–288, 1996.

S. A. van de Geer. High-dimensional generalized linear models and the Lasso. *Ann. Statist.*, 36(2): 614–645, 2008. ISSN 0090-5364.

S. A. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the Lasso. *Electron. J. Stat.*, 3:1360–1392, 2009. ISSN 1935-7524.

V. Vovk. Competitive on-line statistics. *Internat. Statist. Rev.*, 69:213–248, 2001.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596, 2010.

Journal of Machine Learning Research 14 (2013) 771-800 Submitted 4/11; Revised 9/12; Published 3/13

Semi-Supervised Learning Using Greedy Max-Cut

Jun Wang WANGJUN@US.IBM.COM
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598, USA

Tony Jebara JEBARA@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
New York, NY 10027, USA

Shih-Fu Chang SFCHANG@EE.COLUMBIA.EDU
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA

Editor: Mikhail Belkin

Abstract

Graph-based semi-supervised learning (*SSL*) methods play an increasingly important role in practical machine learning systems, particularly in agnostic settings when no parametric information or other prior knowledge is available about the data distribution. Given the constructed graph represented by a weight matrix, transductive inference is used to propagate known labels to predict the values of all unlabeled vertices. Designing a robust label diffusion algorithm for such graphs is a widely studied problem and various methods have recently been suggested. Many of these can be formalized as regularized function estimation through the minimization of a quadratic cost. However, most existing label diffusion methods minimize a univariate cost with the classification function as the only variable of interest. Since the observed labels seed the diffusion process, such univariate frameworks are extremely sensitive to the initial label choice and any label noise. To alleviate the dependency on the initial observed labels, this article proposes a bivariate formulation for graph-based *SSL*, where both the binary label information and a continuous classification function are arguments of the optimization. This bivariate formulation is shown to be equivalent to a linearly constrained Max-Cut problem. Finally an efficient solution via greedy gradient Max-Cut (*GGMC*) is derived which gradually assigns unlabeled vertices to each class with minimum connectivity. Once convergence guarantees are established, this greedy Max-Cut based *SSL* is applied on both artificial and standard benchmark data sets where it obtains superior classification accuracy compared to existing state-of-the-art *SSL* methods. Moreover, *GGMC* shows robustness with respect to the graph construction method and maintains high accuracy over extensive experiments with various edge linking and weighting schemes.

Keywords: graph transduction, semi-supervised learning, bivariate formulation, mixed integer programming, greedy Max-Cut

1. Introduction

In many real applications, labeled samples are scarce but unlabeled samples are abundant. Paradigms that consider both labeled and unlabeled data, that is, semi-supervised learning (*SSL*) methods, have

been increasingly explored in practical machine learning systems. While many semi-supervised learning approaches estimate a smooth function over labeled and unlabeled examples, this article presents a novel approach which emphasizes a *bivariate* optimization problem over the classification function *and* the labels. Prior to describing the method in detail, we briefly mention other *SSL* methods and previous work to motivate this article's contributions.

One of the earliest examples of the empirical advantages of *SSL* was co-training, a method first developed for text mining problems (Blum and Mitchell, 1998) and later extended in various forms to other applications (Chawla and Karakoulas, 2005; Goldman and Zhou, 2000). Therein, multiple classifiers are first estimated using conditionally independent feature sets of training data. The performance advantages of this method rely heavily on the existence of independent and complementary classifiers. Theoretical results show that some mild assumptions on the underlying data distribution are sufficient for co-training to work (Balcan et al., 2005; Wang and Zhou, 2010). However, performance can dramatically degrade if the classifiers do not complement each other or the independence assumption does not hold (Krogel and Scheffer, 2004). Though co-training is conceptually similar to semi-supervised learning due to the way it incorporates unlabeled data, the classifier training procedure itself is often supervised.

The extension of traditional supervised support vector machines (*SVMs*) to the semi-supervised scenario is another widely used *SSL* algorithm. Instead of maximizing separation (via a maximum-margin hyperplane) over training data as in standard *SVMs*, semi-supervised *SVMs* (*S3VMs*) estimate a hyperplane to balance maximum-margin partitioning of labeled data while encouraging a separation through low-density regions of the data (Vapnik, 1998). For example, transductive support vector machines (TSVMs) were developed as one of the earliest incarnations of semi-supervised *SVMs* (Joachims, 1999).[1] Various optimization techniques have been applied to solve *S3VMs* (Chapelle et al., 2008), resulting in a wide range of methods, such as low density separation (Chapelle and Zien, 2005), semi-definite programming based methods (Bie and Cristianini, 2004; Xu et al., 2008), and a branch-and-bound based approach (Chapelle et al., 2007).

Another family of *SSL* methods known as graph-based approaches have recently become popular due to their high accuracy and computational efficiency. Graph-based semi-supervised learning (*GSSL*) treats both labeled and unlabeled samples from a data set as vertices in a graph and builds pairwise edges between these vertices which are weighted by the affinity between the corresponding samples. The small portion of vertices with labels are then used by *SSL* methods to perform graph partition or information propagation to predict labels for unlabeled vertices. For instance, the graph mincuts approach formulates the label prediction as a graph cut problem (Blum and Chawla, 2001; Blum et al., 2004). Other *GSSL* methods, like graph transductive learning, formulate the problem as regularized function estimation over an undirected weighted graph. These methods optimize a trade-off between the accuracy of the classification function on labeled samples and a regularization term that favors a smooth function. The weighted graph and the optimal function ultimately propagate label information from labeled data to unlabeled data to produce transductive predictions. Popular algorithms for *GSSL* include graph cuts (Blum and Chawla, 2001; Blum et al., 2004; Joachims, 2003; Kveton et al., 2010), graph random walks (Azran, 2007; Szummer and Jaakkola, 2002), manifold regularization (Belkin et al., 2005, 2006; Sindhwani et al., 2008, 2005), and graph regularization (Zhou et al., 2004; Zhu et al., 2003). Comprehensive survey articles have also been disseminated (Zhu, 2005).

1. It is actually more appropriate to call this method a semi-supervised *SVM* since the learned classifier is indeed inductive (Zhu and Goldberg, 2009).

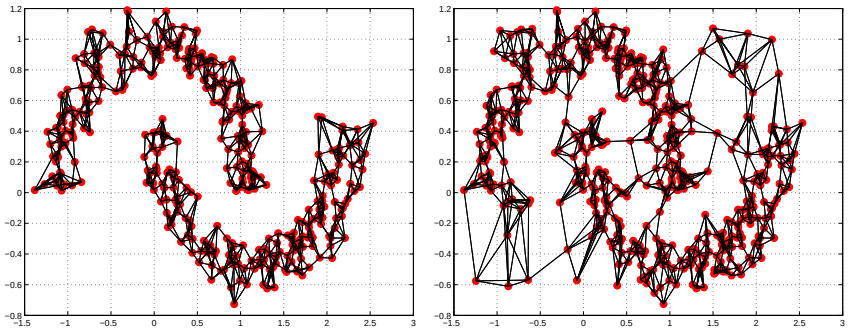
For some synthetic and real data problems, *GSSL* approaches do achieve promising performance. However, previous research has identified several realistic settings and labeling situations where this performance can be compromised (Wang et al., 2008b). In particular, both the graph construction methodology and the label initialization conditions can significantly impact prediction accuracy (Jebara et al., 2009). For a well-constructed graph such as the one shown in Figure 1(a), many *GSSL* methods produce satisfactory predictions. However, for graphs involving non-separable manifold structure as shown in Figure 1(b), prediction accuracy may deteriorate. Even if one assumes that the graph structures used in the above methods faithfully describe the data manifold, *GSSL* algorithms may still be misled by problems in the label information. Figure 3 depicts several cases where the label information leads to invalid graph transduction solutions for all the aforementioned algorithms.

In order to handle such challenging labeling conditions, we first extend the existing *GSSL* formulation by casting it as a *bivariate* optimization problem over the classification function *and* the labels. Then we demonstrate that minimizing the mixed bivariate cost function can be reduced to a pure integer programming problem that is equivalent to a constrained Max-Cut problem. Though semi-definite programming can be used to obtain approximate solutions, these are impractical due to scalability issues. Instead, an efficient greedy gradient Max-Cut (*GGMC*) solution is developed which remedies the instability previous methods seem to have vis-a-vis the initial labeling conditions on the graph. In the proposed greedy solution, initial labels simply act as initial values of the graph cut which is incrementally refined until convergence. During each iteration of the greedy search, the optimal unlabeled vertex is assigned to the labeled subset with minimum connectivity to maximally preserve cross-subset edge weight. Finally, an overall cut is produced after placing the unlabeled vertices into one of the label sets. It is then straightforward to obtain the final label prediction from the graph cut result. Note that this greedy gradient Max-Cut solution is equivalent to alternating between minimization of the cost over the label matrix and minimization of the cost over the prediction function. Moreover, to alleviate dependencies on the initialization of the cut (the given labels), a re-weighting of the connectivity between unlabeled vertices and labeled subsets is proposed. This re-weighting performs a within-class normalization using vertex degree as well as a between-class normalization using class prior information. We demonstrate that the greedy gradient Max-Cut based graph transduction produces significantly better performance on both artificial and real data sets.

The remainder of this paper is organized as the follows. Section 2 provides a brief background of graph-based *SSL* and discusses some open issues. In Section 3, we present our bivariate graph transduction framework, followed by the theoretical proof of its equivalence with the constrained Max-Cut problem in Section 4. In addition, a greedy gradient Max-Cut algorithm is proposed. Section 5 provides experimental validation for the algorithm on both toy and real classification data sets. Comparisons with leading semi-supervised methods are made. Concluding remarks and discussions are then provided in Section 6.

2. Background and Open Issues

In this section, we introduce some notation and then revisit two critical components of graph-based *SSL*: graph construction and label propagation. Subsequently, we discuss some challenging issues such as *SSL*'s sensitivity to graph construction and label initialization.



(a) (b)

Figure 1: Examples of constructed k-nearest-neighbors (kNN) graphs with $k = 5$ on the artificial two moon data set for a) the completely separable case; and b) the non-separable case with noisy samples.

2.1 Notations

We first summarize the notation used in this article. Assume we are given *iid* (independent and identically distributed) labeled samples $\{(\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_l, z_l)\}$ as well as unlabeled samples $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$ drawn from a distribution $p(\mathbf{x}, z)$. Define the set of labeled inputs as $\mathbf{X}_l = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ with cardinality $|\mathbf{X}_l| = l$ and the set of unlabeled inputs $\mathbf{X}_u = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$ with cardinality $|\mathbf{X}_u| = u$. The labeled set $\mathbf{X}_l$ is associated with labels $\mathcal{Z}_l = \{z_1, \cdots, z_l\}$, where $z_i \in \{1, \cdots, c\}, i = 1, 2, \cdots, l$. The goal of semi-supervised learning is to infer the missing labels $\{z_{l+1}, \cdots, z_n\}$ corresponding to the unlabeled data $\{\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$, where typically $l << n$ ($l + u = n$). A crucial component of *GSSL* is the estimation of a weighted sparse graph $\mathcal{G}$ from the input data $\mathbf{X} = \mathbf{X}_l \cup \mathbf{X}_u$. Subsequently, a labeling algorithm uses $\mathcal{G}$ and the known labels $\mathcal{Z}_l = \{z_1, \ldots, z_l\}$ to provide estimates $\hat{\mathcal{Z}}_u = \{\hat{z}_{l+1}, \ldots, \hat{z}_{l+u}\}$ which try to approximate the true labels $\mathcal{Z}_u = \{z_{l+1}, \ldots, z_{l+u}\}$ as measured by an appropriately chosen loss function.

In this article, assume the undirected graph converted from the data $\mathbf{X}$ is represented by $\mathcal{G} = \{\mathbf{X}, \mathbf{E}\}$, where the set of vertices is $\mathbf{X} = \{\mathbf{x}_i\}$ and the set of edges is $\mathbf{E} = \{e_{ij}\}$. Each sample $\mathbf{x}_i$ is treated as a vertex and the weight of edge e_{ij} is w_{ij}. Typically, one uses a kernel function $k(\cdot)$ over pairs of points to compute weights. The weights for edges are used to build a weight matrix which is denoted by $\mathbf{W} = \{w_{ij}\}$. Similarly, the vertex degree matrix $\mathbf{D} = diag([d_1, \cdots, d_n])$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The graph Laplacian is defined as $\mathbf{\Delta} = \mathbf{D} - \mathbf{W}$ and the normalized graph Laplacian is

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}.$$

The graph Laplacian and its normalized version can be viewed as operators on the space of functions f which can be used to define a regularization measure of smoothness over strongly-connected regions in a graph (Chung and Biggs, 1997). For example, the smoothness measurement of functions f using $\mathbf{L}$ over a graph is defined as

$$\langle f, \mathbf{L}f \rangle = \sum_i \sum_j w_{ij} \left\| \frac{f(\mathbf{x}_i)}{\sqrt{d_i}} - \frac{f(\mathbf{x}_j)}{\sqrt{d_j}} \right\|^2.$$

Finally, the label information is formulated as a label matrix $\mathbf{Y} = \{y_{ij}\} \in \mathbb{B}^{n\times c}$, where $y_{ij} = 1$ if sample $\mathbf{x}_i$ is associated with label j for $j \in \{1,2,\cdots,c\}$, that is, $z_i = j$, and $y_{ij} = 0$ otherwise. For single label problems (as opposed to multi-label problems), the constraints $\sum_{j=1}^{c} y_{ij} = 1$ are also imposed. Moreover, we will often refer to row and column vectors of such matrices, for instance, the i'th row and j'th column vectors of $\mathbf{Y}$ are denoted as $\mathbf{Y}_{i\cdot}$ and $\mathbf{Y}_{\cdot j}$, respectively. Let $\mathbf{F} = f(\mathbf{X})$ be the values of classification function over the data set $\mathbf{X}$. Most of the *GSSL* methods then use the graph quantity $\mathbf{W}$ as well as the known labels to recover a continuous classification function $\mathbf{F} \in \mathbb{R}^{n\times c}$ by minimizing a predefined cost on the graph.

2.2 Graph Construction for Semi-Supervised Learning

To estimate $\hat{\mathcal{Z}}_u = \{\hat{z}_{l+1},\ldots,\hat{z}_{l+u}\}$ using $\mathcal{G}$ and the known labels $\mathcal{Z}_l = \{z_1,\ldots,z_l\}$, we first convert the data points $\mathbf{X} = \mathbf{X}_l \cup \mathbf{X}_u$ into a graph $\mathcal{G} = \{\mathbf{X},\mathbf{E},\mathbf{W}\}$. This section discusses the graph construction method, $\mathbf{X} \to \mathcal{G}$, in detail. Given input data $\mathbf{X}$ with cardinality $|\mathbf{X}| = l+u$, graph construction produces a graph $\mathcal{G}$ consisting of $n = l+u$ vertices where each vertex is associated with the sample $\mathbf{x}_i$. The estimation of $\mathcal{G}$ from $\mathbf{X}$ usually proceeds in two steps.

The first step is to compute a score between all pairs of vertices using a similarity function. This creates a full adjacency matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$ is computed using kernel function $k(\cdot)$ to measure sample similarity. Subsequently, in the second step of graph construction, the matrix $\mathbf{K}$ is sparsified and reweighted to produce the final matrix $\mathbf{W}$. Sparsification is important since it leads to improved efficiency, better accuracy, and robustness to noise in the label inference stage. Furthermore, the kernel function $k(\cdot)$ is often only locally useful as a similarity and does not recover reliable weights between pairs of samples that are relatively far apart.

2.2.1 GRAPH SPARSIFICATION

Starting with the fully connected matrix $\mathbf{K}$, sparsification removes edges by recovering a binary matrix $\mathbf{B} \in \mathbb{B}^{n\times n}$ where $\mathbf{B}_{ij} = 1$ indicates that an edge is present between sample $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{B}_{ij} = 0$ indicates the edge is absent (assume $\mathbf{B}_{ii} = 0$ unless otherwise noted). Here we will primarily investigate two graph sparsification algorithms: neighborhood approaches including the k-nearest and ε neighbors algorithms, and matching approaches such as b-matching (BM) (Edmonds and Johnson, 2003). All such methods operate on the matrix $\mathbf{K}$ or, equivalently, the distance matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$ obtained from $\mathbf{K}$ element-wise as $\mathbf{H}_{ij} = \sqrt{\mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij}}$.

Sparsification via Neighborhood Methods: There are two typical ways to build a neighborhood graph: the ε-neighborhood graph connecting samples within a distance of ε, and the kNN (k-nearest-neighbors) graph connecting k closest samples. Recent studies show the dramatic influences that different neighborhood methods have on clustering techniques (Carreira-Perpinán and Zemel, 2005; Maier et al., 2009). In practice, the kNN graph remains a more common approach since it is more adaptive to scale variation and data density anomalies while an improper threshold value in the ε-neighborhood graph may result in disconnected components or subgraphs in the data set or even isolated singleton vertices, as shown in Figure 2(b). In this article, we often use kNN neighborhood graphs since the ε-neighborhood graphs provide consistently weaker performance. In the remainder of this article, we will use neighborhood and kNN neighborhood graph interchangeably without specific declaration.

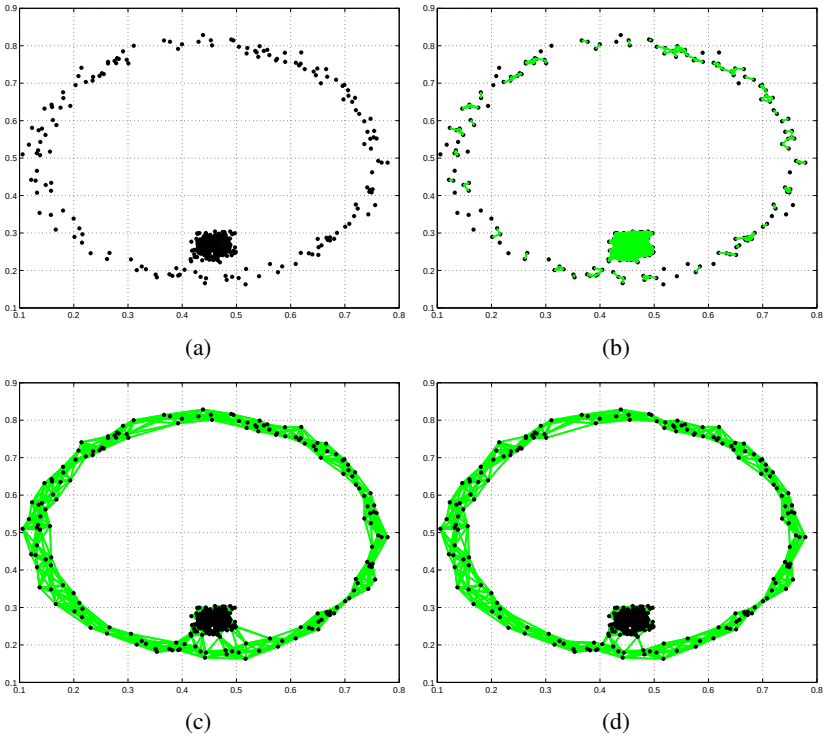


Figure 2: The synthetic data set used for demonstrating different graph construction approaches. a) The synthetic data; b) The ε-nearest neighbor graph; c) The k-nearest neighbor graph; d) The b-matched graph.

More specifically, the k-nearest neighbor graph is a graph in which two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by an edge if the distance $\mathbf{H}_{ij}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ is within or equal k-th smallest among the distances from $\mathbf{x}_i$ to other samples in $\mathbf{X}$. Roughly speaking, the k-nearest neighbors algorithm starts with a matrix $\hat{\mathbf{B}}$ of all zeros and for each point, searches for the k closest points to it (without considering itself). If a point j is one of the k closest neighbors to i, then we set $\hat{\mathbf{B}}_{ij} = 1$. It is straightforward to show that k-nearest neighbors search solves the following optimization problem:

$$\begin{aligned} &\min_{\hat{\mathbf{B}} \in \mathbb{B}} \sum_{ij} \hat{\mathbf{B}}_{ij} \mathbf{H}_{ij} \\ &s.t. \sum_j \hat{\mathbf{B}}_{ij} = k, \hat{\mathbf{B}}_{ii} = 0, \forall i, j \in 1, \ldots, n. \end{aligned} \tag{1}$$

The final solution of Equation (1) is produced by symmetrizing $\hat{\mathbf{B}}$ as follows $\mathbf{B}_{ij} = \max(\hat{\mathbf{B}}_{ij}, \hat{\mathbf{B}}_{ji})$.[2] This greedy algorithm is in fact not solving a well defined optimization problem over symmetric binary matrices. In addition, since it produces a symmetric matrix only via the *ad hoc* maximization over $\hat{\mathbf{B}}$ and its transpose, the solution $\mathbf{B}$ it produces does not satisfy the equality $\sum_k \mathbf{B}_{ij} = k$, but, rather, only satisfies the inequality $\sum_j \mathbf{B}_{ij} \geq k$. Ironically, despite conventional wisdom and the nomenclature, the k-nearest neighbors algorithm is producing an undirected subgraph with more

2. It is possible to replace the maximization operator with minimization to produce a symmetric matrix, yet in the setting $\mathbf{B} = \min(\hat{\mathbf{B}}, \hat{\mathbf{B}}^\top)$ the solution $\mathbf{B}$ only satisfies the inequality $\sum_j \mathbf{B}_{ij} \leq k$ and not the desired equality.

than k neighbors for each vertex. This motivates researchers to investigate the b-matching algorithm which actually achieves the desired output.

Sparsification via b-Matching: The b-matching problem generalizes maximum weight matching, that is, the linear assignment problem, where the objective is to find the binary matrix to minimize the optimization problem

$$\begin{aligned} &\min_{\mathbf{B}\in\mathbb{B}} \sum_{ij} \mathbf{B}_{ij}\mathbf{H}_{ij} \\ &s.t. \sum_j \mathbf{B}_{ij} = b, \mathbf{B}_{ii} = 0, \mathbf{B}_{ij} = \mathbf{B}_{ji}, \forall i, j \in 1,\ldots,n. \end{aligned} \tag{2}$$

achieving symmetry directly without post-processing. Here, the symmetric solution is recovered up-front by enforcing the additional constraints $\mathbf{B}_{ij} = \mathbf{B}_{ji}$. The matrix then satisfies the equality $\sum_j \mathbf{B}_{ij} = \sum_i \mathbf{B}_{ij} = b$ strictly. The solution to Equation (2) is not quite as straightforward or efficient as the greedy k-nearest neighbors algorithm. A polynomial time $O(bn^3)$ solution has been known, yet recent advances show that much faster alternatives are possible via (guaranteed) loopy belief propagation (Huang and Jebara, 2007).

Compared with the neighborhood graphs, the b-matching graph is balanced or b-regular. In other words, each vertex in the b-matched graph has exactly b edges connecting it to other vertices. This advantage plays a key role when conducting label propagation on typical samples $\mathbf{X}$ which are unevenly and non-uniformly distributed. Our previous work applied b-matching to construct graphs for semi-supervised learning tasks and demonstrated the superior performance over some unevenly sampled data (Jebara et al., 2009). For example, in Figure 2, this data set clearly contains two clusters of points, a dense Gaussian cluster surrounded by a ring cluster. Furthermore, the cluster data is unevenly sampled; one cluster is dense and the other is fairly sparse. In this example, the k-nearest neighbor graph constantly generates many cross-cluster edges while b-matching efficiently alleviates this problem by removing most of the improper edges. The example clearly shows that the b-matching technique produces regular graphs which could overcome the drawback of cross-structure linkages often generated by nearest neighbor methods. This intuitive study confirms the importance of graph construction methods and advocates b-matching as a valuable alternative to k-nearest neighbors, a method that many practitioners expect to produce regular undirected graphs, though in practice often generates irregular graphs.

2.2.2 GRAPH EDGE RE-WEIGHTING

Once a graph has been sparsified and a binary matrix $\mathbf{B}$ is computed and used to delete unwanted edges, several procedures can then be used to update the weights in the matrix $\mathbf{K}$ to produce a final set of edge weights $\mathbf{W}$. Specifically, whenever $\mathbf{B}_{ij} = 0$, the edge weight is also $w_{ij} = 0$; however, $\mathbf{B}_{ij} = 1$ implies that $w_{ij} \geq 0$. Two popular approaches are considered here for estimating the non-zero components of $\mathbf{W}$.

Binary Weighting: The simplest approach for building the weighted graph is the *binary* weighting approach, where all the linked edges in the graph are given the weight 1 and the edge weights of disconnected vertices are given the weight 0. In other words, this setting simply uses $\mathbf{W} = \mathbf{B}$. However, this uniform weight on graph edges can be sensitive, particularly if some of the graph vertices were improperly connected by the sparsification procedure (either the neighborhood based procedures or the b-matching procedure).

Gaussian Kernel Weighting: An alternative approach is *Gaussian kernel* weighting which is often applied to modulate sample similarity. Therein, the edge weight between two connected

samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed as:

$$w_{ij} = \mathbf{B}_{ij} \exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right),$$

where the function $d(\mathbf{x}_i, \mathbf{x}_j)$ evaluates the dissimilarity of samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is the kernel bandwidth parameter. There are many choices for the distance function $d(\cdot)$ including any ℓ_p distance, χ^2 distance, and cosine distance (Zhu, 2005; Belkin et al., 2005; Jebara et al., 2009).

This final step in the graph construction procedure ensures that the unlabeled data $\mathbf{X}$ has now been converted into a graph $\mathcal{G}$ with a weighted sparse undirected adjacency matrix $\mathbf{W}$. Given this graph and some initial label information $\mathbf{Y}_l$, any of the current popular algorithms for graph based SSL can be used to solve the labeling problem.

2.3 Univariate Graph Regularization Framework

Given the constructed graph $\mathcal{G} = \{\mathbf{X}, \mathbf{E}\}$, whose geometric structure is represented by the weight matrix $\mathbf{W}$, the label inference task is to diffuse the known labels $\mathcal{Z}_l$ to all the unlabeled vertices $\mathbf{X}_u$ in the graph and estimate $\hat{\mathcal{Z}}_u$. Designing a robust label diffusion algorithm for such graphs is a widely studied problem (Chapelle et al., 2006; Zhu, 2005; Zhu and Goldberg, 2009).

Here we are particularly interested in a category of approaches, which estimate the prediction function $\mathbf{F} \in \mathbb{R}^{n \times c}$ by minimizing a quadratic cost defined over the graph. The cost function typically involves a trade-off between the smoothness of the function over the graph of both labeled and unlabeled data (consistency of the predictions on closely connected vertices) and the accuracy of the function at fitting the label information on the labeled vertices. Approaches like the Gaussian fields and harmonic functions (*GFHF*) method (Zhu et al., 2003) and the local and global consistency (*LGC*) method (Zhou et al., 2004) fall into this category, so does our previous method of graph transduction via alternating minimization (Wang et al., 2008b).

Both *LGC* and *GFHF* define a cost function Q that involves the combined contribution of two penalty terms: the global smoothness Q_{smooth} and local fitting accuracy Q_{fit}. The final prediction function $\mathbf{F}$ is obtained by minimizing the cost function as:

$$\mathbf{F}^* = \arg\min_{\mathbf{F} \in \mathbb{R}^{n \times c}} Q(\mathbf{F}) = \arg\min_{\mathbf{F} \in \mathbb{R}^{n \times c}} \left(Q_{smooth}(\mathbf{F}) + Q_{fit}(\mathbf{F})\right). \tag{3}$$

A natural formulation of the above cost function is *LGC* (Zhou et al., 2004) which uses an elastic regularizer framework as follows

$$Q(\mathbf{F}) = \|\mathbf{F}\|_{\mathcal{G}}^2 + \frac{\mu}{2}\|\mathbf{F} - \mathbf{Y}\|^2. \tag{4}$$

The first term $\|\mathbf{F}\|_{\mathcal{G}}^2$ represents function smoothness over graph $\mathcal{G}$ and $\|\mathbf{F} - \mathbf{Y}\|^2$ measures the empirical loss on given labeled samples. Specifically, in *LGC*, the function smoothness is defined using the semi-inner product

$$Q_{smooth} = \|\mathbf{F}\|_{\mathcal{G}}^2 = \frac{1}{2}\langle \mathbf{F}, \mathbf{L}\mathbf{F} \rangle = \frac{1}{2}\mathrm{tr}(\mathbf{F}^\top \mathbf{L}\mathbf{F}).$$

Note that the coefficient μ in Equation (4) balances global smoothness and local fitting terms. If we set $\mu = \infty$ and use a standard graph Laplacian quantity $\mathbf{\Delta}$ for the smoothness term, the above

framework reduces to the harmonic function formulation (Zhu et al., 2003). More precisely, the cost function only preserves the smoothness term as

$$Q(\mathbf{F}) = \mathrm{tr}(\mathbf{F}^\top \boldsymbol{\Delta} \mathbf{F}). \tag{5}$$

Meanwhile, the harmonic function $\mathbf{F}$ minimizing the above cost also satisfies two conditions:

$$\begin{aligned} &\frac{\partial Q}{\partial \mathbf{F}_u} = \boldsymbol{\Delta}\mathbf{F}_u = 0, \\ &\mathbf{F}_l = \mathbf{Y}_l, \end{aligned}$$

where $\mathbf{F}_l, \mathbf{F}_u$ are the function values of $f(\cdot)$ over labeled and unlabeled vertices, that is, $\mathbf{F}_l = f(\mathbf{X}_l)$, $\mathbf{F}_u = f(\mathbf{X}_u)$, and $\mathbf{F} = [\mathbf{F}_l \ \ \mathbf{F}_u]^\top$. The first equation above denotes the zero derivative of the object function on the unlabeled data and the second equation clamps the function value on the given label value $\mathbf{Y}_l$. Both *LGC* and *GFHF* are univariate regularization frameworks where the continuous prediction function is treated as the only variable in the optimization procedure. The optimal solutions for Equation (4) and Equation (5) are easily obtained by solving a linear system.

2.4 Open Issues

Existing graph-based *SSL* methods hinge on having good label information and an appropriately constructed graph (Wang et al., 2008b; Liu et al., 2012). But the heuristic design of the graph may result in suboptimal inference. In addition, the label propagation procedure can easily be misled if there exist excessive noise or outliers in the initial labeled set. Finally, in *iid* settings, the difference between empirically estimated class proportions and their true expected value is bounded (Huang and Jebara, 2010). However, practical annotation procedures are not necessarily *iid* and labeled data may have empirical class frequencies that deviate significantly from the expected class ratios. These degenerate situations seem to plague real world problems and compromise the performance of many state-of-the-art *SSL* algorithms. We next discuss some open issues which occur often in graph construction and label propagation, two critical components of all *GSSL* algorithms.

2.4.1 SENSITIVITY TO GRAPH CONSTRUCTION

As shown in Figure 1(a), a well-built graph obtained from separable manifolds of data will achieve good results with most existing *GSSL* approaches. However, practical applications often produce non-separable graphs as shown in Figure 1(b). In addition to the widely used kNN graph, we showed that b-matching could be used successfully for graph construction (Jebara et al., 2009). But both kNN graphs and b-matched graphs are heuristics and require the careful selection of the parameter k or b which controls the number of links incident to each vertex in the graph. Moreover, edge reweighing on the sparse graph often also requires exploration forcing the user to select kernels and various kernel parameters. All these heuristic steps in graph design require extra effort from the user and demand some level of familiarity with the data domain.

2.4.2 SENSITIVITY TO LABEL NOISE

Most of the existing *GSSL* methods are based on an univariate quadratic regularization framework which relies heavily on the quality of the initially assigned labels. For certain synthetic and real data problems, such graph transduction approaches achieve promising performance. However, several realistic labeling conditions produce unsatisfactory performance (Wang et al., 2008b). Even if

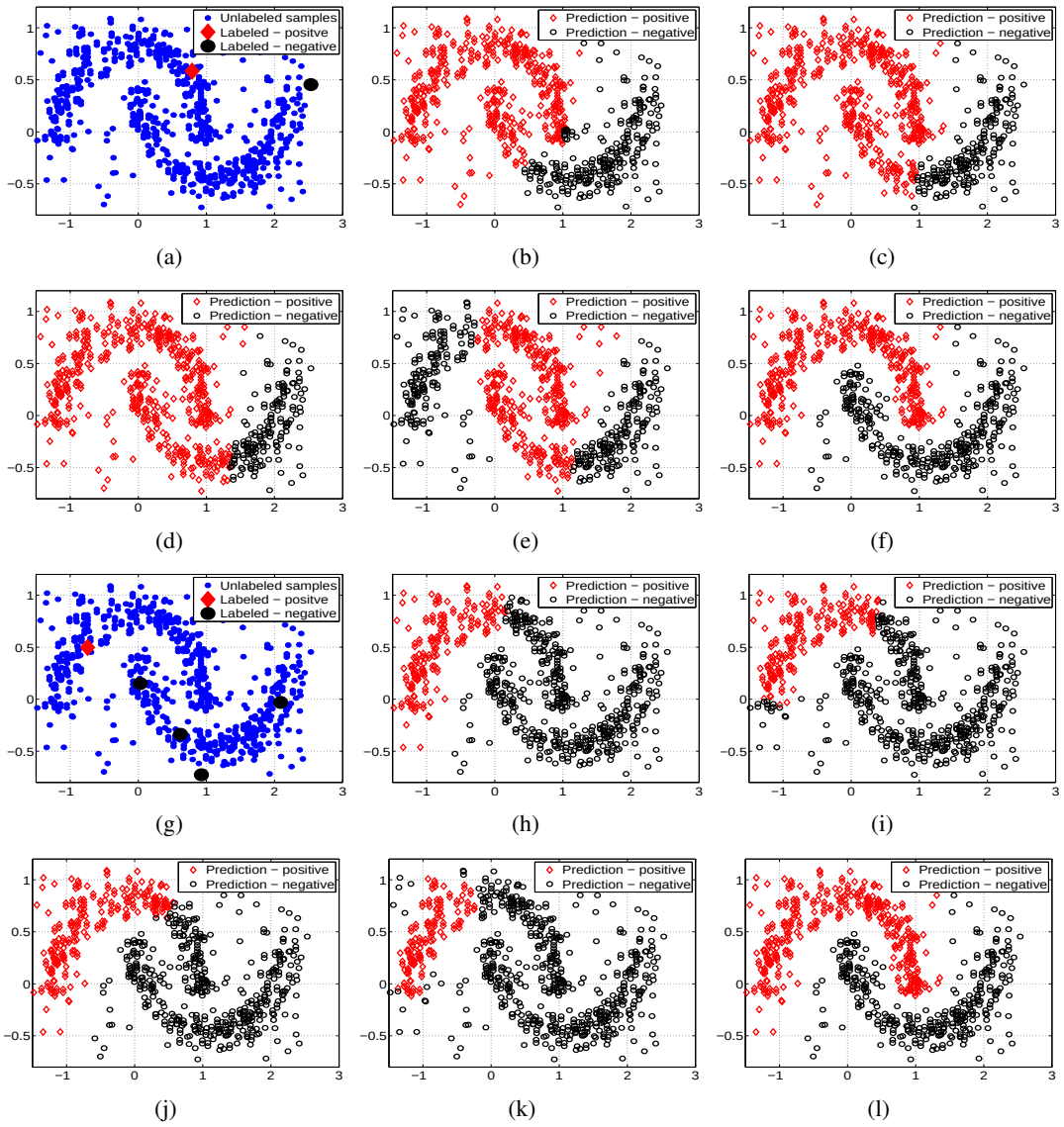


Figure 3: Examples illustrating the sensitivity of graph-based *SSL* to adverse labeling conditions. Particularly challenging conditions are shown in (a) where an uninformative label on an outlier sample is the only negative label (denoted by a black circle) and in (g) where imbalanced labeling is involved. Prediction results are shown for the *GFHF* method (Zhu et al., 2003) in (b) and (h), the *LGC* method (Zhou et al., 2004) in (c) and (i), the *LapSVM* method (Belkin et al., 2006) in (d) and (j), the *TSVM* method (Joachims, 1999) in (e) and (k); and our method in (f) and (l).

the graph is perfectly constructed from the data, problematic initial labels under practical situations can easily deteriorate the performance of *SSL* prediction. Figure 3 provides examples depicting imbalanced and noisy labels that lead to invalid graph transduction solutions for all the aforementioned algorithms. The first labeling problem involves uninformative labels (Figure 3(a)). The only

negative label (dark circle) is located in an outlier region where the low density connectivity limits its diffusion to the rest of the graph. The leading *SSL* methods classify the majority of unlabeled nodes in the graph as positive (Figure 3(b)-Figure 3(e)). Such conditions are frequent in real problems like content-based image retrieval (CBIR) where the visual query example is not necessarily representative of the class. Another difficult case is due to imbalanced labeling. There, the ratio of training labels is disproportionate to the underlying class proportions. For example, Figure 3(g) depicts two half-circles with an almost equal number of samples. However, since the training labels contain three negative samples and only one positive example, the *SSL* predictions are strongly biased towards the negative class (see Figures 3(h) to 3(k)). This imbalanced labeling situation occurs frequently in realistic problems such as the annotation of microscopic images (Wang et al., 2008a). Therein, the human labeler favors certain cellular phenotypes due to domain-specific biological hypotheses. To tackle these issues, we next propose a novel bivariate framework for graph-based *SSL* and describe an efficient algorithm that achieves it via alternating minimization.

3. Bivariate Framework for Graph-Based *SSL*

We first propose an extension to the existing graph regularization-based *SSL* formulations by casting the problem as a *bivariate* optimization over both the classification function and the unknown labels. Then we demonstrate that the minimization of this bivariate cost reduces to a linearly constrained binary integer programming (BIP) problem. This problem can be approximated via semi-definite programming yet this approach is impractical due to scalability issues. We instead explore a fast method which alternates minimization of the cost over the label matrix and the prediction function.

3.1 The Cost Function

Recall the univariate regularization formulation for graph-based *SSL* in Equation (3). Also note that the optimization problem in existing approaches such as *LGC* and *GFHF* can be broken up into separate parallel problems since the cost function decomposes into additive terms that only depend on individual columns of the prediction matrix $\mathbf{F}$ (Wang et al., 2008a). Such a decomposition reveals that biases may arise if the input labels are disproportionately imbalanced. In addition, when the graph contains background noise and makes class manifolds non-separable (as in Figure 1(b)), these existing graph transduction approaches fail to output reasonable classification results.

Since the univariate framework treats the initial label information as a constant, we propose a novel bivariate optimization framework that explicitly optimizes over both the classification function $\mathbf{F}$ and the binary label matrix $\mathbf{Y}$:

$$\begin{aligned}(\mathbf{F}^*,\mathbf{Y}^*) \quad &= \arg\min_{\mathbf{F}\in\mathbb{R}^{n\times c},\mathbf{Y}\in\mathbb{B}^{n\times c}} Q(\mathbf{F},\mathbf{Y}) \\ &s.t. \quad y_{ij}\in\{0,1\}, \\ &\quad \textstyle\sum_{j=1}^{c} y_{ij}=1, \\ &\quad y_{ij}=1, \ \text{for} \ \ z_i=j, \ \ j=1,\cdots,c.,\end{aligned}$$

where $\mathbb{B}^{n\times c}$ is the set of all binary matrices $\mathbf{Y}$ of size $n\times c$. For a labeled sample $\mathbf{x}_i\in\mathbf{X}_l$, $y_{ij}=1$ if $z_i=j$, and the constraint $\sum_{j=1}^{c} y_{ij}=1$ indicates that this a single label prediction problem. We specify the cost function as

$$Q(\mathbf{F},\mathbf{Y})=\frac{1}{2}\mathrm{tr}\left(\mathbf{F}^\top\mathbf{L}\mathbf{F}+\mu(\mathbf{F}-\mathbf{Y})^\top(\mathbf{F}-\mathbf{Y})\right). \tag{6}$$

Finally, rewriting the cost as a summation (Zhou et al., 2004) reveals a more intuitive formulation where

$$Q(\mathbf{F},\mathbf{Y}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left\|\frac{\mathbf{F}_{i\cdot}}{\sqrt{d_i}} - \frac{\mathbf{F}_{j\cdot}}{\sqrt{d_j}}\right\|^2 + \frac{\mu}{2}\sum_{i=1}^{n}\|\mathbf{F}_{i\cdot}-\mathbf{Y}_{i\cdot}\|^2.$$

3.2 Reduction to a Univariate Problem

In the new graph regularization framework proposed above, the cost function involves two variables to be optimized. Simultaneously recovering both solutions is intractable due to the mixed integer programming problem over binary $\mathbf{Y}$ and continuous $\mathbf{F}$. To solve the issue, we first show how to reduce the original mixed problem to a univariate optimization problem with respect to the label variable $\mathbf{Y}$.

$\mathbf{F}$ *optimization step:*
In each loop with $\mathbf{Y}$ fixed, the classification function $\mathbf{F} \in \mathbb{R}^{n\times c}$ is continuous and the cost function is convex, allowing the minimum to be recovered by setting the partial derivative to zero:

$$\begin{aligned}&\frac{\partial Q}{\partial \mathbf{F}^*} = 0 \Longrightarrow \mathbf{L}\mathbf{F}^* + \mu(\mathbf{F}^*-\mathbf{Y}) = 0\\ &\Longrightarrow \mathbf{F}^* = (\mathbf{L}/\mu+\mathbf{I})^{-1}\mathbf{Y} = \mathbf{P}\mathbf{Y},\end{aligned} \tag{7}$$

where we denote the $\mathbf{P}$ matrix as

$$\mathbf{P} = (\mathbf{L}/\mu+\mathbf{I})^{-1},$$

and name it the *propagation matrix* since it is used to derived a prediction function $\mathbf{F}$ given a label matrix $\mathbf{Y}$. Because the graph is often symmetric, it is easy to show that the graph Laplacian $\mathbf{L}$ and the propagation matrix $\mathbf{P}$ are both symmetric.

$\mathbf{Y}$ *optimization step:*
Next replace $\mathbf{F}$ in Equation (6) by its optimal value $\mathbf{F}^*$ from the solution of Equation (7). This yields

$$\begin{aligned}Q(\mathbf{Y}) &= \frac{1}{2}\mathrm{tr}(\mathbf{Y}^\top\mathbf{P}^\top\mathbf{L}\mathbf{P}\mathbf{Y} + \mu(\mathbf{P}Y-\mathbf{Y})^\top(\mathbf{P}\mathbf{Y}-\mathbf{Y}))\\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top\left[\mathbf{P}^\top\mathbf{L}\mathbf{P}+\mu(\mathbf{P}^\top-\mathbf{I})(\mathbf{P}-\mathbf{I})\right]\mathbf{Y}\right) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top\mathbf{A}\mathbf{Y}\right),\end{aligned}$$

where we group all the constant parts in the above equation and define

$$\mathbf{A} = \mathbf{P}^\top\mathbf{L}\mathbf{P}+\mu(\mathbf{P}^\top-\mathbf{I})(\mathbf{P}-\mathbf{I}) = \mathbf{P}^\top\mathbf{L}\mathbf{P}+\mu(\mathbf{P}-\mathbf{I})^2.$$

The final optimization problem becomes

$$\begin{aligned}&\mathbf{Y}^* = \arg\min\frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top\mathbf{A}\mathbf{Y}\right)\\ &s.t.\quad y_{ij}\in\{0,1\},\\ &\qquad \textstyle\sum_j y_{ij} = 1,\ \ j=1,\cdots,c\\ &\qquad y_{ij} = 1, \text{for}\ \ z_i = j,\ \ j=1,\cdots,c.\end{aligned} \tag{8}$$

The first constraint produces a binary integer problem and the second one $\sum_j y_{ij} = 1$ produces a single assignment constraint, that is, each vertex can only be assigned one class label. The third group of constraints encodes the initial label information in the variable $\mathbf{Y}$. Since the binary matrix $\mathbf{Y} \in \mathbb{B}^{n \times c}$ is subject to linear constraints of the form $\sum_j y_{ij} = 1$ and initial labeling conditions, the optimization in Equation (8) requires solving a linearly constrained binary integer programming (BIP) problem which is NP hard (Cook, 1971; Karp, 1972).

3.3 Incorporating Label Normalization

A straightforward approach to solving the minimization problem in Equation (8) is to use the gradient to greedily update the label variable $\mathbf{Y}$. However, this may produce biased classification results in practice since, at each iteration, the class with more labels will be preferred and will propagate more quickly to the unlabeled examples. This arises in practice (as in Figure 3) and is due to the fact that $\mathbf{Y}$ starts off sparse and contains many unknown entries. To compensate for this bias during label propagation, we propose using a normalized label variable $\tilde{\mathbf{Y}} = \mathbf{\Lambda} Y$ for computing the cost function in Equation (6) as

$$\begin{aligned} Q &= \frac{1}{2}\mathrm{tr}\left(\mathbf{F}^\top \mathbf{L}\mathbf{F} + \mu(\mathbf{F}-\tilde{\mathbf{Y}})^\top(\mathbf{F}-\tilde{\mathbf{Y}})\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{F}^\top \mathbf{L}\mathbf{F} + \mu(\mathbf{F}-\mathbf{\Lambda}\mathbf{Y})^\top(\mathbf{F}-\mathbf{\Lambda}\mathbf{Y})\right). \end{aligned} \tag{9}$$

The diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda}) = \mathrm{diag}([\lambda_1, \cdots, \lambda_n])$ is introduced to re-weight or re-balance the influence of labels from different classes as it modulates the label importance based on node degree. The value of λ_i $(i = 1, \cdots, n)$ is computed using the vertex degree d_i and label information

$$\lambda_i = \begin{cases} p_j \cdot \frac{d_i}{\sum_k y_{kj} d_k} & : \quad y_{ij} = 1 \\ 0 & : \quad \text{otherwise}, \end{cases} \tag{10}$$

where p_j is the prior of class j and is subject to the constraint $\sum_{j=1}^{c} p_j = 1$. The value of p_j can be either estimated from the labeled training set or simply set to be uniform $p_j = 1/c$ $(j = 1, \cdots, c)$ in agnostic situations (when no better prior is available or if the labeled data is plagued by biased sampling). Using the normalized label matrix $\tilde{\mathbf{Y}}$ in the bivariate formulation allows labeled nodes with high degrees to contribute more during the label propagation process. However, the total diffusion of each class is kept equal (for agnostic settings with no priors available) or proportional to the class prior (for the setting with prior information). Therefore, the influence of different classes is balanced even if the given class labels are imbalanced. If class proportion information is known, it can be integrated by scaling the diffusion with the appropriate prior. In other words, the label normalization attempts to enforce simple concentration inequalities which, in the *iid* case require the predicted label results to concentrate around the underlying class ratios (Huang and Jebara, 2010). This intuition is in line with prior work that uses class proportion information in transductive inference where class proportion is enforced as a hard constraint (Chapelle et al., 2007) or as a regularizer (Mann and McCallum, 2007).

3.4 Alternating Minimization Procedure

To solve the above refined problem, we proposed an alternating minimization algorithm (Wang et al., 2008b). Briefly, starting with Equation (9) and repeating the similar derivation as in Section 3.2, we

obtain the optimal solution $\mathbf{F}^*$ and the final cost function with respect to label variable $\mathbf{Y}$ as

$$\mathbf{F}^* = \mathbf{P}\tilde{\mathbf{Y}} = \mathbf{P}\mathbf{\Lambda}\mathbf{Y}, \tag{11}$$

$$Q = \frac{1}{2}\mathrm{tr}\left(\tilde{\mathbf{Y}}^\top \mathbf{A}\tilde{\mathbf{Y}}\right) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top \mathbf{\Lambda}\mathbf{A}\mathbf{\Lambda}\mathbf{Y}\right). \tag{12}$$

Instead of finding the global optimum $\mathbf{Y}^*$, we only take an incremental step in each iteration to modify a single entry in $\mathbf{Y}$. Namely in each iteration, we find the optimal position (i^*, j^*) in the matrix $\mathbf{Y}$ and change the binary value of $y_{i^* j*}$ from 0 to 1. To do this, we find the direction with the largest negative gradient guiding our choice of binary step on $\mathbf{Y}$. Specifically, we evaluate $\| \bigtriangledown Q_{\mathbf{Y}} \|$ and find the largest negative value to determine (i^*, j^*).

Note that the setting $y_{i^* j^*} = 1$ is equivalent to modifying the normalized label matrix $\tilde{\mathbf{Y}}$ by setting $\tilde{y}_{i^*, j^*} = \varepsilon_{i^*}, 0 < \varepsilon_{i^*} < 1$, and $\mathbf{Y}, \tilde{\mathbf{Y}}$ can be converted from each other componentwise. Thus, the greedy optimization of Q with respect to $\mathbf{Y}$ is equivalent to greedy minimization of Q with respect to $\tilde{\mathbf{Y}}$. More formally, we derive the gradient of the above loss function $\nabla_{\tilde{\mathbf{Y}}} Q = \frac{\partial Q}{\partial \tilde{\mathbf{Y}}}$ and recover it with respect to $\mathbf{Y}$ as:

$$\frac{\partial Q}{\partial \tilde{\mathbf{Y}}} = \mathbf{A}\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{\Lambda}\mathbf{Y}. \tag{13}$$

As described earlier, we search the gradient matrix $\nabla_{\tilde{\mathbf{Y}}} Q$ to find the minimal element

$$(i^*, j^*) = \arg\min_{\mathbf{x}_i \in \mathcal{X}_u, 1 \le j \le c} \nabla_{\tilde{y}_{ij}} Q.$$

Because of the binary nature of $\mathbf{Y}$, we simply set $y_{i^* j^*} = 1$ instead of making a continuous update. Accordingly, the node weight matrix $\mathbf{\Lambda}^{t+1}$ can be recalculated with the updated $\mathbf{Y}^{t+1}$ in the iteration $t+1$. The update of $\mathbf{Y}$ is greedy and thus it could backtrack from predicted labels in previous iterations without convergence guarantees. We propose a straightforward way to guarantee convergence and avoid backtracking or unstable oscillation in the greedy propagation process: once an unlabeled point has been labeled, its labeling can no longer be changed. Thus, we remove the most recently labeled point (i^*, j^*) from future consideration and conduct search over the remaining unlabeled data only. In other words, to avoid retroactively changing predicted labels, the labeled vertex $\mathbf{x}_{i^*}$ is removed from $\mathbf{X}_u$ and added to $\mathbf{X}_l$.

Note that although the optimal $\mathbf{F}^*$ can be computed using Equation (11), this need not be done explicitly. Instead, the new value is implicitly used in Equation (14) only to update $\mathbf{Y}$. In the following, we summarize the update rules from step t to $t+1$ in the alternating minimization scheme.

. Compute gradient matrix:

$$(\nabla_{\tilde{\mathbf{Y}}} Q)^t = \mathbf{A}\tilde{\mathbf{Y}}^t = \mathbf{A}\mathbf{\Lambda}^t\mathbf{Y}^t, \mathbf{\Lambda}^t = \mathrm{diag}(\boldsymbol{\lambda}^t).$$

. Update one label:

$$\begin{aligned} (i^*, j^*) &= \arg\min_{\mathbf{x}_i \in \mathbf{X}_u, 1 \le j \le c} (\nabla_{\tilde{y}_{ij}} Q)^t, \\ y_{i^* j^*}^{t+1} &= 1. \end{aligned}$$

. Update label normalization matrix:

$$\lambda^{t+1} \quad = \quad \begin{cases} \frac{\mathbf{D}_{ii}}{\sum_k \mathbf{Y}_{kj}^{t+1}\mathbf{D}_{kk}} & : \quad y_{ij}^{t+1} = 1 \\ 0 & : \quad \text{otherwise.} \end{cases}$$

. Update the list of labeled and unlabeled data:

$$\mathbf{X}_l^{t+1} \longleftarrow \mathbf{X}_l^t + \mathbf{x}_{i^*} \;\; ; \;\; \mathbf{X}_u^{t+1} \longleftarrow \mathbf{X}_u^t - \mathbf{x}_{i^*}.$$

Starting with a few given labels, the method iteratively and greedily updates the label matrix $\mathbf{Y}$ to derive new labels in each iteration. The newly obtained labels are then use in the next iteration. Notice that the label normalization vector is re-computed for each iteration due to the change of label set. Although the original objective is formed in a bivariate manner in Equation 6, the above alternating optimization procedure drives the prediction of new labels without explicitly calculating $\mathbf{F}^*$ as is done in other graph transduction methods like *LGC* and *GFHF*. This unique feature makes the proposed algorithm very efficient since we only update the gradient matrix $\nabla_{\tilde{\mathbf{Y}}} Q$ for prediction new labels in each iteration.

Due to the greedy assignment step and the lack of back-tracking, the algorithm can repeat the alternating minimization (or the gradient computation) at most $n-l$ times. Each minimization step over $\mathbf{F}$ and $\mathbf{Y}$ requires $O(n^2)$ complexity and, thus, the total complexity of the above greedy algorithm is $O(n^3)$. However, the update of the graph gradient can be done efficiently by modifying only a single entry in $\mathbf{Y}$ per iteration. This further reduces the computational cost down to $O(n^2)$. Empirically, the value of the loss function Q decreases rapidly in the the first dozen iterations and steadily converges afterward (Wang et al., 2009). This phenomenon indicates that early stopping strategy could be applied to speed up the training and prediction (Melacci and Belkin, 2011). Once the first few iterations are completed, the new labels are added and the standard propagation step can be used to predict the optimal $\mathbf{F}^*$ as indicated in Equation (11) over the whole graph in one step. The details of the algorithm, namely *graph transduction via alternating minimization*, can be referred to Wang et al. (2008b). In the following section, we provide a greedy Max-Cut based solution, which essentially interprets the above alternating minimization procedure from a graph cut view.

4. Greedy Max-Cut for Semi-Supervised Learning

In this section, we introduce a connection between the proposed bivariate graph transduction framework and the well-known maximum cut problem. Then, a greedy gradient based Max-Cut solution will be developed and related to the above alternating minimization algorithm.

4.1 Equivalence to a Constrained Max-Cut Problem

Recall the optimization problem defined in Equation (8) which is exactly a linearly constrained binary integer programming (BIP) problem. In the case of a two-class problem, this optimization will be reduced to a weighted Max-Cut problem over the graph $\mathcal{G}_{\mathbf{A}} = \{\mathbf{X}, \mathbf{A}\}$ subject to linear constraints. The cost function in Equation (8) can be rewritten as

$$Q(\mathbf{Y}) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top \mathbf{A}\mathbf{Y}\right) = \frac{1}{2}\mathrm{tr}\left(\mathbf{A}\mathbf{Y}\mathbf{Y}^\top\right) = \frac{1}{2}\mathrm{tr}\left(\mathbf{A}\mathbf{R}\right),$$

where $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{R} = \mathbf{Y}\mathbf{Y}^\top$. Considering the constraints $\sum_j y_{ij} = 1$ and $\mathbf{Y} \in \mathbb{B}^{n\times 2}$ for a two-class problem, we let

$$\mathbf{Y} = [\mathbf{y} \;\; \mathbf{e}-\mathbf{y}],$$

where $\mathbf{y} \in \mathbb{B}^n$ (i.e., $\mathbf{y} = \{y_i\}, y_i \in \{0,1\}, i = 1,\cdots,n$) and $\mathbf{e} = [1,1,\cdots,1]^\top$ are column vectors. Then rewrite $\mathbf{R}$ as

$$\begin{aligned} \mathbf{R} &= \mathbf{Y}\mathbf{Y}^\top = [\mathbf{y} \;\; \mathbf{e}-\mathbf{y}][\mathbf{y} \;\; \mathbf{e}-\mathbf{y}]^\top \\ &= \mathbf{e}\mathbf{e}^\top - \mathbf{y}(\mathbf{e}^\top - \mathbf{y}^\top) - (\mathbf{e}-\mathbf{y})\mathbf{y}^\top. \end{aligned} \tag{14}$$

Now rewrite the cost function in Equation (14) by replacing $\mathbf{R}$ with Equation (14)

$$Q(\mathbf{y}) = \frac{1}{2}\mathrm{tr}\left(\mathbf{A}\left[\mathbf{e}\mathbf{e}^\top - \mathbf{y}(\mathbf{e}^\top - \mathbf{y}^\top) - (\mathbf{e}-\mathbf{y})\mathbf{y}^\top\right]\right).$$

Since $\mathbf{e}\mathbf{e}^\top$ is the all-ones matrix, we obtain

$$\frac{1}{2}\mathrm{tr}\left(\mathbf{A}\mathbf{e}\mathbf{e}^\top\right) = \frac{1}{2}\sum_i\sum_j \mathbf{A}_{ij}.$$

It is easy to show that $\mathbf{A}$ is symmetric and

$$\mathbf{y}(\mathbf{e}^\top - \mathbf{y}^\top) = [(\mathbf{e}-\mathbf{y})\mathbf{y}^\top]^\top.$$

Next, simplify the cost function Q as

$$\begin{aligned} Q(\mathbf{y}) &= \frac{1}{2}\mathrm{tr}\left(\mathbf{A}\mathbf{e}\mathbf{e}^\top\right) - \mathrm{tr}\left[(\mathbf{e}^\top - \mathbf{y}^\top)\mathbf{A}\mathbf{y}\right] \\ &= \frac{1}{2}\mathrm{tr}\left(\mathbf{A}\mathbf{e}\mathbf{e}^\top\right) - \mathbf{y}^\top\mathbf{A}(\mathbf{e}-\mathbf{y}). \end{aligned}$$

Since the first part is a constant, the optimal value $\mathbf{y}^*$ of the above minimization problem is the argument of the maximization problem

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} Q(\mathbf{y}) = \arg\max_{\mathbf{y}} \mathbf{y}^\top\mathbf{A}(\mathbf{e}-\mathbf{y}).$$

Define a new function $f(\mathbf{y})$ as

$$f(\mathbf{y}) = \mathbf{y}^\top\mathbf{A}(\mathbf{e}-\mathbf{y}).$$

Again, the variable $\mathbf{y} \in \mathbb{B}^n$ is a binary vector and $\mathbf{e} = [1,1,\cdots,1]^\top$ is the unit column vector. Now we show that maximization of the above function $\max_{\mathbf{y}} f(\mathbf{y})$ is exactly a Max-Cut problem if we treat the symmetric matrix $\mathbf{A}$ as the weighted adjacency matrix of an undirected graph $\mathcal{G}_\mathbf{A} = \{V_\mathbf{A}, \mathbf{A}\}$. Note that the diagonal elements of $\mathbf{A}$ could be non-zero $\mathbf{A}_{ii} \neq 0, i = 1,2,\cdots,n$, which indicates the undirected graph $\mathcal{G}_\mathbf{A}$ has self-connected nodes. Assume $\mathbf{A} = \mathbf{A}^0 + \mathbf{A}^\Lambda$, where $\mathbf{A}^0$ is the matrix obtained by zeroing the diagonal elements of $\mathbf{A}$ and $\mathbf{A}^\Lambda$ is a diagonal matrix with $\mathbf{A}^\Lambda_{ii} = \mathbf{A}_{ii}, \mathbf{A}^\Lambda_{ij} = 0, i,j = 1,2,\cdots,n, i \neq j$. It is straightforward to show that the the function $f(\mathbf{y})$ can be written as

$$f(\mathbf{y}) = \mathbf{y}^\top(\mathbf{A}^0 + \mathbf{A}^\Lambda)(\mathbf{e}-\mathbf{y}) = \mathbf{y}^\top\mathbf{A}^0(\mathbf{e}-\mathbf{y}).$$

In other words, the non-zero elements in $\mathbf{A}$ do not affect the value of $f(\mathbf{y})$. Therefore, in the rest of this article, we can assume that the matrix $\mathbf{A}$ has zero diagonal elements unless the text specifies otherwise.

Since $\mathbf{y} = \{y_1, y_2, \cdots, y_n\}$ is a binary vector, each setting of $\mathbf{y}$ partitions the vertex set $V_\mathbf{A}$ in the graph $\mathcal{G}_\mathbf{A}$ into two disjoint subsets (S_1, S_2). In other words, the two subsets $S_1 = \{v_i | y_i = 1\}$ and $S_2 = \{v_i | y_i = 0\}$ satisfy $S_1 \cup S_2 = V_\mathbf{A}$ and $S_1 \cap S_2 = \emptyset$. The maximization problem can then be written as

$$\max f(\mathbf{y}) = \max \sum_{i,j} a_{ij} \cdot y_i (1 - y_j) = \max \frac{1}{2} \sum_{\substack{v_i \in S_1 \\ v_j \in S_2}} a_{ij}.$$

Because each binary vector $\mathbf{y}$ resulting in a partition (S_1, S_2) over the graph $\mathcal{G}_\mathbf{A}$ and $f(\mathbf{y})$ is a corresponding *cut*, the above maximization $\max f(\mathbf{y})$ is easily recognized as a Max-Cut problem (Deza and Laurent, 2009). However, in graph based semi-supervised learning, the variable $\mathbf{y}$ is partially specified by the initial label values. This given label information can be interpreted as a set of linear constraints on the Max-Cut problem. Thus, the optimal solution can achieved by solving a linearly constrained Max-Cut problem (Karp, 1972). In addition, we also show that a multi-class problem equals a Max K-Cut problem ($K = c$) (refer to Appendix A). Note that previous work used min-cut over the original data graph $\mathcal{G} = \{\mathbf{X}, \mathbf{W}\}$ to perform semi-supervised learning (Blum and Chawla, 2001; Blum et al., 2004). A key difference of the above formulation lies in the fact that we perform max-cut over the transformed graph $\mathcal{G}_\mathbf{A} = \{\mathbf{X}, \mathbf{A}\}$.

However, since there is no guarantee that the weights on the graph $\mathcal{G}_\mathbf{A}$ are non-negative, solutions to the Max-Cut problem can be difficult to find (Barahona et al., 1988). Therefore, in the following subsection, we will propose a gradient greedy solution to efficiently solve the above Max-Cut problem, which can be treated as a different view of the previous alternating minimization solution.

4.2 Label Propagation by Gradient Greedy Max-Cut

For the standard Max-Cut problem, many approximation techniques have been developed, including the most remarkable Goemans-Williamson algorithm using semidefinite programming (Goemans and Williamson, 1994, 1995). However, applying these guaranteed approximation schemes to solve the constrained Max-Cut problem for $\mathbf{Y}$ mentioned above is infeasible due to the constraints on initial labels. Furthermore, there is no guarantee that all edge weights a_{ij} of the graph $\mathcal{G}_\mathbf{A}$ are non-negative, a fundamental requirement in solving a standard Max-Cut problem (Goemans and Williamson, 1995). Instead, here we use a greedy gradient based strategy to find local optima by assigning each unlabeled vertex to the label set with minimum connectivity to maximize cross-set edge weights iteratively.

The greedy Max-Cut algorithm randomly selects unlabeled vertices and places each of them into the appropriate class subset depending on the edges between this unlabeled vertex and the vertices in the labeled subset. Given the label information, the initial label set for class j can be constructed as $\mathcal{S}_j = \{\mathbf{x}_i | y_{ij} = 1\}$ or $\mathcal{S}_j = \{\mathbf{x}_i | z_i = j\}$, $i = 1, 2, \cdots, n; j = 1, 2, \cdots, c$. Define the following as the connectivity between unlabeled vertex $\mathbf{x}_i$ and labeled subset $\mathcal{S}_j$

$$c_{ij} = \sum_{m=1}^{n} a_{im} y_{mj} = \mathbf{A}_{i.} \mathbf{Y}_{.j}, \tag{15}$$

Algorithm 1 Greedy Max-Cut for Label Propagation

Input: the graph $\mathcal{G}_{\mathbf{A}} = \{\mathbf{X}, \mathbf{A}\}$, the given labeled vertex $\mathbf{X}_l$, and initial labels $\mathbf{Y}$;
Initialization:
obtain the initial cut $\{\mathcal{S}_j\}$ by assigning the labeled vertex $\mathbf{X}_l$ to each subset:

$$\mathcal{S}_j = \{x_i | y_{ij} = 1\}, j = 1, 2, \cdots, c$$

unlabeled vertex set $\mathbf{X}_u = \mathbf{X} \setminus \mathbf{X}_l$;
repeat
randomly select an unlabeled vertex $\mathbf{x}_i \in \mathbf{X}_u$
compute the connectivity $c_{ij}, j = 1, 2, \cdots, c$
place the vertex to the labeled subject $\mathcal{S}_{j^*}$:

$$j^* = \arg\min_j c_{ij}$$

add $\mathbf{x}_i$ to $\mathbf{X}_l$: $\mathbf{X}_l \longleftarrow \mathbf{X}_l + \mathbf{x}_i$;
remove $\mathbf{x}_i$ from $\mathbf{X}_u$: $\mathbf{X}_u \longleftarrow \mathbf{X}_u - \mathbf{x}_i$;
until $\mathbf{X}_u = \emptyset$
Output: the final cut and the corresponding labeled subsets $\mathcal{S}_j, j = 1, 2, \cdots, c$

where $\mathbf{A}_{i\cdot}$ is the i'th row vector of $\mathbf{A}$ and $\mathbf{Y}_{\cdot j}$ is the j'th column vector of $\mathbf{Y}$. Intuitively, c_{ij} represents the sum of edge weights between vertex $\mathbf{x}_i$ and label set $\mathcal{S}_j$ given the graph $\mathcal{G}_{\mathbf{A}}$ with edge weights $\mathbf{A}$. Based on this definition, a straightforward local search for the maximum cut involves placing each unlabeled vertex $\mathbf{x}_i \in \mathbf{X}_u$ in the labeled subset $\mathcal{S}_j$ with minimum connectivity c_{ij} to maximize the cross-set edge weights as shown in Algorithm (1). In order to achieve a good solution Algorithm (1) should be run multiple times with different random seeds after which the best cut overall is output (Mathieu and Schudy, 2008).

While the above method is computationally cumbersome, it still does not resolve the issue of undesired local optima and may generate biased cuts. According to the definition in Equation (15), the initialized labels determine the connectivity between unlabeled vertices and labeled subsets. If the computed connectivity is negative, the above random search will prefer assigning unlabeled vertices to the label set with the most labeled vertices which results in biased partitioning. Such biased partitioning also occurs in minimum cut problems over an undirected graph with positive weights (Shi and Malik, 2000). Other label initialization problems may also produce a poor cut. For example, the numbers of labels from different classes may deviate from the underlying class proportions. Alternatively, the labeled vertices may be outliers and lie in regions of the graph with low density. Such labels often lead to weak label prediction results (Wang et al., 2008a). Furthermore, the algorithm's random selection of an unlabeled vertex results in unstable predictions since the chosen unlabeled vertex $\mathbf{x}_i$ could have equally low connectivity to multiple label subsets $\mathcal{S}_j$.

To address the aforementioned issues, we first modify the original definition of connectivity to alleviate label imbalance across different classes. A weighted connectivity is computed as

$$c_{ij} = p_j \cdot \sum_{m=1}^{n} \lambda_m a_{im} y_{mj} = p_j \cdot \mathbf{\Lambda} \mathbf{A}_{i\cdot} \mathbf{Y}_{\cdot j}. \tag{16}$$

The diagonal matrix $\mathbf{\Lambda} = \text{diag}([\lambda_1, \lambda_2, \cdots, \lambda_n]$ is called the label weight matrix as in Equation (10)

$$\lambda_i = \begin{cases} d_i / d_{\mathcal{S}_j} & : \text{ if } \mathbf{x}_i \in \mathcal{S}_j, j = 1, \cdots, c \\ 0 & : \text{ otherwise,} \end{cases}$$

Algorithm 2 Greedy Gradient based Max-Cut for Label Propagation

Input: the graph $\mathcal{G}_{\mathbf{A}} = \{\mathbf{X}, \mathbf{A}\}$ and the given labeled vertex $\mathbf{X}_l$, and initial label $\mathbf{Y}$;
Initialization:
obtain the initial cut $\{\mathcal{S}_j\}$ through assigning the labeled vertex $\mathbf{X}_l$ to each subset:

$$\mathcal{S}_j = \{x_i | y_{ij} = 1\}, j = 1, 2, \cdots, c$$

unlabeled vertex set $\mathbf{X}_u = \mathbf{X} \setminus \mathbf{X}_l$;
repeat
  for all $j = 0$ to $|\mathbf{X}_u|$ **do**
    compute weighted connectivity:

$$c_{ij} = \sum_{k=1}^{n} \lambda_i a_{ik} y_{kj}, \mathbf{x}_i \in \mathbf{X}_u, j = 1, \cdots, c$$

  end for
  update the cut $\{\mathcal{S}_i\}$ by placing the vertex $\mathbf{x}_{i^*}$ to the $\mathcal{S}_{j^*}$'th subset:

$$(i^*, j^*) = \arg \min_{i,j,\mathbf{x}_i \in \mathbf{X}_u} c_{ij}$$

  add $\mathbf{x}_i$ to $\mathbf{X}_l$: $\mathbf{X}_l \longleftarrow \mathbf{X}_l + \mathbf{x}_i$;
  remove $\mathbf{x}_i$ from $\mathbf{X}_u$: $\mathbf{X}_u \longleftarrow \mathbf{X}_u - \mathbf{x}_i$;
until $\mathbf{X}_u = \emptyset$
Output: the final cut and the corresponding labeled subsets $\mathcal{S}_j, j = 1, 2, \cdots, c$

where $d_{\mathcal{S}_j} = \sum_{\mathbf{x}_m \in \mathcal{S}_j} d_m$ is the sum of the degrees of the vertices in the label set $\mathcal{S}_j$. This heuristic setting weights the importance of each label based on degree which alleviates the adverse impact of outliers. If we ignore class priors, this definition of $\mathbf{\Lambda}$ coincides with the one in Equation (10).

Finally, to handle any instability due to the random search algorithm, we propose a greedy gradient search approach where the most beneficial vertex is assigned to the label set with minimum connectivity. In other words, we first compute the connectivity matrix $\mathbf{C} = \{c_{ij}\} \in \mathbb{R}^{n \times c}$ that gives the connectivity between all unlabeled vertices to existing label sets

$$\mathbf{C} = \mathbf{A}\mathbf{\Lambda}\mathbf{Y}.$$

Then we examine $\mathbf{C}$ to identify the element (i^*, j^*) with minimum value as

$$(i^*, j^*) = \arg \min_{i,j:\mathbf{x}_i \in \mathbf{X}_u} c_{ij}.$$

This means that the unlabeled vertex $\mathbf{x}_{i^*}$ has the least connectivity with label set $\mathcal{S}_{j^*}$. Then, we update the labeled set $\mathcal{S}_{j^*}$ by adding vertex $\mathbf{x}_{i^*}$ as one greedy step to maximize the cross-set edge weights. This greedy search can be repeated until all the unlabeled vertices are assigned to labeled sets. In each iteration of the greedy cut process, the weighted connectivity of all unlabeled vertices to labeled sets is re-computed. Then the vertex with minimum connectivity is placed in the proper labeled set. The algorithm is summarized in Algorithm (2).

The connectivity matrix $\mathbf{C}$ can also be viewed as the gradient of the cost function Q in Equation (12) with respect to $\tilde{\mathbf{Y}}$. This is precisely the same setting used in Equation (13) of the alternating minimization algorithm

$$\mathbf{C} = \frac{\partial Q}{\partial \tilde{\mathbf{Y}}} = \mathbf{A}\mathbf{\Lambda}\mathbf{Y}.$$

We name the algorithm *greedy gradient Max-Cut* (*GGMC*) since, in the greedy step, the unlabeled vertices are assigned labels in a manner that reduces the value of Q along the direction of the steepest descent. Consider both the variables $\mathbf{Y}$ and $\mathbf{F}$ in the original bivariate formulation in Equation (9). The greedy Max-Cut method is equivalent to the alternating minimization procedure discussed earlier. Unlike graph-cut based *SSL* methods such as mincuts (Blum and Chawla, 2001; Blum et al., 2004), our *GGMC* algorithm tends to generate more natural graph cuts and avoid biased solutions since it uses a weighted connectivity matrix. This allows it to effectively handle the issues mentioned earlier and, in practice, achieve significant gains in accuracy while retaining efficiency.

4.3 Complexity and Speed Up

Assume the graph has $n = |\mathbf{X}|$ vertices and a subset $\mathbf{X}_l$ with $l = |\mathbf{X}_l|$ labeled vertices (where $l \ll n$). The greedy gradient algorithm terminates after at most $n - l \simeq n$ iterations. In each iteration of the greedy gradient algorithm, the connectivity matrix $\mathbf{C}$ is updated by a matrix multiplication (an $n \times n$-matrix is multiplied by a $n \times c$-matrix). Hence, the complexity of the greedy algorithm is $O(cn^3)$.

However, the greedy algorithm can be greatly accelerated in practice. For example, the computation of the connectivity in Equation (16) can be done incrementally after assigning each new unlabeled vertex to a certain label set. This circumvents the re-calculation of all the entries in the $\mathbf{C}$ matrix. Assume in the t'th iteration the connectivity is $\mathbf{C}^t$ and an unlabeled vertex $\mathbf{x}_i$ with degree d_i is assigned to the labeled set $\mathcal{S}_j$. Clearly, for all remaining unlabeled vertices, the connectivity to the labeled sets remains unchanged except for the j'th labeled set. In other words, only the j'th column of $\mathbf{C}$ needs updating. This update is performed incrementally via

$$\mathbf{C}_{.j}^{t+1} = \frac{d_{\mathcal{S}_j^t}}{d_{\mathcal{S}_j^{t+1}}} \mathbf{C}_{.j}^t + \frac{d_i}{d_{\mathcal{S}_j^{t+1}}} \mathbf{A}_{.i},$$

where $d_{\mathcal{S}_j^{t+1}} = d_{\mathcal{S}_j^t} + d_i$ is the sum of the degrees of the labeled vertices after assigning $\mathbf{x}_i$ to the labeled set $\mathcal{S}_j$. This incremental update reduces the complexity of the greedy gradient search algorithm to $O(n^2)$.

5. Experiments

In this section, we demonstrate the superiority of the proposed *GGMC* method over state-of-the-art semi-supervised learning methods using both synthetic and real data. Previous work showed that *LapSVM* and *LapRLS* outperform other semi-supervised approaches such as Transductive SVMs *TSVM* (Joachims, 1999) and ∇*TSVM* (Chapelle and Zien, 2005). Therefore, we limit our comparisons to only the *LapRLS*, *LapSVM* (Sindhwani et al., 2005; Belkin et al., 2005), *LGC* (Zhou et al., 2004) and *GFHF* (Zhu et al., 2003). To set various hyper-parameters such as γ_I, γ_r in *LapRLS* and *LapSVM*, we followed the default configurations used in the literature. Similarly, for *GGMC* and *LGC*, we set the hyper-parameter $\mu = 0.01$ across all data sets. For the computational cost, *GGMC*, *LGC* and *GFHF* required very similar run-times to output a prediction. However, *LapRLS* and *LapSVM* need significant longer training time, especially for multiple-class problems since multiple rounds of one-vs-all training have to be performed.

As in Section 2, any real implementation of graph-based *SSL* needs a graph construction method algorithm that builds a graph from the training data $\mathbf{X}$. This is then followed by a sparsification

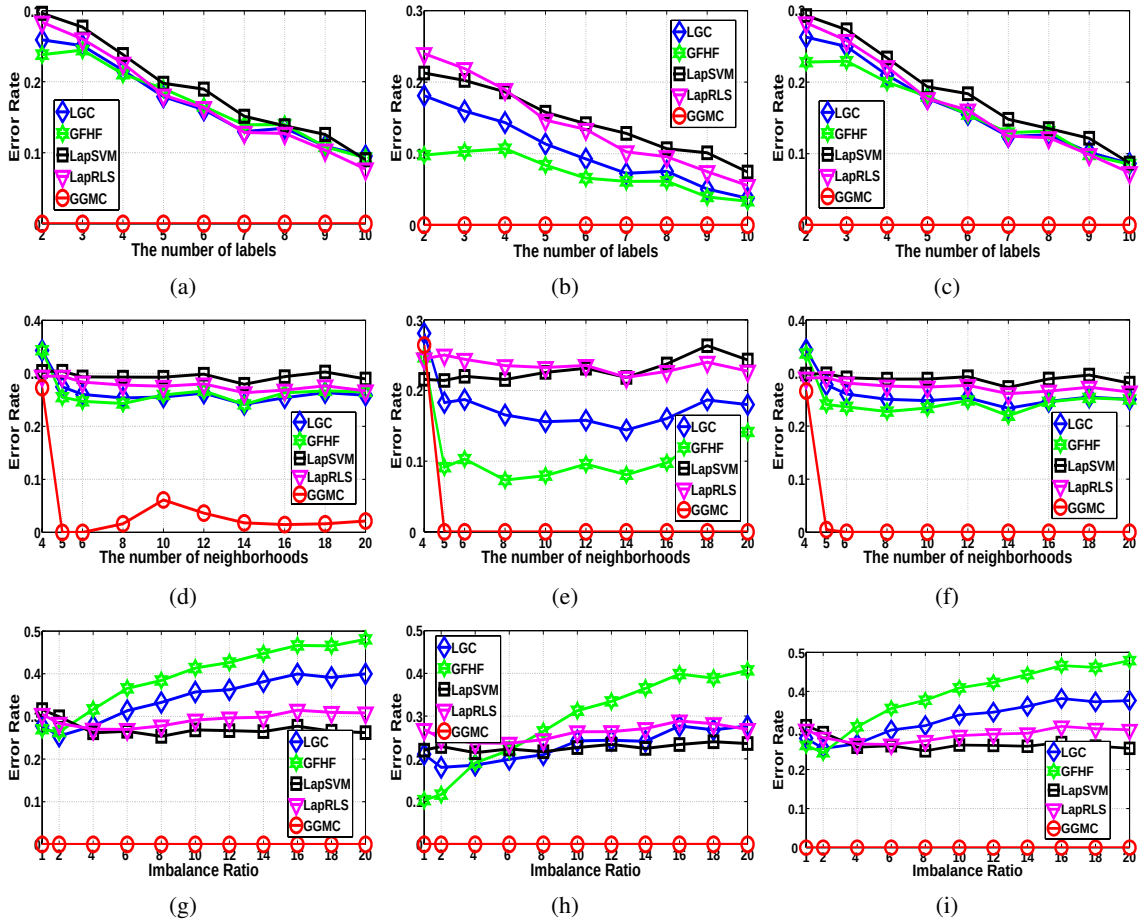
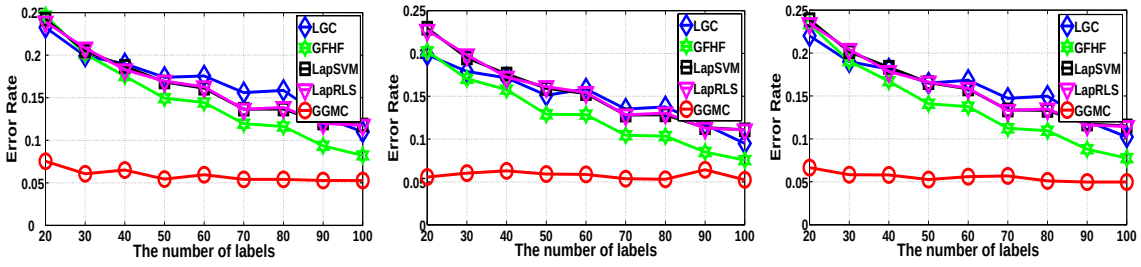


Figure 4: Experimental results on the noisy two-moon data set simulating different graph construction approaches and label conditions. Figures a) d) g) use binary weighting. Figures b) e) h) use fixed Gaussian kernel weighting. Figures c) f) i) use adaptive Gaussian kernel weighting. Figures a) b) c) vary the number of labels. Figures d) e) f) vary the value of k in the graph construction. Figures g) h) i) vary the label imbalance ratio.

procedure to generate the sparse connectivity matrix $\mathbf{B}$ and a weighting procedure to obtain weights on the edges in $\mathbf{B}$. In these experiments, we used the same graph construction procedure for all the *SSL* algorithms. The sparsification was done using the standard k-nearest-neighbors approach and the edge weighting involved either *binary weighting* or *Gaussian kernel weighting*. In the latter case, the ℓ_2 distance $d_{\ell_2}(\mathbf{x}_i, \mathbf{x}_j)$ is used and the kernel bandwidth σ is estimated in two different ways. The first estimate uses a fixed σ defined as the average distance between each selected sample and its k'th nearest neighbor (Chapelle et al., 2006). In addition, a second adaptive approach is also considered which locally estimates the parameter σ to the mean distance in the k-nearest neighborhoods of the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ (Wang et al., 2008a).



(a) (b) (c)

Figure 5: Experimental results on the USPS digits data set under varying levels of labeling using: a) binary weighting; b) fixed Gaussian kernel weighting and c) adaptive Gaussian kernel weighting.

5.1 Noisy Two-Moon Data Set

We first compared *GGMC* with several representative *SSL* algorithms using the noisy two-moon data set shown in Figure 3. Despite the near-perfect classification results reported on clean versions of this data set (Sindhwani et al., 2005; Zhou et al., 2004), small amounts of noise quickly degrade the performance of previous algorithms. In Figure 3, two separable manifolds containing 600 two-dimensional points are mixed with 100 noisy outlier samples. The noise foils previous methods which are sensitive to the locations of the initial labels, disproportional sampling from the classes, and outlier noise. All experiments are repeated with 100 independent folds with random sampling to show the average error rate of each algorithm.

The first group of experiments varies the number of labels provided to the algorithms. We uniformly used $k = 6$ in the k-nearest-neighbors graph construction and applied the aforementioned three edge-weighting schemes. The average error rates of the predictions on the unlabeled points is shown in Figures 4(a), 4(b) and 4(c). These correspond to binary edge weighting, fixed Gaussian kernel edge weighting, and adaptive Gaussian kernel edge weighting, respectively. The results clearly show that *GGMC* is robust to the size of the label set and and generates perfect prediction results for all three edge-weighting schemes.

The second group of experiments demonstrate the influence of the number of edges (i.e., the value of k) in the graph construction method. We varied the value of k from 4 to 20 and Figures 4(d), 4(e), and 4(f) show results for the different edge-weighting schemes. Once again, *GGMC* achieves significantly better performance in most cases.

Finally, we studied the effect of imbalanced labeling on the *SSL* algorithms. We fix one class to have only one label and then randomly select r labels from the other classes. Here, r indicates the imbalance ratio and we study the range $1 \leq r \leq 20$. Figures 4(g), 4(h), and 4(i) show the results with different edge-weighting schemes. Clearly, *GGMC* is insensitive to the imbalance since it computes a per-class label weight normalization which compensates directly for differences in label proportions.

In summary, Figure 4 depicted the performance advantages of *GGMC* relative to *LGC*, *GFHF*, *LapRLS*, and *LapSVM* methods. We clearly see that the four previous algorithms are sensitive to the initial labeling conditions and none of them produces perfect prediction. Furthermore, the error rates of *LGC* and *GFHF* increase significantly when labeling becomes imbalanced, even if many

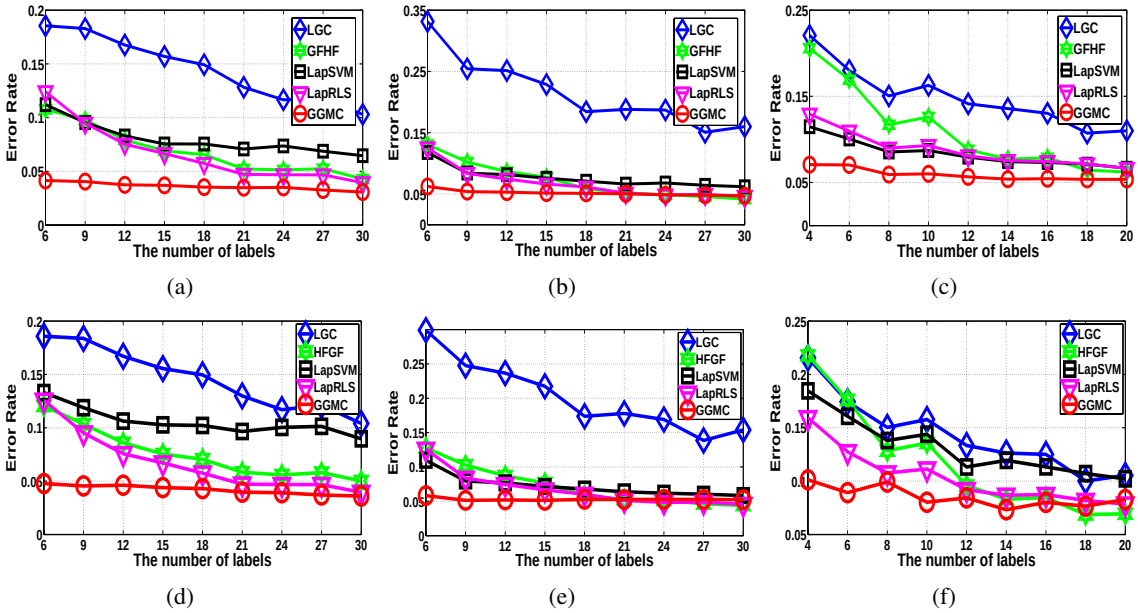


Figure 6: Performance of *LGC*, *GFHF*, *LapRLS*, *LapSVM*, and *GGMC* algorithms using the UCI data sets. The horizontal axis is the number of training labels provided while the vertical axis is the average error rate achieved over 100 random folds. Results are based on k-nearest neighbor graphs shown in a) for the Iris data set, in b) for the Wine data set and in c) for the Breast Cancer data set. Results are based on b-matched graphs shown in d) for the Iris data set, in e) for the Wine data set and in f) for the Breast Cancer data set.

labels are made available. However, *GGMC* achieves high accuracy regardless of the imbalance ratio and the size of the label set. Furthermore, *GGMC* remains robust to the graph construction procedure and the edge-weighting strategy.

5.2 Handwritten Digit Data Set

We also evaluated the algorithms in an image recognition task where handwritten digits in the USPS database are to be annotated with the appropriate label $\{0,1,\ldots,9\}$. The data set contains gray scale handwritten digit images involving 16×16 pixels. We randomly sampled a subset of 4000 samples from the data. For all the constructed graphs, we used the k-nearest-neighbors algorithm with $k=6$ and tried the three different edge-weighting schemes above. We varied the total number of labels from 20 to 100 while guaranteeing that each digit class had at least one label. For each setting, the average error rate was computed over 20 random folds.

The experimental results are shown in Figures 5(a), 5(b), and 5(c) which correspond to the three different edge-weighting schemes. As usual, *GGMC* significantly improves classification accuracy relative to other approaches, especially when few labeled training examples are available. The average error rates of *GGMC* are consistently low with small standard deviations. This demonstrates that the *GGMC* method is less sensitive to the number and locations of the initial training labels.

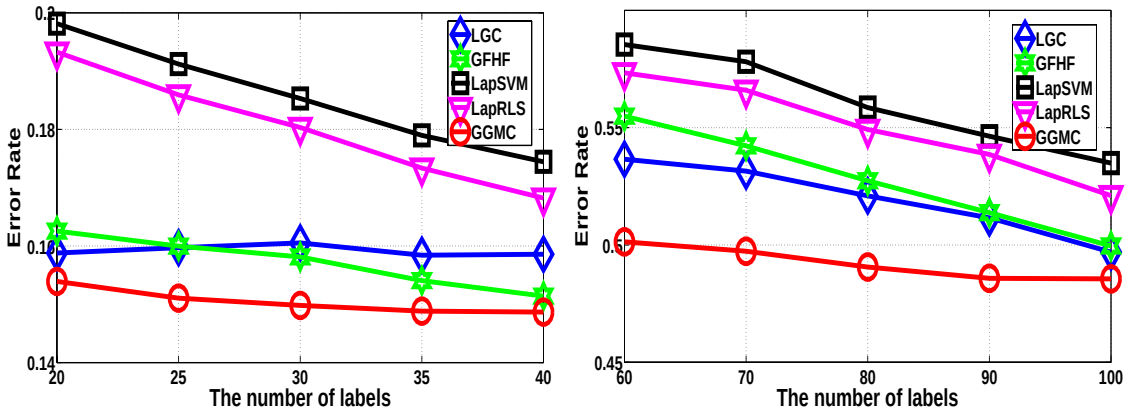


Figure 7: Performance of *LGC*, *GFHF*, *LapRLS*, *LapSVM*, and *GGMC* algorithms using the COIL-20 and Animal data sets. The horizontal axis is the number of training labels provided while the vertical axis is the average error rate. Results are shown in a) for the COIL-210 object data set, and in b) for the Animal data set.

5.3 UCI Data Sets

We tested *GGMC* and the other algorithms on benchmark data sets from the UCI Machine Learning Repository (Frank and Asuncion, 2010). Specifically, we used the Iris, Wine, and Breast Cancer data sets. The numerical attributes of the data sets are all normalized to span the range $[0,1]$. For all three data sets, we used a k-nearest-neighbors graph construction procedure with $k = 6$ and explored the Gaussian kernel with fixed bandwidth as the edge-weighting scheme, where the bandwidth is set as the average distance between each selected sample and its k'th nearest neighbor.

Figure 6 shows the performance of the various *SSL* algorithms. The vertical axis is the average error rate computed over 100 random folds and the horizontal axis shows the number of labeled samples provided at training time. Besides using the k-nearest neighbor graphs, we also evaluated the perform using the b-matched graphs on this data set. The *GGMC* method significantly outperforms other algorithms in most test cases, especially when little labeled data is available.

5.4 COIL-20 Object Images

We investigated the object recognition problem using the well-known Columbia Object Image Library (COIL-20), which contains 1440 gray-scale images of 20 objects (Nene et al., 1996). The images sequences were obtained when the objects were placed on a turntable table with black background, where one image was taken for each 5-degree interval. As with UCI data sets, we constructed kNN graphs with $k = 6$ and used a fixed bandwidth for edge weighting. The number of given labels from all object categories was varied from 20 to 40 with the guarantee that each object class has at least one label. Figure 7(a) shows the performance curves in terms of the average error rate of 100 random tests, where *GGMC* outperformed all other methods.

5.5 NEC Animal Data Set

The NEC Animal data set contains sequences of images of 60 toy animals and has been used as a benchmark data set for image and video classification (Mobahi et al., 2009). Each toy animal has around 72 images taken at different poses. The data set contains a total of 4371 images, each of size 580×480 pixels. In the experiments, the images were re-sized to 96×72 pixels and the grey intensity was used as the feature representation. The previous graph construction methodology was followed and algorithm performance was evaluated using the average error rate across 100 random folds with the number of initial labels varying from 60 to 100. In the experiments, the *GGMC* method again achieved the best performance among all tested methods. In particular, when given very sparse labels, that is, one label per class, *GGMC* produced significantly lower error rates.

6. Conclusion and Discussion

The performance of existing graph-based *SSL* methods depends heavily on the availability of accurate initial labels and good connectivity structure. Otherwise, performance can significantly degrade if labels are not distributed evenly across classes, if the initial label locations are biased, or if noise and outliers corrupt the underlying manifold structure. These problems arise in many real world data sets and limit the performance of state-of-the-art *SSL* algorithms. Furthermore, several heuristic choices in the *SSL* approach require considerable exploratory work by the practitioner before the methods perform well in a specific problem domain.

This article addressed these shortcomings and proposed a novel graph-based semi-supervised learning method named greedy gradient Max-Cut (*GGMC*). Our main contributions include:

1. Extending the existing univariate quadratic regularization framework to an optimization over both label matrix and classification function. Such an extension allows us to treat input labels as part of the optimization problem and thereby alleviate *SSL*'s sensitivity to initial labels.

2. Demonstrating that the bivariate formulation is actually a mixed integer programming problem which can be reduced to a binary integer programming (BIP) problem. In addition, we show that an alternating minimization procedure can be used to derive a locally optimal solution.

3. Proving that the proposed bivariate formulation is equivalent to a Max-Cut problem for the two-class case and proving that it is equivalent to a Maximum K-cut problem for the multi-class case. In addition, we proposed an efficient solution with $O(n^2)$ complexity. This greedy gradient Max-Cut (*GGMC*) solution presents a different interpretation for the alternating minimization procedure from a graph cut view.

Unlike other graph-cut based *SSL* methods such as min-cut (Blum and Chawla, 2001; Blum et al., 2004), the proposed *GGMC* algorithm tends to generate more natural graph cuts and avoids biased solutions. In addition, it uses a weighted connectivity matrix to normalize the label matrix. The result is a solution that can cope with all the aforementioned degeneracies. It improves accuracy in practice while remaining efficient. Future work will extend the proposed methods to out-of-sample settings where additional data points are added to the prediction problem without requiring a full retraining procedure. Another interesting extension of the bivariate framework is active learning which can potentially reduce the amount of labels necessary for accurate prediction (Goldberg et al., 2011).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1117631. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. This work was also partially supported by a Google Research Award and by the Department of Homeland Security under Grant No. N66001-09-C-0080.

Appendix A. Multi-Class Case as a Max K-Cut Problem

Here, we show that K-class bivariate graph transduction is equivalent to a Max K-Cut problem. If the number of classes is K, the label variable $\mathbf{Y}$ is a $n \times K$ matrix denoting the classification result. Therein, $\mathbf{Y}_{ij} = 1$ indicates that vertex $\mathbf{x}_i$ is assigned the label j. We rewrite the cost function in Equation (8) as

$$Q(\mathbf{Y}) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^\top \mathbf{A}\mathbf{Y}\right) = \frac{1}{2}\sum_{k=1}^{K} \mathbf{Y}_{.k}^\top \mathbf{A}\mathbf{Y}_{.k}.$$

Let $\mathbf{y}_k = \mathbf{Y}_{.k}$ be a column vector of $\mathbf{Y}$. Let the non-zero elements in $\mathbf{y}_k$ denote the vertices in subset S_k, where $k = 1, 2, \cdots, K$, $S_1 \cup S_2 \cup \cdots \cup S_K = V_\mathbf{A}$, and $S_m \cap S_n = \emptyset$ if $m \neq n$. Then the above cost function is equivalent to

$$\begin{aligned} Q(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K) &= \tfrac{1}{2}\sum_{k=1}^{K} \mathbf{y}_k^\top \mathbf{A}\mathbf{y}_k = \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in S_k \\ i<j}} \mathbf{A}_{ij} \\ &= \sum_{i<j} \mathbf{A}_{ij} - \sum_{m=1}^{K-1} \sum_{n=m+1}^{K} \sum_{\substack{\mathbf{x}_i \in S_m \\ \mathbf{x}_j \in S_n}} \mathbf{A}_{ij}. \end{aligned}$$

Therefore the original minimization problem is equivalent to maximizing the sum of the weight of the edges between the disjoint sets S_k, that is, the maximum K-cut problem

$$\max_{S_1,\ldots,S_K} \sum_{m=1}^{K-1} \sum_{n=m+1}^{K} \sum_{\substack{\mathbf{x}_i \in S_m \\ \mathbf{x}_j \in S_n}} \mathbf{A}_{ij}.$$

References

A. Azran. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *Proceedings of the International Conference on Machine Learning*, pages 49–56, Corvalis, Oregon, 2007. ACM.

M.-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: towards bridging theory and practice. In *Advances in Neural Information Processing Systems*, volume 17, pages 89–96. 2005.

F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, Barbados, January 2005.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a Geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

T. D. Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems*, volume 16. 2004.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of International Conference on Machine Learning*, pages 19–26, San Francisco, CA, USA, 2001.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, Madison, Wisconsin, United States, 1998. ACM.

A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 13–20, Banff, Alberta, Canada, 2004.

M. Carreira-Perpinán and R. S. Zemel. Proximity graphs for clustering and manifold learning. In *Advances in neural information processing systems*, volume 17, pages 225–232. 2005.

O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, Barbados, January 2005.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

O. Chapelle, V. Sindhwani, and S. S. Keerthi. Branch and bound for semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, volume 19, pages 217–224. Cambridge, MA, 2007.

O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *The Journal of Machine Learning Research*, 9:203–233, 2008.

N. V. Chawla and G. Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23(1):331–366, 2005.

F. R. K. Chung and N. Biggs. *Spectral Graph Theory*. American Mathematical Society Providence, RI, 1997.

S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, Shaker Heights, Ohio, United States, 1971. ACM.

M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer Verlag, 2009.

J. Edmonds and E. Johnson. Matching: A well-solved class of integer linear programs. *Combinatorial OptimizationEureka, You Shrink!*, pages 27–30, 2003.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

M. X. Goemans and D. P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 422–431, Montreal, Quebec, Canada, 1994.

M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

A.B. Goldberg, X. Zhu, A. Furger, and J.M. Xu. Oasis: Online active semi-supervised learning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, pages 327–334, 2000.

B. Huang and T. Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In *Int. Workshop on Artificial Intelligence and Statistics*, 2007.

B. Huang and T. Jebara. Collaborative filtering via rating concentration. In Y.W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume Volume 9 of JMLR: W&CP, pages 334–341, May 13-15 2010.

T. Jebara, J. Wang, and S.-F. Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 441–448, 2009.

T. Joachims. Transductive inference for text classifcation using support vector machines. In *Proceedings of International Conference on Machine Learning*, pages 200–209, 1999.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of International Conference on Machine Learning*, pages 290–297, 2003.

R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 43:85–103, 1972.

M.-A. Krogel and T. Scheffer. Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57(1):61–81, 2004.

B. Kveton, M. Valko, A. Rahimi, and L. Huang. Semi-supervised learning with max-margin graph cuts. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 421–428, 2010.

W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

M. Maier, U. von Luxburg, and M. Hein. Influence of graph construction on graph-based clustering measures. In *Advances in Neural Information Processing Systems*, volume 22, pages 1025–1032. 2009.

G. S. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of International Conference on Machine Learning*, pages 593–600, Corvalis, Oregon, 2007.

C. Mathieu and W. Schudy. Yet another algorithm for dense max cut: go greedy. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 176–182, 2008.

S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184, 2011.

H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744, 2009.

S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (coil-20). *Dept. Comput. Sci., Columbia Univ., New York.[Online] http://www. cs. columbia. edu/CAVE/coil-20.html*, 1996.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 824–831, Bonn, Germany, 2005.

V. Sindhwani, J. Hu, and A. Mojsilovic. Regularized co-clustering with dual supervision. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 976–983. MIT Press, 2008.

M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, pages 945–952. 2002.

V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

J. Wang, S.-F. Chang, X. Zhou, and T. C. S. Wong. Active Microscopic Cellular Image Annotation by Superposable Graph Transduction with Imbalanced Labels. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Alaska, USA, June 2008a.

J. Wang, T. Jebara, and S.-F. Chang. Graph transduction via alternating minimization. In *Proceedings of International Conference on Machine Learning*, pages 1144–1151, Helsinki, Finland, 2008b.

J. Wang, Y.-G. Jiang, and S.-F. Chang. Label diagnosis through self tuning for web image search. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, USA, June 2009.

W. Wang and Z.-H. Zhou. A new analysis of co-training. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1135–1142, Haifa, Israel, June 2010.

Z. Xu, R. Jin, J. Zhu, I. King, and M. Lyu. Efficient convex relaxation for transductive support vector machine. volume 21, pages 1641–1648. 2008.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 321–328. 2004.

X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

X. Zhu and A. B. Goldberg. *Introduction to Semi-supervised Learning*. Morgan & Claypool Publishers, 2009.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of International Conference on Machine Learning*, pages 912–919, 2003.

Journal of Machine Learning Research 14 (2013) 801-805 Submitted 9/12; Revised 2/13; Published 3/13

MLPACK: A Scalable C++ Machine Learning Library

Ryan R. Curtin RYAN.CURTIN@CC.GATECH.EDU
James R. Cline JAMES.CLINE@GATECH.EDU
N. P. Slagle NPSLAGLE@GMAIL.COM
William B. March MARCH@GATECH.EDU
Parikshit Ram P.RAM@GATECH.EDU
Nishant A. Mehta NICHE@CC.GATECH.EDU
Alexander G. Gray AGRAY@CC.GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Editor: Balázs Kégl

Abstract

MLPACK is a state-of-the-art, scalable, multi-platform C++ machine learning library released in late 2011 offering both a simple, consistent API accessible to novice users and high performance and flexibility to expert users by leveraging modern features of C++. MLPACK provides cutting-edge algorithms whose benchmarks exhibit far better performance than other leading machine learning libraries. MLPACK version 1.0.3, licensed under the LGPL, is available at `http://www.mlpack.org`.

Keywords: C++, dual-tree algorithms, machine learning software, open source software, large-scale learning

1. Introduction and Goals

Though several machine learning libraries are freely available online, few, if any, offer efficient algorithms to the average user. For instance, the popular Weka toolkit (Hall et al., 2009) emphasizes ease of use but scales poorly; the distributed Apache Mahout library offers scalability at a cost of higher overhead (such as clusters and powerful servers often unavailable to the average user). Also, few libraries offer breadth; for instance, libsvm (Chang and Lin, 2011) and the Tilburg Memory-Based Learner (TiMBL) are highly scalable and accessible yet each offer only a single method.

MLPACK, intended to be the machine learning analog to the general-purpose LAPACK linear algebra library, aims to combine efficiency and accessibility. Written in C++, MLPACK uses the highly efficient Armadillo matrix library (Sanderson, 2010) and is freely available under the GNU Lesser General Public License (LGPL). Through the use of C++ templates, MLPACK both eliminates unnecessary copying of data sets and performs expression optimizations unavailable in other languages. Also, MLPACK is, to our knowledge, unique among existing libraries in using generic programming features of C++ to allow customization of the available machine learning methods without incurring performance penalties.

In addition, users ranging from students to experts should find the consistent, intuitive interface of MLPACK to be highly accessible. Finally, the source code provides references and comprehensive documentation.

Four major goals of the development team of MLPACK are

- to implement **scalable, fast** machine learning algorithms,
- to design an **intuitive, consistent, and simple** API for non-expert users,
- to implement a **variety** of machine learning methods, and
- to provide **cutting-edge** machine learning algorithms unavailable elsewhere.

This paper offers both an introduction to the simple and extensible API and a glimpse of the superior performance of the library.

2. Package Overview

Each algorithm available in MLPACK features both a set of C++ library functions and a standalone command-line executable. Version 1.0.3 includes the following methods:

- nearest/furthest neighbor search with cover trees or kd-trees (k-nearest-neighbors)
- range search with cover trees or kd-trees
- Gaussian mixture models (GMMs)
- hidden Markov models (HMMs)
- LARS / Lasso regression
- k-means clustering
- fast hierarchical clustering (Euclidean MST calculation)[1] (March et al., 2010)
- kernel PCA (and regular PCA)
- local coordinate coding[1] (Yu et al., 2009)
- sparse coding using dictionary learning
- RADICAL (Robust, Accurate, Direct ICA aLgorithm) (Learned-Miller and Fisher, 2003)
- maximum variance unfolding (MVU) via LRSDP[1] (Burer and Monteiro, 2003)
- the naive Bayes classifier
- density estimation trees[1] (Ram and Gray, 2011)

The development team manages MLPACK with Subversion and the Trac bug reporting system, allowing easy downloads and simple bug reporting. The entire development process is transparent, so any interested user can easily contribute to the library. MLPACK can compile from source on Linux, Mac OS, and Windows; currently, different Linux distributions are reviewing MLPACK for inclusion in their package managers, which will allow users to install MLPACK without needing to compile from source.

3. A Consistent, Simple API

MLPACK features a highly accessible API, both in style (such as consistent naming schemes and coding conventions) and ease of use (such as templated defaults), as well as stringent documentation standards. Consequently, a new user can execute algorithms out-of-the-box often with little or no adjustment to parameters, while the seasoned expert can expect extreme flexibility in algorithmic

1. This algorithm is not available in any other comparable software package.

Data Set	MLPACK	Weka	Shogun	MATLAB	mlpy	sklearn
wine	**0.0003**	0.0621	0.0277	0.0021	0.0025	0.0008
cloud	**0.0069**	0.1174	0.5000	0.0210	0.3520	0.0192
wine-qual	**0.0290**	0.8868	4.3617	0.6465	4.0431	0.1668
isolet	**13.0197**	213.4735	37.6190	46.9518	52.0437	46.8016
miniboone	**20.2045**	216.1469	2351.4637	1088.1127	3219.2696	714.2385
yp-msd	**5430.0478**	>9000.0000	>9000.0000	>9000.0000	>9000.0000	>9000.0000
corel	**4.9716**	14.4264	555.9600	60.8496	209.5056	160.4597
covtype	**14.3449**	45.9912	>9000.0000	>9000.0000	>9000.0000	651.6259
mnist	**2719.8087**	>9000.0000	3536.4477	4838.6747	5192.3586	5363.9650
randu	**1020.9142**	2665.0921	>9000.0000	1679.2893	>9000.0000	8780.0176

Table 1: *k*-NN benchmarks (in seconds).

Data Set	**wine**	**cloud**	**wine-qual**	**isolet**	**miniboone**
UCI Name	Wine	Cloud	Wine Quality	ISOLET	MiniBooNE
Size	178x13	2048x10	6497x11	7797x617	130064x50

Data Set	**yp-msd**	**corel**	**covtype**	**mnist**	**randu**
UCI Name	YearPredictionMSD	Corel	Covertype	*N/A*	*N/A*
Size	515345x90	37749x32	581082x54	70000x784	1000000x10

Table 2: Benchmark data set sizes.

tuning. For example, the following line initializes an object which will perform the standard k-means clustering in Euclidean space:

```
KMeans<> k();
```

However, an expert user could easily use the Manhattan distance, a different cluster initialization policy, and allow empty clusters:

```
KMeans<ManhattanDistance, KMeansPlusPlusInitialization, AllowEmptyClusters> k();
```

Users can implement these custom classes in their code, then simply link against the MLPACK library, requiring no modification within the MLPACK library. In addition to this flexibility, Armadillo 3.4.0 includes sparse matrix support; sparse matrices can be used in place of dense matrices for the appropriate MLPACK methods.

4. Benchmarks

To demonstrate the efficiency of the algorithms implemented in MLPACK, we present a comparison of the running times of *k*-nearest-neighbors and the *k*-means clustering algorithm from MLPACK, Weka (Hall et al., 2009), MATLAB, the Shogun Toolkit (Sonnenburg et al., 2010), mlpy (Albanese et al., 2012), and scikit.learn ('sklearn') (Pedregosa et al., 2011), using a modest consumer-grade workstation containing an AMD Phenom II X6 1100T processor clocked at 3.3 GHz and 8 GB of RAM.

Eight data sets from the UCI data sets repository (Frank and Asuncion, 2010) are used; the MNIST handwritten digit database is also used ('mnist') (LeCun et al., 2001), as well as a uniformly distributed random data set ('randu'). Information on the sizes of these ten data sets appears in Table 2. Data set loading time is not included in the benchmarks. Each test was run 5 times; the average is shown in the results.

Data Set	Clusters	MLPACK	Shogun	MATLAB	sklearn
wine	3	**0.0006**	0.0073	0.0055	0.0064
cloud	5	**0.0036**	0.1240	0.0194	0.1753
wine-qual	7	**0.0221**	0.6030	0.0987	4.0407
isolet	26	**4.9762**	8.5093	54.7463	7.0902
miniboone	2	**0.1853**	8.0206	0.7221	*memory*
yp-msd	10	**34.8223**	135.8853	269.7302	*memory*
corel	10	**0.4672**	2.4237	1.6318	*memory*
covtype	7	**13.5997**	71.1283	54.9034	*memory*
mnist	10	**80.2092**	163.7513	133.9970	*memory*
randu	75	**727.1498**	7443.2675	3117.5177	*memory*

Table 3: k-means benchmarks (in seconds).

k-NN was run with each library on each data set, with $k = 3$. The results for each library and each data set appears in Table 1. The k-means algorithm was run with the same starting centroids for each library, and 1000 iterations maximum. The number of clusters k was chosen to reflect the structure of the data set. Benchmarks for k-means are given in Table 3. Weka and mlpy are excluded because they do not allow specification of the starting centroids. '*memory*' indicates that the system ran out of memory during the test.

MLPACK's k-nearest neighbors and k-means are faster than the competitors in all test cases. Benchmarks for other methods, omitted due to space constraints, also show similar speedups over competing implementations.

5. Future Plans and Conclusion

The favorable benchmarks exhibited above are not necessarily the global optimum; MLPACK's active development team includes several core developers and many contributors. Because MLPACK is open-source, contributions from outsiders are welcome, including feature requests and bug reports. Thus, the performance, extensibility, and breadth of algorithms within MLPACK are all certain to improve.

The first releases of MLPACK lacked parallelism, but experimental parallel code using OpenMP is currently in testing. This parallel support must maintain a simple API and avoid large, reverse-incompatible API changes. Other useful planned features include using on-disk databases (rather than requiring loading the data set entirely into RAM) and validation of saved models (such as trees or distributions). Refactoring work continues on existing code, providing more flexible abstractions and greater extensibility. Nevertheless, MLPACK's future growth will mostly be the addition of new machine learning methods; since the original release (1.0.0), there are five new methods. Forthcoming methods include approximate nearest neighbors, locality-sensitive hashing (LSH), and support vector machines (SVMs).

In conclusion, we have shown that MLPACK is a state-of-the-art C++ machine learning library which leverages the powerful C++ concept of generic programming to give excellent performance on large data sets.

Acknowledgments

A full list of developers and researchers (other than the authors) who have contributed significantly to MLPACK are Sterling Peet, Vlad Grantcharov, Ajinkya Kale, Dongryeol Lee, Chip Mappus, Hua Ouyang, Long Quoc Tran, Noah Kauffman, Rajendran Mohan, and Trironk Kiatkungwanglai.

References

Davide Albanese, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman, and Cesare Furlanello. mlpy: Machine Learning PYThon. 2012. Project homepage at `http://mlpy.fbk.eu/`.

Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Andrew J. Frank and Arthur Asuncion. UCI machine learning repository [`http://archive.ics.uci.edu/ml`], 2010. University of California, Irvine, School of Information and Computer Sciences.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

Erik G. Learned-Miller and John W. Fisher. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, December 2003. ISSN 1532-4435.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

William B. March, Parikshit Ram, and Alexander G. Gray. Fast Euclidean minimum spanning tree: algorithm, analysis, and applications. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 603–612, 2010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Parikshit Ram and Alexander G. Gray. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 627–635, New York, NY, USA, 2011. ACM.

Conrad Sanderson. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA, 2010.

Soeren Sonnenburg, Gunnar Raetsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtech Franc. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11:1799–1802, June 2010.

Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 2223–2231, 2009.

Journal of Machine Learning Research 14 (2013) 807-841 Submitted 3/12; Revised 10/12; Published 3/13

Greedy Sparsity-Constrained Optimization

Sohail Bahmani SBAHMANI@CMU.EDU
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA

Bhiksha Raj BHIKSHA@CS.CMU.EDU
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA

Petros T. Boufounos PETROSB@MERL.COM
Mitsubishi Electric Research Laboratories
201 Broadway
Boston, MA 02139, USA

Editor: Francis Bach

Abstract

Sparsity-constrained optimization has wide applicability in machine learning, statistics, and signal processing problems such as feature selection and Compressed Sensing. A vast body of work has studied the sparsity-constrained optimization from theoretical, algorithmic, and application aspects in the context of sparse estimation in linear models where the fidelity of the estimate is measured by the squared error. In contrast, relatively less effort has been made in the study of sparsity-constrained optimization in cases where nonlinear models are involved or the cost function is not quadratic. In this paper we propose a greedy algorithm, Gradient Support Pursuit (GraSP), to approximate sparse minima of cost functions of arbitrary form. Should a cost function have a Stable Restricted Hessian (SRH) or a Stable Restricted Linearization (SRL), both of which are introduced in this paper, our algorithm is guaranteed to produce a sparse vector within a bounded distance from the true sparse optimum. Our approach generalizes known results for quadratic cost functions that arise in sparse linear regression and Compressed Sensing. We also evaluate the performance of GraSP through numerical simulations on synthetic and real data, where the algorithm is employed for sparse logistic regression with and without ℓ_2-regularization.

Keywords: sparsity, optimization, compressed sensing, greedy algorithm

1. Introduction

The demand for high-dimensional data analysis has grown significantly over the past decade by the emergence of applications such as social networking, bioinformatics, and mathematical finance. In these applications data samples often have thousands of features using which an underlying parameter must be inferred or predicted. In many circumstances the number of collected samples is significantly smaller than the dimensionality of the data, rendering any inference from the data ill-posed. However, it is widely acknowledged that the data sets that need to be processed usually

exhibit significant structure, which sparsity models are often able to capture. This structure can be exploited for robust regression and hypothesis testing, model reduction and variable selection, and more efficient signal acquisition in *underdetermined* regimes. Estimation of parameters with sparse structure is usually cast as an optimization problem, formulated according to specific application requirements. Developing techniques that are robust and computationally tractable to solve these optimization problems, even only approximately, is therefore critical.

In particular, theoretical and application aspects of sparse estimation in linear models have been studied extensively in areas such as signal processing, machine learning, and statistics. However, sparse estimation in problems where nonlinear models are involved have received comparatively little attention. Most of the work in this area extend the use of the ℓ_1-norm as a regularizer, effective to induce sparse solutions in linear regression, to problems with nonlinear models (see, e.g., Bunea, 2008; van de Geer, 2008; Kakade et al., 2010; Negahban et al., 2009). As a special case, logistic regression with ℓ_1 and elastic net regularization are studied by Bunea (2008). Furthermore, Kakade et al. (2010) have studied the accuracy of sparse estimation through ℓ_1-regularization for the exponential family distributions. A more general frame of study is proposed and analyzed by Negahban et al. (2009) where regularization with "decomposable" norms is considered in M-estimation problems. To provide the accuracy guarantees, these works generalize the Restricted Eigenvalue condition (Bickel et al., 2009) to ensure that the loss function is strongly convex over a restriction of its domain. We would like to emphasize that these sufficient conditions generally hold with proper constants and with high probability only if one assumes that the true parameter is bounded. This fact is more apparent in some of the mentioned work (e.g., Bunea, 2008; Kakade et al., 2010), while in some others (e.g., Negahban et al., 2009) the assumption is not explicitly stated. We will elaborate on this matter in Section 2. Tewari et al. (2011) also proposed a coordinate-descent type algorithm for minimization of a convex and smooth objective over the convex signal/parameter models introduced in Chandrasekaran et al. (2012). This formulation includes the ℓ_1-constrained minimization as a special case, and the algorithm is shown to converge to the minimum in objective value similar to the standard results in convex optimization.

Furthermore, Shalev-Shwartz et al. (2010) proposed a number of greedy that sparsify a given estimate at the cost of relatively small increase of the objective function. However, their algorithms are not stand-alone. A generalization of Compressed Sensing is also proposed in Blumensath (2010), where the linear measurement operator is replaced by a nonlinear operator that applies to the sparse signal. Considering the norm of the residual error as the objective, Blumensath (2010) shows that if the objective satisfies certain sufficient conditions, the sparse signal can be accurately estimated by a generalization of the Iterative Hard Thresholding algorithm (Blumensath and Davies, 2009). The formulation of Blumensath (2010), however, has a limited scope because the metric of error is defined using a norm. For instance, the formulation does not apply to objectives such as the logistic loss. More recently, Jalali et al. (2011) studied a forward-backward algorithm using a variant of the sufficient conditions introduced in Negahban et al. (2009). Similar to our work, the main result in Jalali et al. (2011) imposes conditions on the function as restricted to sparse inputs whose non-zeros are fewer than a multiple of the target sparsity level. The multiplier used in their results has an *objective-dependent* value and is never less than 10. Furthermore, the multiplier is important in their analysis not only for determining the stopping condition of the algorithm, but also in the lower bound assumed for the minimal magnitude of the non-zero entries. In contrast, the multiplier in our results is fixed at 4, independent of the objective function itself, and we make no assumptions about the magnitudes of the non-zero entries.

This paper presents an extended version with improved guarantees of our prior work in Bahmani et al. (2011), where we proposed a greedy algorithm, the Gradient Support Pursuit (GraSP), for sparse estimation problems that arise in applications with general nonlinear models. We prove the accuracy of GraSP for a class of cost functions that have a *Stable Restricted Hessian* (SRH). The SRH, introduced in Bahmani et al. (2011), characterizes the functions whose restriction to sparse canonical subspaces have well-conditioned Hessian matrices. Similarly, we analyze the GraSP algorithm for non-smooth functions that have a *Stable Restricted Linearization* (SRL), a property introduced in this paper, analogous to SRH. The analysis and the guarantees for smooth and non-smooth cost functions are similar, except for less stringent conditions derived for smooth cost functions due to properties of symmetric Hessian matrices. We also prove that the SRH holds for the case of the ℓ_2-penalized logistic loss function.

1.1 Notation

In the remainder of this paper we use the notation listed in Table 1.

1.2 Paper Outline

In Section 2 we provide a background on sparse parameter estimation which serves as an overview of prior work. In Section 3 we state the general formulation of the problem and present our algorithm. Conditions that characterize the cost functions and the main accuracy guarantees of our algorithm are provided in Section 3 as well. The guarantees of the algorithm are proved in Appendices A and B. As an example where our algorithm can be applied, ℓ_2-regularized logistic regression is studied in Section 4. Some experimental results for logistic regression with sparsity constraints are presented in Section 5. Finally, Section 6 discusses the results and concludes.

2. Background

We first briefly review sparse estimation problems studied in the literature.

2.1 Sparse Linear Regression and Compressed Sensing

The special case of sparse estimation in linear models has gained significant attention under the title of Compressed Sensing (CS) (Donoho, 2006). In standard CS problems the aim is to estimate a sparse vector $\mathbf{x}^\star$ from noisy linear measurements $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \mathbf{e}$, where $\mathbf{A}$ is a known $n \times p$ measurement matrix with $n \ll p$ and $\mathbf{e}$ is the additive measurement noise. To find the sparsest estimate in this *underdetermined* problem that is consistent with the measurements $\mathbf{y}$ one needs to solve the optimization problem

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \ \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \varepsilon, \tag{1}$$

where ε is a given upper bound for $\|\mathbf{e}\|_2$ (Candès et al., 2006). In the absence of noise (i.e., when $\varepsilon = 0$), if $\mathbf{x}^\star$ is s-sparse (i.e., it has at most s nonzero entries) one merely needs every $2s$ columns of $\mathbf{A}$ to be linearly independent to guarantee exact recovery (Donoho and Elad, 2003). Unfortunately, the ideal solver (1) is computationally NP-hard in general (Natarajan, 1995) and one must seek approximate solvers instead.

It is shown in Candès et al. (2006) that under certain conditions, minimizing the ℓ_1-norm as a convex proxy for the ℓ_0-norm yields accurate estimates of $\mathbf{x}^\star$. The resulting approximate solver

Symbol	Description
$[n]$	the set $\{1,2,\ldots,n\}$ for any $n \in \mathbb{N}$
I	calligraphic letters denote sets unless stated otherwise (e.g., $\mathcal{N}\left(\mu,\sigma^2\right)$ denotes a normal distribution)
I^c	complement of set I
$\mathbf{v}$	bold face small letters denote column vectors in $\mathbb{R}^b$ for some $b \in \mathbb{N}$
$\|\mathbf{v}\|_q$	the ℓ_q-norm of vector $\mathbf{v}$, that is $\left(\sum_{i=1}^{b} \|v_i\|^q\right)^{1/q}$, for a real number $q \geq 1$
$\|\mathbf{v}\|_0$	the "ℓ_0-norm" of vector $\mathbf{v}$ that merely counts its nonzero entries
$\mathbf{v}\|_I$	depending on the context 1. restriction of vector $\mathbf{v}$ to the rows indicated by indices in I, or 2. a vector that equals $\mathbf{v}$ except for coordinates in I^c where it is zero
$\mathbf{v}_r$	the best r-term approximation of vector $\mathbf{v}$
$\mathrm{supp}\,(\mathbf{v})$	the support set (i.e., indices of the non-zero entries) of $\mathbf{v}$
$\mathbf{M}$	bold face capital letters denote matrices in $\mathbb{R}^{a\times b}$ for some $a,b \in \mathbb{N}$
$\mathbf{M}^{\mathrm{T}}$	transpose of matrix $\mathbf{M}$
$\mathbf{M}^{\dagger}$	pseudo-inverse of matrix $\mathbf{M}$
$\mathbf{M}_I$	restriction of matrix $\mathbf{M}$ to the columns enumerated by I
$\|\mathbf{M}\|$	the operator norm of matrix $\mathbf{M}$ which is equal to $\sqrt{\lambda_{\max}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right)}$
$\mathbf{I}$	the identity matrix
$\mathbf{P}_I$	restriction of the identity matrix to the columns indicated by I
$\mathbf{1}$	column vector of all ones
$\mathbb{E}[\cdot]$	expectation
$\mathbf{H}_f(\cdot)$	Hessian of the function f

Table 1: Notation used in this paper

basically returns the solution to the convex optimization problem

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \varepsilon, \tag{2}$$

The required conditions for approximate equivalence of (1) and (2), however, generally hold only if measurements are collected at a higher rate. Ideally, one merely needs $n = O(s)$ measurements to estimate $\mathbf{x}^\star$, but $n = O(s \log \frac{p}{s})$ measurements are necessary for the accuracy of (2) to be guaranteed.

The convex program (2) can be solved in polynomial time using interior point methods. However, these methods do not scale well as the size of the problem grows. Therefore, several first-order convex optimization methods are developed and analyzed as more efficient alternatives (see, e.g., Beck and Teboulle, 2009; Agarwal et al., 2010). Another category of low-complexity algorithms in CS are the non-convex *greedy pursuits* including Orthogonal Matching Pursuit (OMP) (Pati et al., 1993; Tropp and Gilbert, 2007), Compressive Sampling Matching Pursuit (CoSaMP) (Needell and Tropp, 2009), Iterative Hard Thresholding (IHT) (Blumensath and Davies, 2009), and Subspace Pursuit (Dai and Milenkovic, 2009) to name a few. These greedy algorithms implicitly approximate the solution to the ℓ_0-constrained least squares problem

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_0 \leq s. \tag{3}$$

The main theme of these iterative algorithms is to use the residual error from the previous iteration to successively approximate the position of non-zero entries and estimate their values. These algorithms have shown to exhibit accuracy guarantees similar to those of convex optimization methods, though with more stringent requirements.

As mentioned above, to guarantee accuracy of the CS algorithms the measurement matrix should meet certain conditions such as *incoherence* (Donoho and Huo, 2001), Restricted Isometry Property (RIP) (Candès et al., 2006), Nullspace Property (Cohen et al., 2009), etc. Among these conditions RIP is the most commonly used and the best understood condition.

Matrix $\mathbf{A}$ is said to satisfy the RIP of order k—in its symmetric form—with constant δ_k, if $\delta_k < 1$ is the smallest number that

$$(1 - \delta_k) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k) \|\mathbf{x}\|_2^2$$

holds for all k-sparse vectors $\mathbf{x}$. Several CS algorithms are shown to produce accurate solutions provided that the measurement matrix has a sufficiently small RIP constant of order ck with c being a small integer. For example, solving (2) is guaranteed to yield an accurate estimate of s-sparse $\mathbf{x}^\star$ if $\delta_{2s} < \sqrt{2} - 1$ (Candès, 2008). Interested readers can find the best known RIP-based accuracy guarantees for some of the CS algorithms in Foucart (2012).

2.2 Beyond Linear Models

The CS reconstruction algorithms attempt to provide a sparse vector that incurs only a small squared error which measures consistency of the solution versus the acquired data. While this measure of discrepancy is often desirable for signal processing applications, it is not the appropriate choice for a variety of other applications. For example, in statistics and machine learning the logistic loss function is also commonly used in regression and classification problems (see Liu et al., 2009, and references therein). Thus, it is desirable to develop theory and algorithms that apply to a broader class of optimization problems with sparsity constraints.

The existing studies on this subject are mostly in the context of statistical estimation. The majority of these studies consider the cost function to be convex everywhere and rely on the ℓ_1-regularization as the means to induce sparsity in the solution. For example, Kakade et al. (2010) have shown that for the exponential family of distributions maximum likelihood estimation with ℓ_1-regularization yields accurate estimates of the underlying sparse parameter. Furthermore, Negahban et al. have developed a unifying framework for analyzing statistical accuracy of *M-estimators* regularized by "decomposable" norms in (Negahban et al., 2009). In particular, in their work ℓ_1-regularization is applied to Generalized Linear Models (GLM) (Dobson and Barnett, 2008) and shown to guarantee a bounded distance between the estimate and the true statistical parameter. To establish this error bound they introduced the notion of *Restricted Strong Convexity* (RSC), which basically requires a lower bound on the curvature of the cost function around the true parameter in a restricted set of directions. The achieved error bound in this framework is inversely proportional to this curvature bound. Furthermore, Agarwal et al. (2010) have studied Projected Gradient Descent as a method to solve ℓ_1-constrained optimization problems and established accuracy guarantees using a slightly different notion of RSC and *Restricted Smoothness* (RSM).

Note that the guarantees provided for majority of the ℓ_1-regularization algorithms presume that the true parameter is bounded, albeit implicitly. For instance, the error bound for ℓ_1-regularized logistic regression is recognized by Bunea (2008) to be dependent on the true parameter (Bunea, 2008, Assumption A, Theorem 2.4, and the remark that succeeds them). Moreover, the result proposed by Kakade et al. (2010) implicitly requires the true parameter to have a sufficiently short length to allow the choice of the desirable regularization coefficient (Kakade et al., 2010, Theorems 4.2 and 4.5). Negahban et al. (2009) also assume that the true parameter is inside the unit ball to establish the required condition for their analysis of ℓ_1-regularized GLM, although this restriction is not explicitly stated (see the longer version of Negahban et al., 2009, p. 37). We can better understand why restricting the length of the true parameter may generally be inevitable by viewing these estimation problems from the perspective of empirical processes and their convergence. The empirical processes, including those considered in the studies mentioned above, are generally good approximations of their corresponding expected process (see Vapnik, 1998, chap. 5 and van de Geer, 2000). Therefore, if the expected process is not strongly convex over an unbounded, but perhaps otherwise restricted, set the corresponding empirical process cannot be strongly convex over the same set. This reasoning applies in many cases including the studies mentioned above, where it would be impossible to achieve the desired restricted strong convexity properties—with high probability—if the true parameter is allowed to be unbounded.

Furthermore, the methods that rely on the ℓ_1-norm are known to result in sparse solutions, but, as mentioned in Kakade et al. (2010), the sparsity of these solutions is not known to be optimal in general. One can intuit this fact from definitions of RSC and RSM. These two properties bound the curvature of the function from below and above in a restricted set of directions around the true optimum. For quadratic cost functions, such as squared error, these curvature bounds are absolute constants. As stated before, for more general cost functions such as the loss functions in GLMs, however, these constants will depend on the location of the true optimum. Consequently, depending on the location of the true optimum these error bounds could be extremely large, albeit finite. When error bounds are significantly large, the sparsity of the solution obtained by ℓ_1-regularization may not be satisfactory. This motivates investigation of algorithms that do not rely on ℓ_1-norm to induce sparsity.

3. Problem Formulation and the GraSP Algorithm

As seen in Section 2.1, in standard CS the squared error $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ is used to measure fidelity of the estimate. While this is appropriate for a large number of signal acquisition applications, it is not the right cost in other fields. Thus, the significant advances in CS cannot readily be applied in these fields when estimation or prediction of sparse parameters become necessary. In this paper we focus on a generalization of (3) where a generic cost function replaces the squared error. Specifically, for the cost function $f : \mathbb{R}^p \mapsto \mathbb{R}$, it is desirable to approximate

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } \|\mathbf{x}\|_0 \leq s. \tag{4}$$

We propose the Gradient Support Pursuit (GraSP) algorithm, which is inspired by and generalizes the CoSaMP algorithm, to approximate the solution to (4) for a broader class of cost functions.

Of course, even for a simple quadratic objective, (4) can have combinatorial complexity and become NP-hard. However, similar to the results of CS, knowing that the cost function obeys certain properties allows us to obtain accurate estimates through tractable algorithms. To guarantee that GraSP yields accurate solutions and is a tractable algorithm, we also require the cost function to have certain properties that will be described in Section 3.2. These properties are analogous to and generalize the RIP in the standard CS framework. For smooth cost functions we introduce the notion of a Stable Restricted Hessian (SRH) and for non-smooth cost functions we introduce the Stable Restricted Linearization (SRL). Both of these properties basically bound the Bregman divergence of the cost function restricted to sparse canonical subspaces. However, the analysis based on the SRH is facilitated by matrix algebra that results in somewhat less restrictive requirements for the cost function.

3.1 Algorithm Description

Algorithm 1: The GraSP algorithm

input : $f(\cdot)$ and s
output: $\widehat{\mathbf{x}}$

initialize: $\widehat{\mathbf{x}} = 0$
repeat
  compute local gradient: $\mathbf{z} = \nabla f(\widehat{\mathbf{x}})$
  identify directions: $\mathcal{Z} = \operatorname{supp}(\mathbf{z}_{2s})$
  merge supports: $\mathcal{T} = \mathcal{Z} \cup \operatorname{supp}(\widehat{\mathbf{x}})$
  minimize over support: $\mathbf{b} = \arg\min f(\mathbf{x})$ s.t. $\mathbf{x}|_{\mathcal{T}^c} = \mathbf{0}$
  prune estimate: $\widehat{\mathbf{x}} = \mathbf{b}_s$
until *halting condition holds*

GraSP is an iterative algorithm, summarized in Algorithm 1, that maintains and updates an estimate $\widehat{\mathbf{x}}$ of the sparse optimum at every iteration. The first step in each iteration, $\mathbf{z} = \nabla f(\widehat{\mathbf{x}})$, evaluates the gradient of the cost function at the current estimate. For nonsmooth functions, instead of the gradient we use a *restricted subgradient* $\mathbf{z} = \nabla_f(\widehat{\mathbf{x}})$ defined in Section 3.2. Then $2s$ coordinates of the vector $\mathbf{z}$ that have the largest magnitude are chosen as the directions in which pursuing the minimization will be most effective. Their indices, denoted by $\mathcal{Z} = \operatorname{supp}(\mathbf{z}_{2s})$, are then merged

with the support of the current estimate to obtain $\mathcal{T} = \mathcal{Z} \cup \text{supp}(\widehat{\mathbf{x}})$. The combined support is a set of at most $3s$ indices over which the function f is minimized to produce an intermediate estimate $\mathbf{b} = \arg\min f(\mathbf{x})$ s.t. $\mathbf{x}|_{\mathcal{T}^c} = 0$. The estimate $\widehat{\mathbf{x}}$ is then updated as the best s-term approximation of the intermediate estimate $\mathbf{b}$. The iterations terminate once certain condition, for instance, on the change of the cost function or the change of the estimated minimum from the previous iteration, holds.

In the special case where the squared error $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ is the cost function, GraSP reduces to CoSaMP. Specifically, the gradient step reduces to the proxy step $\mathbf{z} = \mathbf{A}^{\mathrm{T}}(\mathbf{y} - \mathbf{A}\widehat{\mathbf{x}})$ and minimization over the restricted support reduces to the constrained pseudoinverse step $\mathbf{b}|_{\mathcal{T}} = \mathbf{A}_{\mathcal{T}}^{\dagger}\mathbf{y}$, $\mathbf{b}|_{\mathcal{T}^c} = \mathbf{0}$ in CoSaMP.

Variants Although in this paper we only analyze the standard form of GraSP outlined in Algorithm 1, other variants of the algorithm can also be studied. Below we list some of these variants.

1. *Debiasing*: In this variant, instead of performing a hard thresholding on the vector $\mathbf{b}$, the objective is minimized restricted to the support set of $\mathbf{b}_s$ to obtain the new iterate:

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } \text{supp}(\mathbf{x}) \subseteq \text{supp}(\mathbf{b}_s).$$

2. *Restricted Newton Step*: To reduce the computations in each iteration, the minimization that yields $\mathbf{b}$, we can set $\mathbf{b}|_{\mathcal{T}^c} = \mathbf{0}$ and take a restricted Newton step as

$$\mathbf{b}|_{\mathcal{T}} = \widehat{\mathbf{x}}|_{\mathcal{T}} - \kappa \left(\mathbf{P}_{\mathcal{T}}^{\mathrm{T}} \mathbf{H}_f(\widehat{\mathbf{x}}) \mathbf{P}_{\mathcal{T}}\right)^{-1} \widehat{\mathbf{x}}|_{\mathcal{T}},$$

   where $\kappa > 0$ is a step-size. Of course, here we are assuming that the restricted Hessian, $\mathbf{P}_{\mathcal{T}}^{\mathrm{T}} \mathbf{H}_f(\widehat{\mathbf{x}}) \mathbf{P}_{\mathcal{T}}$, is invertible.

3. *Restricted Gradient Descent*: The minimization step can be relaxed even further by applying a restricted gradient descent. In this approach, we again set $\mathbf{b}|_{\mathcal{T}^c} = \mathbf{0}$ and

$$\mathbf{b}|_{\mathcal{T}} = \widehat{\mathbf{x}}|_{\mathcal{T}} - \kappa \nabla f(\widehat{\mathbf{x}})|_{\mathcal{T}}.$$

   Since $\mathcal{T}$ contains both the support set of $\widehat{\mathbf{x}}$ and the $2s$-largest entries of $\nabla f(\widehat{\mathbf{x}})$, it is easy to show that each iteration of this alternative method is equivalent to a standard gradient descent followed by a hard thresholding. In particular, if the squared error is the cost function as in standard CS, this variant reduces to the IHT algorithm.

3.2 Sparse Reconstruction Conditions

In what follows we characterize the functions for which accuracy of GraSP can be guaranteed. For twice continuously differentiable functions we rely on Stable Restricted Hessian (SRH), while for non-smooth cost functions we introduce the Stable Restricted Linearization (SRL). These properties that are analogous to the RIP in the standard CS framework, basically require that the curvature of the cost function over the sparse subspaces can be bounded locally from above and below such that the corresponding bounds have the same order. Below we provide precise definitions of these two properties.

Definition 1 (Stable Restricted Hessian)**.** Suppose that f is a twice continuously differentiable function whose Hessian is denoted by $\mathbf{H}_f(\cdot)$. Furthermore, let

$$A_k(\mathbf{x}) = \sup\left\{\Delta^{\mathrm{T}}\mathbf{H}_f(\mathbf{x})\Delta \;\middle|\; |\mathrm{supp}(\mathbf{x})\cup\mathrm{supp}(\Delta)| \le k, \|\Delta\|_2 = 1\right\} \tag{5}$$

and

$$B_k(\mathbf{x}) = \inf\left\{\Delta^{\mathrm{T}}\mathbf{H}_f(\mathbf{x})\Delta \;\middle|\; |\mathrm{supp}(\mathbf{x})\cup\mathrm{supp}(\Delta)| \le k, \|\Delta\|_2 = 1\right\}, \tag{6}$$

for all k-sparse vectors $\mathbf{x}$. Then f is said to have a Stable Restricted Hessian (SRH) with constant μ_k, or in short μ_k-SRH, if $1 \le \frac{A_k(\mathbf{x})}{B_k(\mathbf{x})} \le \mu_k$.

Remark 1. Since the Hessian of f is symmetric, an equivalent for Definition 1 is that a twice continuously differentiable function f has μ_k-SRH if the condition number of $\mathbf{P}_{\mathcal{K}}\mathbf{H}_f(\mathbf{x})\mathbf{P}_{\mathcal{K}}^{\mathrm{T}}$ is not greater than μ_k for all k-sparse vectors $\mathbf{x}$ and sets $\mathcal{K}\subseteq[p]$ with $|\mathrm{supp}(\mathbf{x})\cup\mathcal{K}| \le k$.

In the special case when the cost function is the squared error as in (3), we can write $\mathbf{H}_f(\mathbf{x}) = \mathbf{A}^{\mathrm{T}}\mathbf{A}$ which is constant. The SRH condition then requires

$$B_k\|\Delta\|_2^2 \le \|\mathbf{A}\Delta\|_2^2 \le A_k\|\Delta\|_2^2$$

to hold for all k-sparse vectors Δ with $A_k/B_k \le \mu_k$. Therefore, in this special case the SRH condition essentially becomes equivalent to the RIP condition.

Remark 2. Note that the functions that satisfy the SRH are convex over canonical sparse subspaces, but they are not necessarily convex everywhere. The following two examples describe some non-convex functions that have SRH.

Example 1. Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}$, where $\mathbf{Q} = 2\times\mathbf{1}\mathbf{1}^{\mathrm{T}} - \mathbf{I}$. Obviously, we have $\mathbf{H}_f(\mathbf{x}) = \mathbf{Q}$. Therefore, (5) and (6) determine the extreme eigenvalues across all of the $k\times k$ symmetric submatrices of $\mathbf{Q}$. Note that the diagonal entries of $\mathbf{Q}$ are all equal to one, while its off-diagonal entries are all equal to two. Therefore, for any 1-sparse signal $\mathbf{u}$ we have $\mathbf{u}^{\mathrm{T}}\mathbf{Q}\mathbf{u} = \|\mathbf{u}\|_2^2$, meaning that f has μ_1-SRH with $\mu_1 = 1$. However, for $\mathbf{u} = [1,-1,0,\ldots,0]^{\mathrm{T}}$ we have $\mathbf{u}^{\mathrm{T}}\mathbf{Q}\mathrm{u} < 0$, which means that the Hessian of f is not positive semi-definite (i.e., f is not convex).

Example 2. Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2 + Cx_1x_2\cdots x_{k+1}$ where the dimensionality of $\mathbf{x}$ is greater than k. It is obvious that this function is convex for k-sparse vectors as $x_1x_2\cdots x_{k+1} = 0$ for any k-sparse vector. So we can easily verify that f satisfies SRH of order k. However, for $x_1 = x_2 = \cdots = x_{k+1} = t$ and $x_i = 0$ for $i > k+1$ the restriction of the Hessian of f to indices in $[k+1]$ (i.e., $\mathbf{P}_{[k+1]}^{\mathrm{T}}\mathbf{H}_f(\mathbf{x})\mathbf{P}_{[k+1]}$) is a matrix with diagonal entries all equal to one and off-diagonal entries all equal to Ct^{k-1}. Let $\mathbf{Q}$ denote this matrix and $\mathbf{u}$ be a unit-norm vector such that $\langle\mathbf{u},\mathbf{1}\rangle = 0$. Then it is straightforward to verify that $\mathbf{u}^{\mathrm{T}}\mathbf{Q}\mathbf{u} = 1 - Ct^{k-1}$, which can be negative for sufficiently large values of C and t. Therefore, the Hessian of f is not positive semi-definite everywhere, meaning that f is not convex.

To generalize the notion of SRH to the case of nonsmooth functions, first we define the *restricted subgradient* of a function.

Definition 2 (Restricted Subgradient)**.** We say vector $\nabla_f(\mathbf{x})$ is a restricted subgradient of $f:\mathbb{R}^p\mapsto\mathbb{R}$ at point $\mathbf{x}$ if

$$f(\mathbf{x}+\Delta) - f(\mathbf{x}) \ge \left\langle\nabla_f(\mathbf{x}),\Delta\right\rangle$$

holds for all k-sparse vectors Δ.

Remark 3. We introduced the notion of restricted subgradient so that the restrictions imposed on f are as minimal as we need. We acknowledge that the existence of restricted subgradients implies convexity in sparse directions, but it does not imply convexity everywhere.

Remark 4. Obviously, if the function f is convex everywhere, then any subgradient of f determines a restricted subgradient of f as well. In general one may need to invoke the axiom of choice to define the restricted subgradient.

Remark 5. We drop the sparsity level from the notation as it can be understood from the context.

With a slight abuse of terminology we call

$$\mathrm{B}_f\left(\mathbf{x}' \parallel \mathbf{x}\right) = f\left(\mathbf{x}'\right) - f\left(\mathbf{x}\right) - \left\langle \nabla_f\left(\mathbf{x}\right), \mathbf{x}' - \mathbf{x} \right\rangle$$

the restricted Bregman divergence of $f : \mathbb{R}^p \mapsto \mathbb{R}$ between points $\mathbf{x}$ and $\mathbf{x}'$ where $\nabla_f(\cdot)$ gives a restricted subgradient of $f(\cdot)$.

Definition 3 (Stable Restricted Linearization)**.** Let $\mathbf{x}$ be a k-sparse vector in $\mathbb{R}^p$. For function $f : \mathbb{R}^p \mapsto \mathbb{R}$ we define the functions

$$\alpha_k\left(\mathbf{x}\right) = \sup\left\{ \frac{1}{\|\Delta\|_2^2} \mathrm{B}_f\left(\mathbf{x}+\Delta \parallel \mathbf{x}\right) \mid \Delta \neq 0 \text{ and } \left|\mathrm{supp}\left(\mathbf{x}\right) \cup \mathrm{supp}\left(\Delta\right)\right| \leq k \right\}$$

and

$$\beta_k\left(\mathbf{x}\right) = \inf\left\{ \frac{1}{\|\Delta\|_2^2} \mathrm{B}_f\left(\mathbf{x}+\Delta \parallel \mathbf{x}\right) \mid \Delta \neq 0 \text{ and } \left|\mathrm{supp}\left(\mathbf{x}\right) \cup \mathrm{supp}\left(\Delta\right)\right| \leq k \right\}.$$

Then $f(\cdot)$ is said to have a Stable Restricted Linearization with constant μ_k, or μ_k-SRL, if $\frac{\alpha_k(\mathbf{x})}{\beta_k(\mathbf{x})} \leq \mu_k$ for all k-sparse vectors $\mathbf{x}$.

Remark 6. The SRH and SRL conditions are similar to various forms of the Restricted Strong Convexity (RSC) and Restricted Strong Smoothness (RSS) conditions (Negahban et al., 2009; Agarwal et al., 2010; Blumensath, 2010; Jalali et al., 2011; Zhang, 2011) in the sense that they all bound the curvature of the objective function over a restricted set. The SRL condition quantifies the curvature in terms of a (restricted) Bregman divergence similar to RSC and RSS. The quadratic form used in SRH can also be converted to the Bregman divergence form used in RSC and RSS and vice-versa using the mean-value theorem. However, compared to various forms of RSC and RSS conditions SRH and SRL have some important distinctions. The main difference is that the bounds in SRH and SRL conditions are not global constants; only their ratio is required to be bounded globally. Furthermore, unlike the SRH and SRL conditions the variants of RSC and RSS, that are used in convex relaxation methods, are required to hold over a set which is strictly larger than the set of canonical k-sparse vectors.

There is also a subtle but important difference regarding the points where the curvature is evaluated at. Since Negahban et al. (2009) analyze a convex program, rather than an iterative algorithm, they only needed to invoke the RSC and RSS at a neighborhood of the true parameter. In contrast, the other variants of RSC and RSS (see, e.g., Agarwal et al., 2010; Jalali et al., 2011), as well as our SRH and SRL conditions, require the curvature bounds to hold uniformly over a larger set of points, thereby they are more stringent.

3.3 Main Theorems

Now we can state our main results regarding approximation of

$$\mathbf{x}^\star = \arg\min\ f(\mathbf{x}) \text{ s.t. } \|\mathbf{x}\|_0 \leq s, \tag{7}$$

using the GraSP algorithm.

Theorem 1. *Suppose that f is a twice continuously differentiable function that has μ_{4s}-SRH with $\mu_{4s} \leq \frac{1+\sqrt{3}}{2}$. Furthermore, suppose that for some $\varepsilon > 0$ we have $\varepsilon \leq B_{4s}(\mathbf{x})$ for all $4s$-sparse vectors $\mathbf{x}$. Then $\widehat{\mathbf{x}}^{(i)}$, the estimate at the i-th iteration, satisfies*

$$\left\|\widehat{\mathbf{x}}^{(i)} - \mathbf{x}^\star\right\|_2 \leq 2^{-i}\|\mathbf{x}^\star\|_2 + \frac{6+2\sqrt{3}}{\varepsilon}\left\|\nabla f(\mathbf{x}^\star)|_I\right\|_2,$$

where I is the position of the $3s$ largest entries of $\nabla f(\mathbf{x}^\star)$ in magnitude.

Remark 7. Note that this result indicates that $\nabla f(\mathbf{x}^\star)$ determines how accurate the estimate can be. In particular, if the sparse minimum $\mathbf{x}^\star$ is sufficiently close to an unconstrained minimum of f then the estimation error floor is negligible because $\nabla f(\mathbf{x}^\star)$ has small magnitude. This result is analogous to accuracy guarantees for estimation from noisy measurements in CS (Candès et al., 2006; Needell and Tropp, 2009).

Remark 8. As the derivations required to prove Theorem 1 show, the provided accuracy guarantee holds for any s-sparse $\mathbf{x}^\star$, even if it does not obey (7). Obviously, for arbitrary choices of $\mathbf{x}^\star$, $\nabla f(\mathbf{x}^\star)|_I$ may have a large norm that cannot be bounded properly which implies large errors. In statistical estimation problems, often the true parameter that describes the data is chosen as the target parameter $\mathbf{x}^\star$ rather than the minimizer of the average loss function as in (7). In these problems, the approximation error $\|\nabla f(\mathbf{x}^\star)|_I\|_2$ has statistical interpretation and can determine the statistical precision of the problem. This property is easy to verify in linear regression problems. We will also show this for the logistic loss as an example in Section 4.

Nonsmooth cost functions should be treated differently, since we do not have the luxury of working with Hessian matrices for these type of functions. The following theorem provides guarantees that are similar to those of Theorem 1 for nonsmooth cost functions that satisfy the SRL condition.

Theorem 2. *Suppose that f is a function that is not necessarily smooth, but it satisfies μ_{4s}-SRL with $\mu_{4s} \leq \frac{3+\sqrt{3}}{4}$. Furthermore, suppose that for $\beta_{4s}(\cdot)$ in Definition 3 there exists some $\varepsilon > 0$ such that $\beta_{4s}(\mathbf{x}) \geq \varepsilon$ holds for all $4s$-sparse vectors $\mathbf{x}$. Then $\widehat{\mathbf{x}}^{(i)}$, the estimate at the i-th iteration, satisfies*

$$\left\|\widehat{\mathbf{x}}^{(i)} - \mathbf{x}^\star\right\|_2 \leq 2^{-i}\|\mathbf{x}^\star\|_2 + \frac{6+2\sqrt{3}}{\varepsilon}\left\|\nabla_f(\mathbf{x}^\star)\big|_I\right\|_2,$$

where I is the position of the $3s$ largest entries of $\nabla_f(\mathbf{x}^\star)$ in magnitude.

Remark 9. Should the SRH or SRL conditions hold for the objective function, it is straightforward to convert the *point accuracy* guarantees of Theorems 1 and 2, into accuracy guarantees in terms of the objective value. First we can use SRH or SRL to bound the Bregman divergence, or its restricted version defined above, for points $\widehat{\mathbf{x}}^{(i)}$ and $\mathbf{x}^\star$. Then we can obtain a bound for the accuracy of the objective value by invoking the results of the theorems. This indirect approach, however, might not lead to sharp bounds and thus we do not pursue the detailed analysis in this work.

4. Example: Sparse Minimization of ℓ_2-regularized Logistic Regression

One of the models widely used in machine learning and statistics is the logistic model. In this model the relation between the data, represented by a random vector $\mathbf{a} \in \mathbb{R}^p$, and its associated label, represented by a random binary variable $y \in \{0,1\}$, is determined by the conditional probability

$$\Pr\{y \mid \mathbf{a}; \mathbf{x}\} = \frac{\exp(y \langle \mathbf{a}, \mathbf{x} \rangle)}{1 + \exp(\langle \mathbf{a}, \mathbf{x} \rangle)}, \tag{8}$$

where $\mathbf{x}$ denotes a parameter vector. Then, for a set of n independently drawn data samples $\{(\mathbf{a}_i, y_i)\}_{i=1}^n$ the joint likelihood can be written as a function of $\mathbf{x}$. To find the maximum likelihood estimate one should maximize this likelihood function, or equivalently minimize the negative log-likelihood, the logistic loss,

$$g(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(\langle \mathbf{a}_i, \mathbf{x} \rangle)) - y_i \langle \mathbf{a}_i, \mathbf{x} \rangle .$$

It is well-known that $g(\cdot)$ is strictly convex for $p \leq n$ provided that the associated design matrix, $\mathbf{A} = [\mathbf{a}_1 \, \mathbf{a}_2 \, \ldots \, \mathbf{a}_n]^{\mathrm{T}}$, is full-rank. However, in many important applications (e.g., feature selection) the problem can be underdetermined (i.e., $n < p$). In these scenarios the logistic loss is merely convex and it does not have a unique minimum. Furthermore, it is possible, especially in underdetermined problems, that the observed data is *linearly separable*. In that case one can achieve arbitrarily small loss values by tending the parameters to infinity along certain directions. To compensate for these drawbacks the logistic loss is usually regularized by some penalty term (Hastie et al., 2009; Bunea, 2008).

One of the candidates for the penalty function is the (squared) ℓ_2-norm of $\mathbf{x}$ (i.e., $\|\mathbf{x}\|_2^2$). Considering a positive penalty coefficient η the regularized loss is

$$f(\mathbf{x}) = g(\mathbf{x}) + \frac{\eta}{2} \|\mathbf{x}\|_2^2 .$$

For any convex $g(\cdot)$ this regularized loss is guaranteed to be η-strongly convex, thus it has a unique minimum. Furthermore, the penalty term implicitly bounds the length of the minimizer thereby resolving the aforementioned problems. Nevertheless, the ℓ_2 penalty does not promote sparse solutions. Therefore, it is often desirable to impose an explicit sparsity constraint, in addition to the ℓ_2 regularizer.

4.1 Verifying SRH for ℓ_2-regularized Logistic Loss

It is easy to show that the Hessian of the logistic loss at any point $\mathbf{x}$ is given by $\mathbf{H}_g(\mathbf{x}) = \frac{1}{4n} \mathbf{A}^{\mathrm{T}} \Lambda \mathbf{A}$, where Λ is an $n \times n$ diagonal matrix whose diagonal entries are $\Lambda_{ii} = \operatorname{sech}^2 \frac{1}{2} \langle \mathbf{a}_i, \mathbf{x} \rangle$ with $\operatorname{sech}(\cdot)$ denoting the *hyperbolic secant* function. Note that $\mathbf{0} \preccurlyeq \mathbf{H}_g(\mathbf{x}) \preccurlyeq \frac{1}{4n} \mathbf{A}^{\mathrm{T}} \mathbf{A}$. Therefore, if $\mathbf{H}_\eta(\mathbf{x})$ denotes the Hessian of the ℓ_2-regularized logistic loss, we have

$$\forall \mathbf{x}, \Delta \qquad \eta \|\Delta\|_2^2 \leq \Delta^{\mathrm{T}} \mathbf{H}_\eta(\mathbf{x}) \Delta \leq \frac{1}{4n} \|\mathbf{A}\Delta\|_2^2 + \eta \|\Delta\|_2^2 . \tag{9}$$

To verify SRH, the upper and lower bounds achieved at k-sparse vectors Δ are of particular interest. It only remains to find an appropriate upper bound for $\|\mathbf{A}\Delta\|_2^2$ in terms of $\|\Delta\|_2^2$. To this end we use the following result on Chernoff bounds for random matrices due to Tropp (2012).

Theorem 3 (Matrix Chernoff (Tropp, 2012)). *Consider a finite sequence* $\{\mathbf{M}_i\}$ *of* $k \times k$*, independent, random, self-adjoint matrices that satisfy*

$$\mathbf{M}_i \succcurlyeq \mathbf{0} \quad \textit{and} \quad \lambda_{\max}(\mathbf{M}_i) \leq R \quad \textit{almost surely.}$$

Let $\theta_{\max} := \lambda_{\max}\left(\sum_i \mathbb{E}[\mathbf{M}_i]\right)$*. Then for* $\tau \geq 0$*,*

$$\Pr\left\{\lambda_{\max}\left(\sum_i \mathbf{M}_i\right) \geq (1+\tau)\,\theta_{\max}\right\} \leq k \exp\left(\frac{\theta_{\max}}{R}\left(\tau - (1+\tau)\log(1+\tau)\right)\right).$$

As stated before, in a standard logistic model data samples $\{\mathbf{a}_i\}$ are supposed to be independent instances of a random vector $\mathbf{a}$. In order to apply Theorem 3 we need to make the following extra assumptions:

Assumption. For every $\mathcal{J} \subseteq [p]$ with $|\mathcal{J}| = k$,

(i) we have $\left\|\mathbf{a}|_{\mathcal{J}}\right\|_2^2 \leq R$ almost surely, and

(ii) none of the matrices $\mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbb{E}\left[\mathbf{a}\mathbf{a}^{\mathrm{T}}\right]\mathbf{P}_{\mathcal{J}}$ is the zero matrix.

We define $\theta_{\max}^{\mathcal{J}} := \lambda_{\max}\left(\mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbf{C}\mathbf{P}_{\mathcal{J}}\right)$, where $\mathbf{C} = \mathbb{E}\left[\mathbf{a}\mathbf{a}^{\mathrm{T}}\right]$, and let

$$\overline{\theta} := \max_{\mathcal{J}\subseteq[p]\,,\,|\mathcal{J}|=k} \theta_{\max}^{\mathcal{J}} \quad \text{and} \quad \widetilde{\theta} := \min_{\mathcal{J}\subseteq[p]\,,\,|\mathcal{J}|=k} \theta_{\max}^{\mathcal{J}}.$$

To simplify the notation henceforth we let $h(\tau) = (1+\tau)\log(1+\tau) - \tau$.

Corollary 1. *With the above assumptions, if* $n \geq \frac{R}{\widetilde{\theta}h(\tau)}\left(\log k + k\left(1+\log\frac{p}{k}\right) - \log\varepsilon\right)$ *for some* $\tau > 0$ *and* $\varepsilon \in (0,1)$*, then with probability at least* $1-\varepsilon$ *the* ℓ_2*-regularized logistic loss has* μ_k*-SRH with* $\mu_k \leq 1 + \frac{1+\tau}{4\eta}\overline{\theta}$*.*

Proof For any set of k indices $\mathcal{J}$ let $\mathbf{M}_i^{\mathcal{J}} = \mathbf{a}_i|_{\mathcal{J}}\,\mathbf{a}_i|_{\mathcal{J}}^{\mathrm{T}} = \mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbf{a}_i\mathbf{a}_i^{\mathrm{T}}\mathbf{P}_{\mathcal{J}}$. The independence of the vectors $\mathbf{a}_i$ implies that the matrix

$$\begin{aligned}\mathbf{A}_{\mathcal{J}}^{\mathrm{T}}\mathbf{A}_{\mathcal{J}} &= \sum_{i=1}^n \mathbf{a}_i|_{\mathcal{J}}\,\mathbf{a}_i|_{\mathcal{J}}^{\mathrm{T}} \\ &= \sum_{i=1}^n \mathbf{M}_i^{\mathcal{J}}\end{aligned}$$

is a sum of n independent, random, self-adjoint matrices. Assumption (i) implies that $\lambda_{\max}\left(\mathbf{M}_i^{\mathcal{J}}\right) = \left\|\mathbf{a}_i|_{\mathcal{J}}\right\|_2^2 \leq R$ almost surely. Furthermore, we have

$$\begin{aligned}\lambda_{\max}\left(\sum_{i=1}^n \mathbb{E}\left[\mathbf{M}_i^{\mathcal{J}}\right]\right) &= \lambda_{\max}\left(\sum_{i=1}^n \mathbb{E}\left[\mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbf{a}_i\mathbf{a}_i^{\mathrm{T}}\mathbf{P}_{\mathcal{J}}\right]\right) \\ &= \lambda_{\max}\left(\sum_{i=1}^n \mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbb{E}\left[\mathbf{a}_i\mathbf{a}_i^{\mathrm{T}}\right]\mathbf{P}_{\mathcal{J}}\right) \\ &= \lambda_{\max}\left(\sum_{i=1}^n \mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbf{C}\mathbf{P}_{\mathcal{J}}\right) \\ &= n\lambda_{\max}\left(\mathbf{P}_{\mathcal{J}}^{\mathrm{T}}\mathbf{C}\mathbf{P}_{\mathcal{J}}\right) \\ &= n\theta_{\max}^{\mathcal{J}}.\end{aligned}$$

Hence, for any fixed index set $\mathcal{J}$ with $|\mathcal{J}| = k$ we may apply Theorem 3 for $\mathbf{M}_i = \mathbf{M}_i^{\mathcal{J}}$, $\theta_{\max} = n\theta_{\max}^{\mathcal{J}}$, and $\tau > 0$ to obtain

$$\Pr\left\{\lambda_{\max}\left(\sum_{i=1}^{n}\mathbf{M}_i^{\mathcal{J}}\right) \geq (1+\tau)\, n\theta_{\max}^{\mathcal{J}}\right\} \leq k\exp\left(-\frac{n\theta_{\max}^{\mathcal{J}} h(\tau)}{R}\right).$$

Furthermore, we can write

$$\begin{aligned}
\Pr\left\{\lambda_{\max}\left(\mathbf{A}_{\mathcal{J}}^{\mathrm{T}}\mathbf{A}_{\mathcal{J}}\right) \geq (1+\tau)\, n\overline{\theta}\right\} &= \Pr\left\{\lambda_{\max}\left(\sum_{i=1}^{n}\mathbf{M}_i^{\mathcal{J}}\right) \geq (1+\tau)\, n\overline{\theta}\right\} \\
&\leq \Pr\left\{\lambda_{\max}\left(\sum_{i=1}^{n}\mathbf{M}_i^{\mathcal{J}}\right) \geq (1+\tau)\, n\theta_{\max}^{\mathcal{J}}\right\} \\
&\leq k\exp\left(-\frac{n\theta_{\max}^{\mathcal{J}} h(\tau)}{R}\right) \\
&\leq k\exp\left(-\frac{n\widetilde{\theta} h(\tau)}{R}\right). \qquad (10)
\end{aligned}$$

Note that Assumption (ii) guarantees that $\widetilde{\theta} > 0$, and thus the above probability bound will not be vacuous for sufficiently large n. To ensure a uniform guarantee for all $\binom{p}{k}$ possible choices of $\mathcal{J}$ we can use the union bound to obtain

$$\begin{aligned}
\Pr\left\{\bigvee_{\substack{\mathcal{J}\subseteq[p]\\|\mathcal{J}|=k}} \lambda_{\max}\left(\mathbf{A}_{\mathcal{J}}^{\mathrm{T}}\mathbf{A}_{\mathcal{J}}\right) \geq (1+\tau)\, n\overline{\theta}\right\} &\leq \sum_{\substack{\mathcal{J}\subseteq[p]\\|\mathcal{J}|=k}} \Pr\left\{\lambda_{\max}\left(\mathbf{A}_{\mathcal{J}}^{\mathrm{T}}\mathbf{A}_{\mathcal{J}}\right) \geq (1+\tau)\, n\overline{\theta}\right\} \\
&\leq k\binom{p}{k}\exp\left(-\frac{n\widetilde{\theta} h(\tau)}{R}\right) \\
&\leq k\left(\frac{pe}{k}\right)^{k}\exp\left(-\frac{n\widetilde{\theta} h(\tau)}{R}\right) \\
&= \exp\left(\log k + k + k\log\frac{p}{k} - \frac{n\widetilde{\theta} h(\tau)}{R}\right).
\end{aligned}$$

Therefore, for $\varepsilon \in (0,1)$ and $n \geq R\left(\log k + k\left(1+\log\frac{p}{k}\right) - \log\varepsilon\right) / \left(\widetilde{\theta} h(\tau)\right)$ it follows from (9) that for any $\mathbf{x}$ and any k-sparse Δ,

$$\eta\,\|\Delta\|_2^2 \leq \Delta^{\mathrm{T}}\mathbf{H}_{\eta}(\mathbf{x})\,\Delta \leq \left(\eta + \frac{1+\tau}{4}\overline{\theta}\right)\|\Delta\|_2^2$$

holds with probability at least $1-\varepsilon$. Thus, the ℓ_2-regularized logistic loss has an SRH constant $\mu_k \leq 1 + \frac{1+\tau}{4\eta}\overline{\theta}$ with probability $1-\varepsilon$. ■

Remark 10. One implication of this result is that for a regime in which k and p grow sufficiently large while $\frac{p}{k}$ remains constant one can achieve small failure rates provided that $n = \Omega\left(Rk \log \frac{p}{k}\right)$. Note that R is deliberately included in the argument of the order function because in general R depends on k. In other words, the above analysis may require $n = \Omega\left(k^2 \log \frac{p}{k}\right)$ as the sufficient number of observations. This bound is a consequence of using Theorem 3, but to the best of our knowledge, other results regarding the extreme eigenvalues of the average of independent random PSD matrices also yield an n of the same order. If matrix $\mathbf{A}$ has certain additional properties (e.g., independent and sub-Gaussian entries), however, a better rate of $n = \Omega\left(k \log \frac{p}{k}\right)$ can be achieved without using the techniques mentioned above.

Remark 11. The analysis provided here is not specific to the ℓ_2-regularized logistic loss and can be readily extended to any other ℓ_2-regularized GLM loss whose log-partition function has a Lipschitz-continuous derivative.

4.2 Bounding the Approximation Error

We are going to bound $\|\nabla f(\mathbf{x}^\star)|_I\|_2$ which controls the approximation error in the statement of Theorem 1. In the case of case of ℓ_2-regularized logistic loss considered in this section we have

$$\nabla f(\mathbf{x}) = \sum_{i=1}^{n} \left(\frac{1}{1+\exp\left(-\langle \mathbf{a}_i, \mathbf{x}\rangle\right)} - y_i \right) \mathbf{a}_i + \eta \mathbf{x}.$$

Denoting $\frac{1}{1+\exp(-\langle \mathbf{a}_i, \mathbf{x}^\star\rangle)} - y_i$ by v_i for $i = 1, 2, \ldots, n$ then we can deduce

$$\begin{aligned}
\|\nabla f(\mathbf{x}^\star)|_I\|_2 &= \left\| \frac{1}{n} \sum_{i=1}^{n} v_i \, \mathbf{a}_i|_I + \eta \, \mathbf{x}^\star|_I \right\|_2 \\
&= \left\| \frac{1}{n} \mathbf{A}_I^{\mathrm{T}} \mathbf{v} + \eta \, \mathbf{x}^\star|_I \right\|_2 \\
&\leq \frac{1}{n} \left\|\mathbf{A}_I^{\mathrm{T}}\right\| \|\mathbf{v}\|_2 + \eta \, \|\mathbf{x}^\star|_I\|_2 \\
&\leq \frac{1}{\sqrt{n}} \|\mathbf{A}_I\| \sqrt{\frac{1}{n} \sum_{i=1}^{n} v_i^2} + \eta \, \|\mathbf{x}^\star|_I\|_2 ,
\end{aligned}$$

where $\mathbf{v} = [v_1 \, v_2 \ldots v_n]^{\mathrm{T}}$. Note that v_i's are n independent copies of the random variable $v = \frac{1}{1+\exp(-\langle \mathbf{a}, \mathbf{x}^\star\rangle)} - y$ that is zero-mean and always lie in the interval $[-1, 1]$. Therefore, applying the Hoeffding's inequality yields

$$\Pr\left\{ \frac{1}{n} \sum_{i=1}^{n} v_i^2 \geq (1+c)\, \sigma_v^2 \right\} \leq \exp\left(-2nc^2 \sigma_v^4\right),$$

where $\sigma_v^2 = \mathbb{E}\left[v^2\right]$ is the variance of v. Furthermore, using the logistic model (8) we can deduce

$$\begin{aligned}
\sigma_v^2 &= \mathbb{E}\left[v^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[v^2 \mid \mathbf{a}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(y - \mathbb{E}\left[y \mid \mathbf{a}\right]\right)^2 \mid \mathbf{a}\right]\right] \\
&= \mathbb{E}\left[\operatorname{var}\left(y \mid \mathbf{a}\right)\right] \\
&= \mathbb{E}\left[\frac{1}{1+\exp\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle\right)} \times \frac{\exp\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle\right)}{1+\exp\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle\right)}\right] && \text{(because } y \mid \mathbf{a} \sim \text{Bernoulli as in (8))} \\
&= \mathbb{E}\left[\frac{1}{2+\exp\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle\right)+\exp\left(-\langle \mathbf{a}, \mathbf{x}^\star \rangle\right)}\right] \\
&\leq \frac{1}{4} && \text{(because } \exp(t)+\exp(-t) \geq 2\text{).}
\end{aligned}$$

Therefore, we have $\frac{1}{n}\sum_{i=1}^n v_i^2 < \frac{1}{4}$ with high probability. As in the previous subsection one can also bound $\frac{1}{\sqrt{n}}\|\mathbf{A}_I\| = \sqrt{\frac{1}{n}\lambda_{\max}\left(\mathbf{A}_I^{\mathrm{T}}\mathbf{A}_I\right)}$ using (10) with $k = |I| = 3s$. Hence, with high probability we have

$$\|\nabla f(\mathbf{x}^\star)|_I\|_2 \leq \frac{1}{2}\sqrt{(1+\tau)\,\overline{\theta}} + \eta\,\|\mathbf{x}^\star\|_2 .$$

Interestingly, this analysis can also be extended to the GLMs whose log-partition function $\psi(\cdot)$ obeys $0 \leq \psi''(t) \leq C$ for all t with C being a positive constant. For these models the approximation error can be bounded in terms of the variance of $v_\psi = \psi'\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle\right) - y$.

5. Experimental Results

Algorithms that are used for sparsity-constrained estimation or optimization often induce sparsity using different types of regularizations or constraints. Therefore, the *optimized* objective function may vary from one algorithm to another, even though all of these algorithms try to estimate the same sparse parameter and sparsely optimize the same original objective. Because of the discrepancy in the optimized objective functions it is generally difficult to compare performance of these algorithms. Applying algorithms on real data generally produces even less reliable results because of the unmanageable or unknown characteristics of the real data. Nevertheless, we evaluated performance of GraSP for variable selection in the logistic model both on synthetic and real data.

5.1 Synthetic Data

In our simulations the sparse parameter of interest $\mathbf{x}^\star$ is a $p = 1000$ dimensional vector that has $s = 10$ nonzero entries drawn independently from the standard Gaussian distribution. An intercept $c \in \mathbb{R}$ is also considered which is drawn independently of the other parameters according to the standard Gaussian distribution. Each data sample is an independent instance of the random vector $\mathbf{a} = [a_1, a_2, \ldots, a_p]^{\mathrm{T}}$ generated by an autoregressive process (Hamilton, 1994) determined by

$$a_{j+1} = \rho a_j + \sqrt{1-\rho^2}\, z_j, \qquad \text{for all } j \in [p-1]$$

with $a_1 \sim \mathcal{N}(0,1)$, $z_j \sim \mathcal{N}(0,1)$, and $\rho \in [0,1]$ being the correlation parameter. The data model we describe and use above is identical to the experimental model used in Agarwal et al. (2010), except that we adjusted the coefficients to ensure that $\mathbb{E}\left[a_j^2\right] = 1$ for all $j \in [p]$. The data labels, $y \in \{0,1\}$ are then drawn randomly according to the Bernoulli distribution with

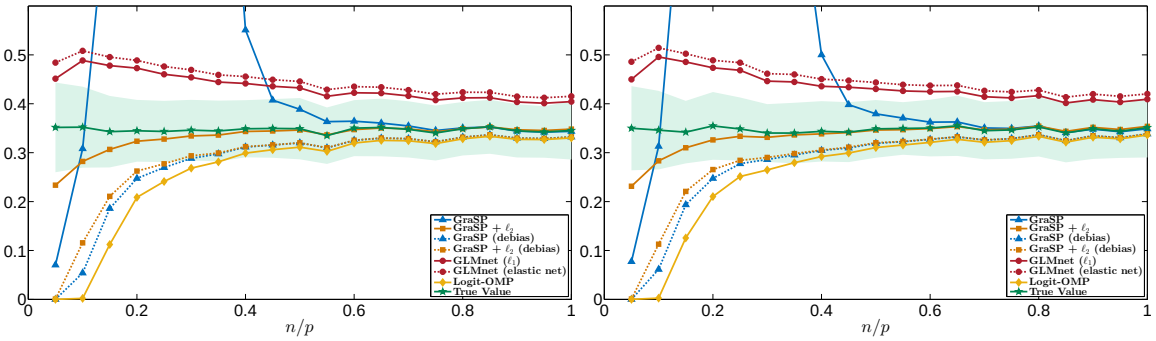
$$\Pr\{y = 0 \mid \mathbf{a}\} = 1/\left(1 + \exp\left(\langle \mathbf{a}, \mathbf{x}^\star \rangle + c\right)\right).$$

We compared GraSP to the LASSO algorithm implemented in the GLMnet package (Friedman et al., 2010), as well as the Orthogonal Matching Pursuit method dubbed Logit-OMP (Lozano et al., 2011). To isolate the effect of ℓ_2-regularization, both LASSO and the basic implementation of GraSP did not consider additional ℓ_2-regularization terms. To analyze the effect of an additional ℓ_2-regularization we also evaluated the performance of GraSP with ℓ_2-regularized logistic loss, as well as the logistic regression with elastic net (i.e., mixed ℓ_1-ℓ_2) penalty also available in the GLMnet package. We configured the GLMnet software to produce s-sparse solutions for a fair comparison. For the elastic net penalty $(1-\omega)\|\mathbf{x}\|_2^2/2 + \omega\|\mathbf{x}\|_1$ we considered the "mixing parameter" ω to be 0.8. For the ℓ_2-regularized logistic loss we considered $\eta = (1-\omega)\sqrt{\frac{\log p}{n}}$. For each choice of the number of measurements n between 50 and 1000 in steps of size 50, and ρ in the set $\left\{0, \frac{1}{3}, \frac{1}{2}, \frac{\sqrt{2}}{2}\right\}$ we generate the data and the associated labels and apply the algorithms. The average performance is measured over 200 trials for each pair of (n, ρ).

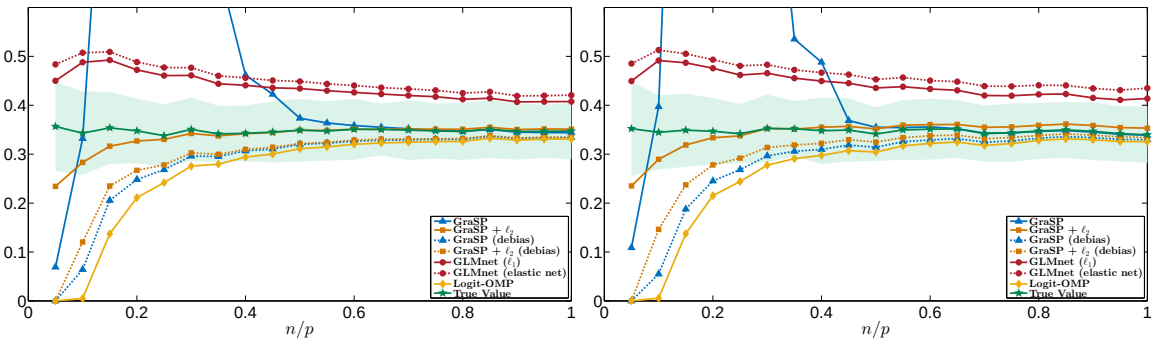
Figure 1 compares the average value of the empirical logistic loss achieved by each of the considered algorithms for a wide range of "sampling ratio" n/p. For GraSP, the curves labelled by GraSP and GraSP + ℓ_2 corresponding to the cases where the algorithm is applied to unregularized and ℓ_2-regularized logistic loss, respectively. Furthermore, the results of GLMnet for the LASSO and the elastic net regularization are labelled by GLMnet (ℓ_1) and GLMnet (elastic net), respectively. The simulation result of the Logit-OMP algorithm is also included. To contrast the obtained results we also provided the average of empirical logistic loss evaluated at the true parameter and one standard deviation above and below this average on the plots. Furthermore, we evaluated performance of GraSP with the debiasing procedure described in Section 3.1.

As can be seen from the figure at lower values of the sampling ratio GraSP is not accurate and does not seem to be converging. This behavior can be explained by the fact that without regularization at low sampling ratios the training data is linearly separable or has very few mislabelled samples. In either case, the value of the loss can vary significantly even in small neighborhoods. Therefore, the algorithm can become too sensitive to the pruning step at the end of each iteration. At larger sampling ratios, however, the loss from GraSP begins to decrease rapidly, becoming effectively identical to the loss at the true parameter for $n/p > 0.7$. The results show that unlike GraSP, Logit-OMP performs gracefully at lower sampling ratios. At higher sampling ratios, however, GraSP appears to yield smaller bias in the loss value. Furthermore, the difference between the loss obtained by the LASSO and the loss at the true parameter never drops below a certain threshold, although the convex method exhibits a more stable behaviour at low sampling ratios.

Interestingly, GraSP becomes more stable at low sampling ratios when the logistic loss is regularized with the ℓ_2-norm. However, this stability comes at the cost of a bias in the loss value at high sampling ratios that is particularly pronounced in Figure 1d. Nevertheless, for all of the tested values of ρ, at low sampling ratios GraSP+ℓ_2 and at high sampling ratios GraSP are consistently closer to the true loss value compared to the other methods. Debiasing the iterates of GraSP also



(a) $\rho = 0$ (b) $\rho = \frac{1}{3}$

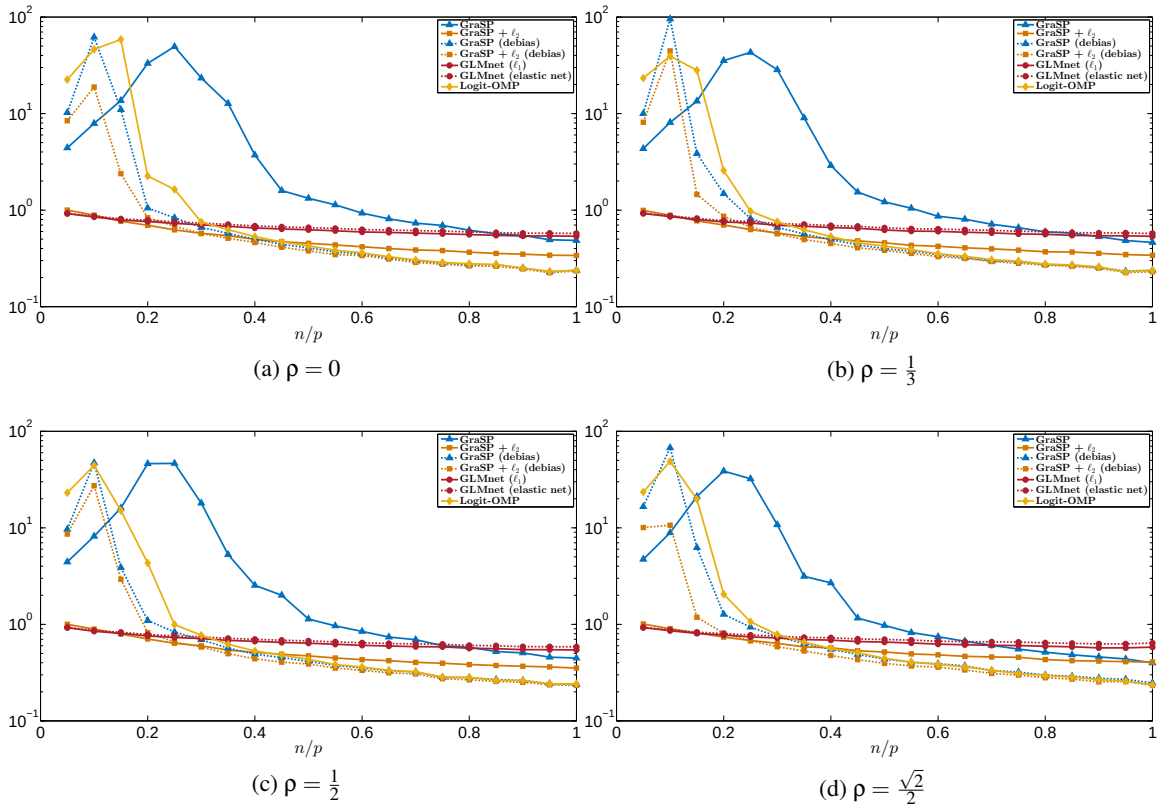


(c) $\rho = \frac{1}{2}$ (d) $\rho = \frac{\sqrt{2}}{2}$

Figure 1: Comparison of the average (empirical) logistic loss at solutions obtained via GraSP, GraSP with ℓ_2-penalty, LASSO, the elastic-net regularization, and Logit-OMP. The results of both GraSP methods with "debiasing" are also included. The average loss at the true parameter and one standard deviation interval around it are plotted as well.

appears to have a stabilizing effect at lower sampling ratios. For GraSP with ℓ_2 regularized cost, the debiasing particularly reduced the undesirable bias at $\rho = \frac{\sqrt{2}}{2}$.

Figure 2 illustrates the performance of the same algorithms in terms of the relative error $\frac{\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2}$ where $\widehat{\mathbf{x}}$ denotes the estimate that the algorithms produce. Not surprisingly, none of the algorithms attain an arbitrarily small relative error. Furthermore, the parameter ρ does not appear to affect the performance of the algorithms significantly. Without the ℓ_2-regularization, at high sampling ratios GraSP provides an estimate that has a comparable error versus the ℓ_1-regularization method. However, for mid to high sampling ratios both GraSP and GLMnet methods are outperformed by Logit-OMP. At low to mid sampling ratios, GraSP is unstable and does not converge to an estimate close to the true parameter. Logit-OMP shows similar behavior at lower sampling ratios. Performance of GraSP changes dramatically once we consider the ℓ_2-regularization and/or the debiasing procedure. With ℓ_2-regularization, GraSP achieves better relative error compared to GLMnet and ordinary GraSP for almost the entire range of tested sampling ratios. Applying the debiasing procedure has improved the performance of both GraSP methods except at very low sampling ratios.



(a) $\rho = 0$ (b) $\rho = \frac{1}{3}$

(c) $\rho = \frac{1}{2}$ (d) $\rho = \frac{\sqrt{2}}{2}$

Figure 2: Comparison of the average relative error (i.e., $\frac{\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2}$) in logarithmic scale at solutions obtained via GraSP, GraSP with ℓ_2-penalty, LASSO, the elastic-net regularization, and Logit-OMP. The results of both GraSP methods with "debiasing" are also included.

These variants of GraSP appear to perform better than Logit-OMP for almost the entire range of n/p.

5.2 Real Data

We also conducted the same simulation on some of the data sets used in NIPS 2003 Workshop on feature extraction (Guyon et al., 2005), namely the ARCENE and DEXTER data sets. The logistic loss values at obtained estimates are reported in Tables 2 and 3. For each data set we applied the sparse logistic regression for a range of sparsity level s. The columns indicated by "G" correspond to different variants of GraSP. Suffixes ℓ_2 and "d" indicate the ℓ_2-regularization and the debiasing are applied, respectively. The columns indicated by ℓ_1 and E-net correspond to the results of the ℓ_1-regularization and the elastic-net regularization methods that are performed using the GLMnet package. The last column contains the result of the Logit-OMP algorithm.

The results for DEXTER data set show that GraSP variants without debiasing and the convex methods achieve comparable loss values in most cases, whereas the convex methods show significantly better performance on the ARCENE data set. Nevertheless, except for a few instances where

s	G	Gd	Gℓ_2	Gℓ_2d	ℓ_1	E-net	Logit-OMP
5	5.89E+01	5.75E-01	2.02E+01	5.24E-01	5.59E-01	6.43E-01	**2.23E-01**
10	3.17E+02	5.43E-01	3.71E+01	4.53E-01	5.10E-01	5.98E-01	**5.31E-07**
15	3.38E+02	6.40E-07	5.94E+00	**1.42E-07**	4.86E-01	5.29E-01	5.31E-07
20	1.21E+02	3.44E-07	8.82E+00	**3.08E-08**	4.52E-01	5.19E-01	5.31E-07
25	9.87E+02	1.13E-07	4.46E+01	**1.35E-08**	4.18E-01	4.96E-01	5.31E-07

Table 2: ARCENE

s	G	Gd	Gℓ_2	Gℓ_2d	ℓ_1	E-net	Logit-OMP
5	7.58E+00	3.28E-01	3.30E+00	2.80E-01	5.75E-01	6.08E-01	**2.64E-01**
10	1.08E+00	1.79E-01	4.33E-01	**1.28E-01**	5.23E-01	5.33E-01	1.79E-01
15	6.06E+00	1.71E-01	3.35E-01	1.17E-01	4.88E-01	4.98E-01	**1.16E-01**
20	1.30E+00	8.84E-02	1.79E-01	8.19E-02	4.27E-01	4.36E-01	**4.60E-02**
25	1.17E+00	**2.51E-07**	2.85E-01	1.17E-02	3.94E-01	4.12E-01	4.62E-03
30	3.04E-01	5.83E-07	2.65E-01	**1.77E-07**	3.70E-01	3.88E-01	2.88E-07
35	6.22E-01	2.08E-07	2.68E-01	**1.19E-07**	3.47E-01	3.72E-01	2.14E-07
40	5.38E-01	2.01E-07	6.30E-02	**1.27E-07**	3.31E-01	3.56E-01	2.14E-07
45	3.29E-01	2.11E-07	1.05E-01	**1.47E-07**	3.16E-01	3.41E-01	2.14E-07
50	2.06E-01	**1.31E-07**	5.66E-02	1.46E-07	2.87E-01	3.11E-01	2.14E-07
55	3.61E-02	**1.20E-07**	8.40E-02	1.31E-07	2.80E-01	2.89E-01	2.14E-07
60	1.18E-01	2.46E-07	5.70E-02	**1.09E-07**	2.66E-01	2.82E-01	2.14E-07
65	1.18E-01	**7.86E-08**	2.87E-02	9.47E-08	2.59E-01	2.75E-01	2.14E-07
70	8.92E-02	1.17E-07	2.23E-02	**8.15E-08**	2.52E-01	2.69E-01	2.14E-07
75	1.03E-01	8.54E-08	3.93E-02	**7.94E-08**	2.45E-01	2.69E-01	2.14E-07

Table 3: DEXTER

Logit-OMP has the best performance, the smallest loss values in both data sets are attained by GraSP methods with debiasing step.

6. Discussion and Conclusion

In many applications understanding high dimensional data or systems that involve these types of data can be reduced to identification of a sparse parameter. For example, in gene selection problems researchers are interested in locating a few genes among thousands of genes that cause or contribute to a particular disease. These problems can usually be cast as sparsity-constrained optimizations. In this paper we introduce a greedy algorithm called the Gradient Support Pursuit(GraSP) as an approximate solver for a wide range of sparsity-constrained optimization problems.

We provide theoretical convergence guarantees based on the notions of a Stable Restricted Hessian (SRH) for smooth cost functions and a Stable Restricted Linearization (SRL) for non-smooth cost functions, both of which are introduced in this paper. Our algorithm generalizes the well-established sparse recovery algorithm CoSaMP that merely applies in linear models with squared error loss. The SRH and SRL also generalize the well-known Restricted Isometry Property for

sparse recovery to the case of cost functions other than the squared error. To provide a concrete example we studied the requirements of GraSP for ℓ_2-regularized logistic loss. Using a similar approach one can verify SRH condition for loss functions that have Lipschitz-continuous gradient that incorporates a broad family of loss functions.

At medium- and large-scale problems computational cost of the GraSP algorithm is mostly affected by the inner convex optimization step whose complexity is polynomial in s. On the other hand, for very large-scale problems, especially with respect to the dimension of the input, p, the running time of the GraSP algorithm will be dominated by evaluation of the function and its gradient, whose computational cost grows with p. This problem is common in algorithms that only have deterministic steps; even ordinary coordinate-descent methods have this limitation (Nesterov, 2012). Similar to improvements gained by using randomization in coordinate-descent methods (Nesterov, 2012), introducing randomization in the GraSP algorithm could reduce its computational complexity at large-scale problems. This extension, however, is beyond the scope of this paper and we leave it for future work.

Appendix A. Iteration Analysis For Smooth Cost Functions

To analyze our algorithm we first establish a series of results on how the algorithm operates on its current estimate, leading to an iteration invariant property on the estimation error. Propositions 1 and 2 are used to prove Lemmas 1 and 2. These Lemmas then are used to prove Lemma 3 that provides an iteration invariant which in turn yields the main result.

Proposition 1. *Let $\mathbf{M}(t)$ be a matrix-valued function such that for all $t \in [0,1]$, $\mathbf{M}(t)$ is symmetric and its eigenvalues lie in interval $[B(t), A(t)]$ with $B(t) > 0$. Then for any vector $\mathbf{v}$ we have*

$$\left(\int_0^1 B(t)\mathrm{d}t\right) \|\mathbf{v}\|_2 \leq \left\| \left(\int_0^1 \mathbf{M}(t)\mathrm{d}t\right) \mathbf{v} \right\|_2 \leq \left(\int_0^1 A(t)\mathrm{d}t\right) \|\mathbf{v}\|_2 .$$

Proof Let $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalue functions defined over the set of symmetric positive-definite matrices, respectively. These functions are in order concave and convex. Therefore, Jensen's inequality yields

$$\lambda_{\min}\left(\int_0^1 \mathbf{M}(t)\mathrm{d}t\right) \geq \int_0^1 \lambda_{\min}(\mathbf{M}(t))\,\mathrm{d}t \geq \int_0^1 B(t)\mathrm{d}t$$

and

$$\lambda_{\max}\left(\int_0^1 \mathbf{M}(t)\mathrm{d}t\right) \leq \int_0^1 \lambda_{\max}(\mathbf{M}(t))\,\mathrm{d}t \leq \int_0^1 A(t)\mathrm{d}t,$$

which imply the desired result. ■

Proposition 2. *Let* $\mathbf{M}(t)$ *be a matrix-valued function such that for all* $t \in [0,1]$ $\mathbf{M}(t)$ *is symmetric and its eigenvalues lie in interval* $[B(t), A(t)]$ *with* $B(t) > 0$*. If* Γ *is a subset of row/column indices of* $\mathbf{M}(\cdot)$ *then for any vector* $\mathbf{v}$ *we have*

$$\left\| \left(\int_0^1 \mathbf{P}_\Gamma^{\mathrm{T}} \mathbf{M}(t) \mathbf{P}_{\Gamma^c} \mathrm{d}t \right) \mathbf{v} \right\|_2 \leq \int_0^1 \frac{A(t) - B(t)}{2} \mathrm{d}t \, \|\mathbf{v}\|_2 .$$

Proof Since $\mathbf{M}(t)$ is symmetric, it is also diagonalizable. Thus, for any vector $\mathbf{v}$ we may write

$$B(t) \|\mathbf{v}\|_2^2 \leq \mathbf{v}^{\mathrm{T}} \mathbf{M}(t) \mathbf{v} \leq A(t) \|\mathbf{v}\|_2^2,$$

and thereby

$$-\frac{A(t) - B(t)}{2} \leq \frac{\mathbf{v}^{\mathrm{T}} \left(\mathbf{M}(t) - \frac{A(t)+B(t)}{2} \mathbf{I} \right) \mathbf{v}}{\|\mathbf{v}\|^2} \leq \frac{A(t) - B(t)}{2}.$$

Since $\mathbf{M}(t) - \frac{A(t)+B(t)}{2}\mathbf{I}$ is also diagonalizable, it follows from the above inequality that

$$\left\| \mathbf{M}(t) - \frac{A(t) + B(t)}{2} \mathbf{I} \right\| \leq \frac{A(t) - B(t)}{2}.$$

Let $\widetilde{\mathbf{M}}(t) = \mathbf{P}_\Gamma^{\mathrm{T}} \mathbf{M}(t) \mathbf{P}_{\Gamma^c}$. Since $\widetilde{\mathbf{M}}(t)$ is a submatrix of $\mathbf{M}(t) - \frac{A(t)+B(t)}{2}\mathbf{I}$ we should have

$$\left\| \widetilde{\mathbf{M}}(t) \right\| \leq \left\| \mathbf{M}(t) - \frac{A(t) + B(t)}{2} \mathbf{I} \right\| \leq \frac{A(t) - B(t)}{2}. \tag{11}$$

Finally, it follows from the convexity of the operator norm, Jensen's inequality, and (11) that

$$\left\| \int_0^1 \widetilde{\mathbf{M}}(t) \, \mathrm{d}t \right\| \leq \int_0^1 \left\| \widetilde{\mathbf{M}}(t) \right\| \mathrm{d}t \leq \int_0^1 \frac{A(t) - B(t)}{2} \mathrm{d}t.$$

■

To simplify notation we introduce functions

$$\alpha_k(\mathbf{p}, \mathbf{q}) = \int_0^1 A_k(t\mathbf{q} + (1-t)\mathbf{p}) \, \mathrm{d}t$$
$$\beta_k(\mathbf{p}, \mathbf{q}) = \int_0^1 B_k(t\mathbf{q} + (1-t)\mathbf{p}) \, \mathrm{d}t$$
$$\gamma_k(\mathbf{p}, \mathbf{q}) = \alpha_k(\mathbf{p}, \mathbf{q}) - \beta_k(\mathbf{p}, \mathbf{q}),$$

where $A_k(\cdot)$ and $B_k(\cdot)$ are defined by (5) and (6), respectively.

Lemma 1. *Let $\mathcal{R}$ denote the set* $\operatorname{supp}(\widehat{\mathbf{x}}-\mathbf{x}^\star)$. *The current estimate* $\widehat{\mathbf{x}}$ *then satisfies*

$$\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2 \leq \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}.$$

Proof Since $\mathcal{Z}=\operatorname{supp}(\mathbf{z}_{2s})$ and $|\mathcal{R}|\leq 2s$ we have $\left\|\mathbf{z}|_{\mathcal{R}}\right\|_2 \leq \|\mathbf{z}|_{\mathcal{Z}}\|_2$ and thereby

$$\left\|\mathbf{z}|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \leq \left\|\mathbf{z}|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2. \tag{12}$$

Furthermore, because $\mathbf{z}=\nabla f(\widehat{\mathbf{x}})$ we can write

$$\begin{aligned}
\left\|\mathbf{z}|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 &\geq \left\|\nabla f(\widehat{\mathbf{x}})|_{\mathcal{R}\setminus\mathcal{Z}}-\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2-\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\
&= \left\|\left(\int_0^1 \mathbf{P}_{\mathcal{R}\setminus\mathcal{Z}}^{\mathrm{T}}\mathbf{H}_f(t\widehat{\mathbf{x}}+(1-t)\mathbf{x}^\star)\,\mathrm{d}t\right)(\widehat{\mathbf{x}}-\mathbf{x}^\star)\right\|_2-\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\
&\geq \left\|\left(\int_0^1 \mathbf{P}_{\mathcal{R}\setminus\mathcal{Z}}^{\mathrm{T}}\mathbf{H}_f(t\widehat{\mathbf{x}}+(1-t)\mathbf{x}^\star)\mathbf{P}_{\mathcal{R}\setminus\mathcal{Z}}\mathrm{d}t\right)(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2-\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\
&\quad -\left\|\left(\int_0^1 \mathbf{P}_{\mathcal{R}\setminus\mathcal{Z}}^{\mathrm{T}}\mathbf{H}_f(t\widehat{\mathbf{x}}+(1-t)\mathbf{x}^\star)\mathbf{P}_{\mathcal{Z}\cap\mathcal{R}}\mathrm{d}t\right)(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}\cap\mathcal{R}}\right\|_2,
\end{aligned}$$

where we split the active coordinates (i.e., $\mathcal{R}$) into the sets $\mathcal{R}\setminus\mathcal{Z}$ and $\mathcal{Z}\cap\mathcal{R}$ to apply the triangle inequality and obtain the last expression. Applying Propositions 1 and 2 yields

$$\begin{aligned}
\left\|\mathbf{z}|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 &\geq \beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2-\frac{\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}\cap\mathcal{R}}\right\|_2-\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\
&\geq \beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2-\frac{\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2-\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2.
\end{aligned} \tag{13}$$

Similarly, we have

$$\begin{aligned}
\left\|\mathbf{z}|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 &\leq \left\|\nabla f(\widehat{\mathbf{x}})|_{\mathcal{Z}\setminus\mathcal{R}}-\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 \\
&= \left\|\left(\int_0^1 \mathbf{P}_{\mathcal{Z}\setminus\mathcal{R}}^{\mathrm{T}}\mathbf{H}_f(t\widehat{\mathbf{x}}+(1-t)\mathbf{x}^\star)\mathbf{P}_{\mathcal{R}}\mathrm{d}t\right)(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 \\
&\leq \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 \\
&= \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2.
\end{aligned} \tag{14}$$

Combining (12), (13), and (14) we obtain

$$\begin{aligned}
\frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 &\geq \left\|\mathbf{z}|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 \\
&\geq \left\|\mathbf{z}|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\
&\geq \beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2-\frac{\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 \\
&\quad -\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2.
\end{aligned}$$

Since $\mathcal{R} = \operatorname{supp}(\widehat{\mathbf{x}}-\mathbf{x}^\star)$, we have $\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 = \|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2$. Hence,

$$\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2 \leq \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|}{\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}.$$

■

Lemma 2. *The vector* $\mathbf{b}$ *given by*

$$\mathbf{b} = \arg\min f(\mathbf{x}) \text{ s.t. } \mathbf{x}|_{\mathcal{T}^c} = 0 \tag{15}$$

satisfies

$$\|\mathbf{x}^\star|_{\mathcal{T}}-\mathbf{b}\|_2 \leq \frac{\|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}+\frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{2\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2.$$

Proof We have

$$\nabla f(\mathbf{x}^\star)-\nabla f(\mathbf{b}) = \int_0^1 \mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\,\mathrm{d}t\,(\mathbf{x}^\star-\mathbf{b}).$$

Furthermore, since $\mathbf{b}$ is the solution to (15) we must have $\nabla f(\mathbf{b})|_{\mathcal{T}} = 0$. Therefore,

$$\begin{aligned}\nabla f(\mathbf{x}^\star)|_{\mathcal{T}} &= \left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\,\mathrm{d}t\right)(\mathbf{x}^\star-\mathbf{b})\\ &= \left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\mathbf{P}_{\mathcal{T}}\,\mathrm{d}t\right)(\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}}\\ &+ \left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\mathbf{P}_{\mathcal{T}^c}\,\mathrm{d}t\right)(\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}^c}.\end{aligned} \tag{16}$$

Since f has μ_{4s}-SRH and $|\mathcal{T}\cup\operatorname{supp}(t\mathbf{x}^\star+(1-t)\mathbf{b})| \leq 4s$ for all $t\in[0,1]$, functions $A_{4s}(\cdot)$ and $B_{4s}(\cdot)$, defined using (5) and (6), exist such that we have

$$B_{4s}(t\mathbf{x}^\star+(1-t)\mathbf{b}) \leq \lambda_{\min}\left(\mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\mathbf{P}_{\mathcal{T}}\right)$$

and

$$A_{4s}(t\mathbf{x}^\star+(1-t)\mathbf{b}) \geq \lambda_{\max}\left(\mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\mathbf{P}_{\mathcal{T}}\right).$$

Thus, from Proposition 1 we obtain

$$\beta_{4s}(\mathbf{b},\mathbf{x}^\star) \leq \lambda_{\min}\left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\mathbf{b})\mathbf{P}_{\mathcal{T}}\,\mathrm{d}t\right)$$

and

$$\alpha_{4s}(\mathbf{b},\mathbf{x}^\star) \geq \lambda_{\max}\left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f\left(t\mathbf{x}^\star+(1-t)\,\mathbf{b}\right)\mathbf{P}_{\mathcal{T}}\mathrm{d}t\right).$$

This result implies that the matrix $\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f\left(t\mathbf{x}^\star+(1-t)\,\mathbf{b}\right)\mathbf{P}_{\mathcal{T}}\mathrm{d}t$, henceforth denoted by $\mathbf{W}$, is invertible and

$$\frac{1}{\alpha_{4s}(\mathbf{b},\mathbf{x}^\star)} \leq \lambda_{\min}\left(\mathbf{W}^{-1}\right) \leq \lambda_{\max}\left(\mathbf{W}^{-1}\right) \leq \frac{1}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}, \tag{17}$$

where we used the fact that $\lambda_{\max}(\mathbf{M})\,\lambda_{\min}\left(\mathbf{M}^{-1}\right)=1$ for any positive-definite matrix $\mathbf{M}$, particularly for $\mathbf{W}$ and $\mathbf{W}^{-1}$. Therefore, by multiplying both sides of (16) by $\mathbf{W}^{-1}$ obtain

$$\mathbf{W}^{-1}\nabla f(\mathbf{x}^\star)|_{\mathcal{T}} = (\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}} + \mathbf{W}^{-1}\left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\,\mathbf{b})\,\mathbf{P}_{\mathcal{T}^c}\mathrm{d}t\right)\mathbf{x}^\star|_{\mathcal{T}^c},$$

where we also used the fact that $(\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}^c} = \mathbf{x}^\star|_{\mathcal{T}^c}$. With $\mathcal{S}^\star=\mathrm{supp}(\mathbf{x}^\star)$, using triangle inequality, (17), and Proposition 2 then we obtain

$$\begin{aligned}
\|\mathbf{x}^\star|_{\mathcal{T}}-\mathbf{b}\|_2 &= \|(\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}}\|_2 \\
&\leq \left\|\mathbf{W}^{-1}\left(\int_0^1 \mathbf{P}_{\mathcal{T}}^{\mathrm{T}}\mathbf{H}_f(t\mathbf{x}^\star+(1-t)\,\mathbf{b})\,\mathbf{P}_{\mathcal{T}^c\cap\mathcal{S}^\star}\mathrm{d}t\right)\mathbf{x}^\star|_{\mathcal{T}^c\cap\mathcal{S}^\star}\right\|_2 + \left\|\mathbf{W}^{-1}\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\right\|_2 \\
&\leq \frac{\|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)} + \frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{2\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2,
\end{aligned}$$

as desired. ■

Lemma 3 (Iteration Invariant)**.** *The estimation error in the current iteration, $\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2$, and that in the next iteration, $\|\mathbf{b}_s-\mathbf{x}^\star\|_2$, are related by the inequality:*

$$\begin{aligned}
\|\mathbf{b}_s-\mathbf{x}^\star\|_2 \leq{}& \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\left(1+\frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\right)\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 \\
&+\left(1+\frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\right)\frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}+\frac{2\|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}.
\end{aligned}$$

Proof Because $\mathcal{Z}\subseteq\mathcal{T}$ we must have $\mathcal{T}^c\subseteq\mathcal{Z}^c$. Therefore, we can write $\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2 = \|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{T}^c}\|_2 \leq \|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2$. Then using Lemma 1 we obtain

$$\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2 \leq \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}. \tag{18}$$

Furthermore,

$$\begin{aligned}
\|\mathbf{b}_s-\mathbf{x}^\star\|_2 &\leq \|\mathbf{b}_s-\mathbf{x}^\star|_{\mathcal{T}}\|_2+\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2 \\
&\leq \|\mathbf{x}^\star|_{\mathcal{T}}-\mathbf{b}\|_2+\|\mathbf{b}_s-\mathbf{b}\|_2+\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2 \leq 2\|\mathbf{x}^\star|_{\mathcal{T}}-\mathbf{b}\|_2+\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2,
\end{aligned} \tag{19}$$

where the last inequality holds because $\|\mathbf{x}^\star|_{\mathcal{T}}\|_0 \leq s$ and $\mathbf{b}_s$ is the best s-term approximation of $\mathbf{b}$. Therefore, using Lemma 2,

$$\|\mathbf{b}_s - \mathbf{x}^\star\|_2 \leq \frac{2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)} \|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2 + \left(1 + \frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\right) \|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2 . \tag{20}$$

Combining (18) and (20) we obtain

$$\begin{aligned}\|\mathbf{b}_s - \mathbf{x}^\star\|_2 \leq & \frac{\gamma_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star) + \gamma_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{2\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)} \left(1 + \frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\right) \|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_2 \\ & + \left(1 + \frac{\gamma_{4s}(\mathbf{b},\mathbf{x}^\star)}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}\right) \frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 + \left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\beta_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)} + \frac{2\|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)}.\end{aligned}$$

■

Using the results above, we can now prove Theorem 1.

***Proof of Theorem* 1.** Using definition 1 it is easy to verify that for $k \leq k'$ and any vector $\mathbf{u}$ we have $A_k(\mathbf{u}) \leq A_{k'}(\mathbf{u})$ and $B_k(\mathbf{u}) \geq B_{k'}(\mathbf{u})$. Consequently, for $k \leq k'$ and any pair of vectors $\mathbf{p}$ and $\mathbf{q}$ we have $\alpha_k(\mathbf{p},\mathbf{q}) \leq \alpha_{k'}(\mathbf{p},\mathbf{q})$, $\beta_k(\mathbf{p},\mathbf{q}) \geq \beta_{k'}(\mathbf{p},\mathbf{q})$, and $\mu_k \leq \mu_{k'}$. Furthermore, for any function that satisfies μ_k-SRH we can write

$$\frac{\alpha_k(\mathbf{p},\mathbf{q})}{\beta_k(\mathbf{p},\mathbf{q})} = \frac{\int_0^1 A_k(t\mathbf{q}+(1-t)\mathbf{p})\,\mathrm{d}t}{\int_0^1 B_k(t\mathbf{q}+(1-t)\mathbf{p})\,\mathrm{d}t} \leq \frac{\int_0^1 \mu_k B_k(t\mathbf{q}+(1-t)\mathbf{p})\,\mathrm{d}t}{\int_0^1 B_k(t\mathbf{q}+(1-t)\mathbf{p})\,\mathrm{d}t} = \mu_k,$$

and thereby $\frac{\gamma_k(\mathbf{p},\mathbf{q})}{\beta_k(\mathbf{p},\mathbf{q})} \leq \mu_k - 1$. Therefore, applying Lemma 3 to the estimate in the i-th iterate of the algorithm shows that

$$\begin{aligned}\left\|\widehat{\mathbf{x}}^{(i)} - \mathbf{x}^\star\right\|_2 &\leq (\mu_{4s}-1)\mu_{4s}\left\|\widehat{\mathbf{x}}^{(i-1)} - \mathbf{x}^\star\right\|_2 + \frac{2\|\nabla f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\beta_{4s}(\mathbf{b},\mathbf{x}^\star)} \\ &\quad + \mu_{4s}\frac{\left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 + \left\|\nabla f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\beta_{2s}\left(\widehat{\mathbf{x}}^{(i-1)},\mathbf{x}^\star\right)} \\ &\leq \left(\mu_{4s}^2 - \mu_{4s}\right)\left\|\widehat{\mathbf{x}}^{(i-1)} - \mathbf{x}^\star\right\|_2 + \frac{2}{\varepsilon}\|\nabla f(\mathbf{x}^\star)|_I\|_2 + \frac{2\mu_{4s}}{\varepsilon}\|\nabla f(\mathbf{x}^\star)|_I\|_2 .\end{aligned}$$

Applying the assumption $\mu_{4s} \leq \frac{1+\sqrt{3}}{2}$ then yields

$$\left\|\widehat{\mathbf{x}}^{(i)} - \mathbf{x}^\star\right\|_2 \leq \frac{1}{2}\left\|\widehat{\mathbf{x}}^{(i-1)} - \mathbf{x}^\star\right\|_2 + \frac{3+\sqrt{3}}{\varepsilon}\|\nabla f(\mathbf{x}^\star)|_I\|_2 .$$

The theorem follows using this inequality recursively. ■

Appendix B. Iteration Analysis For Non-Smooth Cost Functions

In this part we provide analysis of GraSP for non-smooth functions. Definition 3 basically states that for any k-sparse vector $\mathbf{x} \in \mathbb{R}^n$, $\alpha_k(\mathbf{x})$ and $\beta_k(\mathbf{x})$ are in order the smallest and largest values for which

$$\beta_k(\mathbf{x})\|\Delta\|_2^2 \leq \mathrm{B}_f(\mathbf{x}+\Delta \parallel \mathbf{x}) \leq \alpha_k(\mathbf{x})\|\Delta\|_2^2 \tag{21}$$

holds for all vectors $\Delta \in \mathbb{R}^n$ that satisfy $|\mathrm{supp}(\mathbf{x}) \cup \mathrm{supp}(\Delta)| \leq k$. By interchanging $\mathbf{x}$ and $\mathbf{x}+\Delta$ in (21) and using the fact that

$$\mathrm{B}_f(\mathbf{x}+\Delta \parallel \mathbf{x}) + \mathrm{B}_f(\mathbf{x} \parallel \mathbf{x}+\Delta) = \left\langle \nabla f(\mathbf{x}+\Delta) - \nabla f(\mathbf{x}), \Delta \right\rangle$$

one can easily deduce

$$[\beta_k(\mathbf{x}+\Delta)+\beta_k(\mathbf{x})]\|\Delta\|_2^2 \leq \left\langle \nabla f(\mathbf{x}+\Delta) - \nabla f(\mathbf{x}), \Delta \right\rangle \leq [\alpha_k(\mathbf{x}+\Delta)+\alpha_k(\mathbf{x})]\|\Delta\|_2^2. \tag{22}$$

Propositions 3, 4, and 5 establish some basic inequalities regarding the restricted Bregman divergence under SRL assumption. Using these inequalities we prove Lemmas 4 and 5. These two Lemmas are then used to prove an iteration invariant result in Lemma 6 which in turn is used to prove Theorem 2.

Note In Propositions 3, 4, and 5 we assume $\mathbf{x}_1$ and $\mathbf{x}_2$ are two vectors in $\mathbb{R}^n$ such that $|\mathrm{supp}(\mathbf{x}_1) \cup \mathrm{supp}(\mathbf{x}_2)| \leq r$. Furthermore, we use the shorthand $\Delta = \mathbf{x}_1 - \mathbf{x}_2$ and denote $\mathrm{supp}(\Delta)$ by $\mathcal{R}$. We also denote $\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)$ by Δ'. To simplify the notation further the shorthands $\overline{\alpha}_l$, $\overline{\beta}_l$, and $\overline{\gamma}_l$ are used for $\overline{\alpha}_l(\mathbf{x}_1, \mathbf{x}_2) := \alpha_l(\mathbf{x}_1) + \alpha_l(\mathbf{x}_2)$, $\overline{\beta}_l(\mathbf{x}_1, \mathbf{x}_2) := \beta_l(\mathbf{x}_1) + \beta_l(\mathbf{x}_2)$, and $\overline{\gamma}_l(\mathbf{x}_1, \mathbf{x}_2) := \overline{\alpha}_l(\mathbf{x}_1, \mathbf{x}_2) - \overline{\beta}_l(\mathbf{x}_1, \mathbf{x}_2)$, respectively.

Proposition 3. *Let $\mathcal{R}'$ be a subset of $\mathcal{R}$. Then the following inequalities hold.*

$$\left| \overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right| \leq \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2 \tag{23}$$

$$\left| \overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right| \leq \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2$$

Proof Using (21) we can write

$$\beta_r(\mathbf{x}_1) \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 \leq \mathrm{B}_f\left(\mathbf{x}_1 - t\,\Delta|_{\mathcal{R}'} \parallel \mathbf{x}_1\right) \leq \alpha_r(\mathbf{x}_1) \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 \tag{24}$$

$$\beta_r(\mathbf{x}_2) \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 \leq \mathrm{B}_f\left(\mathbf{x}_2 - t\,\Delta|_{\mathcal{R}'} \parallel \mathbf{x}_2\right) \leq \alpha_r(\mathbf{x}_2) \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 \tag{25}$$

and

$$\beta_r(\mathbf{x}_1) \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2 \leq \mathrm{B}_f\left(\mathbf{x}_2 + t\,\Delta|_{\mathcal{R}'} \parallel \mathbf{x}_1\right) \leq \alpha_r(\mathbf{x}_1) \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2 \tag{26}$$

$$\beta_r(\mathbf{x}_2) \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2 \leq \mathrm{B}_f\left(\mathbf{x}_1 - t\,\Delta|_{\mathcal{R}'} \parallel \mathbf{x}_2\right) \leq \alpha_r(\mathbf{x}_2) \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2, \tag{27}$$

where t is an arbitrary real number. Using the definition of the Bregman divergence we can add (24) and (25) to obtain

$$\begin{aligned} \overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 &\leq f\left(\mathbf{x}_1 - t\,\Delta|_{\mathcal{R}'}\right) - f(\mathbf{x}_1) + f\left(\mathbf{x}_2 + t\,\Delta|_{\mathcal{R}'}\right) - f(\mathbf{x}_2) + \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle t \\ &\leq \overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2. \end{aligned} \tag{28}$$

Similarly, (26) and (27) yield

$$\begin{aligned} \overline{\beta}_r \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2 &\leq f\left(\mathbf{x}_1 - t\,\Delta|_{\mathcal{R}'}\right) - f(\mathbf{x}_1) + f\left(\mathbf{x}_2 + t\,\Delta|_{\mathcal{R}'}\right) - f(\mathbf{x}_2) + \left\langle \Delta', \Delta - t\,\Delta|_{\mathcal{R}'} \right\rangle \\ &\leq \overline{\alpha}_r \left\| \Delta - t\,\Delta|_{\mathcal{R}'} \right\|_2^2. \end{aligned} \tag{29}$$

Expanding the quadratic bounds of (29) and using (28) then we obtain

$$0 \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 + 2\left(\overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta, \Delta|_{\mathcal{R}'} \right\rangle \right) t - \overline{\beta}_r \|\Delta\|_2^2 + \left\langle \Delta', \Delta \right\rangle \tag{30}$$

$$0 \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 - 2\left(\overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta, \Delta|_{\mathcal{R}'} \right\rangle \right) t + \overline{\alpha}_r \|\Delta\|_2^2 - \left\langle \Delta', \Delta \right\rangle. \tag{31}$$

It follows from (22), (30), and (31) that

$$0 \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 + 2\left(\overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta, \Delta|_{\mathcal{R}'} \right\rangle \right) t + \overline{\gamma}_r \|\Delta\|_2^2$$

$$0 \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 t^2 - 2\left(\overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta, \Delta|_{\mathcal{R}'} \right\rangle \right) t + \overline{\gamma}_r \|\Delta\|_2^2.$$

These two quadratic inequalities hold for any $t \in \mathbb{R}$ thus their discriminants are not positive, that is,

$$\left(\overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right)^2 - \overline{\gamma}_r^2 \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 \|\Delta\|_2^2 \le 0$$

$$\left(\overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right)^2 - \overline{\gamma}_r^2 \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 \|\Delta\|_2^2 \le 0,$$

which yield the desired result. ■

Proposition 4. *The following inequalities hold for $\mathcal{R}' \subseteq \mathcal{R}$.*

$$\left| \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\alpha}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right| \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2 \tag{32}$$

$$\left| \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\beta}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right| \le \overline{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2$$

Proof From (21) we have

$$\beta_r(\mathbf{x}_1) \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 \le \mathrm{B}_f\left(\mathbf{x}_1 - t\, \Delta'|_{\mathcal{R}'} \,\|\, \mathbf{x}_1 \right) \le \alpha_r(\mathbf{x}_1) \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 \tag{33}$$

$$\beta_r(\mathbf{x}_2) \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 \le \mathrm{B}_f\left(\mathbf{x}_2 + t\, \Delta'|_{\mathcal{R}'} \,\|\, \mathbf{x}_2 \right) \le \alpha_r(\mathbf{x}_2) \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 \tag{34}$$

and

$$\beta_r(\mathbf{x}_1) \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2 \le \mathrm{B}_f\left(\mathbf{x}_2 + t\, \Delta'|_{\mathcal{R}'} \,\|\, \mathbf{x}_1 \right) \le \alpha_r(\mathbf{x}_1) \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2 \tag{35}$$

$$\beta_r(\mathbf{x}_2) \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2 \le \mathrm{B}_f\left(\mathbf{x}_1 - t\, \Delta'|_{\mathcal{R}'} \,\|\, \mathbf{x}_2 \right) \le \alpha_r(\mathbf{x}_2) \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2, \tag{36}$$

for any $t \in \mathbb{R}$. By subtracting the sum of (35) and (36) from that of (33) and (34) we obtain

$$\begin{aligned} \overline{\beta}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 - \overline{\alpha}_r \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2 &\le 2\left\langle \Delta', \Delta'|_{\mathcal{R}'} \right\rangle t - \left\langle \Delta', \Delta \right\rangle \\ &\le \overline{\alpha}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 - \overline{\beta}_r \left\| \Delta - t\, \Delta'|_{\mathcal{R}'} \right\|_2^2. \end{aligned} \tag{37}$$

Expanding the bounds of (37) then yields

$$0 \leq \bar{\gamma}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 + 2\left(\left\langle \Delta', \Delta'|_{\mathcal{R}'} \right\rangle - \overline{\alpha}_r \left\langle \Delta, \Delta'|_{\mathcal{R}'} \right\rangle \right) t + \overline{\alpha}_r \|\Delta\|_2^2 - \langle \Delta', \Delta \rangle$$
$$0 \leq \bar{\gamma}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 - 2\left(\left\langle \Delta', \Delta'|_{\mathcal{R}'} \right\rangle - \overline{\beta}_r \left\langle \Delta, \Delta'|_{\mathcal{R}'} \right\rangle \right) t - \overline{\beta}_r \|\Delta\|_2^2 + \langle \Delta', \Delta \rangle .$$

Note that $\left\langle \Delta', \Delta'|_{\mathcal{R}'} \right\rangle = \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2$ and $\left\langle \Delta, \Delta'|_{\mathcal{R}'} \right\rangle = \left\langle \Delta|_{\mathcal{R}'}, \Delta' \right\rangle$. Therefore, using (22) we obtain

$$0 \leq \bar{\gamma}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 + 2\left(\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\alpha}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right) t + \bar{\gamma}_r \|\Delta\|_2^2 \tag{38}$$
$$0 \leq \bar{\gamma}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 t^2 - 2\left(\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\beta}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right) t + \bar{\gamma}_r \|\Delta\|_2^2 . \tag{39}$$

Since the right-hand sides of (38) and (39) are quadratics in t and always non-negative for all values of $t \in \mathbb{R}$, their discriminants cannot be positive. Thus we have

$$\left(\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\alpha}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right)^2 - \bar{\gamma}_r^2 \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 \|\Delta\|^2 \leq 0$$
$$\left(\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 - \overline{\beta}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right)^2 - \bar{\gamma}_r^2 \left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 \|\Delta\|^2 \leq 0,$$

which yield the desired result. ■

Corollary 2. *The inequality*

$$\left\| \Delta'|_{\mathcal{R}'} \right\|_2 \geq \overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 - \bar{\gamma}_r \left\| \Delta|_{\mathcal{R}\setminus\mathcal{R}'} \right\|_2 ,$$

holds for $\mathcal{R}' \subseteq \mathcal{R}$.

Proof It follows from (32) and (23) that

$$\begin{aligned} -\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 + \overline{\alpha}_r^2 \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 &= -\left\| \Delta'|_{\mathcal{R}'} \right\|_2^2 + \overline{\alpha}_r \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle + \overline{\alpha}_r \left[\overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2^2 - \left\langle \Delta', \Delta|_{\mathcal{R}'} \right\rangle \right] \\ &\leq \bar{\gamma}_r \left\| \Delta'|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2 + \overline{\alpha}_r \bar{\gamma}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 \|\Delta\|_2 . \end{aligned}$$

Therefore, after straightforward calculations we get

$$\begin{aligned} \left\| \Delta'|_{\mathcal{R}'} \right\|_2 &\geq \frac{1}{2}\left(-\bar{\gamma}_r \|\Delta\|_2 + \left| 2\overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 - \bar{\gamma}_r \|\Delta\|_2 \right| \right) \\ &\geq \overline{\alpha}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 - \bar{\gamma}_r \|\Delta\|_2 \\ &\geq \overline{\beta}_r \left\| \Delta|_{\mathcal{R}'} \right\|_2 - \bar{\gamma}_r \left\| \Delta|_{\mathcal{R}\setminus\mathcal{R}'} \right\|_2 . \end{aligned}$$

■

Proposition 5. *Suppose that $\mathcal{K}$ is a subset of $\mathcal{R}^c$ with at most k elements. Then we have*

$$\left\| \Delta' |_{\mathcal{K}} \right\|_2 \leq \overline{\gamma}_{k+r} \|\Delta\|_2 .$$

Proof Using (21) for any $t \in \mathbb{R}$ we can write

$$\beta_{k+r}(\mathbf{x}_1) \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 \leq \mathrm{B}_f \left(\mathbf{x}_1 + t\, \Delta' |_{\mathcal{K}} \,\|\, \mathbf{x}_1 \right) \leq \alpha_{k+r}(\mathbf{x}_1) \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 \tag{40}$$

$$\beta_{k+r}(\mathbf{x}_2) \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 \leq \mathrm{B}_f \left(\mathbf{x}_2 - t\, \Delta' |_{\mathcal{K}} \,\|\, \mathbf{x}_2 \right) \leq \alpha_{k+r}(\mathbf{x}_2) \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 \tag{41}$$

and similarly

$$\beta_{k+r}(\mathbf{x}_1) \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 \leq \mathrm{B}_f \left(\mathbf{x}_2 - t\, \Delta' |_{\mathcal{K}} \,\|\, \mathbf{x}_1 \right) \leq \alpha_{k+r}(\mathbf{x}_1) \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 \tag{42}$$

$$\beta_{k+r}(\mathbf{x}_2) \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 \leq \mathrm{B}_f \left(\mathbf{x}_1 + t\, \Delta' |_{\mathcal{K}} \,\|\, \mathbf{x}_2 \right) \leq \alpha_{k+r}(\mathbf{x}_2) \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 . \tag{43}$$

By subtracting the sum of (42) and (43) from that of (40) and (41) we obtain

$$\begin{aligned} \overline{\beta}_{k+r} \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 - \overline{\alpha}_{k+r} \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 &\leq -2t \left\langle \Delta', \Delta' |_{\mathcal{K}} \right\rangle - \langle \Delta', \Delta \rangle \\ &\leq \overline{\alpha}_{k+r} \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 - \overline{\beta}_{k+r} \left\| \Delta + t\, \Delta' |_{\mathcal{K}} \right\|_2^2 . \end{aligned} \tag{44}$$

Note that $\left\langle \Delta', \Delta' |_{\mathcal{K}} \right\rangle = \left\| \Delta' |_{\mathcal{K}} \right\|_2^2$ and $\left\langle \Delta, \Delta' |_{\mathcal{K}} \right\rangle = 0$. Therefore, (22) and (44) imply

$$0 \leq \overline{\gamma}_{k+r} \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t^2 \pm 2 \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 t + \overline{\gamma}_{k+r} \|\Delta\|_2^2 \tag{45}$$

hold for all $t \in \mathbb{R}$. Hence, as quadratic functions of t, the right-hand side of (45) cannot have a positive discriminant. Thus we must have

$$\left\| \Delta' |_{\mathcal{K}} \right\|_2^4 - \overline{\gamma}_{k+r}^2 \|\Delta\|_2^2 \left\| \Delta' |_{\mathcal{K}} \right\|_2^2 \leq 0,$$

which yields the desired result. ■

Lemma 4. *Let $\mathcal{R}$ denote* $\operatorname{supp}(\widehat{\mathbf{x}} - \mathbf{x}^\star)$. *Then we have*

$$\| (\widehat{\mathbf{x}} - \mathbf{x}^\star) |_{\mathcal{Z}^c} \|_2 \leq \frac{\overline{\gamma}_{2s}(\widehat{\mathbf{x}}, \mathbf{x}^\star) + \overline{\gamma}_{4s}(\widehat{\mathbf{x}}, \mathbf{x}^\star)}{\overline{\beta}_{2s}(\widehat{\mathbf{x}}, \mathbf{x}^\star)} \|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_2 + \frac{\left\| \nabla_f(\mathbf{x}^\star) |_{\mathcal{R} \setminus \mathcal{Z}} \right\|_2 + \left\| \nabla_f(\mathbf{x}^\star) |_{\mathcal{Z} \setminus \mathcal{R}} \right\|_2}{\overline{\beta}_{2s}(\widehat{\mathbf{x}}, \mathbf{x}^\star)} .$$

Proof Given that $\mathcal{Z} = \operatorname{supp}(\mathbf{z}_{2s})$ and $|\mathcal{R}| \leq 2s$ we have $\left\| \mathbf{z} |_{\mathcal{R}} \right\|_2 \leq \| \mathbf{z} |_{\mathcal{Z}} \|_2$. Hence

$$\left\| \mathbf{z} |_{\mathcal{R} \setminus \mathcal{Z}} \right\|_2 \leq \left\| \mathbf{z} |_{\mathcal{Z} \setminus \mathcal{R}} \right\|_2 . \tag{46}$$

Furthermore, using Corollary 2 we can write

$$\begin{aligned}\left\|\mathbf{z}|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 &= \left\|\nabla_f(\widehat{\mathbf{x}})|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\ &\geq \left\|(\nabla_f(\widehat{\mathbf{x}})-\nabla_f(\mathbf{x}^\star))|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 - \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\ &\geq \overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 - \overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\cap\mathcal{Z}}\right\|_2 - \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 \\ &\geq \overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 - \overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 - \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2. \end{aligned} \tag{47}$$

Similarly, using Proposition 5 we have

$$\begin{aligned}\left\|\mathbf{z}|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 = \left\|\nabla_f(\widehat{\mathbf{x}})|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 &\leq \left\|(\nabla_f(\widehat{\mathbf{x}})-\nabla_f(\mathbf{x}^\star))|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 + \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 \\ &\leq \overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 + \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2. \end{aligned} \tag{48}$$

Combining (46), (47), and (48) then yields

$$\begin{aligned}\overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 + \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2 &\geq -\overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\cap\mathcal{Z}}\right\|_2 \\ &\quad + \overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)\left\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 - \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2.\end{aligned}$$

Note that $(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}} = (\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}$. Therefore, we have

$$\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2 \leq \frac{\overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 + \frac{\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2 + \left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}.$$

■

Lemma 5. *The vector* $\mathbf{b}$ *given by*

$$\mathbf{b} = \arg\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t. } \mathbf{x}|_{\mathcal{T}^c} = \mathbf{0} \tag{49}$$

satisfies $\|\mathbf{x}^\star|_{\mathcal{T}} - \mathbf{b}\|_2 \leq \frac{\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{T}}\|_2}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})} + \left(1+\frac{\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2$.

Proof Since $\mathbf{b}$ satisfies (49) we must have $\nabla_f(\mathbf{b})|_{\mathcal{T}} = \mathbf{0}$. Then it follows from Corollary 2 that

$$\begin{aligned}\|\mathbf{x}^\star|_{\mathcal{T}} - \mathbf{b}\|_2 &= \|(\mathbf{x}^\star-\mathbf{b})|_{\mathcal{T}}\|_2 \\ &\leq \frac{\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{T}}\right\|_2}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})} + \frac{\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2.\end{aligned}$$

■

Lemma 6. *The estimation error of the current iterate (i.e.,* $\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2$*) and that of the next iterate (i.e.,* $\|\mathbf{b}_s-\mathbf{x}^\star\|_2$*) are related by the inequality:*

$$\|\mathbf{b}_s-\mathbf{x}^\star\|_2 \le \left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\frac{\overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{\overline{\beta}_{2s}(\widehat{\mathbf{x}}^i,\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{2\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{T}}\right\|_2}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}$$
$$+\left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\frac{\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}.$$

Proof Since $\mathcal{T}^c\subseteq\mathcal{Z}^c$ we have $\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2=\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{T}^c}\|_2\le\|(\widehat{\mathbf{x}}-\mathbf{x}^\star)|_{\mathcal{Z}^c}\|_2$. Therefore, applying Lemma 4 yields

$$\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2\le\frac{\overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}. \tag{50}$$

Furthermore, as showed by (19) during the proof of Lemma 3, we again have

$$\|\mathbf{b}_s-\mathbf{x}^\star\|_2\le 2\|\mathbf{x}^\star|_{\mathcal{T}}-\mathbf{b}\|_2+\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2.$$

Hence, it follows from Lemma 5 that

$$\|\mathbf{b}_s-\mathbf{x}^\star\|_2\le\frac{2\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{T}}\right\|_2}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}+\left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\|\mathbf{x}^\star|_{\mathcal{T}^c}\|_2. \tag{51}$$

Combining (50) and (51) yields

$$\|\mathbf{b}_s-\mathbf{x}^\star\|_2 \le \left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\frac{\overline{\gamma}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)+\overline{\gamma}_{4s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2+\frac{2\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{T}}\right\|_2}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}$$
$$+\left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\frac{\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{R}\setminus\mathcal{Z}}\right\|_2+\left\|\nabla_f(\mathbf{x}^\star)|_{\mathcal{Z}\setminus\mathcal{R}}\right\|_2}{\overline{\beta}_{2s}(\widehat{\mathbf{x}},\mathbf{x}^\star)}.$$

■

***Proof of Theorem* 2.** Let the vectors involved in the j-th iteration of the algorithm be denoted by superscript (j). Given that $\mu_{4s}\le\frac{3+\sqrt{3}}{4}$ we have

$$\frac{\overline{\gamma}_{4s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)}{\overline{\beta}_{4s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)}\le\frac{\sqrt{3}-1}{4}\quad\text{and}\quad 1+\frac{2\overline{\gamma}_{4s}\left(\mathbf{x}^\star,\mathbf{b}^{(j)}\right)}{\overline{\beta}_{4s}\left(\mathbf{x}^\star,\mathbf{b}^{(j)}\right)}\le\frac{1+\sqrt{3}}{2},$$

that yield,

$$\begin{aligned}\left(1+\frac{2\overline{\gamma}_{4s}(\mathbf{x}^\star,\mathbf{b})}{\overline{\beta}_{4s}(\mathbf{x}^\star,\mathbf{b})}\right)\frac{\overline{\gamma}_{2s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)+\overline{\gamma}_{4s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)}{\overline{\beta}_{2s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)} &\le\frac{1+\sqrt{3}}{2}\times\frac{2\overline{\gamma}_{4s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)}{\overline{\beta}_{4s}\left(\widehat{\mathbf{x}}^{(j)},\mathbf{x}^\star\right)}\\ &\le\frac{1+\sqrt{3}}{2}\times\frac{\sqrt{3}-1}{2}\\ &=\frac{1}{2}.\end{aligned}$$

Therefore, it follows from Lemma 6 that

$$\left\|\widehat{\mathbf{x}}^{(j+1)}-\mathbf{x}^\star\right\|_2 \leq \frac{1}{2}\left\|\widehat{\mathbf{x}}^{(j)}-\mathbf{x}^\star\right\|_2 + \frac{3+\sqrt{3}}{\varepsilon}\left\|\nabla_f(\mathbf{x}^\star)\big|_I\right\|_2.$$

Applying this inequality recursively for $j = 0, 1, \cdots, i-1$ then yields

$$\|\widehat{\mathbf{x}}-\mathbf{x}^\star\|_2 \leq 2^{-i}\|\mathbf{x}^\star\|_2 + \frac{6+2\sqrt{3}}{\varepsilon}\left\|\nabla_f(\mathbf{x}^\star)\big|_I\right\|_2,$$

which is the desired result. ■

References

A. Agarwal, S. Negahban, and M. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 37–45. 2010. long version available at `arXiv:1104.4824v1 [stat.ML]`.

S. Bahmani, P. Boufounos, and B. Raj. Greedy sparsity-constrained optimization. In *Conference Record of the Forty-Fifth Asilomar Conference on Signals, Systems, and Computers*, pages 1148–1152, 2011.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

T. Blumensath. Compressed sensing with nonlinear observations. Preprint, 2010. URL `http://users.fmrib.ox.ac.uk/~tblumens/papers/B_Nonlinear.pdf`.

T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, Nov. 2009.

F. Bunea. Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1+\ell_2$ penalization. *Electronic Journal of Statistics*, 2:1153–1194, 2008.

E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592, 2008.

E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, Dec. 2012.

A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *American Mathematical Society*, 22(1):211–231, 2009.

W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009.

A. J. Dobson and A. Barnett. *An Introduction to Generalized Linear Models*. Chapman and Hall/CRC, 3rd edition, May 2008.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001.

S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, volume 13 of *Springer Proceedings in Mathematics*, pages 65–77. Springer New York, 2012.

J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2 2010. ISSN 1548-7660. Software available online at `http://www-stat.stanford.edu/~tibs/glmnet-matlab/`.

I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the NIPS 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT-Press, Cambridge, MA, 2005.

J. D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Verlag, 2009.

A. Jalali, C. C. Johnson, and P. K. Ravikumar. On learning discrete graphical models using greedy methods. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1935–1943. 2011.

S. M. Kakade, O. Shamir, K. Sridharan, and A. Tewari. Learning exponential families in high-dimensions: strong convexity and sparsity. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *JMLR W&CP*, pages 381–388, 2010.

J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 547–556, New York, NY, USA, 2009. ACM.

A. Lozano, G. Swirszcz, and N. Abe. Group orthogonal matching pursuit for logistic regression. In G. Gordon, D. Dunson, and M. Dudik, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR W&CP*, pages 452–460, 2011.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24 (2):227–234, 1995.

D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1348–1356. 2009. long version available at `arXiv:1010.2731v1 [math.ST]`.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, Jan. 2012.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993.

S. Shalev-Shwartz, N. Srebro, and T. Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6):2807–2832, 2010.

A. Tewari, P. K. Ravikumar, and I. S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 882–890. 2011.

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, Aug. 2012.

J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

S. A. van de Geer. *Empirical Processes in M-estimation*. Cambridge University Press, Cambridge, UK, 2000.

S. A. van de Geer. High-dimensional generalized linear models and the Lasso. *The Annals of Statistics*, 36(2):614–645, 2008.

V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

T. Zhang. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Transactions on Information Theory*, 57(9):6215 –6221, Sept. 2011.

Journal of Machine Learning Research 14 (2013) 843-865 Submitted 3/12; Revised 10/12; Published 3/13

Quasi-Newton Methods: A New Direction

Philipp Hennig PHILIPP.HENNIG@TUEBINGEN.MPG.DE
Martin Kiefel MARTIN.KIEFEL@TUEBINGEN.MPG.DE
Department of Empirical Inference
Max Planck Institute for Intelligent Systems
Spemannstraße 38
Tübingen, Germany

Editor: Manfred Opper

Abstract

Four decades after their invention, quasi-Newton methods are still state of the art in unconstrained numerical optimization. Although not usually interpreted thus, these are learning algorithms that fit a local quadratic approximation to the objective function. We show that many, including the most popular, quasi-Newton methods can be interpreted as approximations of Bayesian linear regression under varying prior assumptions. This new notion elucidates some shortcomings of classical algorithms, and lights the way to a novel nonparametric quasi-Newton method, which is able to make more efficient use of available information at computational cost similar to its predecessors.

Keywords: optimization, numerical analysis, probability, Gaussian processes

1. Introduction

Quasi-Newton algorithms are arguably the most popular class of nonlinear numerical optimization methods, used widely in numerical applications not just in machine learning. Their defining property is that they iteratively build estimators B_i for the Hessian $B(x) = \nabla\nabla^\intercal f(x)$ of the objective function $f(x)$, from observations of f's gradient $\nabla f(x)$, at each iteration searching for a local minimum along a line search direction $-B_i^{-1}\nabla f(x)$, an estimate of the eponymous Newton-Raphson search direction. Some of the most widely known members of this family include Broyden's (1965) method, the SR1 formula (Davidon, 1959; Broyden, 1967), the DFP method (Davidon, 1959; Fletcher and Powell, 1963) and the BFGS method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). Decades of continued research effort in this area make it impossible to give even a superficial overview over the available literature. The textbooks by Nocedal and Wright (1999) and Boyd and Vandenberghe (2004) are good modern starting points for readers interested in background. An insightful and extensive contemporary review was compiled by Dennis and Moré (1977). The ubiquity of optimization problems in machine learning has made these algorithms tools of the trade. But, perhaps because they predate machine learning itself, they have rarely been studied as learning algorithms in their own right. This paper offers a probabilistic analysis.

Throughout, let $f : \mathbb{R}^N \to \mathbb{R}$ be a sufficiently regular, not necessarily convex, function; $\nabla f : \mathbb{R}^N \to \mathbb{R}^N$ its gradient; $B : \mathbb{R}^N \to \mathbb{R}^{N\times N}$ its Hessian. We consider iterative algorithms moving from location $x_{\ell-1} \in \mathbb{R}^D$ to location x_ℓ. The algorithm performs consecutive *line searches* along one-dimensional subspaces $x_i(\alpha) = \alpha e_i + x_i^0$, with $\alpha \in \mathbb{R}_+$ and a unit length vector $e_i \in \mathbb{R}^N$ spanning the line search space starting at x_i^0. Evaluations at x_i evince the gradient $\nabla f(x_i)$ (and usually also $f(x_i)$,

though this will not feature in this paper). The goal is to find a candidate x^* for a local minimum: a root $\nabla f(x^*) = 0$ of the gradient.

The derivations of classical quasi-Newton algorithms proceed along the following line of argument: We require an update rule incorporating an observation $\nabla f(x_{i+1})$ into a current estimate B_i to get a new estimate B_{i+1}, subject to the following desiderata:

1. **Low Rank/Cost Updates** Optimization problems regularly have dimensionality above $N \sim 10^3$, even beyond $N \sim 10^6$. To keep computational costs tractable, the update to the estimator B_i for the Hessian should be of the form

$$B_i = B_{i-1} + uCv^\top \qquad \text{with } u, v \in \mathbb{R}^{N\times M}, C \in \mathbb{R}^{M\times M},$$

   with low rank M (usually $M = 1$ or 2), because, by the matrix inversion lemma, its inversion, and multiplication with the gradient has (worst-case) cost $\mathcal{O}(N^2 + NM + M^3)$.

2. **Consistency with Quadratic Model** If f is locally described well to second order, then

$$y_i \equiv \nabla f(x_i) - \nabla f(x_{i-1}) \approx B(x_i)s_i, \tag{1}$$

   with $s_i \equiv x_i - x_{i-1}$. Because this is the fundamental idea behind this family of algorithms, it is also known as *the quasi-Newton equation*.

3. **Symmetry** The Hessian of twice differentiable functions is symmetric; so the estimator should be symmetric, too.

4. **Positive Definiteness** *Convex* functions have positive definite Hessians everywhere. Over time, it has become common conviction that, even for non-convex problems, positive definiteness of the estimator is desirable.

1.1 Outline

This first half of this paper (Section 2) constructs a new conceptual interpretation of quasi-Newton methods. Adopting a probabilistic viewpoint, we interpret the two requirements classically used to derive this family of methods as log likelihood and log prior, both of a specific Gaussian form. Varying the prior covariance and choosing one of two possible likelihoods gives rise to the different members of the family of quasi-Newton methods. A surprising insight arising from this analysis is that the way symmetry and positive definiteness (desiderata 3 and 4 above) are ensured in existing quasi-Newton methods differs from the way one would naïvely choose from the probabilistic perspective. In fact, the posterior arising from the newly identified prior and likelihood assigns nonzero probability mass to non-symmetric (Section 2.1), and to indefinite matrices (Section 2.2). It is only the maximum of the posterior, the estimator used by quasi-Newton methods, that is both symmetric and positive definite. Interestingly, the "proper" probabilistic way to ensure these properties has much higher computational complexity (Sections 2.1 and 3.5).

The second half of the paper (Section 3) uses the insights gained in Section 2 to construct a novel nonparametric Bayesian quasi-Newton algorithm. This replaces the approximate form of desideratum 2 above with an exact, analytic expression. We show that the structural ideas developed in Section 2 extend from the classic parametric formulation to a Gaussian Process model keeping computational cost *linear* in the input dimensionality (it has cost $\mathcal{O}(NM + M^3)$). A further advantage

of the nonparametric formulation is that it allows the use of every gradient observation calculated during a line search instead of just the last one, something that is not easily achievable under the old parametric models.

1.2 Notation

The derivations in the following sections require a compact notation for joint Gaussian probability densities over the elements of matrices. This often requires re-arranging the elements of a matrix $A \in \mathbb{R}^{N\times M}$ into a vector in $\mathbb{R}^{NM}$, which we denote by $\overrightarrow{A}$. We will assume this vectorization operation stacks the rows of A into column vector row by row, not column by column (this choice is relevant, it has effects Equation (3) below). Instead of introducing a new scalar index for the elements of such vectors, it will be convenient to keep the original indices i, j of the matrix A, and interpret them as an index set (ij) of the vector.

Throughout, we make use of the following sum convention: Indices that appear more than once on one side of an equation are summed over, unless they also appear on the other side of the equation. We also extensively use the Kronecker product. Given $A \in \mathbb{R}^{I\times K}$ and $B \in \mathbb{R}^{J\times L}$, the Kronecker product $A \otimes B \in \mathbb{R}^{IJ\times KL}$ has elements

$$(A \otimes B)_{(ij)(k\ell)} = A_{ik}B_{j\ell}. \tag{2}$$

In this notation, using the sum convention defined above, the vectorization of the matrix product AXB can be re-written as

$$\overrightarrow{(AXB)}_{ij} = A_{ik}X_{k\ell}B_{\ell j} = A_{ik}B^{\top}_{j\ell}X_{k\ell} = \left[(A \otimes B^{\top})\overrightarrow{X}\right]_{ij}. \tag{3}$$

Some important properties of Kronecker products are

$$\begin{aligned}
&(A \otimes B)(C \otimes D) = AC \otimes BD, && (A \otimes B)^{-1} = A^{-1} \otimes B^{-1},\\
&\alpha(A \otimes B) = (\alpha A \otimes B) = (A \otimes \alpha B), && \mathrm{rk}(A \otimes B) = \mathrm{rk}A \cdot \mathrm{rk}B,\\
&\mathrm{tr}(A \otimes B) = \mathrm{tr}A \cdot \mathrm{tr}B, && \det(A \otimes B) = \det{}^{\mathrm{rk}(B)}A \cdot \det{}^{\mathrm{rk}(A)}B.
\end{aligned}$$

The identities in the left column directly follow from (2), the less straightforward identities on the right can be found in matrix algebra collections (e.g., Lütkepohl, 1996; Minka, 2000a).

2. Quasi-Newton Methods as Approximate Bayesian Regressors

Aiming for a probabilistic interpretation of quasi-Newton methods, we consider them as regularised maximum likelihood (that is, maximum a posteriori) estimation schemes. The quasi-Newton equation (1) is a likelihood for B. Using $s_i = x_i - x_{i-1}$, we can write it using Dirac's distribution as

$$p(y_i|B, s_i) = \delta(y_i - Bs_i) = \lim_{\beta\to 0} \mathcal{N}\left[y_i; \mathcal{S}_{\triangleright}^{\top}\overrightarrow{B}, (V_{i-1} \otimes \beta)\right], \tag{4}$$

with any arbitrary $N \times N$ matrix V_{i-1}, a scalar β, and the linear operator $\mathcal{S}_{\triangleright} = (I \otimes s_i)$ (the significance of the subscript $\triangleright$ will become clear later). Instead of enforcing this point mass likelihood (4), we could equivalently minimize its negative logarithm

$$-\log p(y_i|B, s_i) = \lim_{\beta\to 0} \tfrac{1}{\beta}(y_i - Bs_i)^{\top}V^{-1}(y_i - Bs_i) + \text{const.}$$

Since the N real numbers in y_i are not sufficient to identify the N^2 numbers in B, classic derivations (Dennis and Moré, 1977; Nocedal and Wright, 1999) choose the estimator minimizing a *regularized* loss,

$$B_i = \operatorname*{argmin}_{B\in\mathbb{R}^{N\times N}} \left\{ \lim_{\beta\to 0} \tfrac{1}{\beta}(y_i - Bs_i)^\top V^{-1}(y_i - Bs_i) + \|B - B_{i-1}\|_{F,V_{i-1}^{-1}} \right\},$$

using the weighted Frobenius norm $\|\cdot\|_{F,V_{i-1}^{-1}}$ from the current best estimate B_{i-1} from previous iterations. The weight in the Frobenius norm is encoded using a positive definite matrix, which we will suggestively call V_{i-1}^{-1} and identify with the V_{i-1} of Equation (4)

$$\begin{aligned} \|B - B_{i-1}\|_{F,V_{i-1}^{-1}} &\equiv \operatorname{tr}(V_{i-1}^{-1}(B - B_{i-1})^\top V_{i-1}^{-1}(B - B_{i-1})) \\ &= (\overrightarrow{B} - \overrightarrow{B}_{i-1})^\top (V_{i-1}^{-1} \otimes V_{i-1}^{-1})(\overrightarrow{B} - \overrightarrow{B}_{i-1}). \end{aligned} \tag{5}$$

The new estimate is the unique matrix B_i minimizing the regularizer subject to Equation (4). Inspecting Equation (5) we see that, up to additive constants, the Frobenius regularizer is the negative logarithm of a Gaussian prior

$$p(B) = \mathcal{N}\left[\overrightarrow{B}; \overrightarrow{B}_{i-1}, \Sigma_{i-1} \equiv (V_{i-1} \otimes V_{i-1})\right]. \tag{6}$$

Gaussian likelihoods are conjugate to Gaussian priors (the sum of quadratic forms is a quadratic form). So the posterior is Gaussian, too, even for the limit case of a Dirac likelihood. We perform the following derivations for finite β, then take the limit at the end. A first form for the posterior can be found by explicitly multiplying the two Gaussians and "completing the square" in the exponent of the product of Gaussians: Posterior covariance and mean are

$$\begin{aligned} \Sigma_\triangleright &= (\Sigma_{i-1}^{-1} + \mathcal{S}_\triangleright(V_{i-1}^{-1} \otimes \beta^{-1})\mathcal{S}_\triangleright^\top)^{-1}, \\ B_\triangleright &= \Sigma_\triangleright(\mathcal{S}_\triangleright(V_{i-1}^{-1} \otimes \beta^{-1})\overrightarrow{Y} + \Sigma_{i-1}^{-1}\overrightarrow{B}_{i-1}). \end{aligned}$$

The following observation is helpful in the search for a more compact form (e.g., Rasmussen and Williams, 2006, Equation 2.12). Because $\Sigma_\triangleright$ is invertible for any finite β,

$$\begin{aligned} \mathcal{S}_\triangleright(V_{i-1}^{-1} \otimes \beta^{-1})(\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright + V_{i-1} \otimes \beta) &= \Sigma_\triangleright^{-1}\Sigma_{i-1}\mathcal{S}_\triangleright, \\ \Sigma_\triangleright\mathcal{S}_\triangleright(V_{i-1}^{-1} \otimes \beta^{-1})(\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright + V_{i-1} \otimes \beta) &= \Sigma_{i-1}\mathcal{S}_\triangleright, \\ \Sigma_\triangleright\mathcal{S}_\triangleright(V_{i-1}^{-1} \otimes \beta^{-1}) &= \Sigma_{i-1}\mathcal{S}_\triangleright(\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1}. \end{aligned}$$

The step from the first to the second line is multiplication from the left by $\Sigma_\triangleright^{-1}$, the one from the second to the third is multiplication from the right by $(\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1}$. Using this result, we re-write the posterior mean, using the Matrix inversion lemma, as

$$\begin{aligned} \overrightarrow{B}_\triangleright &= \Sigma_\triangleright((V_{i-1}^{-1} \otimes \beta^{-1})\mathcal{S}_\triangleright\overrightarrow{Y} + \Sigma_{i-1}^{-1}\overrightarrow{B}_{i-1}) \\ &= \overrightarrow{B}_{i-1} + \Sigma_{i-1}\mathcal{S}_\triangleright(\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright + V_{i-1} \otimes \beta)^{-1} \cdot (\overrightarrow{Y} - \mathcal{S}_\triangleright^\top \overrightarrow{B}_{i-1}). \end{aligned}$$

Now we plug in the explicit expressions for $\mathcal{S}_\triangleright$ and Σ_{i-1}. Note that $\Sigma_{i-1}\mathcal{S}_\triangleright = (V_{i-1} \otimes V_{i-1})(I \otimes s_i) = (V_{i-1} \otimes V_{i-1}s_i)$ and likewise $\mathcal{S}_\triangleright^\top \Sigma_{i-1}\mathcal{S}_\triangleright = (V_{i-1} \otimes s_i^\top V_{i-1}s_i)$. So the posterior has mean and covariance

$$\begin{aligned} B_i &= B_{i-1} + \lim_{\beta \to 0} \frac{(y_i - B_{i-1}s_i)s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i + \beta} && = B_{i-1} + \frac{(y_i - B_{i-1}s_i)s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i} \quad \text{and} \\ \Sigma_i &= V_{i-1} \otimes \left(V_{i-1} - \lim_{\beta \to 0} \frac{V_{i-1}s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i + \beta} \right) && = V_{i-1} \otimes \left(V_{i-1} - \frac{V_{i-1}s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1}s_i} \right) \\ &\equiv V_{i-1} \otimes V_i, \end{aligned} \tag{7}$$

respectively. The new mean is a rank-1 update of the old mean, and the rank of the new covariance Σ_i is one less than that of Σ_{i-1}. The posterior mean has maximum posterior probability (minimal regularized loss), and is thus our new point estimate. Choosing a unit variance prior $\Sigma_{i-1} = I \otimes I$ recovers one of the oldest quasi-Newton algorithms: *Broyden's method* (1965):

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)s_i^\top}{s_i^\top s_i}.$$

Broyden's method does not satisfy the third requirement of Section 1: the updated estimate is, in general, not a symmetric matrix. A supposed remedy for this problem, and in fact the *only* rank-1 update rule that obeys Equation (4) (Dennis and Moré, 1977) is the *symmetric rank 1 (SR1)* method (Davidon, 1959; Broyden, 1967):

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)(y_i - B_{i-1}s_i)^\top}{s_i^\top (y_i - B_{i-1}s_i)}.$$

The SR1 update rule has acquired a controversial reputation (e.g., Nocedal and Wright, 1999, §6.2): While some authors report good results using this method, others note that it is unstable and overly limited. Our Bayesian interpretation identifies the SR1 formula as Gaussian regression with a data-dependent prior variance involving V_{i-1} with

$$V_{i-1}s_i = (y_i - B_{i-1}s_i).$$

Given the explicitly Gaussian prior of Equation (6), there is no rank 1 update rule that gives a symmetric posterior. This blemish of rank-1 updates is also reflected in Equation (7): Uncertainty drops only in the "row", or "primal" subspace of the belief (the right hand side of the Kronecker product in the covariance). While this still means uncertainty goes toward 0 over time, it does so in an asymmetric way.

2.1 Symmetric Estimates, but no Symmetric Beliefs

Many classic quasi-Newton methods provide symmetric estimators for B. Is it possible to encode the Hessians symmetry directly in the probabilistic belief? The proper probabilistic way to do so is to include an additional factor

$$\delta(\Delta\overrightarrow{B} - \overrightarrow{0}) = \lim_{\tau \to 0} \mathcal{N}(\overrightarrow{0}, \Delta\overrightarrow{B}, \tau I) \tag{8}$$

using Δ, the *antisymmetry* operator—the linear map defined through

$$\Delta\overrightarrow{X} = \frac{1}{2}\overrightarrow{(X - X^\top)}.$$

Since this is a linear map, the resulting posterior is analytic, and Gaussian. But the rank of Δ is $1/2 \cdot N(N-1)$ (e.g., Lütkepohl, 1996, §4.3.1, Equations 12 & 20), so the corresponding update rule does not obey the first requirement of Section 1. So, while it is possible to encode symmetry, it is not practical. However, the structure of Equation (7) hints at another idea, which in fact turns out to give rise to the most popular quasi-Newton methods. We introduce a second, *dual* observation (dual, as in "dual vector space", not as in "primal-dual optimization"), using the operator $\mathcal{S}_\triangleleft = (s_i \otimes I)$, the dual of $\mathcal{S}_\triangleright$,

$$p(y_i^\top \,|\, B, s_i^\top) = \delta(y_i^\top - s_i^\top B) = \lim_{\gamma \to 0} \mathcal{N}\left[y_i^\top; \mathcal{S}_\triangleleft^\top \overrightarrow{B}, (\gamma \otimes V_i)\right]. \tag{9}$$

Note that the limit uses V_i, not V_{i-1} as in Equation (4). The posterior has mean

$$\begin{aligned}
\overrightarrow{B}_i &= \overrightarrow{B}_\triangleright + \Sigma_\triangleright \mathcal{S}_\triangleleft \left(K_\triangleleft + \gamma I \otimes V_\triangleright\right)^{-1} \left(\overrightarrow{y_i^\top} - \mathcal{S}_\triangleleft^\top \overrightarrow{B}_\triangleright\right) \\
&= \overrightarrow{B}_{i-1} + \left(I \otimes \frac{V_{i-1} s_i}{s_i^\top V_{i-1} s_i + \beta}\right) \overrightarrow{(y_i - B_{i-1} s_i)} \\
&\quad + (V_{i-1} s_i \otimes V_i)\left[(s_i^\top V_{i-1} s + \gamma) \otimes V_i\right]^{-1} \overrightarrow{\left[y_i^\top - s_i^\top \left(B_{i-1} + \frac{y_i - B_{i-1} s_i}{s_i^\top V_{i-1} s_i + \beta} s_i^\top V_{i-1}\right)\right]}.
\end{aligned}$$

The calculation for the posterior covariance can be reduced to a simple symmetry argument. Expanding the Kronecker products as before, we find that the posterior after both primal and dual observation is a Gaussian with mean and covariance

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1} s_i) s_i^\top V_{i-1}^\top}{s_i^\top V_{i-1} s_i} + \frac{V_{i-1} s_i (y_i - B_{i-1} s_i)^\top}{s_i^\top V_{i-1} s_i} - \frac{V_{i-1} s_i (s_i^\top (y_i - B_{i-1} s_i)) s_i^\top V_{i-1}}{(s_i^\top V_{i-1} s_i)^2}, \tag{10}$$

$$\Sigma_i = \left(V_{i-1} - \frac{V_{i-1} s_i s_i^\top V_{i-1}}{s_i^\top V_{i-1} s_i}\right) \otimes V_i = V_i \otimes V_i. \tag{11}$$

The posterior mean is clearly symmetric if B_{i-1} is symmetric (as V_{i-1} is symmetric by definition). Choosing the unit prior $\Sigma_{i-1} = I \otimes I$ once more, Equation (10) gives what is known as *Powell's* (1970) *symmetric Broyden (PSB)* update. Equation (10) has previously been known to be the most general form of a symmetric rank 2 update obeying the quasi-Newton equation (1) and minimizing a Frobenius regularizer (Dennis and Moré, 1977). This old result is a corollary of our derivations. But note that symmetry only extends to the mean, not the entire belief: In contrast to the posterior generated by Equation (8), samples from this posterior are, with probability 1, not symmetric. Of course, they can be projected into the space of symmetric matrices by applying the symmetrization operator Γ defined by

$$\Gamma \overrightarrow{X} = \frac{1}{2} \overrightarrow{\left(X + X^\top\right)} \qquad (\text{note that } I = \Gamma + \Delta; \Gamma\Delta = 0). \tag{12}$$

Since Γ is a symmetric linear operator, the projection of any Gaussian belief $\mathcal{N}(X; X_0, \Sigma)$ onto the space of symmetric matrices is itself a Gaussian $\mathcal{N}(\Gamma X; \Gamma X_0, \Gamma\Sigma\Gamma)$. But symmetrized samples from the posterior of Equations (10), (11) do not necessarily obey the quasi-Newton Equation (1). While Equation (9) does convey useful information, it is not equivalent to encoding symmetry. It is cheaper, but also weaker, than using the likelihood (8), which encodes the full information afforded by symmetry.

2.2 Positive Definiteness: Meaning or Decoration?

So quasi-Newton methods ensure symmetry in the maximum of the posterior, but not the posterior itself. What about desideratum 4 from Section 1, positive definiteness? Consider choosing $V_{i-1} = B$. The prior (6) then turns into the *non-Gaussian* form

$$p(B) \propto |B|^{-N^2/2} \cdot \exp\left[-\frac{1}{2}\left(N - 2\operatorname{tr}(B_{i-1}B^{-1}) + \operatorname{tr}(B_{i-1}B^{-1}B_{i-1}B^{-1})\right)\right]. \tag{13}$$

This is an intriguing prior. The last term in the exponential has the form of the natural Riemannian metric on the space of positive definite real matrices (Savage, 1982), and may also remind some readers of the Wishart distribution. But the second term in the exponential means this prior is broader than the Wishart. It is not well-defined for degenerate matrices, and it is not clear whether it is proper. It is thus surprising to discover that it engenders the two most popular quasi-Newton methods: If we use the quasi-Newton equation (1) a second time to replace $V_{i-1}s = y$, Equation (13) gives the *DFP* method (Davidon, 1959; Fletcher and Powell, 1963)

$$B_i = B_{i-1} + \frac{(y_i - B_{i-1}s_i)y_i^\top}{s_i^\top y_i} + \frac{y_i(y_i - B_{i-1}s_i)^\top}{y_i^\top s_i} - \frac{y_i(s_i^\top(y_i - B_{i-1}s_i))y_i^\top}{(y_i s_i)^2}.$$

And, if we exchange in the entire preceding derivation $s \leftrightarrow y$, $B \leftrightarrow B^{-1}$, $B_{i-1} \leftrightarrow B_{i-1}^{-1}$, then we arrive at the *BFGS* method (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), which ranks among the most widely used algorithms in numerical optimization. Table 1 gives an overview over the relationships between quasi-Newton methods described so far. It also mentions methods by Greenstadt (1970) and McCormick (see Pearson, 1969) which contain the "missing links" in this table but have not been mentioned so far. These works are also briefly discussed by Dennis and Moré (1977), from where we take these citations.

DFP and BFGS owe much of their popularity to the fact that the updated $B_{i,\mathrm{DFP}}$ and $B_{i,\mathrm{BFGS}}^{-1}$ are guaranteed to be positive definite whenever $B_{i-1,\mathrm{DFP}}$ and $B_{i-1,\mathrm{BFGS}}^{-1}$ are positive definite, respectively, and additionally $y_i^\top s_i > 0$. How helpful is this property? It is relatively straightforward to extend a theorem by Dennis and Moré (1977) to find that, assuming B_{i-1} is positive definite, the posterior mean of Equation (10) is positive definite if, and only if,

$$\begin{aligned} & 0 < (y_i^\top B_{i-1}^{-1} V_{i-1} s_i)^2 + (y_i - B_{i-1}s_i)^\top B_{i-1}^{-1} y_i \cdot s_i^\top V_{i-1} B_{i-1}^{-1} V_{i-1} s_i \\ \Leftrightarrow \quad & 0 < s_i^\top V_{i-1}[B_{i-1}^{-1} y_i y_i^\top B_{i-1}^{-1} - y^\top B_{i-1}^{-1} y_i + s_i^\top y_i] V_{i-1} s_i. \end{aligned}$$

If the prior covariance is not to depend on the data, it is thus impossible to guarantee positive definiteness in this framework—BFGS and DFP circumvent this conceptual issue by choosing $V_{i-1} = B$, then applying Equation (1) a second time. But, even casting aside such philosophical reservations, our analysis also casts doubt upon the efficacy of the way in which DFP and BFGS achieve positive definiteness: Equation (13) does not exclude indefinite matrices; in fact it assigns positive density to every invertible matrix. For example, under a mean $B_{i-1} = I$, the indefinite matrix $B = \operatorname{diag}(1, -1)$ is assigned $p(B) \propto \exp(-2)$. DFP and BFGS achieve positive definiteness, not by including additional information, but by manipulating the prior such that the *MAP estimator* (not the belief) happens to be positive definite. These observations do not rule out any utility of guaranteeing positive definiteness in this way, and the prior (13) deserves closer study. But these results suggest there is less value in the positive definiteness guarantee of DFP and BFGS than previously thought.

likelihood	prior	inferring B	inferring B^{-1}
$y = Bs$	$V = I$	Broyden (1965)	
	$V = B$	Pearson (1969)	McCormick
	$Vs = (y - Bs)$	SR1 (Davidon, 1959)	
$y = Bs \wedge y^\top = Bs^\top$	$V = I$	PSB (Powell, 1970)	Greenstadt (1970)
	$V = B$	DFP	BFGS

Table 1: Overview over probabilistic interpretations of various quasi-Newton methods, based on the combination of prior and likelihood. The "McCormick" entry refers to a note in Pearson (1969), see also Dennis and Moré (1977). The SR1 method is identical for inference on either B or its inverse. The abbreviation DFP stands for Davidon (1959), Fletcher and Powell (1963), while BFGS stands for Broyden (1970), Fletcher (1970), Goldfarb (1970) and Shanno (1970).

2.3 Rank M Updates

The classical quasi-Newton algorithms update the mean of the belief at every step in a rank 2 operation, then, implicitly, reset their uncertainty in the next step, thereby discarding information acquired earlier. Albeit inelegant from a Bayesian point of view, this scheme is still a good idea given other aspects of the framework: Since the quasi-Newton likelihood models the objective function as a quadratic, with constant Hessian everywhere, strict Bayesian inference from this prior would simply average over the Hessian everywhere, which is obviously not a good model. But it is instructive to consider the effect of encoding more than just the most recent observation. It is straightforward to extend Equation (4) to observations (Y, S) from several line searches:

$$Y_{nm} = \nabla_n f(x_{i_m}) - \nabla_n f(x_{i_m - 1}), \qquad S_{nm} = x_{i_m, n} - x_{i_m - 1, n}.$$

Given a prior $p(B) = \mathcal{N}(B; B_0, V_0)$, the Gaussian posterior then has mean and covariance

$$\begin{aligned} B_i &= B_0 + (Y - B_0 S)(S^\top V_0 S)^{-1} S^\top V_0 + V_0 S (S^\top V_0 S)^{-1} (Y - B_0 S)^\top \\ &\quad - V S (S^\top V_0 S)^{-1} (S^\top (Y - B_0 S)) (S^\top V S)^{-1} S^\top V_0, \\ \Sigma_i &= \left(V_0 - V_0 S (S^\top V_0 S)^{-1} S^\top V_0\right) \otimes \left(V_0 - V_0 S (S^\top V_0 S)^{-1} S^\top V_0\right). \end{aligned} \tag{14}$$

Here, the absence of information about the symmetry of the Hessian becomes even more obvious: No matter the prior covariance V_0, because of the term $S^\top Y$ in the second line of Equation (14), the posterior mean is not in general symmetric, *unless* $Y = BS$, (e.g., if the objective function is in fact a quadratic). See Section 4, particularly Figure 3, for a simple experiment with this parametric algorithm.

2.4 Summary

The preceding section showed that quasi-Newton algorithms, including the state-of-the-art BFGS and DFP algorithms, can be interpreted as approximate Bayesian regression from the primal and dual likelihood of Equations (4) and (9) under varying priors, in the following sense: At each quasi-Newton step, fix a Gaussian prior ad hoc, update the mean, then "forget" the covariance update.

Two particularly interesting observations concern the way in which the desiderata of symmetry and positive definiteness of the MAP estimator are achieved in these algorithms. Symmetry is encoded via dual observations, which is a useful but imperfect shortcut. Similarly, positive definiteness is just guaranteed for the mode, not the entire support of the posterior distribution. There may well be a non-obvious value to the "scale-free" prior of Equation (13) (see also Nocedal and Wright, 1999, Equations 6.11–6.13), but our analysis raises doubt on whether the proven good performance of BFGS and DFP is actually down to positive definiteness, or to a different effect involving the broader non-Gaussian prior (13).

3. A Nonparametric Bayesian Quasi-Newton Method

Section 2 used the probabilistic perspective to gain novel insight into classical methods. It showed that quasi-Newton methods can be interpreted as Gaussian regressors using algebraic structure to weaken prior knowledge, in exchange for lower computational cost. In this second part of the paper we depart from the traditional framework to construct a nonparametric, Bayesian quasi-Newton method, de novo. To motivate this effort, notice some other deficiencies of DFP/BFGS not directly connected to computational cost: Equation (4) assumes that the function is (locally) a quadratic. Old observations collected "far" from the current location (in the sense that a second order expansion is a poor approximation) may thus be useless or even harmful. The fact that the function is not quadratic should be part of the model. On an only slightly related point, individual line searches typically involve several evaluations of the objective function f and its gradient; but the algorithms only make use of one of those (the last one). This is clearly wasteful, but even the exact Bayesian parametric algorithm of Section 2.3 has this problem, because the matrix S of several observations along one line search has rank 1, so the inverse of $S^\top V_0 S$ is not defined. The following section will address all these issues, using the framework of nonparametric Gaussian process regression to model the objective function more closely.

3.1 A Nonparametric Prior

Defining a prior for the function $B : \mathbb{R}^N \to \mathbb{R}^{N\times N}$, we choose a set of N^2 correlated Gaussian processes. The mean function is assumed to be an arbitrary integrable function $B_0(x)$ (in our implementation we use a constant function, but the analytic derivations do not need to be so restrictive). The core idea is to assume that the covariance between the element B_{ij} at location[1] $x_\triangledown$ and the entry $B_{k\ell}$ at location $x_\triangle$ is

$$\operatorname{cov}\left(B_{ij}(x_\triangledown), B_{k\ell}(x_\triangle)\right) = k_{ik}(x_\triangledown^\top, x_\triangle^\top) k_{j\ell}(x_\triangledown, x_\triangle) = \left(k \otimes k\right)_{(ij)(k\ell)}(x_\triangledown, x_\triangle)$$

with an $N \times N$ matrix of kernels, k. To give a more concrete intuition: In our implementation we use one joint squared exponential kernel for all elements. I.e.

$$k_{ij}(x_\triangledown, x_\triangle) = V_{ij} \exp\left(-\frac{1}{2}(x_\triangledown - x_\triangle)^\top \Lambda^{-1}(x_\triangledown, x_\triangle)\right) \tag{15}$$

1. We use the notation $x_\triangle$ and $x_\triangledown$ (read "x up" and "x down") to denote two separate, arbitrary elements of the input space. The combinations x_* and x^* or x and x', or x_1 and x_2 are more widely used in the literature. But since this document is heavy on indices, we prefer this notation as it prevents confusion over sub- and superscripts and named indices.

with a positive definite matrix V and length scales Λ. This means

$$\operatorname{cov}(B_{ij}(x_\triangledown^\top), B_{k\ell}(x_\triangle)) = V_{ik}V_{j\ell}\exp\left(-\frac{1}{2}(x_\triangledown - x_\triangle)^\top 2\Lambda^{-1}(x_\triangledown - x_\triangle)\right),$$

and in particular, the marginal variance of any particular local Hessian element is

$$\operatorname{var}(B_{ij}(x_\triangledown)) = \operatorname{cov}(B_{ij}(x_\triangledown), B_{ij}(x_\triangledown)) = V_{ii}V_{jj}.$$

So the prior variance of element B_{ij} is $V_{ii}V_{jj}$, *not* V_{ij}, as one might think at first. Similarly, the length scale on which the elements change is not Λ, but $\Lambda/2$. So it is not possible to encode separate signal scales for the off-diagonal elements of the Hessian in this framework. They are determined entirely by the scales of the diagonal elements. Even so, if V is diagonal, then beliefs between any two different elements of B are independent.

Other kernels can of course be chosen; but it will become clear that an important practical requirement is the ability to efficiently integrate the kernel. This is feasible, though nontrivial, with the squared exponential kernel.

3.2 Line Integral Observations

For the Hessian $B(x)$ of a general function f, the quasi-Newton equation (4) is only a zeroth order approximation (a second-order approximation to f itself), assuming a constant Hessian everywhere. In our treatment, we will replace this approximate statement with its exact version: We observe the value of the *line integral* along the path $r^i : [0,1] \to \mathbb{R}^N$, $r^i(t) = x_{i-1} + t(x_i - x_{i-1})$,

$$Y_{ni} = \sum_m \int_{r^i_m} B_{nm}(x)\,\mathrm{d}x_m.$$

Note that, for scalar fields ϕ_i with $B_{im} = \nabla_m \phi_i$, such as the gradient $\phi_i = \nabla_i f$, it follows from the chain rule that (the following derivations again use the sum convention defined in Section 1.2)

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_i(r^j(t)) = \nabla_m \phi_i(r^j(t))\frac{\partial r^j_m(t)}{\partial t} = B_{im}(r^j(t))\partial_t r^j_m(t).$$

Thus, our line integral obeys

$$\begin{aligned} Y_{ij} &= \int_{r^j} B_{im}(x)\,\mathrm{d}x_m = \int_0^1 B_{im}(r^j(t))\cdot\partial_t r^j_m(t)\,\mathrm{d}t \\ &= \int_0^1 \partial_t \phi_i(r^j(t))\,\mathrm{d}t = \phi_i(r^j(1)) - \phi_i(r^j(0)). \end{aligned} \tag{16}$$

This is the classic result that line integrals over the gradients of scalar fields are independent of the path taken, they only depend on the starting and end points of the path. In particular, our path satisfies $\partial_t r^j_m(t) = S_{jm}$ (its derivative is constant), and our line integral can be written as

$$\begin{aligned} Y_{ij} &= \int_0^1 B_{im}(r^j(t))S_{jm}\,\mathrm{d}t = \delta_{ik}\cdot S_{jm}\int_0^1 B_{km}(t^j)\,\mathrm{d}t^j, \\ \overrightarrow{Y} &= \left[I \otimes \left(S^\top \odot \int_t\right)\right]\overrightarrow{B} \equiv \mathfrak{S}_\triangleright \overrightarrow{B}. \end{aligned} \tag{17}$$

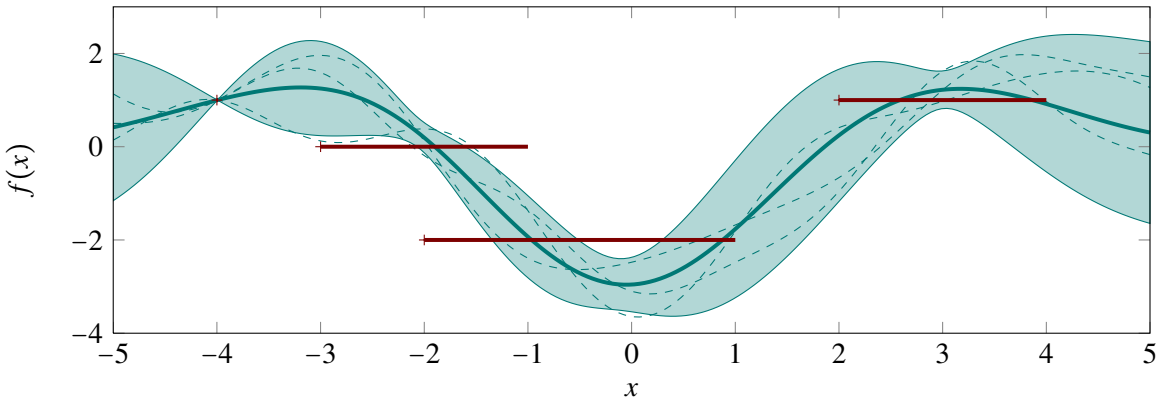


Figure 1: One-dimensional Gaussian process inference from integral observations (squared exponential kernel). Four observations, average values (integral value divided by length of integration region) and integration regions denoted by bars. Posterior mean in thick green, two standard deviations as shaded region, three samples as dashed lines. The left-most integral is over a very small region, which essentially reduces to the classical case of a local observation. Corresponding integrals over the mean, and each sample, are consistent with the integral observations.

where $\odot$ denotes the Hadamard, or element-wise, product $(a \odot b)_{k\ell} = a_{k\ell} b_{k\ell}$. In words: For every projection $S_{j:}$ of the *rows* of B, there are N one-dimensional *functions* $(Bs)_i(t^j) : \mathbb{R} \to \mathbb{R}$. Each of those functions are integrated from 0 to 1 (this is an affine projection onto the space of integrals over $[0,1]$). This amounts to taking *each component* $B_{ij}(t)$ of each projection and applying the integral-projection—hence the Hadamard product. We write the likelihood as

$$p(Y \,|\, B(x), \mathfrak{S}_{\triangleright}) = \lim_{\beta \to 0} \mathcal{N}\left[Y; \mathfrak{S}_{\triangleright}^{\top} \overrightarrow{B}, (k \otimes \beta I_M)\right],$$

using the linear operator $\mathfrak{S}_{\triangleright}$ defined in Equation (17). An interesting aspect to note is that, while path-independence holds for the ground-truth integrals of Equations (16), the prior covariance of Equation (15) does not encode this fact. The prior used here is more conservative than necessary, in the sense that it assigns nonzero probability mass on algebraically impossible functions, in exchange for lower computational cost. This is not unlike the aspects of parametric quasi-Newton methods discussed in Sections 2.1 and 2.2, where nonzero probability mass is assigned to the algebraically impossible case of non-symmetric Hessians. See Section 3.5 for more on this issue.

3.3 Gaussian Process Inference from Integral Observations

Because the Gaussian exponential family is closed under linear transformations, Gaussian process inference is analytic under any linear operator. Since integration is a linear operation, it is a corollary that Gaussian process inference is possible, in closed form, from integral observations. Nevertheless, this idea has only rarely been used in the literature (e.g., by Minka, 2000b). So we briefly digress here to introduce it in detail. Let there be a function $f(x) : \mathbb{R} \to \mathbb{R}$ (extension to multi-

variate functions is straightforward). Assume a Gaussian process prior, with mean function $\mu(x)$, covariance function (kernel) k. We observe, up to Gaussian noise, the value of a definite integral

$$y = \xi + \int_a^b f(x)\,\mathrm{d}x; \qquad \xi \sim \mathcal{N}(0, \sigma^2).$$

What is the posterior? We construct the answer from the finite-dimensional case, then take the Riemann limit. Consider observing the noisy weighted sum of N Gaussian variables, with weights δm_i:

$$y = \xi + \sum_i^N \delta m_i f_i \equiv \xi + m^\top f; \qquad p(f) = \mathcal{N}(\mu; K).$$

The posterior can be found as above, by "completing the square"

$$p(f\,|\,y) = \mathcal{N}(\Psi(K^{-1}\mu + \sigma^{-2} m y), \Psi)$$

with the covariance

$$\Psi = (K^{-1} + \sigma^{-2} m m^\top)^{-1} = K - \frac{K m m^\top K}{\sigma^2 + m^\top K m}.$$

Now consider the limit transition $N \to \infty$, such that the weights δm_i converge to a measure $m(x_i)\,\mathrm{d}x_i$ ($m = 1$ is a special case). We get

$$Km = K_{ij} m_j \to \int_a^b k(x_i, x_j)\,\mathrm{d}m(x_j) \quad \text{and}$$
$$m^\top K m = m_i K_{ij} m_j \to \iint_a^b k(x_i, x_j)\,\mathrm{d}m(x_i)\,\mathrm{d}m(x_j).$$

The mean has the form

$$\mu - \frac{K m m^\top \mu}{\sigma^2 + m^\top K m} + \sigma^{-2} K m \left(1 - \frac{m^\top K m}{\sigma^2 + m^\top K m}\right) y = \mu + K m \left(\frac{y - m^\top \mu}{\sigma^2 + m^\top K m}\right)$$

which, in the limit, transforms to

$$\mu + \frac{y - \int_a^b \mu(\tilde{x})\,\mathrm{d}m(\tilde{x})}{\sigma^2 + \iint_a^b k(x_i, x_j)\,\mathrm{d}m(x_i)\,\mathrm{d}m(x_j)} \int_a^b k(x_i, x_j)\,\mathrm{d}m(x_i).$$

Figure 1 gives a toy 1D example for intuition.

3.4 Posterior on Hessians

Using an argument entirely analogous to that of Section 2, we find that the primal posterior after M observations has mean function

$$\begin{aligned}
&\overrightarrow{B}_\triangleright(x_\forall) \\
&= \overrightarrow{B}_0(x_\forall) + (\Sigma \mathfrak{S}_\triangleright)(x_\forall)(K + (k \otimes \beta I))^{-1}(Y - \mathfrak{S}_\triangleright^\top \overrightarrow{B}_0) \\
&= \overrightarrow{B}_0 + \left[k \otimes k\left(S \odot \int_t\right)\right]\left(k \otimes \left(S \odot \int\right)^\top k\left(S \odot \int\right) + k \otimes \beta I\right)^{-1}(Y - \mathfrak{S}_\triangleright^\top \overrightarrow{B}_0).
\end{aligned}$$

The terms of this equation can be further identified using the Gram matrix

$$\begin{aligned}\left(S \odot \int\right)^{\top} k \left(S \odot \int\right)\Big|_{j\ell} &= \int_0^1 \mathrm{d}t^j \int_0^1 \mathrm{d}t^\ell S_k^j k_{km}(x_\forall(t^j), x_\triangle(t^\ell)) S_m^\ell \\ &= S_{jk}\left[\iint_0^1 k_{km}(x_\forall(t^j), x_\triangle(t^\ell))\,\mathrm{d}t^j\,\mathrm{d}t^\ell\right] S_{m\ell} \\ &\equiv \mathfrak{K} \in \mathbb{R}^{M\times M},\end{aligned} \tag{18}$$

the integrated kernel map

$$\begin{aligned}k\left(S \odot \int\right)\Big|_{kj}(x_\forall) &= \int_0^1 k_{km}(x_\forall, x_\triangle(t^j))\,\mathrm{d}t^j S_{jm} \\ &\equiv \mathfrak{k}(x_\forall) \in \{\mathbb{R}^N \to \mathbb{R}^{N\times M}\},\end{aligned} \tag{19}$$

and the integrated mean function

$$\mathfrak{S}_{\triangleright}^{\top} \overrightarrow{B}_0\Big|_{mk} = S_{jk} \int_0^1 B^0_{mj}(x(t^k))\,\mathrm{d}t^k \equiv \mathfrak{B} \in \mathbb{R}^{N\times M}. \tag{20}$$

These objects are homologous to concepts in canonical Gaussian process inference: $\mathfrak{B}_{0,nm}$ is the n-th mean prediction along the m-th line integral observation. $\mathfrak{k}_{nm}(x_\forall)$ is the covariance between the n-th column of the Hessian at location $x_\forall$ and the m-th line-integral observation. $\mathfrak{K}_{pq}$ is the covariance between the p-th and q-th line integral observations. The derivations for the covariance are similar and contain the same terms. Together with the dual observation, we arrive at a posterior, which has mean and covariance functions

$$\begin{aligned}B_\diamond(x_\forall) = B_0(x_\forall) &+ (Y - \mathfrak{B}_0)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\forall) + \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}(Y - \mathfrak{B}_0)^{\top} \\ &- \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}S^{\top}(Y - \mathfrak{B}_0)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\forall), \\ \Sigma_\diamond(x_\forall, x_\triangle) = \left[k(x_\forall^{\top}, x_\triangle^{\top}) - \mathfrak{k}(x_\forall^{\top})\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\triangle)\right] &\otimes \left[k(x_\forall, x_\triangle) - \mathfrak{k}(x_\forall)\mathfrak{K}^{-1}\mathfrak{k}^{\top}(x_\triangle)\right].\end{aligned}$$

The actual numerical realisation of this nonparametric method involves relatively tedious algebraic derivations, which can be found in Appendix A.

An important aspect is that, because k is a positive definite kernel, unless two observations are exactly identical, $\mathfrak{K}$ has full rank M (the number of function evaluations), even if several observations take place within one shared 1-dimensional subspace. So it is possible to make full use of *all* function evaluations made during line searches, not just the last one, as in the parametric setting of existing quasi-Newton methods. Figure 2 uses another toy setting to give an intuition for why this matters. Just as in Section 2.3, it is clear that the posterior mean is not in general a symmetric matrix. So we project into the space of symmetric matrices using the arguments surrounding Equation (12).

A downside is that evaluating the mean function involves finding the inverse of $\mathfrak{K}$, at cost $\mathcal{O}(M^3)$. Two aspects of numerical optimization make this issue less problematic than one might think. First, solving an optimization problem takes finite time, often just a few hundred evaluations; so the cubic cost in M is often manageable. Where it is not, note that, because optimization proceeds along a trajectory through the parameter space, old observations tend to have low covariance with the Hessian at the current location, and thus a small effect on the local mean estimate. So they can often simply be ignored. The simplest possible way to do so is to just throw away all observations

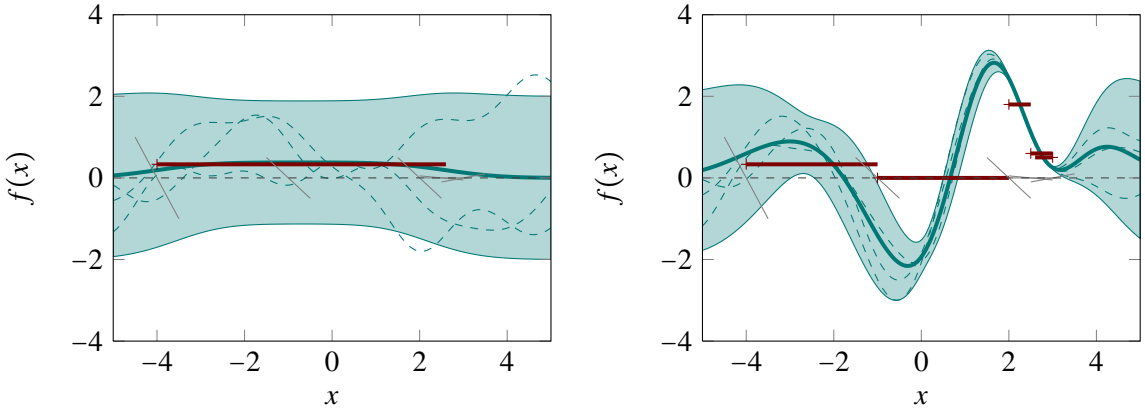


Figure 2: Simulated line search, a toy problem to elucidate why it helps to use all line search observations instead of only the first and last ones. Observations at locations $X = [-4; -1; 2; 2.5; 3; 2.6]$, observed 1D Gradients of $\nabla f(X) = [-2; -1; -1; -0.1; 0.2; -0.001]$. **Left:** Traditional inference based on only the first and last observations. **Right:** Our nonparametric model can use all observations. Gaussian process posterior with thick mean and two standard deviations marginal variance as shaded region, as well as three samples as dashed lines. Effective observations $y_i/(x_i - x_{i-1})$ as bars. Gradient values as thin lines on the abscissa for intuition. The posterior from all observations captures much more structure, and in particular a different mean estimate at the end of the line search ($x = 2.6$), where its value defines the next search direction.

older than some memory bound M_0. This is the approach of the L-BFGS method (Nocedal, 1980). Since the regression framework quantifies the contribution of each observation to the prediction, in the vector $\natural\mathfrak{K}^{-1}$, we can also use the relative sizes of these elements to order past observations and discard those ranked below M_0.

3.5 Diversion: Naïve Gaussian Regression is Too Costly

The discussion in Section 2 established that, with a few caveats (Section 2.2), quasi-Newton methods are Gaussian regressors; and we then extended to nonparametric Gaussian process inference. Importantly, the prior from Section 3.1 is over the elements of the Hessian, and gradient observations are integrals of this function. One may wonder why we did not just start with a Gaussian process prior on the objective function f and used observations of the gradient to infer the Hessian directly from there. This is possible because differentiation, like integration, is a linear operation: Under a Gaussian process prior on f with kernel k^f and mean function μ^f, the mean function of the prior belief over the gradient is $\mu_{\nabla f} = \nabla \mu^f$, and the covariance between elements of ∇f at two

different points $x^{\triangledown}$ and $x^{\triangle}$ is (Rasmussen and Williams, 2006, §9.4)

$$\operatorname{cov}\left(\frac{\partial f(x^{\triangledown})}{\partial x_i^{\triangle}}, \frac{\partial f(x^{\triangledown})}{\partial x_j^{\triangledown}}\right) = \frac{\partial^2 k(x^{\triangledown}, x^{\triangle})}{\partial x_i^{\triangle}, \partial x_j^{\triangledown}}, \qquad \text{which, for an SE kernel, is} \tag{21}$$
$$= \left(\frac{1}{\lambda_j^2}\delta_{ij} + \frac{(x_i^{\triangle} - x_i^{\triangledown})(x_j^{\triangle} - x_j^{\triangledown})}{\lambda_i^2 \lambda_j^2}\right) k_{SE}(x^{\triangle}, x^{\triangledown}).$$

The covariance between elements of the Hessian and elements of the gradient is

$$\operatorname{cov}\left(\frac{\partial^2 f(x^{\triangledown})}{\partial x_i^{\triangle}\, \mathrm{d}x_k^{\triangle}}, \frac{\partial f(x^{\triangledown})}{\partial x_j^{\triangledown}}\right) = \frac{\partial^2 k(x^{\triangledown}, x^{\triangle})}{\partial x_i^{\triangle}, \partial x_j^{\triangledown}}, \qquad \text{which, for an SE kernel, is}$$
$$= \left(\frac{\delta_{ik}(x_j^{\triangle} - x_j^{\triangledown}) + \delta_{jk}(x_i^{\triangle} - x_i^{\triangledown})}{\lambda_i^2 \lambda_j^2} - \frac{(x_k^{\triangle} - x_k^{\triangledown})}{\lambda_k^2}\right) k_{SE}(x^{\triangle}, x^{\triangledown}).$$

So, given observations of the gradient at M points x^m, we can evaluate the mean over the elements of the Hessian $B(x^*)$ as

$$\hat{B}_{ik}(x^*) = \mu_B(x^*) + \operatorname{cov}(B_{ik}(x^*), \nabla_j f(x^m)) K^{-1}_{(jm)(\ell q)} (\nabla_\ell f(x^q) - \mu_{\nabla_\ell f}(x^q)),$$

with a Gram matrix $K \in \mathbb{R}^{MN \times MN}$ of elements $K_{(jm)(\ell q)} = \operatorname{cov}(\nabla_j f(x^m), \nabla_\ell f(x^q))$. From Equation (21) we see that this Gram matrix has specific structure, so it might be possible to construct its inverse faster than in $\mathcal{O}(M^3N^3)$. But even then, this scheme would only provide a belief over the *elements* of B. Since Newton's method requires the *inverse* of B, this mean prediction would still have to be inverted, at cost $\mathcal{O}(N^3)$. This would defeat the point of a quasi-Newton method: constructing a low-rank estimate of the Hessian, and thus a fast estimate of its inverse. If N is small enough to allow for general (cubic) inversion of $\hat{B}$, we might as well just calculate the true Hessian of f and invert that instead. So quasi-Newton methods are not "just" standard Gaussian regression on Hessians. Their key advantage stems from the weaker prior assumptions, as discussed in Sections 2.1 and 2.2, which allow the construction of a low-rank estimate.

4. Experiments

The calculations required by nonparametric quasi-Newton algorithm using the squared-exponential kernel involve exponential functions, error functions, and numerical integrals (see Appendix A for details). A side-effect of these is that this algorithm has slightly lower numerical precision than its predecessors. This issue becomes clear when minimizing quadratic functions (Figure 3), whose constant Hessian voids the modeling advantage of the nonparametric method:[2] The nonparametric algorithm behaves more regularly initially, but towards the end of the optimization process the numerical conditioning of the kernel calculations begins to play a role, offering an advantage to the better conditioned older methods. In real, non-quadratic optimization problems, however, this problem only arises close to the end of optimization, when the algorithm is very close to the optimum. In

2. This is only a diagnostic example. Quadratic functions, whose optimization amounts to solving a positive-definite linear program, are not a realistic use-case for quasi-Newton methods, parametric or not. Specialised methods, like the method of conjugate gradients (Hestenes and Stiefel, 1952), or plain Cholesky decomposition for low-dimensional cases, are better suited for this simple setting.

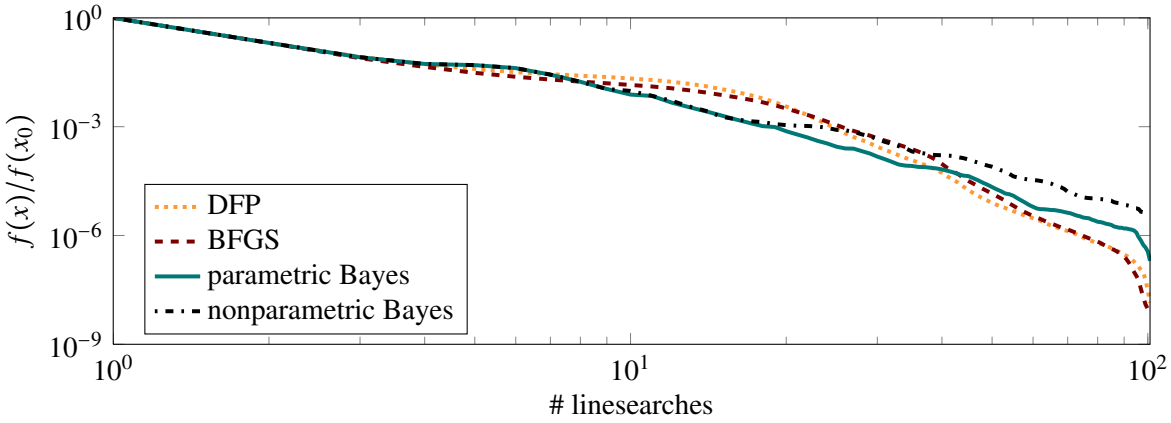


Figure 3: Minimization of a 100-dimensional quadratic. All algorithms shared the same line search method. Averages over 20 sampled problems (see text for details). The two dashed lines in this log-log plot mark linear and quadratic convergence. The Bayesian algorithms converge more regularly and faster initially, but suffer from bad numerical conditioning toward the end of the optimization.

this small region, a local quadratic approximation is valid and the Hessian is essentially constant. In our practical implementation, we thus check for convergence, then pass the learned inverse Hessian to the better conditioned BFGS for the final few steps, in which the learned Hessian barely changes.

For intuition, Figure 4 shows results from a popular two-dimensional test problem—Rosenbrock's polynomial. The plot also shows the mean belief on one element of the Hessian. The availability of this explicit estimate for the entire function is an additional benefit of the nonparametric method.

In problems where the Hessian is not constant everywhere, the nonparametric Bayesian optimizer can sometimes offer drastic advantages over the classical alternatives. Figure 5, left, shows averages of experiments on a 200-dimensional domain. The objective functions is a prior over hyperparameters of a Gaussian process regressor: the logarithm of products of Gamma distributions, with different parameters for each dimension. The right part of the figure shows that the performance advantage is not always so drastic. It was gathered on the corresponding posterior after the addition of 10 datapoints per problem. This makes the objective function less regular, meaning that the optimal Newton path to the minimum has more complex shape, and more line searches are necessary to converge to the minimum.

Figure 6 shows performance on a set of low-dimensional but challenging set of problems: Functions sampled from a Gaussian process with quadratic prior mean, after conditioning on 10 observations of the function's Hessian (drawn separately from a Wishart distribution, to ensure positive definiteness). In all experiments, however, the Bayesian algorithm performs at least as good, and regularly better than its classical competitors. For numerical optimization, even performance gains of a few percent are valuable, because optimization is such a widely encountered problem. Speeding up quasi-Newton methods by 10% means speeding up large parts of machine learning by that amount. Our experiments show that, at least in some cases, the new algorithm offers improvements much beyond that.

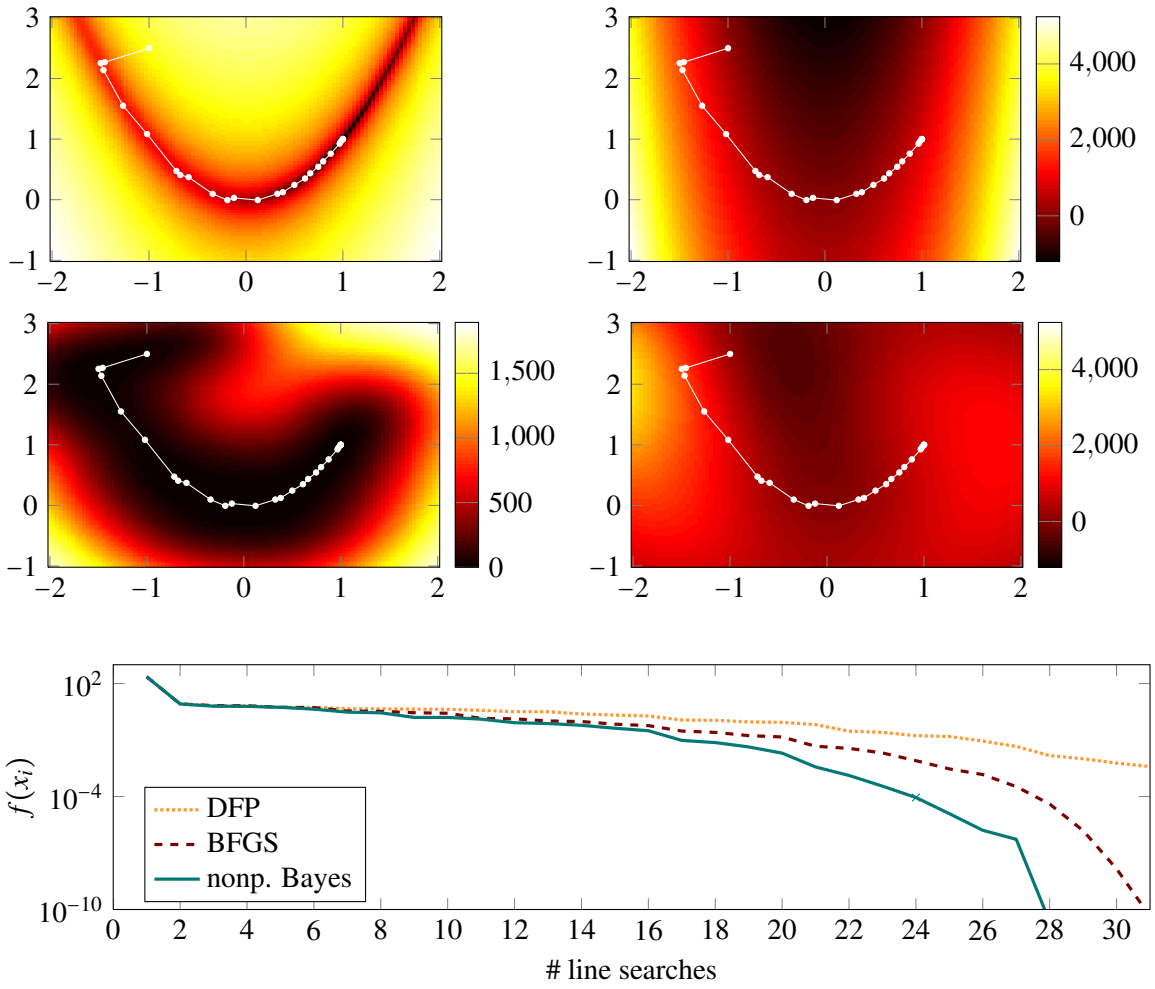


Figure 4: Minimizing Rosenbrock's polynomial, a non-convex function with unique minimum at (1,1). All algorithms start from (-1,2.5). **Top left:** Function values, line search trajectory of the Bayesian algorithm in white. **Top Right:** True value of the (1,1) element of the Hessian (other elements have less interesting structure). **Middle Row:** Two times marginal posterior standard deviation (a.k.a. posterior uncertainty, left) and mean estimate (right) of the Bayesian regressor. Comparing the top right and middle right plots shows good agreement in the regions visited by the algorithm. **Bottom:** function value as function of the number of line searches. The cross after 24 line searches marks the point where the Bayesian method switches to a local parametric model for numerical stability.

4.1 Cost

As pointed out above, the computational complexity of this algorithm, given a diagonal prior mean, is $\mathcal{O}(NM + M^3)$ per update of the search direction, where M is the number of function evaluations used to build the model (which can be controlled ad hoc within the algorithm by excluding redun-

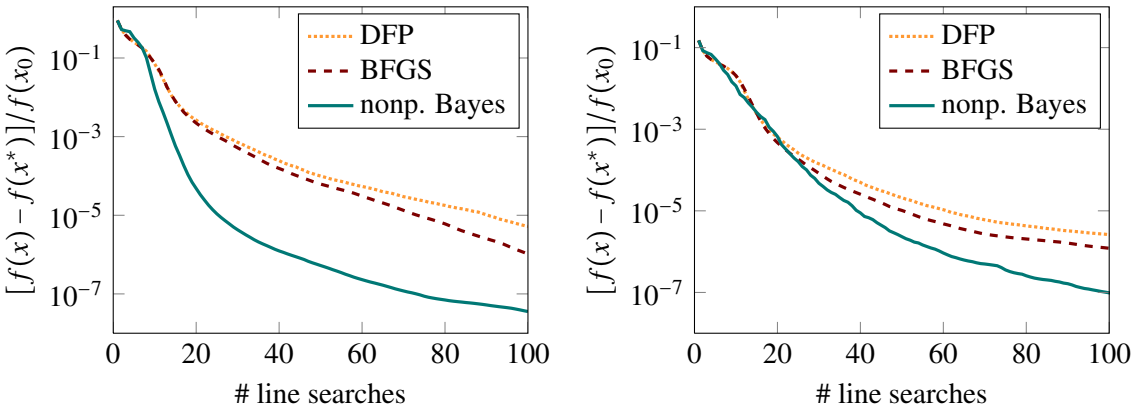


Figure 5: **Left:** Minimizing the 200-dimensional (Gamma) prior over the hyperparameters of a Gaussian process regression modell. **Right:** Minimizing the corresponding *posterior* after the addition of 10 datapoints sampled from the correct model. The datapoints create a more complicated optimization problem in which line searches tend to be shorter, thus reducing the advantage of the Bayesian method gained from superior Hessian estimates. Averages over 20 sampled problems; plotted is the relative distance from initial function value (shared by all algorithms) to the minimum, as a function of the number of line searches (all algorithms use the same line search method).

dant or irrelevant evaluations). This compares to $\mathcal{O}(NM)$ for the corresponding cases of DFP and BFGS. Although the overhead created by the squared-exponential integrals is nontrivial, we found the computational demands of our implementation manageable: In our experiments, the cost of constructing and inverting the matrix $\mathfrak{K}$ was negligible, and could, in very time-sensitive settings, be further reduced by a more efficient implementation.

5. Outlook

In this paper we primarily focused on a better understanding for quasi-Newton methods. For an intuition on the potential of Bayesian formulations of numerical optimization, apart from the new nonparametric algorithm derived in Section 3 and tested in Section 4, consider some potential for future work: Perhaps the most obvious insight is that Gaussian process regression is trivial to extend to noisy evaluations. An upcoming conference paper (Hennig, 2013) will study how this can be used to construct optimizers robust to noise. Repeated integration, and non-Gaussian likelihoods in combination with approximate inference, may allow optimization without gradients, and from only gradient sign observations, respectively. Structured and hierarchical priors are a third direction, offering new avenues for optimization of very high-dimensional functions.

6. Conclusion

We have shown that the most popular quasi-Newton algorithms can be interpreted as approximations to Bayesian regression under Gaussian and other priors. This deepens our understanding of these algorithms. In particular, it emerged that symmetry in the estimators of SR1, PSB, DFP and BFGS,

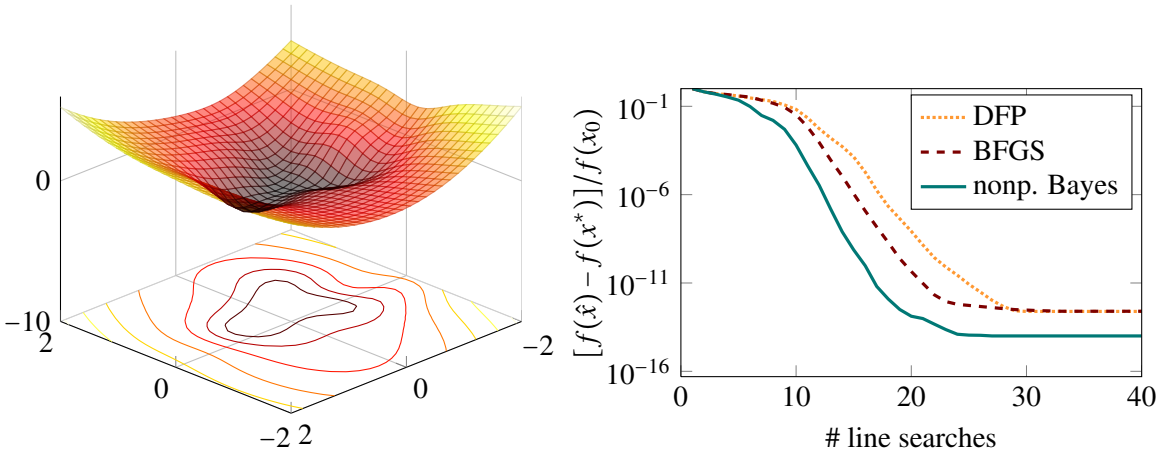


Figure 6: Minimizing randomly generated 4-dimensional analytic functions. **Left:** For illustration. One slice through the $(x, y, 0, 0)$ plane of one of the sampled functions. Neither starting point of the search nor the found optimum lie within this slice, and are thus not shown. **Right:** function values achieved by three numerical optimizers as a function of the number of line searches. All algorithms shared the same line search routine. Plotted is the difference between best function value achieved by any of the optimizers for each function, normalized by the initial function value (which is identical for all algorithms). The lines are averages of the logarithmic values from 10 iid. experiments.

and positive definiteness in those of DFP and BFGS, are encoded in approximate ways which do not capture all available prior information but allow for low computational cost.

As a parallel result, our analysis also gives rise to a new class of Bayesian nonparametric quasi-Newton algorithms. These use a kernel model to learn from all observations in each line-search, explicitly track uncertainty, and thus achieve faster convergence towards the true Hessian. While the new methods are not trivial to understand and implement, their computational cost lies within a constant of that of their predecessors. Our research implementation is available at `http://www.probabilistic-optimization.org/Newton.html`.

Acknowledgments

We would like to thank Christian Schuler, Tom Minka, and Carl Rasmussen and the anonymous reviewers for helpful comments. MK's work is supported by a grant from Microsoft Research Ltd.

Appendix A. Numerical Implementation

As mentioned above, for a concrete implementation, we chose to use the squared exponential kernel (15), and a constant mean function assigning $B_0(x_\forall) = I$ everywhere. It is another advantage of the Bayesian formulation that prior assumptions are directly accessible for analysis: The squared expo-

nential prior amounts to the assumption that the elements of the Hessian vary independently over the parameter space, on one unique set of length-scales Λ. Multiple length scales could be modeled using sums of kernels, but our implementation does not currently offer this option. Changing the length scales Λ amounts to a form of pre-conditioning. The fact that this can be done automatically using methods from machine learning is another advantage of a Bayesian formulation. A naïve approach for such an optimization would be to optimize the hyperparameters by type-II maximum likelihood, as is often done in standard Gaussian process regression. Since this amounts to an optimization problem itself, though, one might hope to find closed form estimators. We will not dwell further on this issue here, leaving it for future work.

A numerical challenge in the implementation arises from the required integrals over squared exponentials. Of the three objects in Equations (18), (19), and (20) only the last one, $\mathfrak{B}$, is truly straightforward, thanks to our choice of constant mean function. The other two will be derived in this section. For this purpose, it is helpful to use an explicit notation for individual line searches: We change the index set from m to (jh): Let observation y_m have been taken as the h-th observation of line search number j. If the line search proceeded along unit direction e_j and started from x_{0j}, then the h-th observation was the difference between the gradients at locations $x_{0j} + (\eta_h - \nu_h)e_j$ and $x_{0j} + \nu_h e_j$.

A.1 $\mathfrak{k}$

The elements of the $N \times M$ matrix $\mathfrak{k}(x_\forall)$ are, (the ellipses are placeholders for the second, analogous part of quadratic forms)

$$\begin{aligned}
\mathfrak{k}^j_{nh}(x_\forall) &= (\eta_h - \nu_h) V_{nm} e^j_m \int_0^1 \exp\left[-\frac{1}{2}(x_\forall - (\nu_h e_j + x_{0j}) - (\eta_h - \nu_{h-1}) t e_j)^\top \Lambda^{-1} \dots \right] \mathrm{d}t \\
&= (Ve^j)_n \exp\left(-\frac{c - b^2/a^2}{2} \right) \frac{1}{a} \int_b^{(\eta_h - \nu_h)a + b/a} \exp\left(-\frac{u^2}{2} \right) \mathrm{d}u \\
&= (Ve^j)_n \exp\left(-\frac{c - b^2/a^2}{2} \right) \sqrt{\frac{\pi}{2a^2}} \left[\operatorname{erf}\left(\frac{(\eta_h - \nu_h)a^2 + b}{\sqrt{2a^2}} \right) - \operatorname{erf}\left(\frac{b}{\sqrt{2a^2}} \right) \right],
\end{aligned}$$

with

$$\begin{aligned}
a &= \sqrt{e_j^\top \Lambda^{-1} e_j} \\
b &= e_j^\top \Lambda^{-1} (x_{0j} + \nu_h e_j - x_\forall) \\
c &= x_\forall^\top \Lambda^{-1} x_\forall - 2 x_\forall \Lambda^{-1} (x_{0j} + \nu_h e_j) + (x_{0j} + \nu_h e_j)^\top \Lambda^{-1} (x_{0j} + \nu_h e_j).
\end{aligned}$$

This involves the error function, for which good double-precision approximations are widely available.

A.2 $\mathfrak{K}$

The $M \times M$ matrix $\mathfrak{K}$ has two types of elements. Along its block diagonal lie covariance between observations collected as part of the same line search. These have the form

$$\begin{aligned}\mathfrak{K}^{ii}_{hk} &= (\eta_h - \nu_h)(\eta_k - \nu_k)(e_i^\top V e_i)\theta^2 \\ &\quad \iint_0^1 \exp\left[-\frac{1}{2}[(\eta_h - \nu_h)t_h e_i + \nu_h e_i + x_{0i} - (\alpha_k - \nu_k)t_k e_i - \nu_k e_i - x_{0i}]^\top \Lambda^{-1}[\dots] \right] \mathrm{d}t_h \, \mathrm{d}t_k \\ &= e_i^\top V e_i \theta^2 \int_{\nu_h}^{\eta_h} \int_{\nu_k}^{\alpha_k} \exp\left[-\frac{(u_h - u_k)^2}{2\sigma_i^2} \right] \mathrm{d}u_h \, \mathrm{d}u_k.\end{aligned}$$

So these terms are double integrals over a one-dimensional squared exponential. Such integrals can be integrated by parts, leading to an analytic expression that only involves error functions and exponential functions (Peltonen, 2012).

The most challenging calculations involve elements of $\mathfrak{K}$ describing the covariance between observations made along different line search directions. We make use, once more, of the closure of the Gaussian exponential family under linear maps, to write

$$\begin{aligned}\mathfrak{K}^{ij}_{hk} =& (\eta_h - \nu_h)(\eta_k - \nu_k) e_j^\top V e_i \\ & \iint_0^1 \exp\left[-\frac{1}{2} \begin{pmatrix} (\eta_h - \nu_h)t_h \\ (\eta_k - \nu_k)t_k \\ 1 \end{pmatrix}^\top \begin{pmatrix} e_j \\ -e_i \\ \nu_h e_j + x_{0j} - \nu_k e_i - x_{0i} \end{pmatrix}^\top \Lambda^{-1} \dots \right] \mathrm{d}t_h \, \mathrm{d}t_k \\ =& \frac{2\pi e_j^\top V e_i \exp[-\frac{1}{2}(c - b^\top A^{-1} b)]}{\sqrt{(1-\rho^2)A_{hh}A_{kk}}} \\ & \left[\Phi(u_{hf}, u_{kf}, \rho) + \Phi(u_{hi}, u_{ki}, \rho) - \Phi(u_{hi}, u_{kf}, \rho) - \Phi(u_{hf}, u_{ki}, \rho) \right]\end{aligned}$$

with the bivariate Gaussian CDF

$$\Phi(b_1, b_2, \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{b_1} \int_{-\infty}^{b_2} \exp[-(x^2 - 2\rho xy + y^2)/2(1-\rho^2)] \, \mathrm{d}x \, \mathrm{d}y$$

and

$$\begin{aligned} A^{-1} &= \begin{pmatrix} e_j^\top \Lambda^{-1} e_j & -e_j^\top \Lambda^{-1} e_i \\ -e_j^\top \Lambda^{-1} e_i & e_i^\top \Lambda^{-1} e_i \end{pmatrix}^{-1} = \frac{\begin{pmatrix} e_i^\top \Lambda^{-1} e_i & e_j^\top \Lambda^{-1} e_i \\ e_j^\top \Lambda^{-1} e_i & e_j^\top \Lambda^{-1} e_j \end{pmatrix}}{e_j^\top \Lambda^{-1} e_j e_i^\top \Lambda^{-1} e_i - (e_j^\top \Lambda^{-1} e_i)^2}, \\ b &= \begin{pmatrix} e_j^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}) \\ -e_i^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}) \end{pmatrix}, \\ c &= (\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i})^\top \Lambda^{-1}(\nu_h e_j + x_{0j} - \nu_k e_i - x_{0i}),\end{aligned}$$

as well as

$$\begin{aligned} \rho &= \frac{e_j^\top \Lambda^{-1} e_i}{\sqrt{e_j^\top \Lambda^{-1} e_j e_i^\top \Lambda^{-1} e_i}}, \\ u_i &= \sqrt{1-\rho^2} \, \mathrm{diag}(\sqrt{[A_{hh}, A_{kk}]}) A^{-1} b, \\ u_f &= \sqrt{1-\rho^2} \, \mathrm{diag}(\sqrt{[A_{hh}, A_{kk}]}) \left[\begin{pmatrix} \eta_h - \nu_h \\ \eta_k - \nu_k \end{pmatrix} + A^{-1} b \right].\end{aligned}$$

Just like in the univariate case, bivariate Gaussian CDFs are not analytic. But single and double precision numerical approximations at acceptable computational cost exist (Genz, 2004).

From Sec. 1, recall that updating the search direction requires the *inverse* of B. Explicit inversion costs $\mathcal{O}(N^3)$, but the inverse can be constructed analytically, from the matrix inversion lemma, in $\mathcal{O}(N^2+NM+M^3)$. Using an argument largely analogous to the derivation of the L-BFGS algorithm (Nocedal, 1980) a diagonal prior mean B_0 lowers cost to $\mathcal{O}(NM+M^3)$, linear in N. Just like L-BFGS, the nonparametric method is thus applicable to problems of even very high dimensionality.

References

S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ Press, 2004.

C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19 (92):577–593, 1965.

C.G. Broyden. Quasi-Newton methods and their application to function minimization. *Math. Comp.*, 21(368):45, 1967.

C.G. Broyden. A new double-rank minimization algorithm. *Notices American Math. Soc*, 16:670, 1969.

C.G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76, 1970.

W.C. Davidon. Variable metric method for minimization. Technical report, Argonne National Laboratories, Ill., 1959.

J.E. Jr Dennis and J.J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, pages 46–89, 1977.

R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317, 1970.

R. Fletcher and M.J.D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.

A. Genz. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260, 2004.

D. Goldfarb. A family of variable metric updates derived by variational means. *Math. Comp.*, 24 (109):23–26, 1970.

J. Greenstadt. Variations on variable-metric methods. *Math. Comp*, 24:1–22, 1970.

P. Hennig. Fast probabilistic optimization from noisy gradients. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

H. Lütkepohl. *Handbook of Matrices*. Wiley, 1996.

T.P. Minka. Old an new linear algebra useful for statistics. Technical report, MIT Media Lab Note, 2000a.

T.P. Minka. Deriving quadrature rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 2000b.

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35(151):773–782, 1980.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

J.D. Pearson. Variable metric methods of minimisation. *The Computer Journal*, 12(2):171–178, 1969.

J. Peltonen. Personal communication, 2012.

M.J.D. Powell. A new algorithm for unconstrained optimization. In O. L. Mangasarian and K. Ritter, editors, *Nonlinear Programming*. AP, 1970.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

R.P. Savage. The space of positive definite matrices and Gromov's invariant. *Transactions of the AMS*, 274(1):239–263, November 1982.

D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24 (111):647–656, 1970.

Journal of Machine Learning Research 14 (2013) 867-897 Submitted 8/12; Revised 2/13; Published 3/13

A Widely Applicable Bayesian Information Criterion

Sumio Watanabe SWATANAB@DIS.TITECH.AC.JP
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
Mailbox G5-19, 4259 Nagatsuta, Midori-ku
Yokohama, Japan 226-8502

Editor: Manfred Opper

Abstract

A statistical model or a learning machine is called regular if the map taking a parameter to a probability distribution is one-to-one and if its Fisher information matrix is always positive definite. If otherwise, it is called singular. In regular statistical models, the Bayes free energy, which is defined by the minus logarithm of Bayes marginal likelihood, can be asymptotically approximated by the Schwarz Bayes information criterion (BIC), whereas in singular models such approximation does not hold.

Recently, it was proved that the Bayes free energy of a singular model is asymptotically given by a generalized formula using a birational invariant, the real log canonical threshold (RLCT), instead of half the number of parameters in BIC. Theoretical values of RLCTs in several statistical models are now being discovered based on algebraic geometrical methodology. However, it has been difficult to estimate the Bayes free energy using only training samples, because an RLCT depends on an unknown true distribution.

In the present paper, we define a widely applicable Bayesian information criterion (WBIC) by the average log likelihood function over the posterior distribution with the inverse temperature $1/\log n$, where n is the number of training samples. We mathematically prove that WBIC has the same asymptotic expansion as the Bayes free energy, even if a statistical model is singular for or unrealizable by a statistical model. Since WBIC can be numerically calculated without any information about a true distribution, it is a generalized version of BIC onto singular statistical models.

Keywords: Bayes marginal likelihood, widely applicable Bayes information criterion

1. Introduction

A statistical model or a learning machine is called regular if the map taking a parameter to a probability distribution is one-to-one and if its Fisher information matrix is always positive definite. If otherwise, it is called singular. Many statistical models and learning machines are not regular but singular, for example, artificial neural networks, normal mixtures, binomial mixtures, reduced rank regressions, Bayesian networks, and hidden Markov models. In general, if a statistical model contains hierarchical layers, hidden variables, or grammatical rules, then it is singular. In other words, if a statistical model is devised so that it extracts hidden structure from a random phenomenon, then it naturally becomes singular. If a statistical model is singular, then the likelihood function cannot be approximated by any normal distribution, resulting that neither AIC, BIC, nor MDL can be used

in statistical model evaluation. Hence constructing singular learning theory is an important issue in both statistics and learning theory.

A statistical model or a learning machine is represented by a probability density function $p(x|w)$ of $x \in \mathbb{R}^N$ for a given parameter $w \in W \subset \mathbb{R}^d$, where W is a set of all parameters. A prior probability density function is denoted by $\varphi(w)$ on W. Assume that training samples $X_1, X_2, ..., X_n$ are independently subject to a probability density function $q(x)$, which is called a true distribution. The log loss function or the minus log likelihood function is defined by

$$L_n(w) = -\frac{1}{n}\sum_{i=1}^{n} \log p(X_i|w). \tag{1}$$

Also the Bayes free energy $\mathcal{F}$ is defined by

$$\mathcal{F} = -\log \int \prod_{i=1}^{n} p(X_i|w)\varphi(w)dw. \tag{2}$$

This value $\mathcal{F}$ can be understood as the minus logarithm of marginal likelihood of a model and a prior, hence it plays an important role in statistical model evaluation. In fact, a model or a prior is often optimized by maximization of the Bayes marginal likelihood (Good, 1965), which is equivalent to minimization of the Bayes free energy.

If a statistical model is regular, then the posterior distribution can be asymptotically approximated by a normal distribution, resulting that

$$\mathcal{F} = nL_n(\hat{w}) + \frac{d}{2}\log n + O_p(1), \tag{3}$$

where $\hat{w}$ is the maximum likelihood estimator, d is the dimension of the parameter space, and n is the number of training samples. The right hand side of Equation (3) is the well-known Schwarz Bayesian information criterion (BIC) (Schwarz, 1978).

If a statistical model is singular, then the posterior distribution is different from any normal distribution, hence the Bayes free energy cannot be approximated by BIC in general. Recently, it was proved that, even if a statistical model is singular,

$$\mathcal{F} = nL_n(w_0) + \lambda \log n + O_p(\log\log n),$$

where w_0 is the parameter that minimizes the Kullback-Leibler distance from a true distribution to a statistical model, and $\lambda > 0$ is a rational number called the real log canonical threshold (RLCT) (Watanabe, 1999, 2001a, 2009, 2010a) .

The birational invariant RLCT, which was firstly found by a research of singular Schwartz distribution (Gelfand and Shilov, 1964), plays an important role in algebraic geometry and algebraic analysis (Bernstein, 1972; Sato and Shintani, 1974; Kashiwara, 1976; Varchenko, 1976; Kollár, 1997; Saito, 2007). In algebraic geometry, it represents a relative property of singularities of a pair of algebraic varieties. In statistical learning theory, it shows the asymptotic behaviors of the Bayes free energy and the generalization loss, which are determined by a pair of an optimal parameter set and a parameter set W.

If a triple of a true distribution, a statistical model, and a prior distribution is fixed, then there is an algebraic geometrical procedure which enables us to find an RLCT (Hironaka, 1964). In

fact, RLCTs for several statistical models and learning machines are being discovered. For example, RLCTs have been studied in artificial neural networks (Watanabe, 2001b; Aoyagi and Nagata, 2012), normal mixtures (Yamazaki and Watanabe, 2003), reduced rank regressions (Aoyagi and Watanabe, 2005), Bayes networks with hidden variables (Rusakov and Geiger, 2005; Zwiernik, 2010, 2011), binomial mixtures, Boltzmann machines (Yamazaki and Watanabe, 2005), and hidden Markov models. To study singular statistical models, new algebraic geometrical theory is constructed (Watanabe, 2009; Drton et al., 2009; Lin, 2011; Király et al., 2012).

Based on such researches, the theoretical behavior of the Bayes free energy is clarified. These results are very important because they indicate the quantitative difference of singular models from regular ones. However, in general, an RLCT depends on an unknown true distribution. In practical applications, we do not know a true distribution, hence we cannot directly apply the theoretical results to statistical model evaluation.

In the present paper, in order to estimate the Bayes free energy without any information about a true distribution, we propose a widely applicable Bayesian information criterion (WBIC) by the following definition:

$$\mathrm{WBIC} = \mathbb{E}_w^{\beta}[nL_n(w)], \quad \beta = \frac{1}{\log n}, \tag{4}$$

where $\mathbb{E}_w^{\beta}[\ ]$ shows the expectation value over the posterior distribution on W that is defined by, for an arbitrary integrable function $G(w)$,

$$\mathbb{E}_w^{\beta}[G(w)] = \frac{\int G(w) \prod_{i=1}^{n} p(X_i|w)^{\beta}\, \varphi(w) dw}{\int \prod_{i=1}^{n} p(X_i|w)^{\beta}\, \varphi(w) dw}. \tag{5}$$

In this definition, $\beta > 0$ is called the inverse temperature. Then the main purpose of this paper is to show

$$\mathcal{F} = \mathrm{WBIC} + O_p(\sqrt{\log n}).$$

To establish mathematical support of WBIC, we prove three theorems. Firstly, in Theorem 3 we show that there exists a unique inverse temperature β^* which satisfies

$$\mathcal{F} = \mathbb{E}_w^{\beta^*}[nL_n(w)].$$

The optimal inverse temperature β^* is a random variable which satisfies the convergence in probability, $\beta^* \log n \to 1$ as $n \to \infty$. Secondly, in Theorem 4 we prove that, even if a statistical model is singular,

$$\mathrm{WBIC} = nL_n(w_0) + \lambda \log n + O_p(\sqrt{\log n}).$$

In other words, WBIC has the same asymptotic behavior as the Bayes free energy even if a statistical model is singular. And lastly, in Theorem 5 we prove that, if a statistical model is regular, then

$$\mathrm{WBIC} = nL_n(\hat{w}) + \frac{d}{2}\log n + O_p(1),$$

which shows WBIC coincides with BIC in regular statistical models. Moreover, it is expected that a computational cost in numerical calculation of WBIC is far smaller than that of the Bayes free

Variable	Name	Equation Number
$\mathcal{F}$	Bayes free energy	Equation (2)
$\mathcal{G}$	Generalization loss	Equation (29)
WBIC	WBIC	Equation (4)
WAIC	WAIC	Equation (30)
$\mathbb{E}_w^{\beta}[\ ]$	posterior average	Equation (5)
β^*	optimal inverse temperature	Equation (18)
$L(w)$	log loss function	Equation (6)
$L_n(w)$	empirical loss	Equation (1)
$K(w)$	Average log likelihood ratio	Equation (7)
$K_n(w)$	empirical log likelihood ratio	Equation (8)
λ	real log canonical threshold	Equation (15)
m	multiplicity	Equation (16)
$Q(K(w), \varphi(w))$	parity of model	Equation (17)
$(\mathcal{M}, g(u), a(x,u), b(u))$	resolution quartet	Theorem 1

Table 1: Variable, Name, and Equation Number

energy. These results show that WBIC is a generalized version of BIC onto singular statistical models and that RLCTs can be estimated even if a true distribution is unknown.

This paper consists of eight sections. In Section 2, we summarize several notations. In Section 3, singular learning theory and the standard representation theorem are introduced. The main theorems and corollaries of this paper are explained in Section 4, which are mathematically proved in Section 5. As the purpose of the present paper is to prove the mathematical support of WBIC, Sections 4 and 5 are the main sections. In section 6, a method how to use WBIC in statistical model evaluation is illustrated using an experimental result. In section 7 and 8, we discuss and conclude the present paper.

2. Statistical Models and Notations

In this section, we summarize several notations. Table 1 shows variables, names, and equation numbers in this paper. The average log loss function $L(w)$ and the entropy of the true distribution S are respectively defined by

$$\begin{aligned} L(w) &= -\int q(x) \log p(x|w) dx, \\ S &= -\int q(x) \log q(x) dx. \end{aligned} \tag{6}$$

Then $L(w) = S + D(q||p_w)$, where $D(q||p_w)$ is the Kullback-Leibler distance defined by

$$D(q||p_w) = \int q(x) \log \frac{q(x)}{p(x|w)} dx.$$

Then $D(q||p_w) \geq 0$, hence $L(w) \geq S$. Moreover, $L(w) = S$ if and only if $p(x|w) = q(x)$.

In this paper, we assume that there exists a parameter w_0 in the interior of W which minimizes $L(w)$,

$$L(w_0) = \min_{w \in W} L(w),$$

where the interior of a set S is the defined by the largest open set that is contained in S. Note that such w_0 is not unique in general, because the map $w \mapsto p(x|w)$ is not one-to-one in general in singular statistical models. We also assume that, for an arbitrary w that satisfies $L(w) = L(w_0)$, $p(x|w)$ is the same probability density function. Let $p_0(x)$ be such a unique probability density function. In general, the set

$$W_0 = \{w \in W; p(x|w) = p_0(x)\}$$

is not a set of single element but an analytic set or an algebraic set with singularities. Let us define a log density ratio function,

$$f(x,w) = \log \frac{p_0(x)}{p(x|w)},$$

which is equivalent to

$$p(x|w) = p_0(x) \exp(-f(x,w)).$$

Two functions $K(w)$ and $K_n(w)$ are respectively defined by

$$K(w) = \int q(x) f(x,w) dx, \tag{7}$$

$$K_n(w) = \frac{1}{n} \sum_{i=1}^{n} f(X_i, w). \tag{8}$$

Then it immediately follows that

$$\begin{aligned} L(w) &= L(w_0) + K(w), \\ L_n(w) &= L_n(w_0) + K_n(w). \end{aligned}$$

The expectation value over all sets of training samples $X_1, X_2, ..., X_n$ is denoted by $\mathbb{E}[\ ]$. For example, $\mathbb{E}[L_n(w)] = L(w)$ and $\mathbb{E}[K_n(w)] = K(w)$. The problem of statistical learning is characterized by the log density ratio function $f(x,w)$. In fact,

$$\mathbb{E}_w^{\beta}[nL_n(w)] = nL_n(w_0) + \mathbb{E}_w^{\beta}[nK_n(w)], \tag{9}$$

$$\mathbb{E}_w^{\beta}[nK_n(w)] = \frac{\int nK_n(w) \exp(-n\beta K_n(w)) \varphi(w) dw}{\int \exp(-n\beta K_n(w)) \varphi(w) dw}. \tag{10}$$

The main purpose of the present paper is to prove

$$\mathcal{F} = nL_n(w_0) + \mathbb{E}_w^{\beta}[nK_n(w)] + O_p(\sqrt{\log n})$$

for $\beta = 1/\log n$.

Definition.

(1) If $q(x) = p_0(x)$, then $q(x)$ is said to be *realizable* by $p(x|w)$. If otherwise, it is said to be *unrealizable*.

(2) If the set W_0 consists of a single element w_0 and if the Hessian matrix

$$J_{ij}(w) = \frac{\partial^2 L}{\partial w_i \partial w_j}(w) \tag{11}$$

at $w = w_0$ is strictly positive definite, then $q(x)$ is said to be *regular* for $p(x|w)$. If otherwise, it is said to be *singular* for $p(x|w)$.

Note that the matrix $J(w)$ is equal to the Hessian matrix of $K(w)$ and that $J(w_0)$ is equal to the Fisher information matrix if the true distribution is realizable by a statistical model. Also note that, if $q(x)$ is realizable by $p(x|w)$, then $K(w)$ is Kullback-Leibler divergence of $q(x)$ and $p(x|w)$. However, if $q(x)$ is not realizable by $p(x|w)$, then it is not.

3. Singular Learning Theory

In this section we summarize singular learning theory. In the present paper, we assume the following conditions.

Fundamental Conditions.

(1) The set of parameters W is a compact set in $\mathbb{R}^d$ whose interior is not the empty set. Its boundary is defined by several analytic functions $\pi_1(w)$, $\pi_2(w)$, ..., $\pi_k(w)$, in other words,

$$W = \{w \in \mathbb{R}^d; \pi_1(w) \geq 0, \pi_2(w) \geq 0, ..., \pi_k(w) \geq 0\}.$$

(2) The prior distribution satisfies $\varphi(w) = \varphi_1(w)\varphi_2(w)$, where $\varphi_1(w) \geq 0$ is an analytic function and $\varphi_2(w) > 0$ is a C^∞-class function.

(3) Let $s \geq 6$ and

$$L^s(q) = \{f(x); \|f\|_s \equiv \Big(\int |f(x)|^s q(x)dx\Big)^{1/s} < \infty\}$$

be a Banach space. There exists an open set $W' \supset W$ such that the map $W' \ni w \mapsto f(x,w)$ is an $L^s(q)$-valued analytic function.

(4) The set W_ε is defined by

$$W_\varepsilon = \{w \in W \ ; \ K(w) \leq \varepsilon\}.$$

It is assumed that there exist constants $\varepsilon, c > 0$ such that

$$(\forall w \in W_\varepsilon) \quad \mathbb{E}_X[f(X,w)] \geq c\, \mathbb{E}_X[f(X,w)^2]. \tag{12}$$

Remark. (1) These conditions allow that the set of optimal parameters

$$W_0 = \{w \in W \ ; \ p(x|w) = p(x|w_0)\} = \{w \in W \ ; \ K(w) = 0\}$$

may contain singularities, and that the Hessian matrix $J(w)$ at $w \in W_0$ is not positive definite. Therefore $K(w)$ can not be approximated by any quadratic form in general.

(2) The condition Equation (12) is satisfied if a true distribution is realizable by or regular for a statistical model (Watanabe, 2010a). If a true distribution is unrealizable by and singular for a statistical model, there is an example which does not satisfy this condition. In the present paper, we study the case when Equation (12) is satisfied.

Lemma 1 *Assume Fundamental Conditions (1)-(4). Let*

$$\beta = \frac{\beta_0}{\log n},$$

where $\beta_0 > 0$ is a constant and let $0 \leq r < 1/2$. Then, as $n \to \infty$,

$$\int_{K(w) \geq 1/n^r} \exp(-n\beta K_n(w))\varphi(w)dw = o_p(\exp(-\sqrt{n})), \tag{13}$$

$$\int_{K(w) \geq 1/n^r} nK_n(w)\exp(-n\beta K_n(w))\varphi(w)dw = o_p(\exp(-\sqrt{n})). \tag{14}$$

The proof of Lemma 1 is given in Section 5.

Let $\varepsilon > 0$ be a sufficiently small constant. Lemma 1 shows that integrals outside of the region W_ε do not affect the expectation value $\mathbb{E}_w^\beta[nK_n(w)]$ asymptotically, because in the following theorems, we prove that integral in the region W_ε have larger orders than the integral outside of W_ε. To study integrals in the region W_ε, we need algebraic geometrical method, because the set $\{w; K(w) = 0\}$ contains singularities in general. There are quite many kinds of singularities, however, the following theorem makes any singularities be a same standard form.

Theorem 1 *(Standard Representation) Assume Fundamental Conditions (1)-(4). Let $\varepsilon > 0$ be a sufficiently small constant. Then there exists an quartet $(\mathcal{M}, g(u), a(x,u), b(u))$, where*
(1) $\mathcal{M}$ is a d-dimensional real analytic manifold,
(2) g is a proper analytic function $g : \mathcal{M} \to W'_\varepsilon$, where W'_ε is the set that is defined by the largest open set contained in W_ε and $g : \{u \in \mathcal{M}; K(g(u)) \neq 0\} \to \{w \in W'_\varepsilon; K(w) \neq 0\}$ is a bijective map,
(3) $a(x,u)$ is an $L^s(q)$-valued analytic function,
(4) and $b(u)$ is an infinitely many times differentiable function which satisfies $b(u) > 0$,
such that the following equations are satisfied in each local coordinate of $\mathcal{M}$:

$$\begin{aligned} K(g(u)) &= u^{2k}, \\ f(x, g(u)) &= u^k a(x,u), \\ \varphi(w)dw &= \varphi(g(u))|g'(u)|du = b(u)|u^h|du, \end{aligned}$$

where $k = (k_1, k_2, ..., k_d)$ and $h = (h_1, h_2, ..., h_d)$ are multi-indices made of nonnegative integers. At least one of k_j is not equal to zero.

Remark. (1) In this theorem, for $u = (u_1, u_2, \cdots, u_d) \in \mathbb{R}^d$, notations u^{2k} and $|u^h|$ respectively represent

$$\begin{aligned} u^{2k} &= u_1^{2k_1} u_2^{2k_2} \cdots u_d^{2k_d}, \\ |u^h| &= |u_1^{h_1} u_2^{h_2} \cdots u_d^{h_d}|. \end{aligned}$$

The singularity $u = 0$ in $u^{2k} = 0$ is said to be normal crossing. Theorem 1 shows that any singularities can be made normal crossing by using an analytic function $w = g(u)$.
(2) A map $w = g(u)$ is said to be proper if, for an arbitrary compact set C, $g^{-1}(C)$ is also compact.
(3) The proof of Theorem 1 is given in Theorem 6.1 of a book (Watanabe, 2009, 2010a). In order to prove this theorem, we need the Hironaka resolution of singularities (Hironaka, 1964; Atiyah, 1970) that is the fundamental theorem in algebraic geometry. The function $w = g(u)$ is often referred to as a resolution map.
(4) In this theorem, a quartet $(k, h, a(x,u), b(u))$ depends on a local coordinate in general. For a given function $K(w)$, there is an algebraic recursive algorithm which enables us to find a resolution

map $w = g(u)$. However, even for a fixed $K(w)$, a resolution map is not unique, resulting that a quartet $(\mathcal{M}, g(u), a(x,u), b(u))$ is not unique.

Definition. (Real Log Canonical Threshold) Let $\{\mathcal{U}_\alpha; \alpha \in \mathcal{A}\}$ be a system of local coordinates of a manifold $\mathcal{M}$,

$$\mathcal{M} = \bigcup_{\alpha \in \mathcal{A}} \mathcal{U}_\alpha.$$

The real log canonical threshold (RLCT) is defined by

$$\lambda = \min_{\alpha \in \mathcal{A}} \min_{j=1}^{d} \Big(\frac{h_j + 1}{2k_j} \Big), \tag{15}$$

where we define $1/k_j = \infty$ if $k_j = 0$. The multiplicity m is defined by

$$m = \max_{\alpha \in \mathcal{A}} \#\Big\{ j; \frac{h_j + 1}{2k_j} = \lambda \Big\}, \tag{16}$$

where $\#S$ shows the number of elements of a set S.

The concept of RLCT is well known in algebraic geometry and statistical learning theory. In the following definition we introduce a parity of a statistical model.

Definition. (Parity of Statistical Model) The support of $\varphi(g(u))$ is defined by

$$\mathrm{supp}\ \varphi(g(u)) = \overline{\{u \in \mathcal{M}\ ;\ g(u) \in W_\varepsilon,\ \ \varphi(g(u)) > 0\}},$$

where $\overline{S}$ shows the closure of a set S. A local coordinate $\mathcal{U}_\alpha$ is said to be an essential local coordinate if both equations

$$\begin{aligned} \lambda &= \min_{j=1}^{d} \Big(\frac{h_j + 1}{2k_j} \Big), \\ m &= \#\{ j; (h_j + 1)/(2k_j) = \lambda \}, \end{aligned}$$

hold in its local coordinate. The set of all essential local coordinates is denoted by $\{\mathcal{U}_\alpha; \alpha \in \mathcal{A}^*\}$. If, for an arbitrary essential local coordinate, there exist both $\delta > 0$ and a natural number j in the set $\{ j\ ;\ (h_j + 1)/(2k_j) = \lambda \}$ such that
(1) k_j is an odd number,
(2) $\{(0,0,..,0,u_j,0,0,..,0)\ ;\ |u_j| < \delta\} \subset \mathrm{supp}\ \varphi(g(u))$,
then we define $Q(K(g(u)), \varphi(g(u))) = 1$. If otherwise, $Q(K(g(u)), \varphi(g(u))) = 0$. If there exists a resolution map $w = g(u)$ such that $Q(K(g(u)), \varphi(g(u))) = 1$, then we define

$$Q(K(w), \varphi(w)) = 1. \tag{17}$$

If otherwise $Q(K(w), \varphi(w)) = 0$. If $Q(K(w), \varphi(w)) = 1$, then the parity of a statistical model is said to be odd, otherwise even.

It was proved in Theorem 2.4 of a book (Watanabe, 2009) that, for a given set (q, p, φ), λ and m are independent of a choice of a resolution map. Such a value is called a birational invariant. The RLCT is a birational invariant.

Lemma 2 *If a true distribution $q(x)$ is realizable by a statistical model $p(x|w)$, then the value $Q(K(g(u)), \varphi(g(u)))$ is independent of a choice of a resolution map $w = g(u)$.*

Proof of this lemma is shown in Section 5. Lemma 2 indicates that, if a true distribution is realizable by a statistical model, then $Q(K(g(u)),\varphi(g(u)))$ is a birational invariant. The present paper proposes a conjecture that $Q(K(g(u)),\varphi(g(u)))$ is a birational invariant in general. By Lemma 2, this conjecture is proved if we can show the proposition that, for an arbitrary nonnegative analytic function $K(w)$, there exist $q(x)$ and $p(x|w)$ such that $K(w)$ is the Kullback-Leibler distance from $q(x)$ to $p(x|w)$.

Example. Let $w=(a,b,c)\in\mathbb{R}^3$ and

$$K(w)=(ab+c)^2+a^2b^4,$$

which is the Kullback-Leibler distance of a neural network model in Example 1.6 of a book (Watanabe, 2009), where a true distribution is realizable by a statistical model. The prior $\varphi(w)$ is defined by some nonzero function on a sufficiently large compact set. In a singular statistical model, compactness of the prior is necessary in general, because, if the parameter set is not compact, then it is not easy to mathematically treat the integration on the neighborhood among the infinite point.

Let a system of local coordinates be

$$\mathcal{U}_i=\{(a_i,b_i,c_i)\in\mathbb{R}^3\}\quad (i=1,2,3,4).$$

A resolution map $g:\mathcal{U}_1\cup\mathcal{U}_2\cup\mathcal{U}_3\cup\mathcal{U}_4\to\mathbb{R}^3$ in each local coordinate is defined by

$$\begin{aligned}
a=a_1c_1,\quad & b=b_1,\quad & c=c_1,\\
a=a_2,\quad & b=b_2c_2,\quad & c=a_2(1-b_2)c_2,\\
a=a_3,\quad & b=b_3,\quad & c=a_3b_3(b_3c_3-1),\\
a=a_4,\quad & b=b_4c_4,\quad & c=a_4b_4c_4(c_4-1).
\end{aligned}$$

This map g is made of recursive blowing-ups whose centers are smooth manifolds, hence it is one-to-one as a map $g:\{u;K(g(u))>0\}\to\{w;K(w)>0\}$. Then

$$\begin{aligned}
K(a,b,c) &= c_1^2\{(a_1b_1+1)^2+a_1^2b_1^4\}=a_2^2c_2^2(1+b_2^2c_2^2)\\
&= a_3^2b_3^4(c_3^2+1)=a_4^2b_4^2c_4^4(1+b_4^2).
\end{aligned}$$

Therefore integration over W can be calculated using integration over the manifold. The Jacobian determinant $|g'(u)|$ is

$$\begin{aligned}
|g'(u)| &= |c_1|=|a_2c_2|\\
&= |a_3b_3^2|=|a_4b_4c_4|^2.
\end{aligned}$$

In other words, in each local coordinate,

$$\begin{aligned}
(k_1,k_2,k_3) &= (0,0,1),(1,0,1),(1,2,0),(1,1,2),\\
(h_1,h_2,h_3) &= (0,0,1),(1,0,1),(1,2,0),(2,2,2).
\end{aligned}$$

Therefore

$$\Big(\frac{h_1+1}{2k_1},\frac{h_2+1}{2k_2},\frac{h_2+1}{2k_2}\Big)=(\infty,\infty,1),(1,0,1),(1,\frac{3}{4},\infty),(\frac{3}{2},\frac{3}{2},\frac{3}{4}).$$

The smallest value among them is $3/4$ which is equal to λ and the multiplicty is $m=1$. The essential local coordinates are $\mathcal{U}_3$ and $\mathcal{U}_4$. In $\mathcal{U}_3$ and $\mathcal{U}_4$, the sets $\{u_j\,;\,(h_j+1)/(2k_j)=3/4\}$ are respectively $\{u_2\}$ and $\{u_3\}$, where $2k_j=4$ in both cases. Consequently, both k_j are even, thus the parity is given by $Q(K(w),\varphi(w))=0$.

Lemma 3 *Assume that the Fundamental Conditions (1)-(4) are satisfied and that a true distribution $q(x)$ is regular for a statistical model $p(x|w)$. If w_0 is contained in the interior of W and if $\varphi(w_0) > 0$, then*

$$\lambda = \frac{d}{2}, \quad m = 1,$$

and

$$Q(K(w), \varphi(w)) = 1.$$

Proof of this lemma is shown in Section 5.

Theorem 2 *Assume that the Fundamental Conditions (1)-(4) are satisfied. Then the following holds.*

$$\mathcal{F} = nL_n(w_0) + \lambda \log n - (m-1)\log\log n + R_n,$$

where λ is a real log canonical threshold, m is its multiplicity, and $\{R_n\}$ is a sequence of random variables which converges to a random variable in law, when $n \to \infty$.

Theorem 2 was proved in the previous papers. In the case when $q(x)$ is realizable by and singular for $p(x|w)$, the expectation value of $\mathcal{F}$ is given by algebraic analysis (Watanabe, 2001a). The asymptotic behavior of $\mathcal{F}$ as a random variable was shown in a book (Watanabe, 2009). These results were generalized (Watanabe, 2010a) for the case that $q(x)$ is unrealizable.

Remark. In practical applications, we do not know the true distribution, hence λ and m are unknown. Therefore, we can not directly apply Theorem 2 to such cases. The main purpose of the present paper is to make a new method how to estimate $\mathcal{F}$ even if the true distribution is unknown.

4. Main Results

In this section, we introduce the main results of the present paper.

Theorem 3 (*Unique Existence of the Optimal Parameter*) *Assume that $L_n(w)$ is not a constant function of w. Then the followings hold.*
(1) The value $\mathbb{E}_w^{\beta}[nL_n(w)]$ is a decreasing function of β.
(2) There exists a unique β^* $(0 < \beta^* < 1)$ which satisfies

$$\mathcal{F} = \mathbb{E}_w^{\beta^*}[nL_n(w)]. \tag{18}$$

Note that the function $L_n(w)$ is not a constant function in an ordinary statistical model with probability one. The Proof of Theorem 3 is given in Section 5. Based on this theorem, we define the optimal inverse temperature.

Definition. The unique parameter β^* that satisfies Equation (18) is called the optimal inverse temperature.

In general, the optimal inverse temperature β^* depends on a true distribution $q(x)$, a statistical model $p(x|w)$, a prior $\varphi(w)$, and training samples. Therefore β^* is a random variable. In the present paper, we study its probabilistic behavior. Theorem 4 is a mathematical base for such a purpose.

Theorem 4 (*Main Theorem*) *Assume Fundamental Conditions (1)-(4) and that*

$$\beta = \frac{\beta_0}{\log n},$$

where β_0 is a constant. Then there exists a random variable U_n such that

$$\mathbb{E}_w^{\beta}[nL_n(w)] = nL_n(w_0) + \frac{\lambda \log n}{\beta_0} + U_n\sqrt{\frac{\lambda \log n}{2\beta_0}} + O_p(1),$$

where λ is the real log canonical threshold and $\{U_n\}$ is a sequence of random variables, which satisfies $\mathbb{E}[U_n] = 0$, converges to a Gaussian random variable in law as $n \to \infty$. Moreover, if a true distribution $q(x)$ is realizable by a statistical model $p(x|w)$, then $\mathbb{E}[(U_n)^2] < 1$.

The proof of Theorem 4 is given in Section 5. Theorem 4 with $\beta_0 = 1$ shows that

$$\mathrm{WBIC} = nL_n(w_0) + \lambda \log n + U_n\sqrt{\frac{\lambda \log n}{2}} + O_p(1),$$

whose first two main terms are equal to those of $\mathcal{F}$ in Theorem 2. From Theorem 4 and its proof, three important corollaries are derived.

Corollary 1 *If the parity of a statistical model is odd, $Q(K(w), \varphi(w)) = 1$, then $U_n = 0$.*

Corollary 2 *Let β^* be the optimal inverse temperature. Then*

$$\beta^* = \frac{1}{\log n}\Big(1 + \frac{U_n}{\sqrt{2\lambda \log n}} + o_p\Big(\frac{1}{\sqrt{\log n}}\Big)\Big).$$

Corollary 3 *Let $\beta_1 = \beta_{01}/\log n$ and $\beta_2 = \beta_{02}/\log n$, where β_{01} and β_{02} are positive constants. Then the convergence in probability*

$$\frac{\mathbb{E}_w^{\beta_1}[nL_n(w)] - \mathbb{E}_w^{\beta_2}[nL_n(w)]}{1/\beta_1 - 1/\beta_2} \to \lambda \tag{19}$$

holds as $n \to \infty$, where λ is the real log canonical threshold.

Proofs of these corollaries are given in Section 5. Note that, if the expectation value $\mathbb{E}_w^{\beta_1}[\ ]$ is calculated by some numerical method, then $\mathbb{E}_w^{\beta_2}[\ ]$ can be estimated using $\mathbb{E}_w^{\beta_1}[\ ]$ by using

$$\mathbb{E}_w^{\beta_2}[nL_n(w)] = \frac{\mathbb{E}_w^{\beta_1}[nL_n(w)\exp(-(\beta_2 - \beta_1)nL_n(w))]}{\mathbb{E}_w^{\beta_1}[\exp(-(\beta_2 - \beta_1)nL_n(w))]}. \tag{20}$$

Therefore RLCT can be estimated by the same computational cost as WBIC. In Bayes estimation, the posterior distribution is often approximated by some numerical method. If we know theoretical values of RLCTs, then we can confirm the approximated posterior distribution by comparing theoretical values with estimated ones.

The well-known Schwarz BIC is defined by

$$\mathrm{BIC} = nL_n(\hat{w}) + \frac{d}{2}\log n,$$

where $\hat{w}$ is the maximum likelihood estimator. WBIC can be understood as the generalized BIC onto singular statistical models, because it satisfies the following theorem.

Theorem 5 *If a true distribution $q(x)$ is regular for a statistical model $p(x|w)$, then*

$$\mathrm{WBIC} = nL_n(\hat{w}) + \frac{d}{2}\log n + o_p(1).$$

Proof of Theorem 5 is given in Section 5. This theorem shows that the difference of WBIC and BIC is smaller than a constant order term, if a true distribution is regular for a statistical model. This theorem holds even if a true distribution $q(x)$ is unrealizable by $p(x|w)$.

Remark. Since the set of parameters W was assumed to be compact, it is proved in Main Theorem 6.4 of a book (Watanabe, 2009) that $nL_n(w_0) - nL_n(\hat{w})$ is a constant order random variable in general. If a true distribution is regular for and realizable by a statistical model, its average is asymptotically equal to $d/2$, where d is the dimension of parameter. If a true distribution is singular for a statistical model, then it is sometimes much larger than $d/2$, because it is asymptotically equal to the maximum value of the Gaussian process. Whether replacement of $nL_n(w_0)$ by $nL(\hat{w})$ is appropriate or not depends on the statistical model and its singularities (Drton, 2009).

5. Proofs of Main Results

In this section, we prove the main theorems and corollaries.

5.1 Proof of Lemma 1

Let us define an empirical process,

$$\eta_n(w) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(K(w) - f(X_i, w)).$$

It was proved in Theorem 5.9 and 5.10 of a book (Watanabe, 2009) that $\eta_n(w)$ converges to a random process in law and

$$\|\eta_n\| \equiv \sup_{w\in W}|\eta_n(w)|$$

also converges to a random variable in law. If $K(w) \geq 1/n^r$, then

$$\begin{aligned} nK_n(w) &= nK(w) - \sqrt{n}\,\eta_n(w) \\ &\geq n^{1-r} - \sqrt{n}\,\|\eta_n\|. \end{aligned}$$

By the condition $1-r > 1/2$ and $\beta = \beta_0/\log n$,

$$\begin{aligned} \exp(\sqrt{n})\int_{K(w)\geq 1/n^r}\exp(-n\beta K_n(w))\varphi(w)dw \\ \leq \exp(-n^{1-r}\beta + \sqrt{n} + \sqrt{n}\beta\|\eta_n\|), \end{aligned}$$

which converges to zero in probability, which shows Equation (13). Then, let us prove Equation (14). Since the set of parameter W is compact, $\|K\| \equiv \sup_w K(w) < \infty$. Therefore,

$$\begin{aligned} |nK_n(w)| &\leq n\|K\| + \sqrt{n}\|\eta_n\| \\ &= n\,(\|K\| + \|\eta_n\|/\sqrt{n}). \end{aligned}$$

Hence

$$\begin{aligned}
&\exp(\sqrt{n})\int_{K(w)\geq 1/n^r}|nK_n(w)|\exp(-n\beta K_n(w))\varphi(w)dw\\
&\leq (\|K\|+\|\eta_n\|/\sqrt{n})\\
&\times\exp(-n^{1-r}\beta+\sqrt{n}+\sqrt{n}\beta\|\eta_n\|+\log n),
\end{aligned}$$

which converges to zero in probability. (Q.E.D.)

5.2 Proof of Lemma 3

Without loss of generality, we can assume $w_0=0$. Since $q(x)$ is regular for $p(x|w)$, there exists w^* such that

$$K(w)=\frac{1}{2}w\cdot J(w^*)w,$$

where $J(w)$ is given in Equation (11). Since $J(w_0)$ is a strictly positive definite matrix, there exists $\varepsilon>0$ such that, if $K(w)\leq\varepsilon$, then $J(w^*)$ is positive definite. Let ℓ_1 and ℓ_2 be respectively the minimum and maximum eigen values of $\{J(w^*);K(w)\leq\varepsilon\}$. Then

$$\frac{1}{4}\ell_1\sum_{j=1}^d w_j^2\leq\frac{1}{2}w\cdot J(w^*)w\leq\ell_2\sum_{j=1}^d w_j^2.$$

By using a blow-up $g:\mathcal{U}_1\cup\cdots\cup\mathcal{U}_d\to W$ which is represented on each local coordinate $\mathcal{U}_i=(u_{i1},u_{i2},...,u_{id})$,

$$\begin{aligned}
w_i &= u_{ii},\\
w_j &= u_{ii}u_{ij}\quad(j\neq i),
\end{aligned}$$

it follows that

$$\frac{\ell_1\,u_{ii}^2}{4}(1+\sum_{j\neq i}u_{ij}^2)\leq\frac{u_{ii}^2}{2}(\hat{u},J(w^*)\hat{u})\leq\ell_2\,u_{ii}^2(1+\sum_{j\neq i}u_{ij}^2),$$

where $\hat{u}_{ij}=u_{ij}$ $(j\neq i)$ and $\hat{u}_{ii}=1$. These inequalities show that $k_i=1$ in $\mathcal{U}_i$, therefore $Q(K(w),\varphi(w))=1$. The Jacobian determinant of the blow-up is

$$|g'(u)|=|u_{ii}|^{d-1},$$

hence $\lambda=d/2$ and $m=1$. (Q.E.D.)

5.3 Proof of Theorem 3

Let us define a function $F_n(\beta)$ of $\beta>0$ by

$$F_n(\beta)=-\log\int\prod_{i=1}^n p(X_i|w)^\beta\varphi(w)dw.$$

Then, by the definition, $\mathcal{F}=F_n(1)$ and

$$\begin{aligned}
F_n'(\beta) &= \mathbb{E}_w^\beta[nL_n(w)],\\
F_n''(\beta) &= -\mathbb{E}_w^\beta[(nL_n(w))^2]+\mathbb{E}_w^\beta[nL_n(w)]^2.
\end{aligned}$$

By the Cauchy-Schwarz inequality and the assumption that $L_n(w)$ is not a constant function,

$$F_n''(\beta) < 0,$$

which shows (1). Since $F_n(0) = 0$,

$$\mathcal{F} = F_n(1) = \int_0^1 F_n'(\beta) d\beta.$$

By using the mean value theorem, there exists β^* $(0 < \beta^* < 1)$ such that

$$\mathcal{F} = F_n'(\beta^*) = \mathbb{E}_w^{\beta^*}[nL_n(w)].$$

Here $F_n'(\beta)$ is a decreasing function, β^* is unique, which completes Theorem 3. (Q.E.D.)

5.4 First Preparation for Proof of Theorem 4

In this subsection, we prepare the proof of Theorem 4. By using Equation (9) and Equation (10), the proof of Theorem 4 results in evaluating $E_w^{\beta}[nK_n(w)]$. By Lemma 1,

$$E_w^{\beta}[nK_n(w)] = \frac{B_n + o_p(\exp(-\sqrt{n}))}{A_n + o_p(\exp(-\sqrt{n}))}, \tag{21}$$

where A_n and B_n are respectively defined by

$$A_n = \int_{K(w)<\varepsilon} \exp(-n\beta K_n(w))\varphi(w) dw, \tag{22}$$

$$B_n = \int_{K(w)<\varepsilon} nK_n(w) \exp(-n\beta K_n(w))\varphi(w) dw. \tag{23}$$

By Theorem 1, an integral over $\{w \in W; K(w) < \varepsilon\}$ is equal to that over $\mathcal{M}$. For a given set of local coordinates $\{\mathcal{U}_\alpha\}$ of $\mathcal{M}$, there exists a set of C^∞ class functions $\{\varphi_\alpha(g(u))\}$ such that, for an arbitrary $u \in \mathcal{M}$,

$$\sum_{\alpha \in \mathcal{A}} \varphi_\alpha(g(u)) = \varphi(g(u)).$$

By using this fact, for an arbitrary integrable function $G(w)$,

$$\int_{K(w)<\varepsilon} G(w)\varphi(w) dw = \sum_{\alpha \in \mathcal{A}} \int_{\mathcal{U}_\alpha} G(g(u))\varphi_\alpha(g(u))|g'(u)| du.$$

Without loss of generality, we can assume that $\overline{\mathcal{U}_\alpha \cap \mathrm{supp}\, \varphi(g(u))}$ is isomorphic to $[-1,1]^d$. Moreover, by Theorem 1, there exists a function $b_\alpha(u) > 0$ such that

$$\varphi_\alpha(g(u))|g'(u)| = |u^h| b_\alpha(u),$$

in each local coordinate. Consequently,

$$\int_{K(w)<\varepsilon} G(w)\varphi(w) dw = \sum_{\alpha \in \mathcal{A}} \int_{[-1,1]^d} du\, G(g(u))\, |u^h|\, b_\alpha(u).$$

In each local coordinate,

$$K(g(u)) = u^{2k}.$$

We define a function $\xi_n(u)$ by

$$\xi_n(u) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{u^k - a(X_i, u)\}.$$

Then

$$K_n(g(u)) = u^{2k} - \frac{1}{\sqrt{n}} u^k \xi_n(u).$$

Note that

$$u^k = \int a(x,u) q(x) dx$$

holds, because

$$u^{2k} = K(g(u)) = \int f(x, g(u)) q(x) dx = u^k \int a(x,u) q(x) dx.$$

Therefore, for an arbitrary u,

$$\mathbb{E}[\xi_n(u)] = 0.$$

The function $\xi_n(u)$ can be understood as a random process on $\mathcal{M}$. On Fundamental Conditions (1)-(4), it is proved in Theorem 6.1, Theorem 6.2, and Theorem 6.3 of a book (Watanabe, 2009) that (1) $\xi_n(u)$ converges to a Gaussian random process $\xi(u)$ in law and

$$\mathbb{E}[\sup_u \xi_n(u)^2] \to \mathbb{E}[\sup_u \xi(u)^2].$$

(2) If $q(x)$ is realizable by $p(x|w)$, and if $u^{2k} = 0$, then

$$\mathbb{E}[\xi_n(u)^2] = \mathbb{E}_X[a(X,u)^2] = 2. \tag{24}$$

By using the random process $\xi_n(u)$, the two random variables A_n and B_n can be represented by integrals over $\mathcal{M}$,

$$\begin{aligned} A_n &= \sum_{\alpha \in \mathcal{A}} \int_{[-1,1]^d} du \, \exp(-n\beta u^{2k} + \sqrt{n}\beta u^k \xi_n(u)) |u^h| b_\alpha(u), \\ B_n &= \sum_{\alpha \in \mathcal{A}} \int_{[-1,1]^d} du \, (nu^{2k} - \sqrt{n} u^k \xi_n(u)) \\ &\quad \times \exp(-n\beta u^{2k} + \sqrt{n}\beta u^k \xi_n(u)) |u^h| b_\alpha(u) \end{aligned}$$

To prove Theorem 4, we study asymptotics of these two random variables.

5.5 Second Preparation for Proof of Theorem 4

To evaluate two integrals A_n and B_n as $n \to \infty$, we have to study the asymptotic behavior of the following Schwartz distribution,

$$\delta(t - u^{2k}) \, |u|^h$$

for $t \to 0$. Without loss of generality, we can assume that, in each essential local coordinate,

$$\lambda = \frac{h_1+1}{2k_1} = \frac{h_2+1}{2k_2} = \cdots = \frac{h_m+1}{2k_m} < \frac{h_j+1}{2k_j},$$

for an arbitrary j such that $m < j \leq d$. A variable $u \in \mathbb{R}^d$ is denoted by

$$u = (u_a, u_b) \in \mathbb{R}^m \times \mathbb{R}^{d-m}.$$

We define a measure du^* by

$$du^* = \frac{(\prod_{j=1}^{m} \delta(u_j))\ (\prod_{j=m+1}^{d} (u_j)^{\mu_j})\ du}{2^m\ (m-1)!\ (\prod_{j=1}^{m} k_j)}, \tag{25}$$

where $\delta(\ )$ is the Dirac delta function, and $\mu = (\mu_{m+1}, \mu_2, ..., \mu_d)$ is a multi-index defined by

$$\mu_j = -2\lambda k_j + h_j \quad (m+1 \leq j \leq d).$$

Then $\mu_j > -1$, hence Equation (25) defines a measure on $\mathcal{M}$. The support of du^* is $\{u = (u_a, u_b)\ ;\ u_a = 0\}$.

Definition. Let σ be a d-dimensional variable made of ± 1. We use the notation,

$$\sigma = (\sigma_1, \sigma_2, ..., \sigma_d) \in \mathbb{R}^d$$

where $\sigma_j = \pm 1$. The set of all such variables is denoted by $S(d)$.

$$S(d) = \{\sigma\ ;\ \sigma_j = \pm 1\ (1 \leq j \leq d)\}.$$

Also we use the notation

$$\sigma u = (\sigma_1 u_1, \sigma_2 u_2, ..., \sigma_d u_d) \in \mathbb{R}^d.$$

Then $(\sigma u)^k = \sigma^k u^k$ and $(\sigma u)^{2k} = u^{2k}$. By using this notation, we can derive the asymptotic behavior of $\delta(t - u^{2k})|u^h|$ for $t \to 0$.

Lemma 4 *Let $G(u^{2k}, u^k, u)$ be a real-valued C_1-class function of (u^{2k}, u^k, u) $(u \in \mathbb{R}^d)$. The following asymptotic expansion holds as $t \to +0$,*

$$\begin{aligned}
&\int_{[-1,1]^d} du\ \delta(t-u^{2k})|u|^h G(u^{2k}, u^k, u) \\
&= t^{\lambda-1}(-\log t)^{m-1} \sum_{\sigma \in S(d)} \int_{[0,1]^d} du^*\ G(t, \sigma^k \sqrt{t}, u) \\
&+ O\Big(t^{\lambda-1}(-\log t)^{m-2}\Big),
\end{aligned} \tag{26}$$

where du^* is a measure defined by Equation (25).

(Proof of Lemma 4) Let $Y(t)$ be the left hand side of Equation (26). Then

$$\begin{aligned} Y(t) &= \sum_{\sigma\in S(d)} \int_{[0,1]^d} \delta(t-(\sigma u)^{2k})|\sigma u|^h G((\sigma u)^{2k},(\sigma u)^k,\sigma u)d(\sigma u) \\ &= \sum_{\sigma\in S(d)} \int_{[0,1]^d} \delta(t-u^{2k})|u|^h G(t,\sigma^k\sqrt{t},u)du. \end{aligned}$$

By using Theorem 4.6 of a book (Watanabe, 2009), if $u\in[0,1]^d$, then

$$\begin{aligned} \delta(t-u^{2k})|u|^h du &= t^{\lambda-1}(-\log t)^{m-1}du^* \\ &\quad +O(t^{\lambda-1}(-\log t)^{m-2}). \end{aligned}$$

By applying this relation to $Y(t)$, we obtain Lemma 4. (Q.E.D.)

5.6 Proof of Lemma 2

Let $\Phi(w)>0$ be an arbitrary C^∞ class function on W_ε. Let $Y(t,\Phi)$ $(t>0)$ be a function defined by

$$Y(t,\Phi)\equiv\int_{K(w)<\varepsilon}\delta(t-K(w))f(x,w)\Phi(w)\varphi(w)dw,$$

whose value is independent of a choice of a resolution map. By using a resolution map $w=g(u)$,

$$Y(t,\Phi)=\sum_{\alpha\in\mathcal{A}}\sum_{\sigma\in S(d)}\int_{[-1,1]^d}du\,\delta(t-u^{2k})\,u^k\,|u|^h a(x,u)\Phi(g(u))b_\alpha(u)du.$$

By Lemma 4, and $\sigma=(\sigma_a,\sigma_b)$,

$$\begin{aligned} Y(t,\Phi) &= t^{\lambda-1/2}(\log t)^{m-1}\sum_{\alpha\in\mathcal{A}^*}\sum_{\sigma_a\in S(m)}(\sigma_a)^k\sum_{\sigma_b\in S(d-m)}(\sigma_b)^k \\ &\quad\times\int_{[0,1]^d}du^*\,a(x,\sigma u)\,\Phi(g(\sigma u))\,b_\alpha(\sigma u) \\ &\quad +O(t^{\lambda-1/2}(\log t)^{m-2}). \end{aligned}$$

By the assumption that a true distribution is realizable by a statistical model, Equation (24) shows that there exists x such that $a(x,u)\neq 0$ for $u^{2k}=0$. On the support of du^*,

$$\sigma u=(\sigma_a u_a,\sigma_b u_b)=(0,\sigma_b u_b),$$

consequently the main order term of $Y(t,\Phi)$ is determined by $\Phi(0,u_b)$.

In order to prove Lemma, it is sufficient to prove that $Q(K(g(u)),\varphi(g(u)))=1$ is equivalent to the proposition that, for an arbitrary Φ, the main term of $Y(t,\Phi)$ is equal to zero.

First, assume that $Q(K(g(u)),\varphi(g(u)))=1$. Then at least one k_j $(1\leq j\leq m)$ is odd, σ_a^k takes both values ±1, hence

$$\sum_{\sigma_a\in S(m)}\sigma_a^k=0,$$

which shows that the coefficient of the main order term in $Y(t,\Phi)$ $(t\to+0)$ is zero for an arbitrary $\Phi(w)$. Second, assume that $Q(K(g(u)),\varphi(g(u)))=0$. Then all k_j $(1\leq j\leq m)$ are even, hence

$$\sum_{\sigma_a\in S(m)}\sigma_a^k=\sum_{\sigma_a\in S(m)}1\neq 0.$$

Then there exists a function $\Phi(w)$ such that the main order term is not equal to zero. Therefore $Q(K(g(u)),\varphi(g(u)))$ does not depend on the resolution map. (Q.E.D.)

5.7 Proof of Theorem 4

In this subsection, we prove Theorem 4 using the foregoing preparations. We need to study A_n and B_n in Equation (22) and Equation (23). Firstly, we study A_n.

$$\begin{aligned} A_n &= \sum_{\alpha\in\mathcal{A}}\int_{[-1,1]^d} du\ \exp(-n\beta u^{2k}+\beta\sqrt{n}u^k\xi_n(u))|u|^h b_\alpha(u) \\ &= \sum_{\alpha\in\mathcal{A}}\int_{[-1,1]^d} du \int_0^\infty dt\ \delta(t-u^{2k})|u|^h b_\alpha(u) \\ &\quad \times\exp(-n\beta u^{2k}+\beta\sqrt{n}u^k\xi_n(u)). \end{aligned}$$

By substitution $t := t/(n\beta)$ and $dt := dt/(n\beta)$,

$$\begin{aligned} A_n &= \sum_{\alpha\in\mathcal{A}}\int_{[-1,1]^d} b_\alpha(u)du \int_0^\infty \frac{dt}{n\beta}\delta\Big(\frac{t}{n\beta}-u^{2k}\Big)|u|^h \\ &\quad \times\exp(-n\beta u^{2k}+\beta\sqrt{n}u^k\xi_n(u)). \end{aligned}$$

For simple notations, we use

$$\begin{aligned} \int_{\mathcal{M}} du^* &\equiv \sum_{\alpha\in\mathcal{A}^*}\sum_{\sigma\in S(d)}\int_{[0,1]^d} b_\alpha(u)\,du^*, \\ \xi_n^*(u) &\equiv \sigma^k\xi_n(u), \end{aligned}$$

where $\{\mathcal{U}_\alpha\ ;\ \alpha\in\mathcal{A}^*\}$ is the set of all essential local coordinates. Then by using Lemma 4, $\delta(t/n\beta - u^{2k})$ can be asymptotically expanded for $n\beta\to 0$, hence

$$\begin{aligned} A_n &= \int_{\mathcal{M}} du^* \int_0^\infty \frac{dt}{n\beta}\Big(\frac{t}{n\beta}\Big)^{\lambda-1}\Big(-\log(\frac{t}{n\beta})\Big)^{m-1} \\ &\quad \times\exp(-t+\sqrt{\beta t}\,\xi_n^*(u)) + O_p(\frac{(\log(n\beta))^{m-2}}{(n\beta)^\lambda}) \\ &= \frac{(\log(n\beta))^{m-1}}{(n\beta)^\lambda}\int_{\mathcal{M}} du^* \int_0^\infty dt\ t^{\lambda-1}\exp(-t)\ \exp(\sqrt{\beta t}\,\xi_n^*(u)) \\ &\quad +O_p(\frac{(\log(n\beta))^{m-2}}{(n\beta)^\lambda}). \end{aligned}$$

Since $\beta=\beta_0/\log n\to 0$,

$$\exp(\sqrt{\beta t}\,\xi_n^*(u)) = 1+\sqrt{\beta t}\,\xi_n^*(u)+O_p(\beta).$$

By using the gamma function,

$$\Gamma(\lambda)=\int_0^\infty t^{\lambda-1}\ \exp(-t)\,dt,$$

it follows that

$$\begin{aligned} A_n &= \frac{(\log(n\beta))^{m-1}}{(n\beta)^\lambda}\Big\{\Gamma(\lambda)\Big(\int_{\mathcal{M}} du^*\Big)+\sqrt{\beta}\Gamma(\lambda+\frac{1}{2})\Big(\int_{\mathcal{M}} du^*\xi_n^*(u)\Big)\Big\} \\ &\quad +O_p(\frac{(\log(n\beta))^{m-2}}{(n\beta)^\lambda}). \end{aligned}$$

Secondly, B_n can be calculated by the same way,

$$\begin{aligned} B_n &= \sum_{\alpha\in\mathcal{A}} \int_{[-1,1]^d} du \int_0^\infty dt\ \delta(t-u^{2k})|u|^h b_\alpha(u) \\ &\quad \times (nu^{2k} - \sqrt{n}u^k\xi_n(u)) \exp(-n\beta u^{2k} + \beta\sqrt{n}u^k\xi_n(u)). \end{aligned}$$

By substitution $t := t/(n\beta)$ and $dt := dt/(n\beta)$ and Lemma 4,

$$\begin{aligned} B_n &= \int_{\mathcal{M}} du^* \int_0^\infty \frac{dt}{n\beta} \left(\frac{t}{n\beta}\right)^{\lambda-1} \left(-\log(\frac{t}{n\beta})\right)^{m-1} \\ &\quad \times \frac{1}{\beta}(t - \sqrt{\beta t}\ \xi_n^*(u)) \exp(-t + \sqrt{\beta t}\ \xi_n^*(u)) + O_p(\frac{(\log(n\beta))^{m-2}}{\beta(n\beta)^\lambda}) \\ &= \frac{(\log(n\beta))^{m-1}}{\beta(n\beta)^\lambda} \int_{\mathcal{M}} du^* \int_0^\infty t^{\lambda-1}(t - \sqrt{\beta t}\ \xi_n^*(u)) \exp(-t) \\ &\quad \times \exp(\sqrt{\beta t}\ \xi_n^*(u)) + O_p(\frac{(\log(n\beta))^{m-2}}{\beta(n\beta)^\lambda}). \end{aligned}$$

Therefore,

$$\begin{aligned} B_n &= \frac{(\log(n\beta))^{m-1}}{\beta(n\beta)^\lambda} \Big\{ \Gamma(\lambda+1)\Big(\int_{\mathcal{M}} du^*\Big) + \sqrt{\beta}\,\Gamma(\lambda+\frac{3}{2})\Big(\int_{\mathcal{M}} du^*\xi_n^*(u)\Big) \\ &\quad -\sqrt{\beta}\,\Gamma(\lambda+\frac{1}{2})\Big(\int_{\mathcal{M}} du^*\xi_n^*(u)\Big)\Big\} + O_p(\frac{(\log(n\beta))^{m-2}}{\beta(n\beta)^\lambda}). \end{aligned}$$

Let us define a random variable Θ by

$$\Theta = \frac{\int_{\mathcal{M}} du^*\xi_n^*(u)}{\int_{\mathcal{M}} du^*}. \tag{27}$$

By applying results of A_n and B_n to Equation (21),

$$\mathbb{E}_w^\beta[nK_n(w)] = \frac{1}{\beta} \times \frac{\Gamma(\lambda+1) + \sqrt{\beta}\,\Theta\,\{\Gamma(\lambda+3/2) - \Gamma(\lambda+1/2)\}}{\Gamma(\lambda) + \sqrt{\beta}\,\Theta\Gamma(\lambda+1/2)} + O_p(1).$$

Note that, if a, b, c, d are constants and $\beta \to 0$,

$$\frac{c+\sqrt{\beta}\,d}{a+\sqrt{\beta}\,b} = \frac{c}{a} + \sqrt{\beta}\left(\frac{ad-bc}{a^2}\right) + O(\beta).$$

Then by using an identity,

$$\frac{\Gamma(\lambda)(\Gamma(\lambda+3/2) - \Gamma(\lambda+1/2)) - \Gamma(\lambda+1)\Gamma(\lambda+1/2)}{\Gamma(\lambda)^2} = -\frac{\Gamma(\lambda+1/2)}{2\Gamma(\lambda)},$$

we obtain

$$\mathbb{E}_w^\beta[nK_n(w)] = \frac{1}{\beta}\frac{\Gamma(\lambda+1)}{\Gamma(\lambda)} - \frac{\Theta}{\sqrt{\beta}}\frac{\Gamma(\lambda+1/2)}{2\Gamma(\lambda)} + O_p(1).$$

A random variable U_n is defined by

$$U_n = -\frac{\Theta\Gamma(\lambda+1/2)}{\sqrt{2\lambda}\Gamma(\lambda)}. \tag{28}$$

Then it follows that

$$\mathbb{E}_w^{\beta}[nK_n(w)] = \frac{\lambda}{\beta} + U_n\sqrt{\frac{\lambda}{2\beta}} + O_p(1).$$

By the definition of $\xi_n(u)$, $\mathbb{E}[\Theta] = 0$, hence $\mathbb{E}[U_n] = 0$. By using Cauchy-Schwarz inequality,

$$\Theta^2 \leq \frac{\int_{\mathcal{M}} du^*\ \xi_n^*(u)^2}{\int_{\mathcal{M}} du^*}.$$

Lastly let us study the case that $q(x)$ is realizable by $p(x|w)$. The support of du^* is contained in $u^{2k} = 0$, hence we can apply Equation (24) to Θ,

$$\mathbb{E}[\Theta^2] \leq \frac{\int_{\mathcal{M}} du^* \mathbb{E}[\xi_n^*(u)^2]}{\int_{\mathcal{M}} du^*} = 2.$$

The gamma function satisfies

$$\frac{\Gamma(\lambda+1/2)}{\Gamma(\lambda)} < \sqrt{\lambda} \quad (\lambda > 0).$$

Hence we obtain

$$\mathbb{E}[(U_n)^2] \leq \frac{\mathbb{E}[\Theta^2]}{2\lambda}\Big(\frac{\Gamma(\lambda+1/2)}{\Gamma(\lambda)}\Big)^2 < 1,$$

which completes Theorem 4. (Q.E.D.)

5.8 Proof of Corollary 1

By definition Equation (27) and Equation (28), it is sufficient to prove $\Theta = 0$, where

$$\Theta = \frac{\sum_{\alpha\in\mathcal{A}^*}\sum_{\sigma\in S(d)}\int_{[0,1]^d} b_\alpha(u)\ du^*\ \sigma^k\ \xi_n(u)}{\sum_{\alpha\in\mathcal{A}^*}\sum_{\sigma\in S(d)}\int_{[0,1]^d} b_\alpha(u)\ du^*}.$$

The support of the measure du^* is contained in the set $\{u = (0, u_b)\}$. We use a notation $\sigma = (\sigma_a, \sigma_b) \in \mathbb{R}^m \times \mathbb{R}^{d-m}$. If $Q(q,p,\varphi) = 1$ then there exists a resolution map $w = g(u)$ such that σ_a^k takes values both $+1$ and -1 in arbitrary local coordinate, hence

$$\sum_{\sigma_a\in S(m)} \sigma_a^k = 0.$$

It follows that

$$\sum_{\sigma\in S(d)} \sigma^k\xi_n(0,u_b) = \sum_{\sigma_b\in S(d-m)} \sigma_b^k\xi_n(0,u_b) \sum_{\sigma_a\in S(m)} \sigma_a^k = 0,$$

therefore, $\Theta = 0$, which completes Corollary 1. (Q.E.D.)

5.9 Proof Corollary 2

By using the optimal inverse temperature β^*, we define $T = 1/(\beta^* \log n)$. By the definition, $\mathcal{F} = \mathbb{E}_w^{\beta^*}[nL_n(w)]$. By using Theorem 2 and Theorem 4,

$$\lambda \log n = T\lambda \log n + U_n \sqrt{T\lambda \log n/2} + V_n = 0,$$

where

$$V_n = O_p(\log\log n).$$

It follows that

$$T + \frac{U_n\sqrt{T}}{\sqrt{2\lambda \log n}} - 1 + \frac{V_n}{\log n} = 0.$$

Therefore,

$$\sqrt{T} = -\frac{U_n}{\sqrt{8\lambda \log n}} + \sqrt{1 + \frac{(U_n)^2}{8\lambda \log n} - \frac{V_n}{\log n}}.$$

Since $U_n = O_p(1)$,

$$\sqrt{T} = 1 - \frac{U_n}{\sqrt{8\lambda \log n}} + o_p(\frac{1}{\sqrt{\lambda \log n}}),$$

resulting that

$$\beta^* \log n = 1 + \frac{U_n}{\sqrt{2\lambda \log n}} + o_p(\frac{1}{\sqrt{\lambda \log n}}),$$

which completes Corollary 2. (Q.E.D.)

5.10 Proof of Corollary 3

By using Theorem 4,

$$\begin{aligned}
\mathbb{E}_w^{\beta_1}[nL_n(w)] &= nL_n(w_0) + \frac{\lambda}{\beta_1} + O_p(\sqrt{\log n}), \\
\mathbb{E}_w^{\beta_2}[nL_n(w)] &= nL_n(w_0) + \frac{\lambda}{\beta_2} + O_p(\sqrt{\log n}).
\end{aligned}$$

Since $(1/\beta_1 - 1/\beta_2) = O_p(\log n)$,

$$\lambda = \frac{\mathbb{E}_w^{\beta_1}[nL_n(w)] - \mathbb{E}_w^{\beta_2}[nL_n(w)]}{1/\beta_1 - 1/\beta_2} + O_p(1/\sqrt{\log n}),$$

which shows Corollary 3. (Q.E.D.)

5.11 Proof Theorem 5

By using Equation (9) and Equation (10),

$$\mathbb{E}_w^{\beta}[nL_n(w)] = nL_n(w_0) + \mathbb{E}_w^{\beta}[nK_n(w)],$$

the proof of Theorem 5 results in evaluating $\mathbb{E}_w^{\beta}[nK_n(w)]$. By Lemma 1 for the case $r = 1/4$,

$$\mathbb{E}_w^{\beta}[nK_n(w)] = \frac{D_n + o_p(\exp(-\sqrt{n}))}{C_n + o_p(\exp(-\sqrt{n}))},$$

where C_n and D_n are respectively defined by

$$\begin{aligned} C_n &= \int_{K<1/n^{1/4}} \exp(-n\beta K_n(w))\varphi(w)dw, \\ D_n &= \int_{K<1/n^{1/4}} nK_n(w)\exp(-n\beta K_n(w))\varphi(w)dw. \end{aligned}$$

If a statistical model is regular, the maximum likelihood estimator $\hat{w}$ converges to w_0 in probability. Let $J_n(w)$ be $d \times d$ matrices defined by

$$(J_n)_{ij}(w) = \frac{\partial^2 K_n}{\partial w_i \partial w_j}(w).$$

There exists a parameter w^* such that

$$K_n(w) = K_n(\hat{w}) + \frac{1}{2}(w-\hat{w}) \cdot J_n(w^*)(w-\hat{w}).$$

Since $\hat{w} \to w_0$ in probability and $K(w) < 1/n^{1/4}$, $w^* \to w_0$ in probability. Then

$$\begin{aligned} &\|J_n(w^*) - J(w_0)\| \leq \|J_n(w^*) - J_n(w_0)\| + \|J_n(w_0) - J(w_0)\| \\ &\leq \|w^* - w_0\| \sup_{K(w)<1/n^{1/4}} \left\| \frac{\partial J_n(w)}{\partial w} \right\| + \|J_n(w_0) - J(w_0)\|, \end{aligned}$$

which converges to zero in probability as $n \to \infty$. Therefore,

$$J_n(w^*) = J(w_0) + o_p(1).$$

Since the model is regular, $J(w_0)$ is a positive definite matrix. Now we define

$$\begin{aligned} C_n &= \exp(-n\beta K_n(\hat{w})) \\ &\quad \times \int_{K(w)<n^{1/4}} \exp(-\frac{n\beta}{2}(w-\hat{w}) \cdot (J(w_0) + o_p(1))(w-\hat{w}))\varphi(w)dw. \end{aligned}$$

By substituting

$$u = \sqrt{n\beta}(w-\hat{w}),$$

it follows that

$$\begin{aligned} C_n &= \exp(-n\beta K_n(\hat{w}))(n\beta)^{-d/2} \\ &\quad \times \int \exp(-\frac{1}{2}u \cdot (J(w_0) + o_p(1))u)\varphi(\hat{w} + \frac{u}{\sqrt{n\beta}})du \\ &= \frac{(2\pi)^{d/2}\exp(-n\beta K_n(\hat{w}))(\varphi(\hat{w}) + o_p(1))}{(n\beta)^{d/2}\det(J(w_0) + o_p(1))^{1/2}}. \end{aligned}$$

H	1	2	3	4	5	6
WBIC_1 Ave.	17899.82	3088.90	71.11	78.21	83.23	87.58
WBIC_1 Std.	1081.30	226.94	3.67	3.78	3.97	4.09
WBIC_2 Ave.	17899.77	3089.03	71.18	75.43	82.54	86.83
WBIC_2 Std.	1081.30	226.97	3.54	3.89	4.03	4.08
BIC Ave.	17899.77	3089.03	71.18	83.47	91.86	94.87
BIC Std.	1081.30	226.97	3.54	3.89	4.03	4.08

Table 2: WBIC and BIC in Model Selection

In the same way,

$$\begin{aligned} D_n &= \exp(-n\beta K_n(\hat{w})) \\ &\quad \times \int_{K(w)<1/n^{1/4}} \Big(nK_n(\hat{w}) + \frac{n}{2}(w-\hat{w}) \cdot (J(w_0)+o_p(1))(w-\hat{w}) \Big) \\ &\quad \times \exp\Big(-\frac{n\beta}{2}(w-\hat{w}) \cdot (J(w_0)+o_p(1))(w-\hat{w}) \Big) \varphi(w) dw \\ &= \frac{(2\pi)^{d/2} \exp(-n\beta K_n(\hat{w}))(\varphi(\hat{w})+o_p(1))}{(n\beta)^{d/2} \det(J(w_0)+o_p(1))^{1/2}} \Big(nK_n(\hat{w}) + \frac{d}{2\beta} + o_p(1) \Big). \end{aligned}$$

Here $nK_n(\hat{w}) = O_p(1)$, because the true distribution is regular for a statistical model. Therefore,

$$\mathbb{E}_w^\beta[nL_n(w)] = nL_n(w_0) + nK_n(\hat{w}) + \frac{d}{2\beta} + o_p(1),$$

which completes Theorem 5. (Q.E.D.)

6. A Method How to Use WBIC

In this section we show a method how to use WBIC in statistical model evaluation. The main theorems have already been mathematically proved, hence WBIC has a theoretical support. The following experiment was conducted not for proving theorems but for illustrating a method how to use it.

6.1 Statistical Model Selection

Firstly, we study model selection by using WBIC.

Let $x \in \mathbb{R}^M$, $y \in \mathbb{R}^N$, and $w = (A, B)$, where A is an $H \times M$ matrix and B is an $N \times H$ matrix. A reduced rank regression model is defined by

$$p(x,y|w) = \frac{r(x)}{(2\pi\sigma^2)^{N/2}} \exp\Big(-\frac{1}{2\sigma^2} \|y - BAx\|^2 \Big),$$

where $r(x)$ is a probability density function of x and σ^2 is the variance of an output. Let $\mathcal{N}_M(0,\Sigma)$ denote the M-dimensional normal distribution with the average zero and the covariance matrix Σ.

In an experiment, we set $\sigma = 0.1$, $r(x) = \mathcal{N}_M(0, 3^2 I)$, where I is the identity matrix, and $\varphi(w) = \mathcal{N}_d(0, 10^2 I)$. The true distribution was fixed as $p(x,y|w_0)$, where $w_0 = (A_0, B_0)$ was determined so

that A_0 and B_0 were respectively an $H_0 \times M$ matrix and an $M \times H_0$ matrix. Note that, in reduced rank regression models, RLCTs and multiplicities were clarified by a research (Aoyagi and Watanabe, 2005) and $Q(K(w), \varphi(w)) = 1$ for arbitrary $q(x)$, $p(x|w)$, and $\varphi(w)$. In the experiment, $M = N = 6$ and the true rank was set as $H_0 = 3$. Each element of A_0 and B_0 was taken from $\mathcal{N}_{\mathrm{J}}(0, 0.2^2)$ and fixed. From the true distribution $p(x,y|w_0)$, 100 sets of $n = 500$ training samples were generated.

The Metropolis method was employed for sampling from the posterior distribution,

$$p(w|X_1, X_2, ..., X_n) \propto \exp(-\beta n L_n(w) + \log \varphi(w)),$$

where $\beta = 1/\log n$. Every Metropolis trial was generated from a normal distribution $\mathcal{N}_d(0, (0.0012)^2 I)$, by which the acceptance probability was 0.1-0.9. First 50000 Metropolis trails were not used. After 50000 trails, $R = 2000$ parameters $\{w_r; r = 1, 2, ..., R\}$ were obtained in every 100 Metropolis steps. The expectation value of a function $G(w)$ over the posterior distribution was approximated by

$$\mathbb{E}_w^{\beta}[G(w)] = \frac{1}{R} \sum_{r=1}^{R} G(w_r).$$

The six statistical models $H = 1, 2, 3, 4, 5, 6$ were compared by the criterion,

$$\mathrm{WBIC} = \mathbb{E}_w^{\beta}[nL_n(w)], \quad (\beta = 1/\log n).$$

To compare these values among several models, we show both WBIC_1, WBIC_2, and BIC in Table 2. In the table, the average and the standard deviation of WBIC_1 defined by

$$\mathrm{WBIC}_1 = \mathrm{WBIC} - nS_n,$$

for 100 independent sets of training samples are shown, where the empirical entropy of the true distribution

$$S_n = -\frac{1}{n} \sum_{i=1}^{n} \log q(X_i)$$

does not depend on a statistical model. Although nS_n does not affect the model selection, its standard deviation is in proportion to $\sqrt{n}$. In order to estimate the standard deviation of the essential part of WBIC, the effect of nS_n was removed. In 100 independent sets of training samples, the true model $H = 3$ was chosen 100 times in this experiment, which demonstrates a typical application method of WBIC.

Also WBIC_2 in Table 2 shows the average and the standard deviation of

$$\mathrm{WBIC}_2 = nL_n(\hat{w}) + \lambda \log n - (m-1) \log \log n - nS_n,$$

where $\hat{w}$ is the maximum likelihood estimator, λ is the real log canonical threshold, and m is the multiplicity. The maximum likelihood estimator $\hat{w} = (\hat{A}, \hat{B})$ is given in reduced rank regression (Anderson, 1951),

$$\hat{B}\hat{A} = VV^t YX^t (XX^t)^{-1},$$

where t shows the transposed matrix and X, Y, and V are matrices defined as follows.
(1) X_{ij} is the i-th coefficient of x_j
(2) Y_{ij} is the i-th coefficient of y_j.

H	1	2	3	4	5	6
Theory λ	5.5	10	13.5	15	16	17
Theory m	1	1	1	2	1	2
Average λ	5.51	9.95	13.49	14.80	15.72	16.55
Std. Dev. λ	0.17	0.31	0.52	0.65	0.66	0.72

Table 3: RLCTs for the case $H_0 = 3$

(3) The matrix V is made of eigen vectors with respect to the maximum H eigen values of the matrix,

$$YX^t(XX^t)^{-1}XY^t.$$

If we know the model that is equal to the true distribution, and if we have theoretical real log canonical threshold and multiplicity, then we can calculate WBIC_2.

The value BIC in Table 2 shows the average and the standard deviation of Shwarz BIC,

$$\mathrm{BIC} = nL_n(\hat{w}) + \frac{d}{2}\log n - nS_n,$$

where d is the essential dimension of the parameter space of the reduced rank regression,

$$d = H(M+N-H).$$

We used this dimension because the number of parameters in reduced rank regression is $H(M+N)$ and it has free dimension H^2. These results show that WBIC is a better approximator of the Bayes free energy than BIC.

6.2 Estimating RLCT

Secondly, we study a method how to estimate an RLCT. By using the same experiment as the foregoing subsection, we estimated RLCTs of reduced rank regression models by using Corollary 3. Based on Equation (19), the estimated RLCT is given by

$$\hat{\lambda} = \frac{\mathbb{E}_w^{\beta_1}[nL_n(w)] - \mathbb{E}_w^{\beta_2}[nL_n(w)]}{1/\beta_1 - 1/\beta_2},$$

where $\beta_1 = 1/\log n$ and $\beta_2 = 1.5/\log n$ and we used Equation (20) in the calculation of $\mathbb{E}_w^{\beta_2}[\ ]$. Theory λ in Table 3 shows the theoretical values of RLCTs of reduced rank regression. For the cases when true distributions are unrealizable by statistical models, RLCTs are given by half the dimension of the parameter space, $\lambda = H(M+N-H)/2$. In Table 3, averages and standard deviations of λ shows estimated RLCTs. The theoretical RLCTs were well estimated. The difference between theory and experimental results was caused by the effect of the smaller order terms than $\log n$. In the case the multiplicity $m = 2$, the term $\log\log n$ also affected the results.

7. Discussion

In this section, we discuss the widely applicable information criterion from three different points of view.

7.1 WAIC and WBIC

Firstly, let us study the difference between the free energy and the generalization error. In the present paper, we study the Bayes free energy $\mathcal{F}$ as the statistical model selection criterion. Its expectation value is given by

$$\mathbb{E}[\mathcal{F}] = nS + \int q(x^n) \log \frac{q(x^n)}{p(x^n)} dx^n,$$

where S is the entropy of the true distribution,

$$\begin{aligned} q(x^n) &= \prod_{i=1}^{n} q(x_i), \\ p(x^n) &= \int \prod_{i=1}^{n} p(x_i|w)\varphi(w) dw, \end{aligned}$$

and $dx^n = dx_1 dx_2 \cdots dx_n$. Hence minimization of $\mathbb{E}[\mathcal{F}]$ is equivalent to minimization of the Kullback-Leibler distance from the $q(x^n)$ to $p(x^n)$.

There is a different model evaluation criterion, which is the generalization loss defined by

$$\mathcal{G} = -\int q(x) \log p^*(x) dx, \tag{29}$$

where $p^*(x)$ is the Bayes predictive distribution defined by $p^*(x) = \mathbb{E}_w^{\beta}[p(x|w)]$, with $\beta = 1$. The expectation value of $\mathcal{G}$ satisfies

$$\mathbb{E}[\mathcal{G}] = S + \mathbb{E}\Big[\int q(x) \log \frac{q(x)}{p^*(x)} dx\Big].$$

Hence minimization of $\mathbb{E}[\mathcal{G}]$ is equivalent to minimization of the Kullback-Leibler distance from $q(x)$ to $p^*(x)$. Both of $\mathcal{F}$ and $\mathcal{G}$ are important in statistics and learning theory, however, they are different criteria.

The well-known model selection criteria AIC and BIC are respectively defined by

$$\begin{aligned} \mathrm{AIC} &= L_n(\hat{w}) + \frac{d}{n}, \\ \mathrm{BIC} &= nL_n(\hat{w}) + \frac{d}{2} \log n. \end{aligned} \tag{30}$$

If a true distribution is realizable by and regular for a statistical model, then

$$\begin{aligned} \mathbb{E}[\mathrm{AIC}] &= \mathbb{E}[\mathcal{G}] + o(\frac{1}{n}), \\ \mathbb{E}[\mathrm{BIC}] &= \mathbb{E}[\mathcal{F}] + O(1). \end{aligned}$$

These relations can be generalized onto singular statistical models. We define WAIC and WBIC by

$$\begin{aligned} \mathrm{WAIC} &= T_n + V_n/n, \\ \mathrm{WBIC} &= \mathbb{E}_w^{\beta}[nL_n(w)], \quad \beta = 1/\log n, \end{aligned}$$

where

$$\begin{aligned} T_n &= -\frac{1}{n}\sum_{i=1}^{n}\log p^*(X_i|w), \\ V_n &= \sum_{i=1}^{n}\Big\{\mathbb{E}_w[(\log p(X_i|w))^2]-\mathbb{E}_w[\log p(X_i|w)]^2\Big\}. \end{aligned}$$

Then, even if a statistical model is unrealizable by and singular for a statistical model,

$$\mathbb{E}[\mathrm{WAIC}] = \mathbb{E}[\mathcal{G}]+O(\frac{1}{n^2}), \tag{31}$$

$$\mathbb{E}[\mathrm{WBIC}] = \mathbb{E}[\mathcal{F}]+O(\log\log n), \tag{32}$$

where Equation (31) was proved in a book (Watanabe, 2009, 2010b), whereas Equation (32) has been proved in the present paper. In fact, the difference between the average leave-one-out cross validation and the average generalization error is in proportion to $1/n^2$ and the difference between the leave-one-out cross validation and WAIC is also in proportion to $1/n^2$. The difference between $\mathbb{E}[\mathrm{WBIC}]$ and $\mathbb{E}[\mathcal{F}]$ is caused by the multiplicity m. If a statistical model is realizable by and regular for a statistical model, WAIC and WBIC respectively coincide with AIC and BIC,

$$\begin{aligned} \mathrm{WAIC} &= \mathrm{AIC}+o_p(\frac{1}{n}), \\ \mathrm{WBIC} &= \mathrm{BIC}+o_p(1). \end{aligned}$$

Theoretical comparison of WAIC and WBIC in singular model selection is an important problem for future study.

7.2 Other Methods How to Evaluate Free Energy

Secondly, we discuss several methods how to numerically evaluate the Bayes free energy. There are three methods other than WBIC.

Firstly, let $\{\beta_j; j=0,1,2,...,J\}$ be a sequence which satisfies

$$0=\beta_0<\beta_1<\cdots<\beta_J=1.$$

Then the Bayes free energy satisfies

$$\mathcal{F}=-\sum_{j=1}^{J}\log\mathbb{E}_w^{\beta_{j-1}}[\exp(-n(\beta_j-\beta_{j-1})L_n(w))].$$

This method can be used without asymptotic theory. We can estimate $\mathcal{F}$, if the number J is sufficiently large and if all expectation values over the posterior distributions $\{\mathbb{E}_w^{\beta_{j-1}}[\ ]\}$ are precisely calculated. The disadvantage of this method is its huge computational costs for accurate calculation. In the present paper, this method is referred to as 'all temperatures method'.

Secondly, the importance sampling method is often used. Let $H(w)$ be a function which approximates $nL_n(w)$. Then, for an arbitrary function $G(w)$, we define an expectation value $\hat{\mathbb{E}}_w[\ ]$ by

$$\hat{\mathbb{E}}_w[G(w)]=\frac{\int G(w)\exp(-H(w))\varphi(w)dw}{\int\exp(-H(w))\varphi(w)dw}.$$

Method	Asymptotics	RLCT	Comput. Cost
All Temperatures	Not used	Not Used	Huge
Importance Sampling	Not used	Not Used	Small
Two-Step	Used	Used	Small
WBIC	Used	Not Used	Small

Table 4: Comparison of Several Methods

Then

$$\begin{aligned}\mathcal{F} &= -\log \hat{\mathbb{E}}_w[\exp(-nL_n(w)+H(w))] \\ &\quad -\log \int \exp(-H(w))\varphi(w)dw,\end{aligned}$$

where the last term is the free energy of $H(w)$. Hence if we find $H(w)$ whose free energy is analytically calculated and if it is easy to generate random samples from $\hat{\mathbb{E}}_w[\ ]$, then $\mathcal{F}$ can be numerically evaluated. The accuracy of this method strongly depends on the choice of $H(w)$.

Thirdly, a two-step method was proposed (Drton, 2010). Assume that we have theoretical values about RLCTs for all cases about true distribution and statistical models. Then, in the first step, a null hypothesis model is chosen by using BIC. In the second step, the optimal model is chosen by using RLCTs with the assumption that the null hypothesis model is a true distribution. If the selected model is different from the null hypothesis model, then the same procedure is recursively applied until the null hypothesis model becomes the optimal model. In this method, asymptotic theory is necessary but RLCTs do not contain fluctuations because they are theoretical values.

Compared with these methods, WBIC needs asymptotic theory but it does not theoretical results about RLCT. The theoretical comparison of these methods is summarized in Table 4.

The effectiveness of a model selection method strongly depends on a statistical condition which is determined by a true distribution, a statistical model, a prior distribution, and a set of training samples. Under some condition, one method may be more effective, however, under the other condition, another may be. The proposed method WBIC gives a new approach in numerical calculation of the Bayes free energy which is more useful with cooperation with the conventional method. It is a future study to clarify which method is recommended in what statistical conditions.

Remark. It is one of the most important problems in Bayes statistics how to make accurate Markov Chain Monte Carlo (MCMC) process. There are several MCMC methods, for example, the Metropolis method, the Gibbs sampler method, the Hybrid Monte Carlo method, and the exchange Monte Carlo method. Numerical calculation of WBIC depends on the accuracy of MCMC process.

7.3 Algebraic Geometry and Statistics

Lastly, let us discuss a relation between algebraic geometry and statistics. In the present paper, we define the parity of a statistical $Q(K(w),\varphi(w))$ and proved that it affects the asymptotic behavior of WBIC. In this subsection we show three mathematical properties of the parity of a statistical model.

Firstly, the parity has a relation to the analytic continuation of $K(w)^{1/2}$. For example, by using blow-up, $(a,b)=(a_1,a_1b_1)=(a_2b_2,b_2)$, it follows that analytic continuation of $(a^2+b^2)^{1/2}$ is given

by

$$(a^2+b^2)^{1/2} = a_1\sqrt{1+b_1^2} = b_2\sqrt{a_2^2+1},$$

which takes both positive and negative values. On the other hand, $(a^4+b^4)^{1/2}$ takes only nonnegative value. The parity indicates such difference.

Secondly, the parity has a relation to statistical model with a restricted parameter set. For example, a statistical model

$$p(x|a) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{(x-a)^2}{2})$$

whose parameter set is given by $\{a \geq 0\}$ is equivalent to a statistical model $p(x|b^2)$ and $\{b \in \mathbb{R}\}$. In other words, a statistical model which has restricted parameter set is statistically equivalent to another even model which has unrestricted parameter set. We have a conjecture that an even statistical model has some relation to a model with a restricted parameter model.

And lastly, the parity has a relation to the difference of $K(w)$ and $K_n(w)$. As is proven (Watanabe, 2001a), the relation

$$-\log\int \exp(-nK_n(w))\varphi(w)dw = -\log\int \exp(-nK(w))\varphi(w)dw + O_p(1)$$

holds independent of the parity of a statistical model. On the other hand, if $\beta = 1/\log n$, then

$$\begin{aligned}\mathbb{E}_w^\beta[nK_n(w)] &= \frac{\int nK(w)\exp(-n\beta K(w))\varphi(w)dw}{\int \exp(-n\beta K(w))\varphi(w)} \\ &\quad + U_n\sqrt{\log n} + O_p(1).\end{aligned}$$

If the parity is odd, then $U_n = 0$, otherwise U_n is not equal to zero in general. This fact shows that the parity shows difference in a fluctuation of the likelihood function.

8. Conclusion

We proposed a widely applicable Bayesian information criterion (WBIC) which can be used even if a true distribution is unrealizable by and singular for a statistical model and proved that WBIC has the same asymptotic expansion as the Bayes free energy. Also we developed a method how to estimate real log canonical thresholds even if a true distribution is unknown.

Acknowledgments

This research was partially supported by the Ministry of Education, Science, Sports and Culture in Japan, Grant-in-Aid for Scientific Research 23500172.

References

T. W. Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Annals of Mathematical Statistics*, 22:327–351, 1951.

M. Aoyagi and K. Nagata. Learning coefficient of generalization error in Bayesian estimation and Vandermonde matrix-type singularity. *Neural Computation*, 24(6):1569–1610, 2012.

M. Aoyagi and S. Watanabe. Stochastic complexities of reduced rank regression in Bayesian estimation. *Neural Networks*, 18(7):924–933, 2005.

M. F. Atiyah. Resolution of singularities and division of distributions. *Communications of Pure and Applied Mathematics*, 13:145–150, 1970.

I. N. J. Bernstein. Analytic continuation of distributions with respect to a parameter. *Functional Analysis and its Applications*, 6(4):26–40, 1972.

M. Drton. Likelihood ratio tests and singularities. *The Annals of Statistics*, 37:979–1012, 2009.

M. Drton. Reduced rank regression. In *Workshop on Singular Learning Theory*. American Institute of Mathematics, 2010.

M. Drton, B. Sturmfels, and S. Sullivant. *Lecures on Algebraic Statistics*. Birkhäuser, Berlin, 2009.

I. M. Gelfand and G. E. Shilov. *Generalized Functions. Volume I: Properties and Operations*. Academic Press, San Diego, 1964.

I. J. Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. MIT Press, Cambridge, 1965.

H. Hironaka. Resolution of singularities of an algebraic variety over a field of characteristic zero. *Annals of Mathematics*, 79:109–326, 1964.

M. Kashiwara. B-functions and holonomic systems. *Inventiones Mathematicae*, 38:33–53, 1976.

F. J. Király, P. Büuau, F. C. Meinecke, D. A. J. Blythe, and K-R Müller. Algebraic geometric comparison of probability distributions. *Journal of Machine Learning Research*, 13:855–903, 2012.

J. Kollár. Singularities of pairs. In *Algebraic Geometry, Santa Cruz 1995, Proceedings of Symposia in Pure Mathematics*, volume 62, pages 221–286. American Mathematical Society, 1997.

S. Lin. *Algebraic Methods for Evaluating Integrals in Bayesian Statistics*. PhD thesis, Ph.D. dissertation, University of California, Berkeley, 2011.

D. Rusakov and D. Geiger. Asymptotic model selection for naive Bayesian network. *Journal of Machine Learning Research*, 6:1–35, 2005.

M. Saito. On real log canonical thresholds. *arXiv:0707.2308v1*, 2007.

M. Sato and T. Shintani. On zeta functions associated with prehomogeneous vector space. *Annals of Mathematics*, 100:131–170, 1974.

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

A. Varchenko. Newton polyhedrons and estimates of oscillatory integrals. *Functional Analysis and its Applications*, 10(3):13–38, 1976.

S. Watanabe. Algebraic analysis for singular statistical estimation. *Lecture Notes in Computer Sciences*, 1720:39–50, 1999.

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

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

S. Watanabe. *Algebraic geometry and statistical learning theory*. Cambridge University Press, Cambridge, UK, 2009.

S. Watanabe. Asymptotic learning curve and renormalizable condition in statistical learning theory. *Journal of Physics Coneference Series*, 233(1), 2010a. 012014. doi: 10.1088/1742-6596/233/1/012014.

S. Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3591, 2010b.

K. Yamazaki and S. Watanabe. Singularities in mixture models and upper bounds of stochastic complexity. *Neural Networks*, 16(7):1029–1038, 2003.

K. Yamazaki and S. Watanabe. Singularities in complete bipartite graph-type boltzmann machines and upper bounds of stochastic complexities. *IEEE Transactions on Neural Networks*, 16(2): 312–324, 2005.

P. Zwiernik. Asymptotic model selection and identifiability of directed tree models with hidden variables. *CRiSM report*, 2010.

P. Zwiernik. An asymptotic behaviour of the marginal likelihood for general markov models. *Journal of Machine Learning Research*, 12:3283–3310, 2011.

Journal of Machine Learning Research 14 (2013) 899-925 Submitted 12/11; Revised 10/12; Published 4/13

Truncated Power Method for Sparse Eigenvalue Problems

Xiao-Tong Yuan XTYUAN1980@GMAIL.COM
Tong Zhang TZHANG@STAT.RUTGERS.EDU
Department of Statistics
Rutgers University
New Jersey, 08816, USA

Editor: Hui Zou

Abstract

This paper considers the sparse eigenvalue problem, which is to extract dominant (largest) sparse eigenvectors with at most k non-zero components. We propose a simple yet effective solution called *truncated power method* that can approximately solve the underlying nonconvex optimization problem. A strong sparse recovery result is proved for the truncated power method, and this theory is our key motivation for developing the new algorithm. The proposed method is tested on applications such as sparse principal component analysis and the densest k-subgraph problem. Extensive experiments on several synthetic and real-world data sets demonstrate the competitive empirical performance of our method.

Keywords: sparse eigenvalue, power method, sparse principal component analysis, densest k-subgraph

1. Introduction

Given a $p \times p$ symmetric positive semidefinite matrix A, the *largest k-sparse eigenvalue* problem aims to maximize the quadratic form $x^\top Ax$ with a sparse unit vector $x \in \mathbb{R}^p$ with no more than k non-zero elements:

$$\lambda_{\max}(A,k) = \max_{x\in\mathbb{R}^p} x^\top Ax, \qquad \text{subject to } \|x\| = 1, \quad \|x\|_0 \le k, \tag{1}$$

where $\|\cdot\|$ denotes the ℓ_2-norm, and $\|\cdot\|_0$ denotes the ℓ_0-norm which counts the number of non-zero entries in a vector. The sparsity is controlled by the values of k and can be viewed as a design parameter. In machine learning applications, for example, principal component analysis, this problem is motivated from the following perturbation formulation of matrix A:

$$A = \bar{A} + E, \tag{2}$$

where A is the empirical covariance matrix, $\bar{A}$ is the true covariance matrix, and E is a random perturbation due to having only a finite number of empirical samples. If we assume that the largest eigenvector $\bar{x}$ of $\bar{A}$ is sparse, then a natural question is to recover $\bar{x}$ from the noisy observation A when the error E is "small". In this context, the problem (1) is also referred to as sparse principal component analysis (sparse PCA).

In general, problem (1) is non-convex. In fact, it is also NP-hard because it can be reduced to the subset selection problem for ordinary least squares regression (Moghaddam et al., 2006), which is

known to be NP-hard. Various researchers have proposed approximate optimization methods: some are based on greedy procedures (e.g., Moghaddam et al., 2006; Jolliffe et al., 2003; d'Aspremont et al., 2008), and some others are based on various types of convex relaxation or reformulation (e.g., d'Aspremont et al., 2007; Zou et al., 2006; Journée et al., 2010). Statistical analysis of sparse PCA has also received significant attention. Under the high dimensional single spike model, Johnstone (2001) proved the consistency of PCA using a subset of features corresponding to the largest sample variances. Under the same single spike model, Amini and Wainwright (2009) established conditions for recovering the non-zero entries of eigenvectors using the convex relaxation method of d'Aspremont et al. (2007). However, these results were concerned with variable selection consistency under a relatively simple and specific example with limited general applicability. More recently, Paul and Johnstone (2012) studied an extension called multiple spike model, and proposed an augmented sparse PCA method for estimating each of the leading eigenvectors and investigated the rate of convergence of their procedure in the high dimensional setting. In another recent work that is independent of ours, Ma (2013) analyzed an iterative thresholding method for recovering the sparse principal subspace. Although it also focused on the multiple spike covariance model, the procedures and techniques considered there are closely related to the method studied in this paper. In addition, Shen et al. (2013) analyzed the consistency of the sparse PCA method of Shen and Huang (2008), and Cai et al. (2012) analyzed the optimal convergence rate of sparse PCA and introduced an adaptive procedure for estimating the principal subspace.

This paper proposes and analyzes a computational procedure called *truncated power iteration method* that approximately solves (1). This method is similar to the classical power method, with an additional truncation operation to ensure sparsity. We show that if the true matrix $\bar{A}$ has a sparse (or approximately sparse) dominant eigenvector $\bar{x}$, then under appropriate assumptions, this algorithm can recover $\bar{x}$ when the spectral norm of sparse submatrices of the perturbation E is small. Moreover, this result can be proved under relative generality without restricting ourselves to the rather specific spike covariance model. Therefore our analysis provides strong theoretical support for this new method, and this differentiates our proposal from previous studies. We have applied the proposed method to sparse PCA and to the densest k-subgraph finding problem (with proper modification). Extensive experiments on synthetic and real-world large-scale data sets demonstrate both the competitive sparse recovering performance and the computational efficiency of our method.

It is worth mentioning that the truncated power method developed in this paper can also be applied to the *smallest k-sparse eigenvalue* problem given by:

$$\lambda_{\min}(A,k) = \min_{x\in\mathbb{R}^p} x^\top Ax, \qquad \text{subject to } \|x\| = 1, \quad \|x\|_0 \le k,$$

which also has many applications in machine learning.

1.1 Notation

Let $\mathbb{S}^p = \{A \in \mathbb{R}^{p\times p} \mid A = A^\top\}$ denote the set of symmetric matrices, and $\mathbb{S}^p_+ = \{A \in \mathbb{S}^p, A \succeq 0\}$ denote the cone of symmetric, positive semidefinite matrices. For any $A \in \mathbb{S}^p$, we denote its eigenvalues by $\lambda_{\min}(A) = \lambda_p(A) \le \cdots \le \lambda_1(A) = \lambda_{\max}(A)$. We use $\rho(A)$ to denote the spectral norm of A, which is $\max\{|\lambda_{\min}(A)|, |\lambda_{\max}(A)|\}$, and define

$$\rho(A,s) := \max\{|\lambda_{\min}(A,s)|, |\lambda_{\max}(A,s)|\}. \tag{3}$$

The i-th entry of vector x is denoted by $[x]_i$ while $[A]_{ij}$ denotes the element on the i-th row and j-th column of matrix A. We denote by A_k any $k\times k$ principal submatrix of A and by A_F the principal

submatrix of A with rows and columns indexed in set F. If necessary, we also denote A_F as the restriction of A on the rows and columns indexed in F. Let $\|x\|_p$ be the ℓ_p-norm of a vector x. In particular, $\|x\|_2 = \sqrt{x^\top x}$ denotes the Euclidean norm, $\|x\|_1 = \sum_{i=1}^d |[x]_i|$ denotes the ℓ_1-norm, and $\|x\|_0 = \#\{j : [x]_j \neq 0\}$ denotes the ℓ_0-norm. For simplicity, we also denote the ℓ_2 norm $\|x\|_2$ by $\|x\|$. In the rest of the paper, we define $Q(x) := x^\top Ax$. We let $\mathrm{supp}(x) := \{j : [x]_j \neq 0\}$ denote the support set of vector x. Given an index set F, we define

$$x(F) := \arg\max_{x \in \mathbb{R}^p} x^\top Ax, \qquad \text{subject to } \|x\| = 1, \quad \mathrm{supp}(x) \subseteq F.$$

Finally, we denote by $I_{p\times p}$ the $p \times p$ identity matrix.

1.2 Paper Organization

The remaining of this paper is organized as follows: §2 describes the truncated power iteration algorithm that approximately solves problem (1). In §3 we analyze the solution quality of the proposed algorithm. §4 evaluates the relevance of our theoretical prediction and the practical performance of the proposed algorithm in applications of sparse PCA and the densest k-subgraph finding problems. We conclude this work and discuss potential extensions in §5.

2. Truncated Power Method

Since $\lambda_{\max}(A,k)$ equals $\lambda_{\max}(A_k^*)$ where A_k^* is the $k \times k$ principal submatrix of A with the largest eigenvalue, one may solve (1) by exhaustively enumerating all subsets of $\{1,\ldots,p\}$ of size k in order to find A_k^*. However, this procedure is impractical even for moderate sized k since the number of subsets is exponential in k.

2.1 Algorithm

Therefore in order to solve the spare eigenvalue problem (1) more efficiently, we consider an iterative procedure based on the standard power method for eigenvalue problems, while maintaining the desired sparsity for the intermediate solutions. The procedure, presented in Algorithm 1, generates a sequence of intermediate k-sparse eigenvectors $x_0, x_1, \ldots$ from an initial sparse approximation x_0. At each time stamp t, the intermediate vector x_{t-1} is multiplied by A, and then the entries are truncated to zeros except for the largest k entries. The resulting vector is then normalized to unit length, which becomes x_t. The cardinality k is a free parameter in the algorithm. If no prior knowledge of sparsity is available, then we have to tune this parameter, for example, through cross-validation. Note that our theory does not require choosing k precisely (see Theorem 4), and thus the tuning is not difficult in practice. At each iteration, the computational complexity is in $O(kp+p)$ which is $O(kp)$ for matrix-vector product Ax_{t-1} and $O(p)$[1] for selecting k largest elements from the obtained vector of length p to get F_t.

Definition 1 *Given a vector x and an index set F, we define the truncation operation $Truncate(x,F)$ to be the vector obtained by restricting x to F, that is*

$$[Truncate(x,F)]_i = \begin{cases} [x]_i & i \in F \\ 0 & otherwise \end{cases}.$$

1. Our actual implementation employs sorting for simplicity, which has a slightly worse complexity of $O(p \ln p)$ instead of $O(p)$.

Algorithm 1: Truncated Power (TPower) Method

Input : matrix $A \in \mathbb{S}^p$, initial vector $x_0 \in \mathbb{R}^p$
Output : x_t
Parameters : cardinality $k \in \{1, ..., p\}$

Let $t = 1$.
repeat
  Compute $x_t' = Ax_{t-1}/\|Ax_{t-1}\|$.
  Let $F_t = \text{supp}(x_t', k)$ be the indices of x_t' with the largest k absolute values.
  Compute $\hat{x}_t = \text{Truncate}(x_t', F_t)$.
  Normalize $x_t = \hat{x}_t/\|\hat{x}_t\|$.
  $t \leftarrow t + 1$.
until *Convergence*;

Remark 2 *Similar to the behavior of traditional power method, if $A \in \mathbb{S}^p_+$, then TPower tries to find the (sparse) eigenvector of A corresponding to the largest eigenvalue. Otherwise, it may find the (sparse) eigenvector with the smallest eigenvalue if $-\lambda_p(A) > \lambda_1(A)$. However, this situation is easily detectable because it can only happen when $\lambda_p(A) < 0$. In such case, we may restart TPower with A replaced by an appropriately shifted version $A + \tilde{\lambda} I_{p\times p}$.*

2.2 Convergence

We now show that when A is positive semidefinite, TPower converges. This claim is a direct consequence of the following proposition.

Proposition 3 *If all $2k \times 2k$ principal submatrix A_{2k} of A are positive semidefinite, then the sequence $\{Q(x_t)\}_{t\geq 1}$ is monotonically increasing, where x_t is obtained from the TPower algorithm.*

Proof Observe that the iterate x_t in TPower solves the following constrained linear optimization problem:

$$x_t = \underset{\|x\|=1, \|x\|_0 \leq k}{\arg\max} L(x; x_{t-1}), \qquad L(x; x_{t-1}) := \langle 2Ax_{t-1}, x - x_{t-1}\rangle.$$

Clearly, $Q(x) - Q(x_{t-1}) = L(x; x_{t-1}) + (x - x_{t-1})^\top A(x - x_{t-1})$. Since $\|x_t - x_{t-1}\|_0 \leq 2k$ and each $2k \times 2k$ principal submatrix of A is positive semidefinite, we have $(x_t - x_{t-1})^\top A(x_t - x_{t-1}) \geq 0$. It follows that $Q(x_t) - Q(x_{t-1}) \geq L(x_t; x_{t-1})$. By the definition of x_t as the maximizer of $L(x; x_{t-1})$ over x (subject to $\|x\| = 1$ and $\|x\|_0 \leq k$), we have $L(x_t; x_{t-1}) \geq L(x_{t-1}; x_{t-1}) = 0$. Therefore $Q(x_t) - Q(x_{t-1}) \geq 0$, which proves the desired result. ■

3. Sparse Recovery Analysis

We consider the general noisy matrix model (2), and are especially interested in the high dimensional situation where the dimension p of A is large. We assume that the noise matrix E is a dense $p \times p$ matrix such that its sparse submatrices have small spectral norm $\rho(E, s)$ (see (3)) for s in the same order of k. We refer to this quantity as *restricted perturbation error*. However, the spectral

norm of the full matrix perturbation error $\rho(E)$ can be large. For example, if the original covariance is corrupted by an additive standard Gaussian iid noise vector, then $\rho(E,s) = O(\sqrt{s\log p/n})$, which grows linearly in $\sqrt{s}$, instead of $\rho(E) = O(\sqrt{p/n})$, which grows linearly in $\sqrt{p}$. The main advantage of the sparse eigenvalue formulation (1) over the standard eigenvalue formulation is that the estimation error of its optimal solution depends on $\rho(E,s)$ with respectively a small $s = O(k)$ rather than $\rho(E)$. This linear dependency on sparsity k instead of the original dimension p is analogous to similar results for sparse regression (or compressive sensing). In fact, the restricted perturbation error considered here is analogous to the idea of restricted isometry property (RIP) considered by Candes and Tao (2005).

The purpose of the section is to show that if matrix $\bar{A}$ has a unique sparse (or approximately sparse) dominant eigenvector, then under suitable conditions, TPower can (approximately) recover this eigenvector from the noisy observation A.

Assumption 1 *Assume that the largest eigenvalue of $\bar{A} \in \mathbb{S}^p$ is $\lambda = \lambda_{\max}(\bar{A}) > 0$ that is non-degenerate, with a gap $\Delta\lambda = \lambda - \max_{j>1}|\lambda_j(\bar{A})|$ between the largest and the remaining eigenvalues. Moreover, assume that the eigenvector $\bar{x}$ corresponding to the dominant eigenvalue λ is sparse with cardinality $\bar{k} = \|\bar{x}\|_0$.*

We want to show that under Assumption 1, if the spectral norm $\rho(E,s)$ of the error matrix is small for an appropriately chosen $s > \bar{k}$, then it is possible to approximately recover $\bar{x}$. Note that in the extreme case of $s = p$, this result follows directly from the standard eigenvalue perturbation analysis (which does not require Assumption 1).

We now state our main result as below, which shows that under appropriate conditions, the TPower method can recover the sparse eigenvector. The final error bound is a direct generalization of standard matrix perturbation result that depends on the full matrix perturbation error $\rho(E)$. Here this quantity is replaced by the restricted perturbation error $\rho(E,s)$.

Theorem 4 *We assume that Assumption 1 holds. Let $s = 2k + \bar{k}$ with $k \geq \bar{k}$. Assume that $\rho(E,s) \leq \Delta\lambda/2$. Define*

$$\gamma(s) := \frac{\lambda - \Delta\lambda + \rho(E,s)}{\lambda - \rho(E,s)} < 1, \qquad \delta(s) := \frac{\sqrt{2}\rho(E,s)}{\sqrt{\rho(E,s)^2 + (\Delta\lambda - 2\rho(E,s))^2}}.$$

If $|x_0^\top \bar{x}| \geq \theta + \delta(s)$ for some $\|x_0\|_0 \leq k$, $\|x_0\| = 1$, and $\theta \in (0,1)$ such that

$$\mu = \sqrt{(1 + 2((\bar{k}/k)^{1/2} + \bar{k}/k))(1 - 0.5\theta(1+\theta)(1-\gamma(s)^2))} < 1, \tag{4}$$

then we either have

$$\sqrt{1 - |x_0^\top \bar{x}|} < \sqrt{10}\delta(s)/(1-\mu), \tag{5}$$

or for all $t \geq 0$

$$\sqrt{1 - |x_t^\top \bar{x}|} \leq \mu^t \sqrt{1 - |x_0^\top \bar{x}|} + \sqrt{10}\delta(s)/(1-\mu). \tag{6}$$

Remark 5 *We only state our result with a relatively simple but easy to understand quantity $\rho(E,s)$, which we refer to as restricted perturbation error. It is analogous to the RIP concept (Candes and Tao, 2005), and is also directly comparable to the traditional full matrix perturbation error $\rho(E)$. While it is possible to obtain sharper results with additional quantities, we intentionally keep the theorem simple so that its consequence is relatively easy to interpret.*

Remark 6 *Although we state the result by assuming that the dominant eigenvector $\bar{x}$ is sparse, the theorem can also be adapted to certain situations that $\bar{x}$ is only approximately sparse. In such case, we simply let $\bar{x}'$ be a $\bar{k}$ sparse approximation of $\bar{x}$. If $\bar{x}'-\bar{x}$ is sufficiently small, then $\bar{x}'$ is the dominant eigenvector of a symmetric matrix $\bar{A}'$ that is close to $\bar{A}$; hence the theorem can be applied with the decomposition $A=\bar{A}'+E'$ where $E'=E+A-\bar{A}'$.*

Note that we did not make any attempt to optimize the constants in Theorem 4, which are relatively large. Therefore in the discussion, we shall ignore the constants, and focus on the main message Theorem 4 conveys. If $\rho(E,s)$ is smaller than the eigen-gap $\Delta\lambda/2>0$, then $\gamma(s)<1$ and $\delta(s)=O(\rho(E,s))$. It is easy to check that for any $k\geq\bar{k}$, if $\gamma(s)$ is sufficiently small then the requirement (4) can be satisfied for a sufficiently small θ of the order $(\bar{k}/k)^{1/2}$. It follows that under appropriate conditions, as long as we can find an initial x_0 such that

$$|x_0^\top\bar{x}|\geq c(\rho(E,s)+(\bar{k}/k)^{1/2})$$

for some constant c, then $1-|x_t^\top\bar{x}|$ converges geometrically until

$$\|x_t-\bar{x}\|=O(\rho(E,s)).$$

This result is similar to the standard eigenvector perturbation result stated in Lemma 10 of Appendix A, except that we replace the spectral error $\rho(E)$ of the full matrix by $\rho(E,s)$ that can be significantly smaller when $s\ll p$. To our knowledge, this is the first sparse recovery result for the sparse eigenvalue problem in a relatively general setting. This theorem can be considered as a strong theoretical justification of the proposed TPower algorithm that distinguishes it from earlier algorithms without theoretical guarantees. Specifically, the replacement of the full matrix perturbation error $\rho(E)$ with $\rho(E,s)$ gives the theoretical insights on why TPower works well in practice.

To illustrate our result, we briefly describe a consequence of the theorem under the single spike covariance model of Johnstone (2001) which was investigated by Amini and Wainwright (2009). We assume that the observations are p dimensional vectors

$$x_i=\bar{x}+\varepsilon,$$

for $i=1,\ldots,n$, where $\varepsilon\sim N(0,I_{p\times p})$. For simplicity, we assume that $\|\bar{x}\|=1$. The true covariance is

$$\bar{A}=\bar{x}\bar{x}^\top+I_{p\times p},$$

and A is the empirical covariance

$$A=\frac{1}{n}\sum_{i=1}^n x_i x_i^\top.$$

Let $E=A-\bar{A}$, then random matrix theory implies that with large probability,

$$\rho(E,s)=O(\sqrt{s\ln p/n}).$$

Now assume that $\max_j|\bar{x}_j|$ is sufficiently large. In this case, we can run TPower with a starting point $x_0=e_j$ for some vector e_j (where e_j is the vector of zeros except the j-th entry being one) so that $|e_j^\top\bar{x}|=|\bar{x}_j|$ is sufficiently large, and the assumption for the initial vector $|x_0^\top\bar{x}|\geq c(\rho(E,s)+$

$(\bar{k}/k)^{1/2})$ is satisfied with $s = O(\bar{k})$. We may run TPower with an appropriate initial vector to obtain an approximate solution x_t of error

$$\|x_t - \bar{x}\| = O(\sqrt{\bar{k}\ln p/n}).$$

This bound is optimal (Cai et al., 2012). Note that our results are not directly comparable to those of Amini and Wainwright (2009), which studied support recovery. Nevertheless, it is worth noting that if $\max_j |\bar{x}_j|$ is sufficiently large, then our result becomes meaningful when $n = O(\bar{k}\ln p)$; however their result requires $n = O(\bar{k}^2 \ln p)$ to be meaningful, although this is for the pessimistic case of $\bar{x}$ having equal nonzero values of $1/\sqrt{\bar{k}}$. Based on a similar spike covariance model, Ma (2013) independently presented and analyzed an iterative thresholding method for recovering sparse orthogonal principal components, using ideas related to what we present in this paper.

Finally we note that if we cannot find an initial vector with large enough value $|x_0^\top \bar{x}|$, then it may be necessary to take a relatively large k so that the requirement $|x_0^\top \bar{x}| \geq c(\rho(E,s) + (\bar{k}/k)^{1/2})$ is satisfied. With such a k, $\rho(E,s)$ may be relatively large and hence the theorem indicates that x_t may not converge to $\bar{x}$ accurately. Nevertheless, as long as $|x_t^\top \bar{x}|$ converges to a value that is not too small (e.g., can be much larger than $|x_0^\top \bar{x}|$), we may reduce k and rerun the algorithm with a k-sparse truncation of x_t as initial vector together with the reduced k. In this two stage process, the vector found from the first stage (with large k) is truncated and normalized, and then used as the initial value of the second stage (with small k). Therefore we may also regard it as an initialization method for TPower. Specially, in the first stage we may run TPower with $k = p$ from arbitrary initialization. In this stage, TPower reduces to the classic power method which outputs the dominant eigenvector x of A. Let $F = \text{supp}(x,k)$ be the indices of x with the largest k absolute values and $x_0 := \text{Truncate}(x,F)/\|\text{Truncate}(x,F)\|$. Let $\theta = x^\top \bar{x} - (\bar{k}/k)^{1/2}\sqrt{1-(x^\top \bar{x})^2} - \delta(s)$. It is implied by Lemma 12 in Appendix A that $x_0^\top \bar{x} \geq \theta + \delta(s)$. Obviously, if $\theta(1+\theta) \geq 8(\bar{k}/k)/((1+4\bar{k}/k)(1-\gamma(s)^2))$, then x_0 will be an initialization suitable for Theorem 1. From this initialization, we can obtain a better solution using the TPower method. In practice, one may use other methods to obtain an approximate x_0 to initialize TPower, not necessarily restricted to running TPower with larger k.

4. Experiments

In this section, we first show numerical results (in §4.1) that confirm the relevance of our theoretical predictions. We then illustrate the effectiveness of TPower method when applied to sparse principal component analysis (sparse PCA) (in §4.2) and the densest k-subgraph (DkS) finding problem (in §4.3). The Matlab code for reproducing the experimental results reported in this section is available from `https://sites.google.com/site/xtyuan1980/publications`.

4.1 Simulation Study

In this experiment, we illustrate the performance of TPower using simulated data. Theorem 4 implies that under appropriate conditions, the estimation error $\sqrt{1-x_t^\top \bar{x}}$ is proportional to $\delta(s)$. By definition, $\delta(s)$ is an increasing function with respect to perturbation error $\rho(E,s)$ and a decreasing function with respect to the gap $\Delta\lambda$ between the largest eigenvalue and the remaining eigenvalues. We will verify the results of Theorem 4 by applying TPower to the following single spike model

with true covariance

$$\bar{A} = \beta \bar{x}\bar{x}^\top + I_{p\times p}$$

and empirical covariance

$$A = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top,$$

where $x_i \sim \mathcal{N}(0,\bar{A})$. For the true covariance matrix $\bar{A}$, its dominant eigenvector is $\bar{x}$ with eigenvalue $\beta+1$, and its eigenvalue gap is $\Delta\lambda = \beta$. For this model, with large probability we have $\rho(E,s) = O(\sqrt{s\ln p/n})$. Therefore, for fixed dimensionality p, the error bound is relevant to the triplet $\{n,\beta,k\}$. In this study, we consider a setup with $p = 1000$, and $\bar{x}$ is a $\bar{k}$-sparse uniform random vector with $\bar{k} = 20$ and $\|\bar{x}\| = 1$. We are interested in the following two cases:

1. Cardinality k is tuned and fixed: we will study how the estimation error is affected by sample size n and eigen-gap β.

2. Cardinality k is varying: for fixed sample size n and eigen-gap β, we will study how the estimation error is affected by cardinality k in the algorithm.

4.1.1 ON INITIALIZATION

Theorem 4 suggests that TPower can benefit from a good initial vector x_0. We initialize x_0 by using the warm-start strategy suggested at the end of §3. In our implementation, this strategy is specialized as follows: we sequentially run TPower with cardinality $\{8k,4k,2k,k\}$, using the (truncated) output from the previous running as the initial vector for the next running. This initialization strategy works satisfactory in our numerical experiments.

4.1.2 TEST I: CARDINALITY k IS TUNED AND FIXED

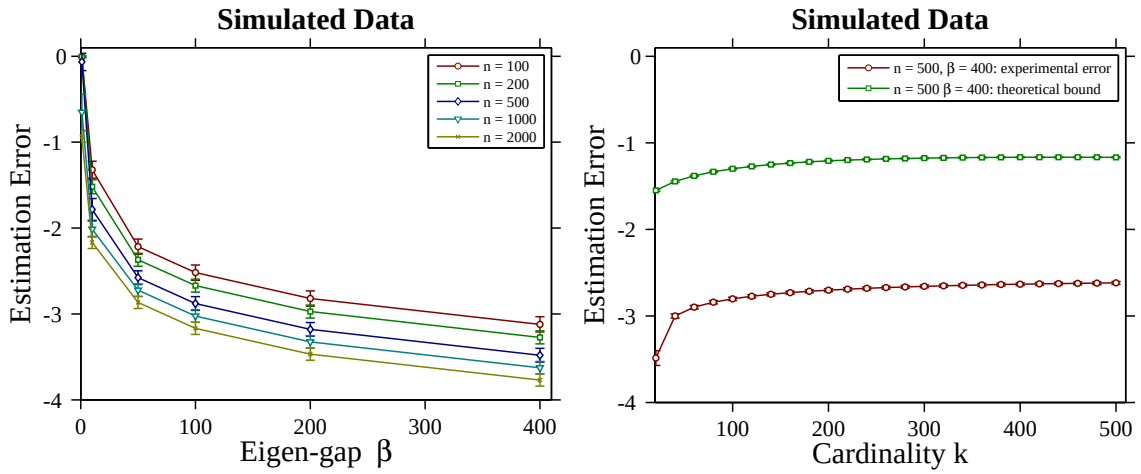
In this case, we test with $n \in \{100,200,500,1000,2000\}$ and $\beta \in \{1,10,50,100,200,400\}$. For each pair $\{n,\beta\}$, we generate 100 empirical covariance matrices and employ the TPower to compute a k-sparse eigenvector $\hat{x}$. For each empirical covariance matrix A, we also generate an independent empirical covariance matrix A_{val} to select k from the candidate set $\mathcal{K} = \{5,10,15,...,50\}$ by maximizing the following criterion:

$$\hat{k} = \arg\max_{k\in\mathcal{K}} \hat{x}(k)^\top A_{val}\hat{x}(k),$$

where $\hat{x}(k)$ is the output of TPower for A under cardinality k. For different pairs (n,β), the tuned values of k could be different. For example, for $(n,\beta) = (100,1)$, $k = 10$ will be selected; while for $(n,\beta) = (100,10)$, $k = 20$ will be selected. Note that Theorem 4 does not require an accurate estimation of k. Figure 1(a) shows the estimation error curves as functions of β under various n. It can be observed that for any fixed n, the estimation error decreases as eigen-gap β increases; and for any fixed β, the estimation error decreases as sample size n increases. This is consistent with the prediction of Theorem 4.

4.1.3 TEST II: CARDINALITY k IS VARYING

In this case, we fix sample size $n = 500$ and eigen-gap $\beta = 400$, and test the values of $k \in \{20...,500\}$ that are at least as large as the true sparsity $\bar{k} = 20$. We generate 100 empirical covariance matrices and employ the TPower to compute a k-sparse eigenvector. Figure 1(b) shows the estimation error



(a) Estimation error *vs.* eigen-gap β (under various sample size n).

(b) Estimation error bound *vs.* cardinality k (with $n = 500$ and $\beta = 400$). Both theoretical and empirical curves are plotted.

Figure 1: Estimation error curves on the simulated data. For better viewing, please see the original pdf file.

curves as functions of k. It can be observed that the estimation error becomes larger as k increases. This is consistent with the prediction of Theorem 4. For a fixed k, provided that the conditions are satisfied in Theorem 4, we can also calculate the theoretical estimation error bound $\sqrt{10}\delta(s)/(1-\mu)$. The curve of the theoretical bound is plotted in the same figure. As predicted by Theorem 4, the theoretical bound curve dominates the empirical error curve. Similar observations are also made for other fixed pairs $\{n, \beta\}$.

4.2 Sparse PCA

Principal component analysis (PCA) is a well established tool for dimensionality reduction and has a wide range of applications in science and engineering where high dimensional data sets are encountered. Sparse principal component analysis (sparse PCA) is an extension of PCA that aims at finding sparse vectors (loading vectors) capturing the maximum amount of variance in the data. In recent years, various researchers have proposed various approaches to directly address the conflicting goals of explaining variance and achieving sparsity in sparse PCA. For instance, greedy search and branch-and-bound methods were investigated by Moghaddam et al. (2006) to solve small instances of sparse PCA exactly and to obtain approximate solutions for larger scale problems. d'Aspremont et al. (2008) proposed the use of greedy forward selection with a certificate of optimality. Another popular technique for sparse PCA is regularized sparse learning. Zou et al. (2006) formulated sparse PCA as a regression-type optimization problem and imposed the Lasso penalty (Tibshirani, 1996) on the regression coefficients. The DSPCA algorithm of d'Aspremont et al. (2007) is an ℓ_1-norm based semidefinite relaxation for sparse PCA. Shen and Huang (2008) resorted to the singular value decomposition (SVD) to compute low-rank matrix approximations of the data matrix under various sparsity-inducing penalties. Mairal et al. (2010) proposed an online

learning method for matrix decomposition with sparsity regularization. More recently, Journée et al. (2010) studied a generalized power method to solve sparse PCA with a certain dual reformulation of the problem. Similar power-truncation-type methods were also considered by Witten et al. (2009) and Ma (2013).

Given a sample covariance matrix, $\Sigma \in \mathbb{S}_+^p$ (or equivalently a centered data matrix $D \in \mathbb{R}^{n\times p}$ with n rows of p-dimensional observations vectors such that $\Sigma = D^\top D$) and the target cardinality k, following the literature (Moghaddam et al., 2006; d'Aspremont et al., 2007, 2008), we formulate sparse PCA as:

$$\hat{x} = \arg\max_{x\in\mathbb{R}^p} x^\top \Sigma x, \qquad \text{subject to } \|x\| = 1, \|x\|_0 \le k. \tag{7}$$

The TPower method proposed in this paper can be directly applied to solve the above problem. One advantage of TPower for Sparse PCA is that it directly addresses the constraint on cardinality k. To find the top m rather than the top one sparse loading vectors, a common approach in the literature (d'Aspremont et al., 2007; Moghaddam et al., 2006; Mackey, 2008) is to use the *iterative deflation* method for PCA: subsequent sparse loading vectors can be obtained by recursively removing the contribution of the previously found loading vectors from the covariance matrix. Here we employ a projection deflation scheme proposed by Mackey (2008), which deflates a vector $\hat{x}$ using the formula:

$$\Sigma' = (I_{p\times p} - \hat{x}\hat{x}^\top)\Sigma(I_{p\times p} - \hat{x}\hat{x}^\top).$$

Obviously, Σ' remains positive semidefinite. Moreover, Σ' is rendered left and right orthogonal to $\hat{x}$.

4.2.1 CONNECTION WITH EXISTING SPARSE PCA METHODS

In the setup of sparse PCA, TPower is closely related to GPower (Journée et al., 2010) and sPCA-rSVD (Shen and Huang, 2008) which share the same spirit of thresholding iteration to make the loading vectors sparse. Indeed, GPower and sPCA-rSVD are identical except for the initialization and post-processing phases (see, e.g., Journée et al., 2010). TPower is most closely related to the GPower$_{\ell_0}$ (Journée et al., 2010, Algorithm 3) in the sense that both are characterized by rank-1 approximation and alternate optimization with hard-thresholding. Indeed, given a data matrix $D \in \mathbb{R}^{n\times p}$, GPower$_{\ell_0}$ solves the following ℓ_0-norm regularized rank-1 approximation problem:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2 + \gamma\|x\|_0, \qquad \text{subject to } \|z\| = 1.$$

GPower$_{\ell_0}$ is essentially a coordinate descent procedure which iterates between updating x and z. Given x_{t-1}, the update of z_t is $z_t = Dx_{t-1}/\|Dx_{t-1}\|$. Given z_t, the update of x_t is a hard-thresholding operation which selects those entries in $D^\top z_t = D^\top Dx_{t-1}/\|Dx_{t-1}\|$ with squared values greater than γ and then normalize the vector after truncation. From the viewpoint of rank-1 approximation, it can be shown that TPower optimizes the following cardinality constrained problem:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2, \qquad \text{subject to } \|z\| = 1,\ \|x\| = 1,\ \|x\|_0 \le k.$$

Indeed, based on the fact that $z = Dx/\|Dx\|$ is optimal at any x, the above problem is identical to the formulation (7). To update x_t, TPower selects the top k entries of $D^\top Dx_{t-1}$ and then normalize the truncated vector. Therefore, we can see that TPower and GPower$_{\ell_0}$ differs in the thresholding manner: the former selects the top k entries in $D^\top Dx_{t-1}$ while the latter preserves those entries in

$D^\top Dx_{t-1}$ with squared values greater than $\gamma\|Dx_{t-1}\|^2$. Another rank-1 approximation formulation was considered by Witten et al. (2009) with ℓ_1-norm ball constraint:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2, \qquad \text{subject to } \|z\| = 1,\ \|x\| = 1,\ \|x\|_1 \le c.$$

Its minimization procedure, called Projected Matrix Decomposition (PMD), alternates between the update of x and the update of z; where the update of x is a soft-thresholding operation.

Our method is also related to the Iterative Thresholding Sparse PCA (ITSPCA) method (Ma, 2013) which concentrates on recovering a sparse subspace of dimension m under the spike model. In particular, when $m = 1$, ITSPCA reduces to a power method with thresholding. However, TPower differs from ITSPCA in the following two aspects. First, the truncation strategy is different: we truncate the vector by preserving the top k largest absolute entries and setting the remaining entries to zeros, while ITSPCA truncates the vector by setting entries below a fixed threshold to zeros. Second, the analysis is different: TPower is analyzed under the matrix perturbation theory and thus is deterministic, while the analysis of ITSPCA focused on the convergence rate under the stochastic multiple spike model.

TPower is essentially a greedy selection method for solving problem (1). In this viewpoint, it is related to PathSPCA (d'Aspremont et al., 2008) which is a forward greedy selection procedure. PathSPCA starts from the empty set and at each iteration it selects the most relevant variable and adds it to the current variable set; it then re-estimates the leading eigenvector on the augmented variable set. Both TPower and PathSPCA output sparse solutions with exact cardinality k.

4.2.2 RESULTS ON TOY DATA SET

To illustrate the sparse recovering performance of TPower, we apply the algorithm to a synthetic data set drawn from a sparse PCA model. We follow the same procedure proposed by Shen and Huang (2008) to generate random data with a covariance matrix having sparse eigenvectors. To this end, a covariance matrix is first synthesized through the eigenvalue decomposition $\Sigma = VDV^\top$, where the first m columns of $V \in \mathbb{R}^{p\times p}$ are pre-specified sparse orthogonal unit vectors. A data matrix $X \in \mathbb{R}^{n\times p}$ is then generated by drawing n samples from a zero-mean normal distribution with covariance matrix Σ, that is $X \sim \mathcal{N}(0,\Sigma)$. The empirical covariance $\hat{\Sigma}$ matrix is then estimated from data X as the input for TPower.

Consider a setup with $p = 500$, $n = 50$, and the first $m = 2$ dominant eigenvectors of Σ are sparse. Here the first two dominant eigenvectors are specified as follows:

$$[v_1]_i = \begin{cases} \frac{1}{\sqrt{10}}, & i = 1,...,10 \\ 0, & \text{otherwise} \end{cases}, \quad [v_2]_i = \begin{cases} \frac{1}{\sqrt{10}}, & i = 11,...,20 \\ 0, & \text{otherwise} \end{cases}.$$

The remaining eigenvectors v_j for $j \ge 3$ are chosen arbitrarily, and the eigenvalues are fixed at the following values:

$$\begin{cases} \lambda_1 = 400, \\ \lambda_2 = 300, \\ \lambda_j = 1, \quad j = 3,...,500. \end{cases}$$

We generate 500 data matrices and employ the TPower method to compute two unit-norm sparse loading vectors $u_1, u_2 \in \mathbb{R}^{500}$, which are hopefully close to v_1 and v_2. Our method is compared

on this data set with a greedy algorithm PathPCA (d'Aspremont et al., 2008), two power-iteration-type methods GPower (Journée et al., 2010) and PMD (Witten et al., 2009), two sparse regression based methods SPCA (Zou et al., 2006) and online SPCA (oSPCA) (Mairal et al., 2010), and the standard PCA. For GPower, we test its two block versions GPower$_{\ell_1,m}$ and GPower$_{\ell_0,m}$ with ℓ_1-norm and ℓ_0-norm penalties, respectively. Here we do not directly compare to two representative sparse PCA algorithms sPCA-rSVD (Shen and Huang, 2008) and DSPCA (d'Aspremont et al., 2007) because the former is shown to be identical to GPower up to initialization and post-processing phases (Journée et al., 2010), while the latter is suggested by the authors as a secondary choice after PathSPCA. All tested algorithms were implemented in Matlab 7.12 running on a desktop. We use the two-stage warm-start strategy for initialization. Similar to the empirical study in the previous section, we tune the cardinality parameter k on independently generated validation matrices.

In this experiment, we regard the true model to be successfully recovered when both quantities $|v_1^\top u_1|$ and $|v_2^\top u_2|$ are greater than 0.99. Table 1 lists the recovering results by the considered methods. It can be observed that TPower, PathPCA, GPower, PMD and oSPCA all successfully recover the ground truth sparse PC vectors with high rate of success. SPCA frequently fails to recover the spares loadings on this data set. The potential reason is that SPCA is initialized with the ordinary principal components which in many random data matrices are far away from the truth sparse solution. Traditional PCA always fails to recover the sparse PC loadings on this data set. The success of TPower and the failure of traditional PCA can be well explained by our sparse recovery result in Theorem 4 (for TPower) in comparison to the traditional eigenvector perturbation theory in Lemma 10 (for traditional PCA), which we have already discussed in §3. However, the success of other methods suggests that it might be possible to prove sparse recovery results similar to Theorem 4 for some of these alternative algorithms. The running time of these algorithms on this data is listed in the last column of Table 1. It can be seen that TPower is among the top efficient solvers.

Algorithms	Parameter	$\lvert v_1^\top u_1\rvert$	$\lvert v_2^\top u_2\rvert$	Prob. of succ.	CPU (in ms)
TPower	$k = 10$	.9998 (.0001)	.9997 (.0002)	1	6.14 (0.76)
PathSPCA	$k = 10$	.9998 (.0001)	.9997 (.0002)	1	77.42 (2.95)
GPower$_{\ell_1,m}$	$\gamma = 0.8$	.9997 (.0016)	.9996 (.0022)	0.99	6.22 (0.30)
GPower$_{\ell_0,m}$	$\gamma = 0.8$	.9997 (.0016)	.9991 (.0117)	0.99	6.07 (0.30)
PMD	$c = 3.0$	.9998 (.0001)	.9997 (.0002)	1	11.97 (0.48)
oSPCA	$\lambda = 3$	.9929 (.0434)	.9923 (.0483)	0.97	24.74 (1.20)
SPCA	$\lambda_1 = 10^{-3}$	.9274 (.0809)	.9250 (.0810)	0.25	799.99 (50.62)
PCA	–	.9146 (.0801)	.9086 (.0790)	0	3.87 (1.59)

Table 1: The quantitative results on a synthetic data set. The values $|v_1^\top u_1|$, $|v_2^\top u_2|$, CPU time (in ms) are in format of mean (std) over 500 running.

4.2.3 RESULTS ON PITPROPS DATA

The PitProps data set (Jeffers, 1967), which consists of 180 observations with 13 measured variables, has been a standard benchmark to evaluate algorithms for sparse PCA (see, e.g., Zou et al., 2006; Shen and Huang, 2008; Journée et al., 2010). Following these previous studies, we also consider to compute the first six sparse PCs of the data. In Table 2, we list the total cardinality and

the proportion of adjusted variance (Zou et al., 2006) explained by six components computed with TPower, PathSPCA (d'Aspremont et al., 2008), GPower, PMD, oSPCA and SPCA. From these results we can see that on this relatively simple data set, TPower, PathSPCA and GPower perform quite similarly and are slightly better than PMD, oSPCA and SPCA.

Table 3 lists the six extracted PCs by TPower with cardinality setting 6-2-1-2-1-1. We can see that the important variables associated with the six PCs are exclusive except for the variable "ringb" which is simultaneously selected by PC1 and PC4. The variable "diaknot" is excluded from all the six PCs. The same loadings are also extracted by both PathSPCA and GPower under the parameters listed in Table 2.

Method	Parameters	Total cardinality	Prop. of explained variance
TPower	cardinalities: 7-2-4-3-5-4	25	0.8887
TPower	cardinalities: 6-2-1-2-1-1	13	0.7978
PathSPCA	cardinalities: 7-2-4-3-5-4	25	0.8834
PathSPCA	cardinalities: 6-2-1-2-1-1	13	0.7978
GPower$_{\ell_1,m}$	$\gamma = 0.22$	26	0.8438
GPower$_{\ell_1,m}$	$\gamma = 0.50$	13	0.7978
PMD	$c = 1.50$	25	0.8244
PMD	$c = 1.10$	13	0.7309
oSPCA	$\lambda = 0.2$	27	0.8351
oSPCA	$\lambda = 0.4$	12	0.6625
SPCA	see Zou et al. (2006)	18	0.7580

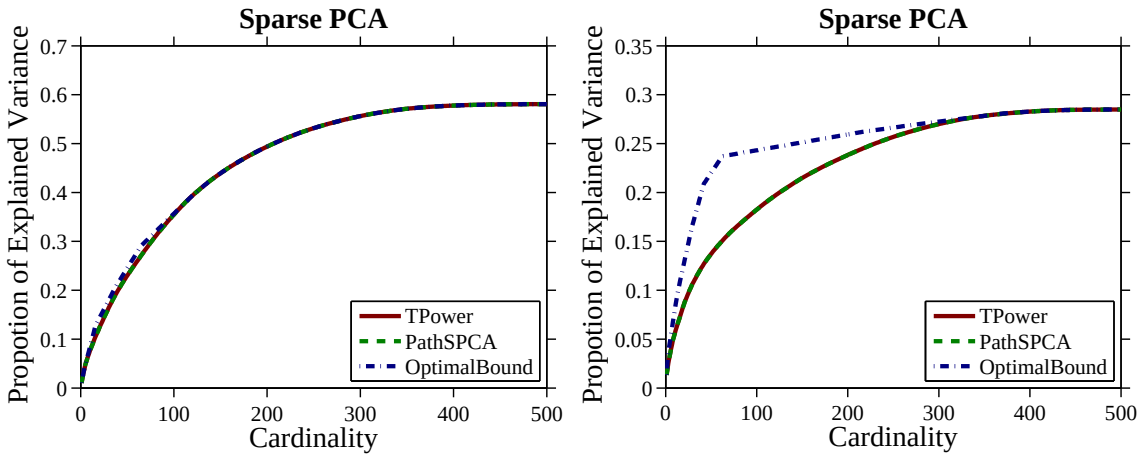
Table 2: The quantitative results on the PitProps data set. The result of SPCA is taken from Zou et al. (2006).

PCs	x_1 topd	x_2 length	x_3 moist	x_4 testsg	x_5 ovensg	x_6 ringt	x_7 ringb	x_8 bowm	x_9 bowd	x_{10} whorls	x_{11} clear	x_{12} knots	x_{13} diaknot
PC1	.4444	.4534	0	0	0	0	.3779	.3415	.4032	.4183	0	0	0
PC2	0	0	.7071	.7071	0	0	0	0	0	0	0	0	0
PC3	0	0	0	0	1.000	0	0	0	0	0	0	0	0
PC4	0	0	0	0	0	.8569	.5154	0	0	0	0	0	0
PC5	0	0	0	0	0	0	0		0	0	1.000	0	0
PC6	0	0	0	0	0	0	0	0	0	0	0	1.000	0

Table 3: The extracted six PCs by TPower on PitProps data set with cardinality setting 6-2-1-2-1-1. Note that in this setting, the extracted significant loadings are non-overlapping except for "ringb". And the variable "diaknot" is excluded from all the six PCs.

4.2.4 RESULTS ON BIOLOGICAL DATA

We have also evaluated the performance of TPower on two gene expression data sets, one is the Colon cancer data from Alon et al. (1999), the other is the Lymphoma data from Alizadeh et al. (2000). Following the experimental setup of d'Aspremont et al. (2008), we consider the 500 genes with the largest variances. We plot the variance versus cardinality tradeoff curves in Figure 2, to-



(a) Colon Cancer (b) Lymphoma

Figure 2: The variance versus cardinality tradeoff curves on two gene expression data sets. For better viewing, please see the original pdf file.

gether with the result from PathSPCA and the upper bounds of optimal values from d'Aspremont et al. (2008). Note that our method performs almost identical to the PathSPCA which is demonstrated to have optimal or very close to optimal solutions in many cardinalities. The computational time of the two methods on both data sets is comparable and is less than two seconds.

4.2.5 SUMMARY

To summarize this group of experiments on sparse PCA, the basic finding is that TPower performs quite competitively in terms of the tradeoff between explained variance and representation sparsity. The performance is comparable or superior to leading methods such as PathSPCA and GPower. It is observed that TPower, PathSPCA and GPower outperform PMD, oSPCA and SPCA on the benchmark data Pitprops. It is not surprising that TPower and GPower behave similarly because both are power-truncation-type method (see the previous §4.2.1). While strong theoretical guarantee can be established for the TPower method, it remains open to show that PathSPCA and GPower have a similar sparse recovery performance.

4.3 Densest k-Subgraph Finding

As another concrete application, we show that with proper modification, TPower can be applied to the densest k-subgraph finding problem. Given an undirected graph $G = (V,E)$, $|V| = n$, and integer $1 \leq k \leq n$, the densest k-subgraph (DkS) problem is to find a set of k vertices with maximum average degree in the subgraph induced by this set. In the weighted version of DkS we are also given nonnegative weights on the edges and the goal is to find a k-vertex induced subgraph of maximum average edge weight. Algorithms for finding DkS are useful tools for analyzing networks. In particular, they have been used to select features for ranking (Geng et al., 2007), to identify cores of communities (Kumar et al., 1999), and to combat link spam (Gibson et al., 2005).

It has been shown that the DkS problem is NP hard for bipartite graphs and chordal graphs (Corneil and Perl, 1984), and even for graphs of maximum degree three (Feige et al., 2001). A large body of algorithms have been proposed based on a variety of techniques including greedy algorithms (Feige et al., 2001; Asahiro et al., 2002; Ravi et al., 1994), linear programming (Billionnet and Roupin, 2004; Khuller and Saha, 2009), and semidefinite programming (Srivastav and Wolf, 1998; Ye and Zhang, 2003). For general k, the algorithm developed by Feige et al. (2001) achieves the best approximation ratio of $O(n^{\varepsilon})$ where $\varepsilon < 1/3$. Ravi et al. (1994) proposed 4-approximation algorithms for weighted DkS on complete graphs for which the weights satisfy the triangle inequality. Liazi et al. (2008) has presented a 3-approximation algorithm for DkS for chordal graphs. Recently, Jiang et al. (2010) proposed to reformulate DkS as a 1-mean clustering problem and developed a 2-approximation to the reformulated clustering problem. Moreover, based on this reformulation, Yang (2010) proposed a $1+\varepsilon$-approximation algorithm with certain exhaustive (and thus expensive) initialization procedure. In general, however, Khot (2006) showed that DkS has no polynomial time approximation scheme (PTAS), assuming that there are no sub-exponential time algorithms for problems in NP.

Mathematically, DkS can be restated as the following binary quadratic programming problem:

$$\max_{\pi\in\mathbb{R}^n} \pi^\top W\pi, \qquad \text{subject to } \pi\in\{1,0\}^n, \|\pi\|_0 = k, \tag{8}$$

where W is the (non-negative weighted) adjacency matrix of G. If G is an undirected graph, then W is symmetric. If G is directed, then W could be asymmetric. In this latter case, from the fact that $\pi^\top W\pi = \pi^\top \frac{W+W^\top}{2}\pi$, we may equivalently solve Problem (8) by replacing W with $\frac{W+W^\top}{2}$. Therefore, in the following discussion, we always assume that the affinity matrix W is symmetric (or G is undirected).

4.3.1 THE TPOWER-DKS ALGORITHM

We propose the TPower-DkS algorithm as an adaptation of TPower to the DkS problem. The process generates a sequence of intermediate vectors $\pi_0, \pi_1, ...$ from a starting vector π_0. At each step t the vector π_{t-1} is multiplied by the matrix W, then π_t is set to be the indicator vector of the top k entries in $W\pi_{t-1}$. The TPower-Dks is outlined in Algorithm 2. The convergence of this algorithm can be justified using the same arguments of bounding optimization as described in §2.2.

Algorithm 2: Truncated Power Method for DkS (TPower-DkS)

Input : $W \in \mathbb{S}^n_+$,, initial vector $\pi_0 \in \mathbb{R}^n$
Output : π_t
Parameters : cardinality $k \in \{1,...,n\}$

Let $t = 1$.
repeat
  Compute $\pi'_t = W\pi_{t-1}$.
  Identify $F_t = \text{supp}(\pi'_t, k)$ the index set of π'_t with top k values.
  Set π_t to be 1 on the index set F_t, and 0 otherwise.
  $t \leftarrow t+1$.
until *Convergence*;

Remark 7 *By relaxing the constraint $\pi \in \{0,1\}^n$ to $\|\pi\| = \sqrt{k}$, we may convert the densest k-subgraph problem* (8) *to the standard sparse eigenvalue problem* (1) *(up to a scaling) and then directly apply TPower (in Algorithm 1) for solution. Our numerical experience shows that such a relaxation strategy also works satisfactory in practice, although is slightly inferior to TPower-DkS (in Algorithm 2) which directly addresses the original problem.*

Remark 8 *As aforementioned that the DkS problem is generally NP-hard. The quality of its approximate solution can be measured by the approximation ratio defined as the output objective to the optimal objective. Recently, Jiang et al. (2010) proposed to reformulate DkS as a 1-mean clustering problem and developed a 2-approximation to the reformulated clustering problem. Moreover, based on this reformulation, Yang (2010) proposed a $1+\varepsilon$-approximation algorithm with certain exhaustive (and thus expensive) initialization procedure. Provided that W is positive semidefinite with equal diagonal elements, trivial derivation shows that TPower-DkS is identical to the method of Jiang et al. (2010). Therefore, the approximation ratio results from Jiang et al. (2010); Yang (2010) can be shared by TPower-DkS in this restricted case.*

Note that in Algorithm 2 we require that W is positive semidefinite. The motivation of this requirement is to guarantee the convexity of the objective in problem (8), and thus the convergence of Algorithm 2 can be justified by the similar arguments in §2.2. In many real-world DkS problems, however, it is often the case that the affinity matrix W is not positive semidefinite. In this case, the objective is non-convex and thus the monotonicity of TPower-DkS does not hold. However, this complication can be circumvented by instead running the algorithm with the shifted quadratic function:

$$\max_{\pi \in \mathbb{R}^n} \pi^\top (W + \tilde{\lambda} I_{p\times p})\pi, \qquad \text{subject to } \pi \in \{0,1\}^n, \|\pi\|_0 = k.$$

where $\tilde{\lambda} > 0$ is large enough such that $\tilde{W} = W + \tilde{\lambda} I_{p\times p} \in \mathbb{S}_+^n$. On the domain of interest, this change only adds a constant term to the objective function. The TPower-DkS, however, produces a different sequence of iterates, and there is a clear tradeoff. If the second term dominates the first term (say by choosing a very large $\tilde{\lambda}$), the objective function becomes approximately a squared norm, and the algorithm tends to terminate in very few iterations. In the limiting case of $\tilde{\lambda} \to \infty$, the method will not move away from the initial iterate. To handle this issue, we propose to gradually increase $\tilde{\lambda}$ during the iterations and we do so only when the monotonicity is violated. To be precise, if at a time instance t, $\pi_t^\top W \pi_t < \pi_{t-1}^\top W \pi_{t-1}$, then we add $\tilde{\lambda} I_{p\times p}$ to W with a gradually increased $\tilde{\lambda}$ by repeating the current iteration with the updated matrix until $\pi_t^\top (W + \tilde{\lambda} I_{p\times p})\pi_t \geq \pi_{t-1}^\top (W + \lambda I_{p\times p})\pi_{t-1}$,[2] which implies $\pi_t^\top W \pi_t \geq \pi_{t-1}^\top W \pi_{t-1}$.

4.3.2 ON INITIALIZATION

Since TPower-DkS is a monotonically increasing procedure, it guarantees to improve the initial point π_0. Basically, any existing approximation DkS method, for example, greedy algorithms (Feige et al., 2001; Ravi et al., 1994), can be used to initialize TPower-DkS. In our numerical experiments, we observe that by simply setting π_0 as the indicator vector of the vertices with the top k (weighted) degrees, our method can achieve very competitive results on all the real-world data sets we have tested on.

2. Note that the inequality $\pi_t^\top (W + \tilde{\lambda} I_{p\times p})\pi_t \geq \pi_{t-1}^\top (W + \tilde{\lambda} I_{p\times p})\pi_{t-1}$ is deemed to be satisfied when $\tilde{\lambda}$ is large enough, for example, when $W + \tilde{\lambda} I_{p\times p} \in \mathbb{S}_+^n$.

4.3.3 RESULTS ON WEB GRAPHS

We have tested TPower on four page-level web graphs: cnr-2000, amazon-2008, ljournal-2008, hollywood-2009, from the WebGraph framework provided by the Laboratory for Web Algorithms.[3] We treated each directed arc as an undirected edge. Table 4 lists the statistics of the data sets used in the experiment.

Graph	Nodes ($\|V\|$)	Total Arcs ($\|E\|$)	Average Degree
cnr-2000	325,557	3,216,152	9.88
amazon-2008	735,323	5,158,388	7.02
ljournal-2008	5,363,260	79,023,142	14.73
hollywood-2009	1,139,905	113,891,327	99.91

Table 4: The statistics of the web graph data sets.

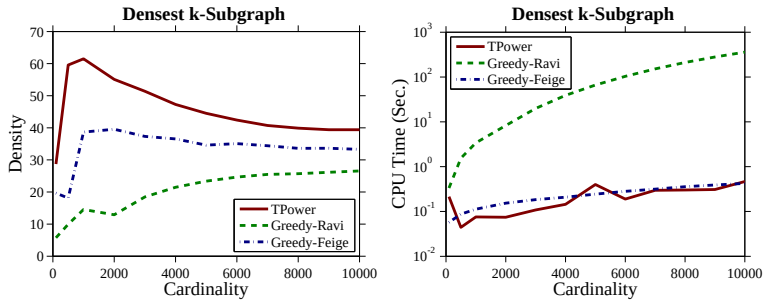
We compare our TPower-DkS method with two greedy methods for the DkS problem. One greedy method is proposed by Ravi et al. (1994) which is referred to as Greedy-Ravi in our experiments. The Greedy-Ravi algorithm works as follows: it starts from a heaviest edge and repeatedly adds a vertex to the current subgraph to maximize the weight of the resulting new subgraph; this process is repeated until k vertices are chosen. The other greedy method is developed by Feige et al. (2001, Procedure 2) which is referred as Greedy-Feige in our experiments. The procedure works as follows: let S denote the $k/2$ vertices with the highest degrees in G; let C denote the $k/2$ vertices in the remaining vertices with largest number of neighbors in S; return $S \cup C$.

Figure 3 shows the density value $\pi^\top W\pi/k$ and CPU time versus the cardinality k. From the density curves we can observe that on cnr-2000, ljournal-2008 and hollywood-2009, TPower-DkS consistently outputs denser subgraphs than the two greedy algorithms, while on amazon-2008, TPower-DkS and Greedy-Ravi are comparable and both are better than Greedy-Feige. For CPU running time, it can be seen from the right column of Figure 3 that Greedy-Feige is the fastest among the three methods while TPower-DkS is only slightly slower. This is due to the fact that TPower-DkS needs iterative matrix-vector products while Greedy-Feige only needs a few degree sorting operations. Although TPower-DkS is slightly slower than Greedy-Feige, it is still quite efficient. For example, on hollywood-2009 which has hundreds of millions of arcs, for each k, Greedy-Feige terminates within about 1 second while TPower terminates within about 10 seconds. The Greedy-Ravi method is however much slower than the other two on all the graphs when k is large.

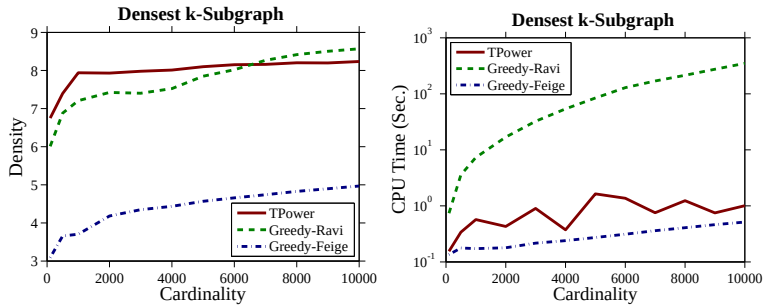
4.3.4 RESULTS ON AIR-TRAVEL ROUTINE

We have applied TPower-DkS to identify subsets of American and Canadian cities that are most easily connected to each other, in terms of estimated commercial airline travel time. The graph[4] is of size $|V| = 456$ and $|E| = 71,959$: the vertices are 456 busiest commercial airports in United States and Canada, while the weight w_{ij} of edge e_{ij} is set to the inverse of the mean time it takes to travel from city i to city j by airline, including estimated stopover delays. Due to the headwind

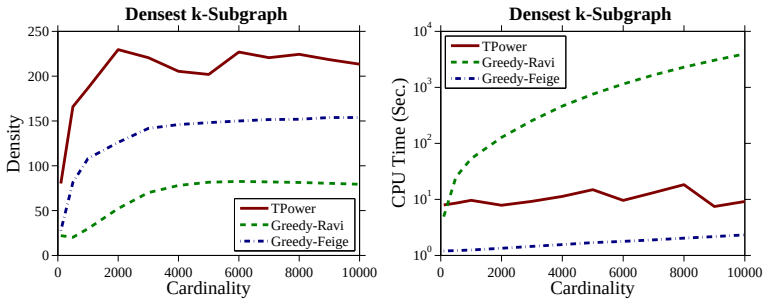
3. These four data sets are publicly available at `http://lae.dsi.unimi.it/datasets.php`.
4. The data is available at `www.psi.toronto.edu/affinitypropogation`.



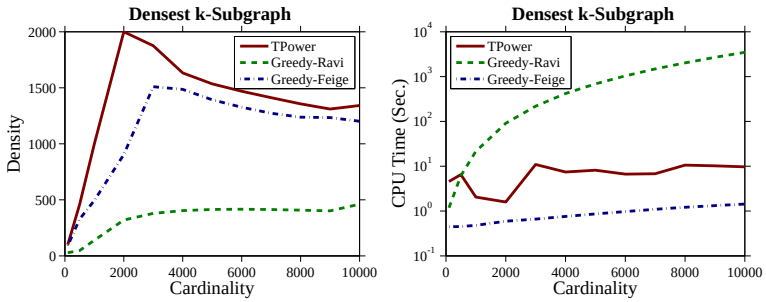
(a) cnr-2000



(b) amazon-2008



(c) ljournal-2008



(d) hollywood-2009

Figure 3: Identifying densest k-subgraph on four web graphs. Left: density curves as a function of cardinality. Right: CPU time (in second) curves as a function of cardinality. For better viewing, please see the original pdf file.

effect, the transit time can depend on the direction of travel; thus 36% of the weight are asymmetric. Figure 4(a) shows a map of air-travel routine.

As in the previous experiment, we compare TPower-DkS to Greedy-Ravi and Greedy-Feige on this data set. For all the three considered algorithms, the densities of k-subgraphs under different k values are shown in Figure 4(b), and the CPU running time curves are given in Figure 4(c). From the former figure we observe that TPower-DkS consistently outperforms the other two greedy algorithms in terms of the density of the extracted k-subgraphs. From the latter figure we can see that TPower-DkS is slightly slower than Greed-Feige but much faster than Greedy-Ravi. Figure 4(d)$\sim$4(f) illustrate the densest k-subgraph with $k = 30$ output by the three algorithms. In each of these three subgraph, the red dot indicates the representing city with the largest (weighted) degree. Both TPower-DkS and Greedy-Feige reveal 30 cities in east US. The former takes *Cleveland* as the representing city while the latter *Cincinnati*. Greedy-Ravi reveals 30 cities in west US and CA and takes *Vancouver* as the representing city. Visual inspection shows that the subgraph recovered by TPower-DkS is the densest among the three.

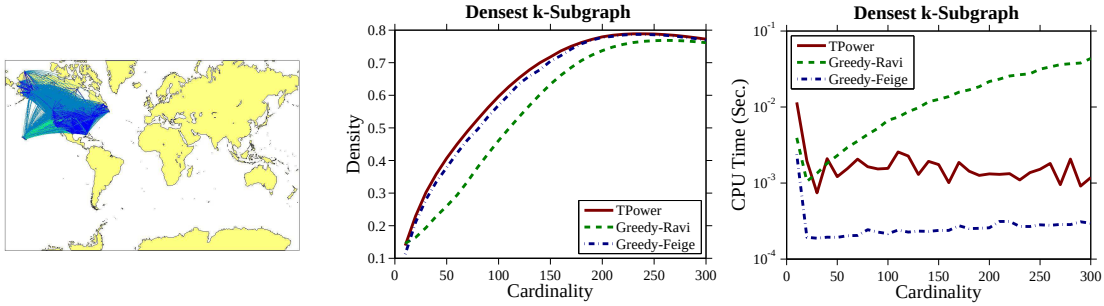
After discovering the densest k-subgraph, we can eliminate their nodes and edges from the graph and then apply the algorithms on the reduced graph to search for the next densest subgraph. This sequential procedure can be repeated to find multiple densest k-subgraphs. Figure 4(g)$\sim$4(i) illustrate sequentially estimated six densest 30-subgraphs by the three considered algorithms. Again, visual inspection shows that our method outputs more geographically compact subsets of cities than the other two. As a quantitative result, the total densities of the six subgraphs discovered by the three algorithms are: 1.14 (TPower-DkS), 0.90 (Greedy-Feige) and 0.99 (Greedy-Ravi), respectively.

5. Conclusion

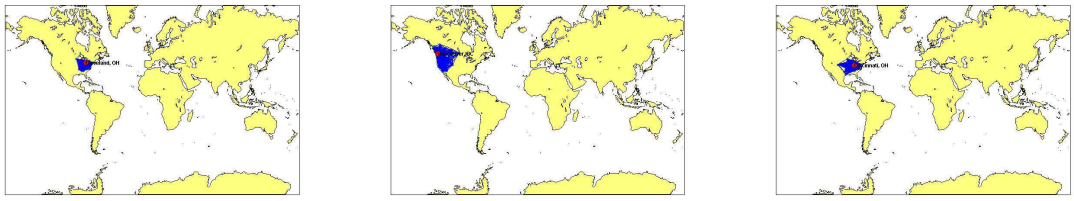
The sparse eigenvalue problem has been widely studied in machine learning with applications such as sparse PCA. TPower is a truncated power iteration method that approximately solves the non-convex sparse eigenvalue problem. Our analysis shows that when the underlying matrix has sparse eigenvectors, under proper conditions TPower can approximately recover the true sparse solution. The theoretical benefit of this method is that with appropriate initialization, the reconstruction quality depends on the restricted matrix perturbation error at size s that is comparable to the sparsity $\bar{k}$, instead of the full matrix dimension p. This explains why this method has good empirical performance. To our knowledge, this is one of the first theoretical results of this kind, although our empirical study suggests that it might be possible to prove related sparse recovery results for some other algorithms we have tested. We have applied TPower to two concrete applications: sparse PCA and the densest k-subgraph finding problem. Extensive experimental results on synthetic and real-world data sets validate the effectiveness and efficiency of the TPower algorithm. To summarize, simply combing power iteration with hard-thresholding truncation provides an accurate and scalable computational method for the sparse eigenvalue problem.

Acknowledgments

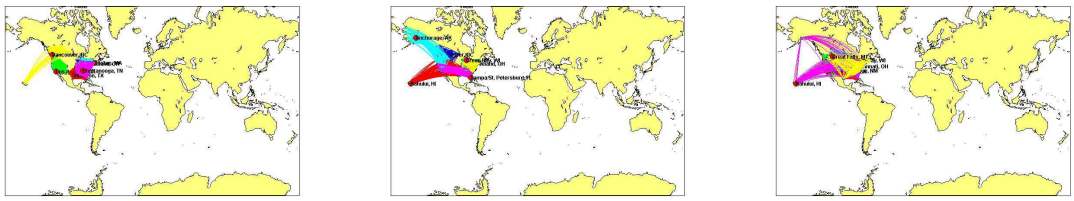
The work is supported by NSF grants DMS-1007527, IIS-1016061, and IIS-1250985.



(a) The air-travel route map (b) Density (c) CPU Time



(d) TPower-DkS (e) Greedy-Ravi (f) Greedy-Feige



(g) TPower-DkS (h) Greedy-Ravi (i) Greedy-Feige

Figure 4: Identifying densest k-subgraph of air-travel routing. Top row: Route map, and the density and CPU time evolving curves. Middle row: The densest 30-subgraph discovered by the three considered algorithms. Bottom row: Sequentially discovered six densest 30-subgraphs by the three considered algorithms. For better viewing, please see the original pdf file.

Appendix A. Proof Of Theorem 4

Our proof employs several technical tools including the perturbation theory of symmetric eigenvalue problem (Lemma 9 and Lemma 10), the convergence analysis of traditional power method (Lemma 11), and the error analysis of hard-thresholding operation (Lemma 12).

We state the following standard result from the perturbation theory of symmetric eigenvalue problem (see, e.g., Golub and Loan, 1996).

Lemma 9 *If B and $B+U$ are $p \times p$ symmetric matrices, then $\forall 1 \leq k \leq p$,*

$$\lambda_k(B) + \lambda_p(U) \leq \lambda_k(B+U) \leq \lambda_k(B) + \lambda_1(U),$$

where $\lambda_k(B)$ denotes the k-th largest eigenvalue of matrix B.

Lemma 10 *Consider set F such that $supp(\bar{x}) \subseteq F$ with $|F| = s$. If $\rho(E,s) \leq \Delta\lambda/2$, then the ratio of the second largest (in absolute value) to the largest eigenvalue of sub matrix A_F is no more than $\gamma(s)$. Moreover,*

$$\|\bar{x}^\top - x(F)\| \leq \delta(s) := \frac{\sqrt{2}\rho(E,s)}{\sqrt{\rho(E,s)^2 + (\Delta\lambda - 2\rho(E,s))^2}}.$$

Proof We may use Lemma 9 with $B = \bar{A}_F$ and $U = E_F$ to obtain

$$\lambda_1(A_F) \geq \lambda_1(\bar{A}_F) + \lambda_p(E_F) \geq \lambda_1(\bar{A}_F) - \rho(E_F) \geq \lambda - \rho(E,s)$$

and $\forall j \geq 2$,

$$|\lambda_j(A_F)| \leq |\lambda_j(\bar{A}_F)| + \rho(E_F) \leq \lambda - \Delta\lambda + \rho(E,s).$$

This implies the first statement of the lemma.

Now let $x(F)$, the largest eigenvector of A_F, be $\alpha\bar{x} + \beta x'$, where $\|\bar{x}\|_2 = \|x'\|_2 = 1$, $\bar{x}^\top x' = 0$ and $\alpha^2 + \beta^2 = 1$, with eigenvalue $\lambda' \geq \lambda - \rho(E,s)$. This implies that

$$\alpha A_F \bar{x} + \beta A_F x' = \lambda'(\alpha\bar{x} + \beta x'),$$

implying

$$\alpha x'^\top A_F \bar{x} + \beta x'^\top A_F x' = \lambda'\beta.$$

That is,

$$|\beta| = |\alpha| \frac{x'^\top A_F \bar{x}}{\lambda' - x'^\top A_F x'} \leq |\alpha| \frac{|x'^\top A_F \bar{x}|}{\lambda' - x'^\top A_F x'} = |\alpha| \frac{|x'^\top E_F \bar{x}|}{\lambda' - x'^\top A_F x'} \leq t|\alpha|,$$

where $t = \rho(E,s)/(\Delta\lambda - 2\rho(E,s))$. This implies that $\alpha^2(1+t^2) \geq \alpha^2 + \beta^2 = 1$, and thus $\alpha^2 \geq 1/(1+t^2)$. Without loss of generality, we may assume that $\alpha > 0$, because otherwise we can replace $\bar{x}$ with $-\bar{x}$. It follows that

$$\|x(F) - \bar{x}\|^2 = 2 - 2x(F)^\top \bar{x} = 2 - 2\alpha \leq 2\frac{\sqrt{1+t^2} - 1}{\sqrt{1+t^2}} \leq \frac{2t^2}{1+t^2}.$$

This implies the desired bound. ■

The following result measures the progress of untruncated power method.

Lemma 11 *Let y be the eigenvector with the largest (in absolute value) eigenvalue of a symmetric matrix A, and let $\gamma < 1$ be the ratio of the second largest to largest eigenvalue in absolute values. Given any x such that $\|x\| = 1$ and $y^\top x > 0$; let $x' = Ax/\|Ax\|$, then*

$$|y^\top x'| \geq |y^\top x|[1 + (1-\gamma^2)(1-(y^\top x)^2)/2].$$

Proof Without loss of generality, we may assume that $\lambda_1(A) = 1$ is the largest eigenvalue in absolute value, and $|\lambda_j(A)| \leq \gamma$ when $j > 1$. We can decompose x as $x = \alpha y + \beta y'$, where $y^\top y' = 0$, $\|y\| = \|y'\| = 1$, and $\alpha^2 + \beta^2 = 1$. Then $|\alpha| = |x^\top y|$. Let $z' = Ay'$, then $\|z'\| \leq \gamma$ and $y^\top z' = 0$. This means $Ax = \alpha y + \beta z'$, and

$$\begin{aligned}
|y^\top x'| =& \frac{|y^\top Ax|}{\|Ax\|} = \frac{|\alpha|}{\sqrt{\alpha^2 + \beta^2\|z'\|^2}} \geq \frac{|\alpha|}{\sqrt{\alpha^2 + \beta^2\gamma^2}} \\
=& \frac{|y^\top x|}{\sqrt{1-(1-\gamma^2)(1-(y^\top x)^2)}} \\
\geq& |y^\top x|\,[1 + (1-\gamma^2)(1-(y^\top x)^2)/2].
\end{aligned}$$

The last inequality is due to $1/\sqrt{1-z} \geq 1 + z/2$ for $z \in [0,1)$. This proves the desired bound. ■

The following lemma quantifies the error introduced by the truncation step in TPower.

Lemma 12 *Consider $\bar{x}$ with $supp(\bar{x}) = \bar{F}$ and $\bar{k} = |\bar{F}|$. Consider y and let $F = supp(y,k)$ be the indices of y with the largest k absolute values. If $\|\bar{x}\| = \|y\| = 1$, then*

$$|Truncate(y,F)^\top \bar{x}| \geq |y^\top \bar{x}| - (\bar{k}/k)^{1/2} \min\left[\sqrt{1-(y^\top \bar{x})^2}, (1 + (\bar{k}/k)^{1/2})\,(1-(y^\top \bar{x})^2)\right].$$

Proof Without loss of generality, we assume that $y^\top \bar{x} = \Delta > 0$. We can also assume that $\Delta > \sqrt{\bar{k}/(\bar{k}+k)}$ because otherwise the right hand side is smaller than zero, and thus the result holds trivially.

Let $F_1 = \bar{F} \setminus F$, and $F_2 = \bar{F} \cap F$, and $F_3 = F \setminus \bar{F}$. Now, let $\bar{\alpha} = \|\bar{x}_{F_1}\|$, $\bar{\beta} = \|\bar{x}_{F_2}\|$, $\alpha = \|y_{F_1}\|$, $\beta = \|y_{F_2}\|$, and $\gamma = \|y_{F_3}\|$. let $k_1 = |F_1|$, $k_2 = |F_2|$, and $k_3 = |F_3|$. It follows that $\alpha^2/k_1 \leq \gamma^2/k_3$. Therefore

$$\Delta^2 \leq [\bar{\alpha}\alpha + \bar{\beta}\beta]^2 \leq \alpha^2 + \beta^2 \leq 1 - \gamma^2 \leq 1 - (k_3/k_1)\alpha^2.$$

This implies that

$$\alpha^2 \leq (k_1/k_3)(1-\Delta^2) \leq (\bar{k}/k)(1-\Delta^2) < \Delta^2, \tag{9}$$

where the second inequality follows from $\bar{k} \leq k$ and the last inequality follows from the assumption $\Delta > \sqrt{\bar{k}/(\bar{k}+k)}$. Now by solving the following inequality for $\bar{\alpha}$

$$\alpha\bar{\alpha} + \sqrt{1-\alpha^2}\sqrt{1-\bar{\alpha}^2} \geq \alpha\bar{\alpha} + \beta\bar{\beta} \geq \Delta$$

under the condition $\Delta > \alpha \geq \alpha\bar{\alpha}$, we obtain that

$$\bar{\alpha} \leq \alpha\Delta + \sqrt{1-\alpha^2}\sqrt{1-\Delta^2} \leq \min\left[1, \alpha + \sqrt{1-\Delta^2}\right] \leq \min\left[1, (1 + (\bar{k}/k)^{1/2})\sqrt{1-\Delta^2}\right], \tag{10}$$

where the second inequality follows from the Cauchy-Schwartz inequality and $\Delta \leq 1$, $\sqrt{1-\alpha^2} \leq 1$, while the last inequality follows from (9). Finally,

$$\begin{aligned} |y^\top \bar{x}| - |\text{Truncate}(y,F)^\top \bar{x}| &\leq |(y - \text{Truncate}(y,F))^\top \bar{x}| \\ &\leq \alpha\bar{\alpha} \leq (\bar{k}/k)^{1/2} \min\left[\sqrt{1-(y^\top \bar{x})^2}, (1+(\bar{k}/k)^{1/2})\,(1-(y^\top \bar{x})^2)\right], \end{aligned}$$

where the last inequality follows from (9) and (10). This leads to the desired bound. ■

Next is our main lemma, which says each step of sparse power method improves eigenvector estimation.

Lemma 13 *Assume that $k \geq \bar{k}$. Let $s = 2k + \bar{k}$. If $|x_{t-1}^\top \bar{x}| > \theta + \delta(s)$, then*

$$\sqrt{1-|\hat{x}_t^\top \bar{x}|} \leq \mu\sqrt{1-|x_{t-1}^\top \bar{x}|} + \sqrt{10}\delta(s).$$

Proof Let $F = F_{t-1} \cup F_t \cup \text{supp}(\bar{x})$. Consider the following vector

$$\tilde{x}_t' = A_F x_{t-1} / \|A_F x_{t-1}\|, \tag{11}$$

where A_F denotes the restriction of A on the rows and columns indexed by F. We note that replacing x_t' with $\tilde{x}_t'$ in Algorithm 1 does not affect the output iteration sequence $\{x_t\}$ because of the sparsity of x_{t-1} and the fact that the truncation operation is invariant to scaling. Therefore for notation simplicity, in the following proof we will simply assume that x_t' is redefined as $x_t' = \tilde{x}_t'$ according to (11).

Without loss of generality and for simplicity, we may assume that $x_t'^\top x(F) \geq 0$ and $x_{t-1}^\top \bar{x} \geq 0$, because otherwise we can simply do appropriate sign changes in the proof. We obtain from Lemma 11 that

$$x_t'^\top x(F) \geq x_{t-1}^\top x(F)\,[1 + (1-\gamma(s)^2)(1-(x_{t-1}^\top x(F))^2)/2].$$

This implies that

$$\begin{aligned} [1 - x_t'^\top x(F)] \leq & [1 - x_{t-1}^\top x(F)]\,[1 - (1-\gamma(s)^2)(1+x_{t-1}^\top x(F))(x_{t-1}^\top x(F))/2] \\ \leq & [1 - x_{t-1}^\top x(F)]\,[1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)], \end{aligned}$$

where in the derivation of the second inequality, we have used Lemma 10 and the assumption of the lemma that implies $x_{t-1}^\top x(F) \geq x_{t-1}^\top \bar{x} - \delta(s) \geq \theta$. We thus have

$$\|x_t' - x(F)\| \leq \|x_{t-1} - x(F)\|\,\sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)}.$$

Therefore using Lemma 10, we have

$$\|x_t' - \bar{x}\| \leq \|x_{t-1} - \bar{x}\|\,\sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)} + 2\delta(s).$$

This is equivalent to

$$\sqrt{1-|x_t'^\top \bar{x}|} \leq \sqrt{1-|x_{t-1}^\top \bar{x}|}\sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)} + \sqrt{2}\delta(s).$$

Next we can apply Lemma 12 and use $k \geq \bar{k}$ to obtain

$$\begin{aligned}\sqrt{1-|\hat{x}_t^\top \bar{x}|} &\leq \sqrt{1-|x_t'^\top \bar{x}| + ((\bar{k}/k)^{1/2}+\bar{k}/k)(1-|x_t'^\top \bar{x}|^2)} \\ &\leq \sqrt{1-|x_t'^\top \bar{x}|}\sqrt{1+2((\bar{k}/k)^{1/2}+\bar{k}/k)} \\ &\leq \mu\sqrt{1-|x_{t-1}{}^\top \bar{x}|} + \sqrt{10}\delta(s).\end{aligned}$$

This proves the second desired inequality. ■

We are now in the position to prove Theorem 4.

Proof of Theorem 4:
Let us distinguish the following two complementary cases:
Case I: $\theta+\delta(s) > 1-10\delta(s)^2/(1-\mu)^2$. In this case, we have that $x_0^\top \bar{x} \geq \theta+\delta(s) > 1-10\delta(s)^2/(1-\mu)^2$ which implies the inequality (5).
Case II: $\theta+\delta(s) \leq 1-10\delta(s)^2/(1-\mu)^2$. In this case, we first prove by induction that for all $t \geq 0$, $x_t^\top \bar{x} \geq \theta+\delta(s)$. This is obviously hold for $t=0$. Assume that $|x_{t-1}^\top \bar{x}| \geq \theta+\delta(s)$. Let us further distinguish the following two cases:

(a) $\sqrt{1-|x_{t-1}^\top \bar{x}|} \geq \sqrt{10}\delta(s)/(1-\mu)$. From Lemma 13 we obtain that

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \sqrt{1-|\hat{x}_t^\top \bar{x}|} \leq \mu\sqrt{1-|x_{t-1}^\top \bar{x}|} + \sqrt{10}\delta(s) \leq \sqrt{1-|x_{t-1}^\top \bar{x}|},$$

where the first inequality follows from $|x_t^\top \bar{x}| = |\hat{x}_t^\top \bar{x}|/\|\hat{x}_t\| \geq |\hat{x}_t^\top \bar{x}|$. This implies $|x_t^\top \bar{x}| \geq |x_{t-1}^\top \bar{x}| \geq \theta+\delta(s)$.

(b) $\sqrt{1-|x_{t-1}^\top \bar{x}|} < \sqrt{10}\delta(s)/(1-\mu)$. Based on the previous argument we have

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \mu\sqrt{1-|x_{t-1}^\top \bar{x}|} + \sqrt{10}\delta(s) < \sqrt{10}\delta(s)/(1-\mu),$$

which implies that $|x_t^\top \bar{x}| > 1-10\delta(s)^2/(1-\mu)^2 \geq \theta+\delta(s)$.

In both cases (a) and (b), we have $|x_t^\top \bar{x}| \geq \theta+\delta(s)$ and this finishes the induction. Therefore, by recursively applying Lemma 13 we have that for all $t \geq 0$

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \mu^t\sqrt{1-|x_0^\top \bar{x}|} + \sqrt{10}\delta(s)/(1-\mu),$$

which is inequality (6). This completes the proof.

References

A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, and A. Rosenwald. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

A. 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. *Cell Biology*, 96:6745–6750, 1999.

A. A. Amini and M. J. Wainwright. High-dimensional analysis of semidefinite relaxiation for sparse principal components. *Annals of Statistics*, 37:2877–2921, 2009.

Y. Asahiro, R. Hassin, and K. Iwama. Complexity of finidng dense subgraphs. *Discrete Applied Mathematics*, 211(1-3):15–26, 2002.

A. Billionnet and F. Roupin. A deterministic algorithm for the densest k-subgraph problem using linear programming. Technical report, Technical Report, No. 486, CEDRIC, CNAM-IIE, Paris, 2004.

T. Cai, Z. Ma, and Y. Wu. Sparse pca: Optimal rates and adaptive estimation. 2012. URL `arxiv.org/pdf/1211.1309v1.pdf`.

E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

D. G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9:27–39, 1984.

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

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

U. Feige, G. Kortsarz, and D. Peleg. The dense *k*-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

X. Geng, T. Liu, T. Qin, and H. Li. Feature selection for ranking. In *Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR'07)*, 2007.

D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05)*, pages 721–732, 2005.

G. H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

J. Jeffers. Two case studies in the application of principal components. *Applied Statistics*, 16(3): 225–236, 1967.

P. Jiang, J. Peng, M. Heath, and R. Yang. Finding densest k-subgraph via 1-mean clustering and low-dimension approximation. Technical report, 2010.

I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, 29:295–327, 2001.

I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

M. Journée, Y. Nesterov, P. Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

S. Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

S. Khuller and B. Saha. On finding dense subgraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pages 597–608, 2009.

R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the 8th World Wide Web Conference (WWW'99)*, pages 403–410, 1999.

M. Liazi, I. Milis, and V. Zissimopoulos. A constant approximation algorithm for the densest k-subgraph problem on chordal graphs. *Information Processing Letters*, 108(1):29–32, 2008.

Z. Ma. Sparse principal component analysis and iterative thresholding. *Annals of Statistics*, to appear, 2013.

L. Mackey. Deflation methods for sparse pca. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS'08)*, 2008.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:10–60, 2010.

B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse lda. In *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, pages 641–648, 2006.

D. Paul and I.M. Johnstone. Augmented sparse principal component analysis for high dimensional data. 2012. URL `arxiv.org/pdf/1202.1242v1.pdf`.

S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42:299–310, 1994.

D. Shen, H. Shen, and J.S. Marron. Consistency of sparse pca in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115:317–333, 2013.

H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)*, pages 181–191, 1998.

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

D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

R. Yang. New approximation methods for solving binary quadratic programming problem. Technical report, Master Thesis, Department of Industrial and Enterprise Systems Engineering, University of Illnois at Urbana-Champaign, 2010.

Y. Y. Ye and J. W. Zhang. Approximation of dense-n/2-subgraph and the complement of min-bisection. *Journal of Global Optimization*, 25:55–73, 2003.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Journal of Machine Learning Research 14 (2013) 927-964 Submitted 5/10; Revised 9/11; Published 4/13

Query Induction with Schema-Guided Pruning Strategies

Joachim Niehren JOACHIM.NIEHREN@INRIA.FR
Jérôme Champavère JEROME.CHAMPAVERE@LIFL.FR
Aurélien Lemay AURELIEN.LEMAY@UNIV-LILLE3.FR
Links Project-Team – INRIA *Lille &* LIFL
Parc Scientifique de la Haute Borne
Park Plaza – Inria Bat. B – 40 Avenue Halley
59650 Villeneuve d'Ascq
France

Rémi Gilleron REMI.GILLERON@UNIV-LILLE3.FR
Magnet Project-Team – INRIA *Lille &* LIFL
Parc Scientifique de la Haute Borne
Park Plaza – Inria Bat. B – 40 Avenue Halley
59650 Villeneuve d'Ascq
France

Editor: Mehryar Mohri

Abstract

Inference algorithms for tree automata that define node selecting queries in unranked trees rely on tree pruning strategies. These impose additional assumptions on node selection that are needed to compensate for small numbers of annotated examples. Pruning-based heuristics in query learning algorithms for Web information extraction often boost the learning quality and speed up the learning process. We will distinguish the class of regular queries that are stable under a given schema-guided pruning strategy, and show that this class is learnable with polynomial time and data. Our learning algorithm is obtained by adding pruning heuristics to the traditional learning algorithm for tree automata from positive and negative examples. While justified by a formal learning model, our learning algorithm for stable queries also performs very well in practice of XML information extraction.

Keywords: XML information extraction, XML schemas, interactive learning, tree automata, grammatical inference

1. Introduction

A fundamental problem of XML information extraction is query induction from annotated examples. The problem is to select "relevant" elements in collections of XML documents, while knowing only few positive and negative examples for relevant elements. The target of this learning problem is thus a query for elements in XML documents.

Most approaches for XML information extraction can be found in the context of Web information extraction. The many learning methods applied there range from statistical classification (Kushmerick, 2000; Gilleron et al., 2006b), hidden Markov models (Freitag and McCallum, 1999), conditional random fields (Pinto et al., 2003; Zhu et al., 2005; Gilleron et al., 2006a), active learning on strings (Muslea et al., 2003), grammatical inference (Raeymaekers et al., 2008; Carme et al.,

2007), to inductive logic programming (Cohen et al., 2002). Queries for information extraction can also be produced by visual programming (Baumgartner et al., 2001) possibly with learning enhancement (Carme et al., 2006a). Unsupervised approaches based on ontology knowledge were proposed recently (Sellers et al., 2011a).

```
<country>
  <name>France</name>
  <city>Paris</city>
  <region>
    <name>Nord−Pas de Calais</name>
    <population>3996588</population>
    <city>Lille</city>
  </region>
  <region>
    <name>Vallée du Rhône</name>
    <city>Lyon</city>
    <city>Valence</city>
  </region>
</country>
```

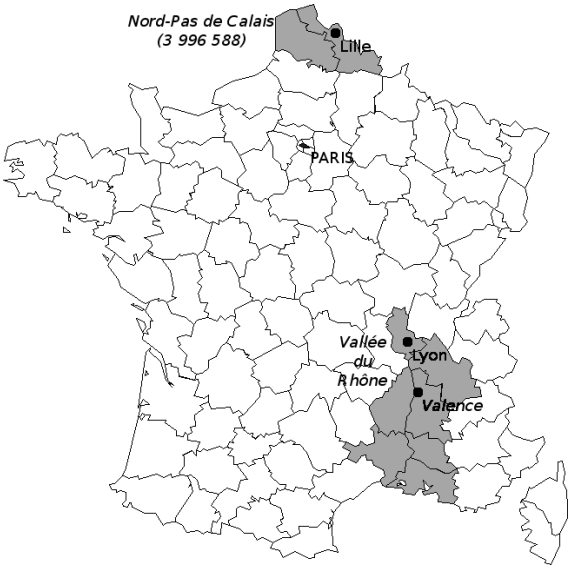


Figure 1: A tiny geographical XML database.

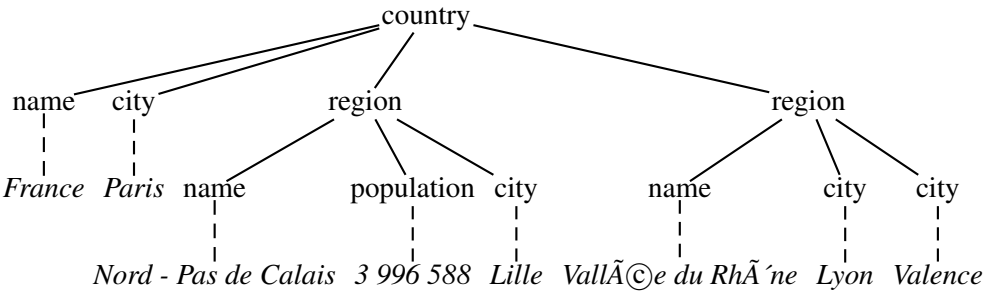


Figure 2: Data tree of the geographical database in Figure 1.

For illustration, we consider in Figure 1 an XML document for a geographical database with information about regions in France. This XML document can be parsed into the data tree in Figure 2. A user may then want to select, for instance, all those regions for which the size of the population is known in the database. This goal can be translated to a query to data trees of geographical databases, which selects all nodes labeled by `region` and having a child labeled by `population`, that is the XPath query `//region[population]`.

However, finding this query requires knowledge on XPath and the XML schema of geographical databases, which is not to be expected from a non-expert user. The difficulties of non-expert users can be solved by supervised query induction. The idea is that the user annotates some examples of elements positively or negatively, meaning that they should be selected or not. This can be done in a graphical interface of the database, as illustrated in Figure 3, or in a Web browser, and then translated to annotations on the XML tree. It can also be done directly by annotating the XML tree.

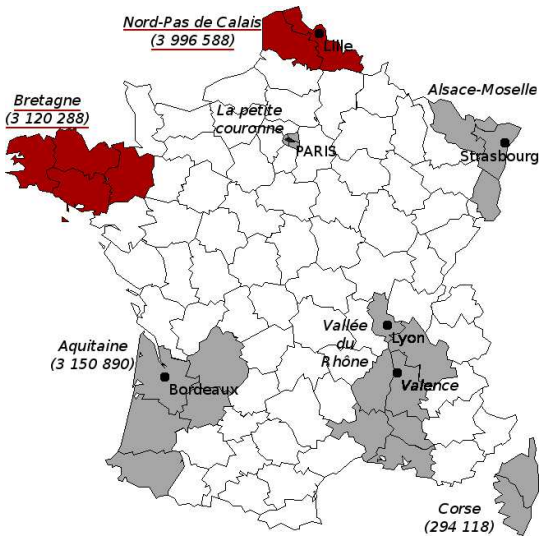


Figure 3: Two positive examples annotated via the user interface.

The query induction algorithm then learns the query from the XML trees with some annotated elements. The main difficulties of query induction are the following:

1. the availability of only few annotated examples,
2. missing semantic information about the meaning of XML tags, and
3. the difficulty to understand natural language texts in data trees.

We do not tackle the third problem in the present paper. Instead, we restrict ourselves to node selection queries in unranked trees without data values. In order to limit the burden of annotation and to obtain relevant annotations quickly, we will rely on interactive learning in Angluin style (Angluin, 1987), where selected nodes proposed by the learner are corrected by the teacher in an annotate-learn-select loop. This corresponds to the usual user interaction loop of Web information extraction systems (see, e.g., Muslea et al., 2003).

In this article, we study the relevance of schema-guided pruning heuristics for query induction.[1] Pruning consists in removing useless information for learning. Schema-less pruning strategies were essential for good quality with few examples of query induction algorithms based on tree automata inference by, for example, Raeymaekers et al. (2008) or Carme et al. (2007), and also for decent efficiency. So far, however, pruning strategies were always defined in ad hoc manners. Thus, our first objective is to define them systematically. Pruning strategies must be aggressive on the one hand side, since the more subtrees are pruned, the more efficient the learning algorithm will be. On the other hand side, pruning strategies must preserve the information required to learn the query. Our second objective is hence to formalize the relationship between pruning strategies and learnable classes of queries. Finally, schemas express semantic information on XML trees which should allow to learn larger classes. Our third objective is thus to use schemas to improve learning algorithms and to understand the relevance of schemas for pruning strategies in particular.

1. Early ideas of the present article were published by Champavère et al. (2008) at ICGI, elaborated in the PhD thesis of Champavère (2010), and seriously revised here.

$$
\begin{array}{ll}
country & \rightarrow name \cdot city \cdot region^* \\
region & \rightarrow name \cdot population^{\varepsilon} \cdot city^+ \\
name & \rightarrow \varepsilon \\
city & \rightarrow \varepsilon \\
population & \rightarrow \varepsilon
\end{array}
$$

Figure 4: Schema *Geo* for a geographical XML database restricted to the description of a country. It is defined by a DTD where country is the label of the root.

Document type definitions (DTDs) are the simplest schemas for XML documents. For instance, let us consider the DTD in Figure 4 for our geographical database example. It says that under the root with label country, there is its name, followed by a city, followed by a possibly empty list of regions. Each region is described by its name, its population but it is not mandatory, and a non empty list of cities. For sake of simplicity, we assume that subtrees with root label name, city and population only contain data values that we do not consider here. For instance, the unranked tree $u = \text{country}(\text{name}, \text{city}, \text{region}(\text{name}, \text{population}, \text{city}), \text{region}(\text{name}, \text{city}, \text{city}))$ is valid for the DTD *Geo*, it is the tree presented in Figure 2 without the data values. Schemas can also be defined by deterministic tree automata. These are finite state machines adapted to trees. They process trees in a bottom-up way (from the leaves to the root) according to a finite set of rules. The result is a tree annotated with states. For instance, a tree automaton associated with the DTD *Geo* can be defined with states $q_{country}$, q_{name}, etc., and a state $q_{invalid}$, and rules expressing the conditions given by the DTD. For an invalid tree, the state $q_{invalid}$ will appear along the run of the automaton and the tree will be rejected, while a valid tree will be processed and annotated by the correct states. For instance the unranked tree u will be annotated as $q_{country}(q_{name}, q_{city}, q_{region}(q_{name}, q_{population}, q_{city}), q_{region}(q_{name}, q_{city}, q_{city}))$, and will be accepted because the root state is $q_{country}$.

We next introduce schema-guided pruning strategies by example and illustrate their relationship to query classes. Pruning strategy *path-only* applied to a node of the XML database keeps only the path leading from the root to the node, and replaces all subtrees adjacent to this path by a special symbol $\top$. Pruning strategy $\textit{path-only}_{Geo}$ keeps that same path but replaces the adjacent subtrees by the state assigned to them by the tree automaton for *Geo*. In Figure 5, the result of applying both pruning strategies to the `name`-child of the `region`-node in Figure 2 is presented. It will turn out that more complex pruning strategies will be needed for ranked trees, in order to deal with the above pruning strategies for unranked trees via binary encoding.

We call a query *stable* for a pruning strategy if the result of applying the strategy to a tree at some selected node does always justify the node's selection. In this case, we call this result the critical region for selecting the node. For illustration, let us consider the query that selects all names of regions whose population is known in our geographic XML database with schema *Geo*, that is the XPath query `//region[population]/name`. Whether a node with label `name` is selected depends only on its local environment, more precisely, on whether its father is labeled by `region` and whether its right-sibling is labeled by `population`. It is easy to see that pruning strategy *path-only* applied to a selected node removes the label of its right sibling. Thus this query is not stable for pruning strategy *path-only*. In contrast, the schema-guided pruning strategy $\textit{path-only}_{Geo}$ stores the label

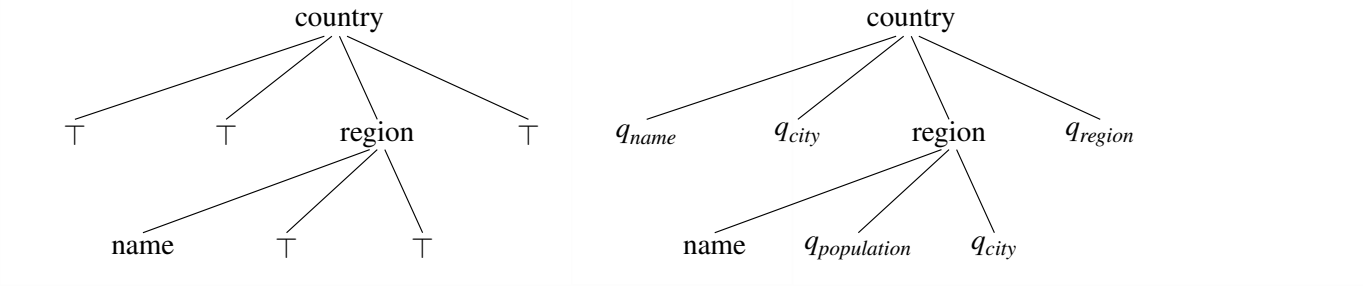


Figure 5: Pruning the tree from Figure 2 at the name-child of the first region-node, by strategies *path-only* (left) and $\textit{path-only}_{Geo}$ (right).

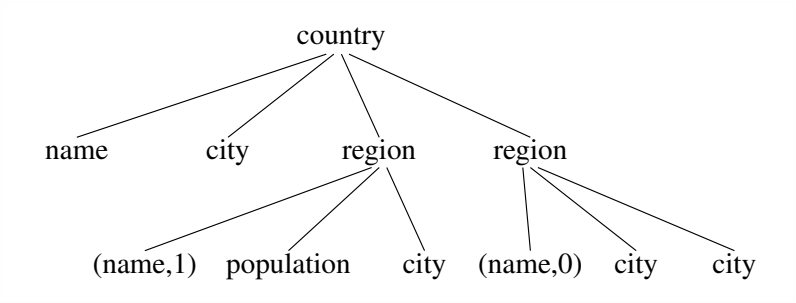


Figure 6: An annotated tree in which the first region name has been annotated positively (it must be selected) and the second region name has been annotated negatively (it must not be selected).

of the right sibling in state $q_{population}$, so that this relevant information is preserved. Therefore, the query `//region[population]/name` is stable for pruning strategy $\textit{path-only}_{Geo}$.

A user of the learning system may want to learn the above XPath query since he does not know how to express it formally. He might not be an XPath expert or might not have access to the schema *Geo* of our geographic database. Such users may still be willing to annotate some selected `name`-nodes by the Boolean value 1 and some rejected nodes by the Boolean value 0 with the help of a graphical interface. An example for a partial annotation of our geographical database that could be obtained this way is given in Figure 6. Pruning strategies can be lifted to pruning functions on positively annotated trees. The easiest way to do so is to keep only the union of the critical regions of all nodes annotated by 1 jointly with their annotations.

Given a pruning strategy σ, the next question is whether σ-stable queries can be learned from σ-*pruned samples* of annotated examples. These are finite sets that contain σ-pruned trees with positive annotations and unpruned trees with negative annotations, such that all annotations are consistent with respect to the target query. For any pruning strategy σ, we will distinguish the class of regular σ-stable queries and show that this class can indeed be identified from σ-pruned

samples. Depending on the pruning strategy, this yields a hierarchy of query classes that is essential for understanding the difficulty of query learning in practice.

1.1 Contributions

We study the impact of pruning strategies on learning algorithms for classes of regular queries for the first time. We recall that XPath queries (without tests on data values) are regular, even if imposing schema restrictions by DTDs or tree automata.

1. We define schema-guided pruning strategies for schemas defined by deterministic bottom-up tree automata, both for ranked and unranked trees. We introduce the notion of stable queries for a given pruning strategy. We show how to characterize stable queries by languages of pruned annotated trees. For regular queries these languages can be recognized by tree automata.

2. We lift pruning strategies to pruning functions that can be applied to positively annotated trees, in order to produce pruned samples for our learning algorithm.

3. We show how to represent pruning functions by means of tree automata. Thereby we define the notion of regular pruning functions. We present an algorithm that decides in polynomial time whether a regular query is stable for a regular pruning function.

4. We present a learning algorithm, based on the $\mathcal{RPNI}$ algorithm (García and Oncina, 1993), that identifies regular σ-stable queries in polynomial time from σ-pruned samples of polynomial cardinality.

5. We present experimental results that confirm the relevance of our query learning algorithm in an interactive learning environment, both for Web information extraction and for XML information extraction.

6. We compare different pruning strategies with respect to their aggressiveness. It turns out that more aggressive pruning strategies yield better learning quality. We also discuss how to select appropriate schema-guided pruning strategies in practice.

1.2 Outline

For sake of clarity, we first give definitions, results and proofs for ranked trees. All results can be lifted to the case of unranked trees via a binary encoding. Experiments will be done on information extraction tasks over unranked trees (HTML trees or XML trees).

Section 2 recalls preliminaries on tree automata for ranked trees and illustrates how to use them as schemas. Section 3 introduces the notion of stable queries for schema-guided pruning strategies and shows that less aggressive pruning strategies give rise to larger classes of stable queries. In Section 4 we show how to lift pruning strategies to pruning functions, by which to prune examples with positive annotations only. We then show how to characterize stable queries for pruning functions by regular languages of pruned annotated examples in a unique manner. Thereby we specify the target languages for the learning algorithm. Section 5 discusses how to define regular stable queries and pruning functions by deterministic tree automata. In Section 6 we present algorithms for testing two consistency properties for regular languages of pruned annotated trees. Section 7 presents our new

learning algorithm for stable queries from pruned annotated examples based on these algorithms. In Section 8, we lift all previous results to unranked trees via a binary encoding. Experimental results on query induction in XML information extraction are discussed in Section 9. Future work and conclusions are presented in Section 10. The appendix completes the proofs of results that are not essential for the main contributions.

1.3 Related Work

There exist two previous approaches for inducing regular languages that can account for the consistency with some domain of the target language. Both of them guarantee that the current hypothesis L of the learning algorithm does satisfy $L \subseteq \mathcal{L}(D)$, were D is a deterministic finite automaton defining the domain, in which the target language must be included. The first dynamic approach is to test language inclusion after each generalization step (Coste et al., 2004). This is the approach we here generalize to query induction. The second static approach is to ensure inclusion (Oncina and Varó, 1996) by typing all states of the current DFA A by the states of the domain DFA D. This means that the target automaton will be a product with the domain automaton D. Formal learnability results were missing for both approaches so far. In the present article, we provide such results for the first time for the dynamic approach. Furthermore, we show that the dynamic approach is feasible in practice even in the case of tree automata, where inclusion testing is more tedious. The static approach performs worse in our applications in most cases.

Schema induction for XML documents was invested by Bex et al. (2006). The interest there is to produce readable DTDs with regular expressions. Query induction, as presented here, usually tries to avoid the induction of schemas, since queries may only rely on parts of the schema. Conversely, however, one might use induced schemas for query induction. Induction of XPath queries was considered by Carme et al. (2006b) and Staworko and Wieczorek (2011). Induction algorithms for top-down deterministic tree transducers were presented by Lemay et al. (2010).

2. Schemas and Tree Automata

In this section, we recall facts on tree automata on ranked trees and illustrate how to use them as schemas. Unranked trees will be treated later on via a binary encoding. Most results we will remind here are standard with the exception of an efficient inclusion test, which will be fundamental for the efficiency of our schema-guided learning algorithm.

Let $\mathbb{N}$ be the set of non-zero natural numbers. A ranked signature is a set Σ equipped with a function rank_Σ from Σ to $\mathbb{N} \cup \{0\}$. For $k \geq 0$, we denote by $\Sigma^{(k)}$ the set $\{f \in \Sigma \mid \text{rank}_\Sigma(f) = k\}$. We write $f^{(k)}$ to indicate that f is of rank k. The set $\mathcal{T}_\Sigma$ of ground terms over Σ is the least set that contains all tuples $f(t_1,\ldots,t_k)$ where $f \in \Sigma^{(k)}$ and $t_1,\ldots,t_k \in \mathcal{T}_\Sigma$. Ground terms in $\mathcal{T}_\Sigma$ are equivalently called ranked Σ-trees or simply trees. As usual we write f instead of the single node tree $f()$ where $f^{(0)} \in \Sigma$ is a constant. We denote by $\mathit{nodes}(t) \subseteq \mathbb{N}^*$ the (prefix-closed) set of nodes of the tree t. The label of a node ν in t is denoted by $t[\nu]$. Given a subset $\Sigma' \subseteq \Sigma$ of labels, we denote by $\mathit{nodes}_{\Sigma'}(t) = \{\nu \in \mathit{nodes}(t) \mid t[\nu] \in \Sigma'\}$ the set of all nodes of t labeled in Σ'.

A *tree automaton* is a tuple $D = (\Sigma, X, R, F)$ where Σ is a finite ranked signature, X is a finite set of states, $F \subseteq X$ is a set of final states, and $R \subseteq \bigcup_{k\geq 0} \Sigma^{(k)} \times X^{k+1}$ is a set of (transition) rules. We denote by $|D|$ the sum of the lengths of the rules of D and call it the size of D. A transition rule $r \in R$ is a tuple $(f^{(k)}, q_1, \ldots, q_k, q)$ where $q, q_1, \ldots, q_k$ are states in X. As usual, we denote such a rule r

by $f(q_1,\ldots,q_k) \to q$ and call $f(q_1,\ldots,q_k)$ its left-hand side. A tree automaton D is (bottom-up) *deterministic* if no two rules of D have the same left-hand side.

The evaluator of D is a function $eval_D : \mathcal{T}_\Sigma \to 2^X$ such that for all trees $f(t_1,\ldots,t_k) \in \mathcal{T}_\Sigma$: $eval_D(f(t_1,\ldots,t_k)) = \{q \mid f(q_1,\ldots,q_k) \to q \text{ in } D, \forall i \in \{1,\ldots,k\} : \ q_i \in eval_D(t_i)\}$. If D is deterministic, then, for any t, the set $eval_D(t)$ contains at most one element. The language recognized by D is the set of all trees in $\mathcal{T}_\Sigma$ that D can evaluate into some final state:

$$\mathcal{L}(D) = \{t \in \mathcal{T}_\Sigma \mid eval_D(t) \cap F \neq \emptyset\}.$$

A tree language $L \subseteq \mathcal{T}_\Sigma$ is *regular* if $L = \mathcal{L}(D)$ for some tree automaton D with signature Σ.

Example 1 *We consider libraries (lib) which contain lists of books (b) with lists of authors (a). We use cons and nil as list constructors as usual for ranked trees, so we use the following signature* $\Sigma_{Lib} = \{lib^{(1)}, b^{(1)}, a^{(0)}, cons^{(2)}, nil^{(0)}\}$. *The unranked library* $lib(b(a,a),(b(a,a),b))$ *with three books of which the last one has no author is represented by the following ranked* Σ*-tree, which is also illustrated graphically on the left of Figure 8:*

$$lib(cons(b(cons(a,cons(a,nil))),cons(b(cons(a,cons(a,nil))),cons(b(nil),nil))))$$

The schema of libraries is defined by the tree automaton Lib' *with signature* Σ_{Lib} *with state set* $X = \{q_{lib}, q_{bs}, q_{as}, q_b, q_a\}$*, final state* q_{lib}*, and the rules:*

$$\begin{array}{llllllllll} lib(q_{bs}) & \to & q_{lib}, & b(q_{as}) & \to & q_b, & nil & \to & q_{bs}, \\ cons(q_b, q_{bs}) & \to & q_{bs}, & nil & \to & q_{as}, & cons(q_a, q_{as}) & \to & q_{as}, \\ a & \to & q_a. & & & & & & \end{array}$$

Note that Lib' *is nondeterministic, since symbol nil may be analyzed as the end of a book-list or as the end of an author-list. Let Lib be the determinization of* Lib'*. This can be done by mapping nil to some new state* $\{q_{as}, q_{bs}\}$*, while adding the necessary rules for this state.*

We will need some basic constructions on tree automata and complexity results which can be found in Comon et al. (2007). Given a tree automaton D, it is decidable in time $O(|D|)$ whether $\mathcal{L}(D) = \emptyset$. Given another tree automaton D' over the same signature, a tree automaton $D \cap D'$ with $\mathcal{L}(D \cap D') = \mathcal{L}(D) \cap \mathcal{L}(D')$ can be computed in time $O(|D| \ |D'|)$.

Our learning algorithm will heavily rely on language inclusion tests for deterministic tree automata. We will use a recent inclusion test, proposed by the authors, which can be implemented highly efficiently and incrementally (w.r.t. adding rules to D'):

Proposition 1 (Efficient Inclusion Test, Champavère et al., 2009) *Let* D' *and* D *be tree automata over* Σ*. If* D *is deterministic then language inclusion* $\mathcal{L}(D') \subseteq \mathcal{L}(D)$ *can be decided in time* $O(|D'| \ |D|)$.

3. Schema-Guided Pruning Strategies

We introduce schema-guided pruning strategies and define a partial aggressiveness order on them. We then present stable queries for a given pruning strategy. We show that less aggressive pruning strategies give rise to larger classes of stable queries.

3.1 Pruned Trees

We fix a ranked signature Σ and a schema D which is a deterministic tree automaton with signature Σ and state set X. We define a *pruned tree* to be a tree over signature $\Sigma \cup X$, that is, the states of D become additional constants. Whenever D is unclear from the context, we will talk of *D-pruned trees* equivalently to pruned trees.

We will write $\mathcal{T}_\Sigma(X)$ instead of $\mathcal{T}_{\Sigma\cup X}$ for the set of pruned trees. It should be noted that schema D can be also abused to recognize pruned trees. In order to do so, it is sufficient to add the rules $q \to q$ for all states $q \in X$. In this way, the evaluator $eval_D$ for unpruned trees can be lifted to pruned trees.

Let t and t' be two pruned trees in $\mathcal{T}_\Sigma(X)$. We say that t *is subsumed by* t', or equivalently that t' *is a D-instance of* t, and write $t \leq t'$ if t' can be obtained from t by replacing occurrences of states q by pruned trees that can be evaluated to q by D. We call a finite set of pruned trees $L \subseteq \mathcal{T}_\Sigma(X)$ *compatible* (with respect to D) if all trees in L have a common instance, that is, if there exists a tree t' such that $t \leq t'$ for all $t \in L$. In this case, L has a least upper bound, that we denote by $\sqcup L$ such that $t \leq \sqcup L$ for all $t \in L$. The existence of least upper bounds follows, since we assume that D is deterministic, so that no subsets of X with two different states $\{q, q'\} \subseteq X$ are compatible. For instance, if $\Sigma = \{f^{(2)}, g^{(1)}, a^{(0)}\}$ and D contains the rules $a \to q'$, $g(q') \to q$, and $f(q,q) \to q''$ where q'' is final then:

$$\sqcup\{f(g(a),q), f(q,g(q')), f(q,q), q''\} = f(g(a),g(q')).$$

A D-completion of a pruned tree $t \in \mathcal{T}_\Sigma(X)$ is a tree $t' \in \mathcal{T}_\Sigma$ such that $t \leq t'$. Every pruned tree t defines a sublanguage of $\mathcal{T}_\Sigma$ that we denote by $compl_D(t)$ containing all its D-completions. Note that $compl_D(q)$ is the set of all trees $t \in \mathcal{T}_\Sigma$ that D can evaluate to q.

3.2 Pruning Strategies

The purpose of a pruning strategy is to remove all parts of a tree that are irrelevant for the selection of a given node. Removed subtrees are always replaced by the state into which they are evaluated by the schema, so that some information about removed subtrees may still be preserved.

Definition 2 *A pruning strategy for D is a function σ that maps any tree $t \in \mathcal{L}(D)$ and node $\nu \in nodes(t)$ to a pruned tree $\sigma(t,\nu) \in \mathcal{T}_\Sigma(X)$ of which t is a D-instance, such that ν is preserved with its label.*

Since schema D is a deterministic tree automaton, there is no choice by which state a subtree is to be replaced. Thus, a pruning strategy can be specified by the subset of nodes rooting pruned subtrees. Let us introduce the four pruning strategies that will be used in the experiments. The pruning strategy *path-only* is total. It prunes away all maximal subtrees that are not ancestors of ν and replaces them by $\top$. The pruning strategy $path\text{-}only_D$ is more restrictive in that it can only be applied to trees satisfying schema D. Note that if $\mathcal{U}$ is the unique universal tree automaton with a single state $\top$ that recognizes all trees, then $path\text{-}only = path\text{-}only_\mathcal{U}$. Pruning strategy $path\text{-}ext_D$ can be applied to any tree satisfying schema D. When applied to a tree t with node ν, the extended path from the root to ν remains unchanged, that is, all children of siblings of nodes on the path to ν are substituted by their state, as well as all children of ν. The pruning strategy *path-ext* is equal to $path\text{-}ext_D$ where $D = \mathcal{U}$.

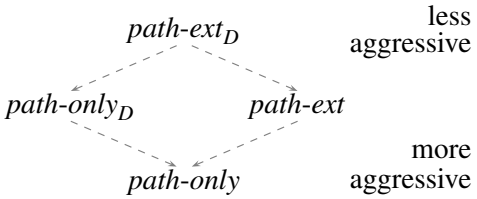


Figure 7: Example pruning strategies ordered by aggressiveness.

Definition 3 *Let σ and σ' be two pruning strategies for schema D and D' respectively such that $\mathcal{L}(D) \subseteq \mathcal{L}(D')$. We call σ less aggressive than σ' if any tree $t \in \mathcal{L}(D)$ with node ν satisfies that $compl_D(\sigma(t,\nu)) \subseteq compl_{D'}(\sigma'(t,\nu))$.*

We will use the following sufficient criterion to prove that σ is less aggressive than σ':

1. Schema D should always evaluate to more informative states than D', that is, for any tree t and state $q \in eval_D(t)$, there exists a state $q' \in eval_{D'}(t)$ such that $compl_D(q) \subseteq compl_{D'}(q')$, and

2. strategy σ should always replace fewer subtrees than σ', that is, for any $t \in \mathcal{L}(D)$ with node ν: $nodes_\Sigma(\sigma(t,\nu)) \supseteq nodes_\Sigma(\sigma'(t,\nu))$.

Clearly, *path-ext*$_D$ is less aggressive than *path-only*$_D$ since *path-only*$_D$ always leaves the minimal number of nodes unchanged, but *path-ext*$_D$ leaves children of ancestor nodes on the path unchanged too. The strategy *path-only* is more aggressive than *path-only*$_D$ since both prune the same nodes away, while *path-only*$_D$ substitutes them by more informative states than *path-only*.

The notion of aggressiveness defines a partial order on pruning strategies, which is illustrated for our example strategies in Figure 7. This order is usually not total since *path-only*$_D$ and *path-ext* are incomparable for most choices of D and Σ.

3.3 Stable Queries

A *query Q with schema D* is a partial function with domain $\mathcal{L}(D)$ which maps trees t from this domain to sets of nodes $Q(t) \subseteq nodes(t)$. Note that queries can be applied to complete trees only. Now, we define the class of those queries that are stable with respect to a given schema-guided pruning strategy.

Definition 4 *Let σ be a pruning strategy and Q a query, both with schema D. We call Q σ-stable if all trees $t \in \mathcal{L}(D)$, selected nodes $\nu \in Q(t)$, and D-completions t' of $\sigma(t,\nu)$ satisfy $\nu \in Q(t')$.*

For a selected node $\nu \in Q(t)$ of a σ-stable query Q, we call $\sigma(t,\nu)$ the critical region of ν in t. Note that we do not define any critical region for rejected nodes, since the definition of stability talks only about selected nodes. The next example illustrates the relevance of selected nodes' critical regions.

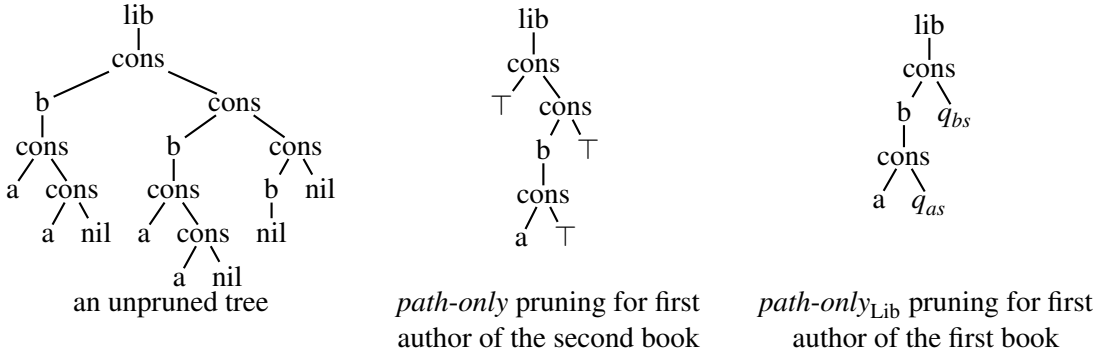


Figure 8: Illustration of "path-only" pruning strategies at the library from Example 1 without schema and with schema Lib.

Example 2 *Let Lib be the deterministic tree automaton recognizing libraries from Example 1. On the left of Figure 8, we present an example library (lib), which contains three books (b), of which the first two have two authors (a), while the last has none. Given that we use ranked trees, lists are build by a binary list constructor (cons), and a constant (nil) for the empty list. We consider the following queries on libraries with this schema:*

Q_1 *selects the first author of all books of the library.*

Q_2 *selects the first author of all books of the library with at least two authors.*

Query Q_1 is stable for both pruning strategies path-only and thus for path-only$_{Lib}$ while query Q_2 is stable only for path-only$_{Lib}$ but not for path-only. This becomes clear when considering the applications of these pruning strategies to the nodes, that correspond to the first authors of the first two books.

The result of applying path-only to the first author of the second book is illustrated in the middle of Figure 8. Note that all the information needed for the selection of this author by Q_1 is preserved. However, since the second author of the same book is pruned away and replaced by $\top$, one cannot know whether there was a second author before, so necessary information is lost for deciding selection by Q_2.

The result of applying path-only$_D$ to the first author of the first book is shown on the right of Figure 8. This time, the pruned tree contains enough information to select the first author of the first book by Q_2, since now, the second author of this book is replaced by the state q_{as} which stands for a nonempty author list.

Our next objective is to restrict pruning strategies to schemas recognizing smaller languages. So let σ' be a pruning strategy with some schema D' such that $\mathcal{L}(D) \subseteq \mathcal{L}(D')$. We now define the pruning strategy $\sigma'_{|D}$ with schema D such that it replaces for all trees $t \in \mathcal{L}(D)$ the same subtrees t_0 as σ' by the unique state in $eval_D(t_0)$, but not by the unique state in $eval_{D'}(t_0)$ as chosen by σ'. Note that we deliberately overload the notation of function restriction here, in that states in images are changed too when restricting domains of pruning strategies.

Proposition 5 *Let σ and σ' be pruning strategies with respective schemas D and D' such that σ is less aggressive than σ'. Any query Q with schema D then satisfies that:*

$$Q \text{ is } \sigma'_{|D}\text{-stable} \Rightarrow Q \text{ is } \sigma\text{-stable.}$$

Proof Let tree t_0 be a D-completion of $\sigma(t,\nu)$ for some selected node $\nu \in Q(t)$. Since σ is less aggressive then σ', t_0 is also a D'-completion of $\sigma'(t,\nu)$. Since $t_0 \in L(D)$ and $\mathcal{L}(D) \subseteq \mathcal{L}(D')$, it is also a D-completion of $\sigma'_{|D}(t,\nu)$. The $\sigma'_{|D}$-stability of Q thus yields $\nu \in Q(t_0)$, so that ν remains selected on any σ-variant of t at ν, which shows that Q is also σ-stable. ■

4. Languages of Annotated Examples for Stable Queries

We will characterize stable queries in terms of languages of pruned positively annotated trees. In order to do so, we will lift pruning strategies to pruning functions that can be applied to example trees with positive annotations. There are several manners to do so, depending of whether only a single positive annotation is permitted or else many of them. Only once the choice of the pruning function is fixed, the characteristic language of a stable query can be defined in a unique manner.

The main idea of the learning algorithm for regular stable queries in Section 7 will be to identify the characteristic languages of the target query from annotated examples. Different methods for lifting pruning strategies to pruning functions will give rise to different target languages and thus to different learning algorithms.

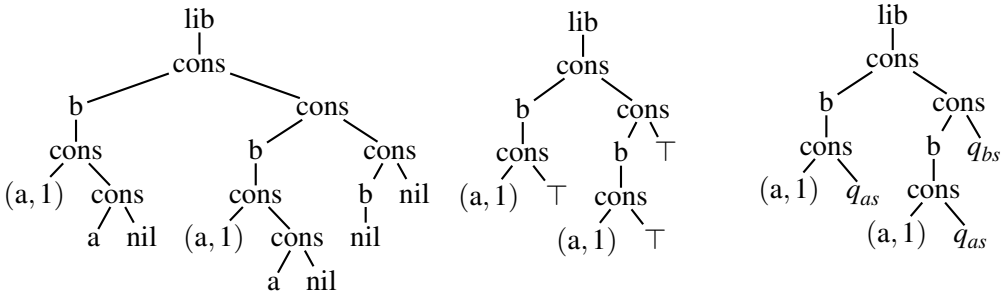
4.1 Annotated Trees

Intuitively, annotated examples for a target query are trees in which some selected nodes are annotated by the Boolean 1 ("true") and some rejected nodes by the Boolean 0 ("false"). We will support partial annotations, so that only few annotations have to be added by a user. Nodes without annotation may either be selected or rejected. Some subtrees may be pruned away and replaced by a state of the schema of the query.

We next formalize the notion of annotated trees (independently of any target query or pruning strategy). Let $\mathbb{B} = \{0,1\}$ be the set of Booleans. As before, we fix a ranked signature Σ and a deterministic tree automaton D with state set X. An *annotated tree* is a tree with ranked signature $\Sigma \cup (\Sigma \times \mathbb{B}) \cup X$, where all $q \in X$ have arity 0 while Boolean annotations preserve the arity. Nodes labeled by states in X are placeholders for subtrees which may contain both selected or rejected nodes. For instance, $(f,0)(a, f((b,1),q))$ is an annotated tree where $f^{(2)}, a^{(0)}, b^{(0)} \in \Sigma$ and $q \in X$. We call an annotated tree *unpruned* if none of its nodes is labeled in X. We call it *positively annotated* if none of its nodes is labeled in $\Sigma \times \{0\}$ and *negatively annotated* if none its nodes is labeled in $\Sigma \times \{1\}$.

An *annotation of a pruned tree* $t \in \mathcal{T}_\Sigma(X)$ is a partial function β mapping a subset of Σ-nodes of t to a Boolean. Let $dom(\beta) \subseteq nodes_\Sigma(t)$ be the domain of β. For instance, the annotation $\beta = [1 \mapsto 1, 2 \mapsto 0]$ maps node 1 to 1 and node 2 to 0. We denote by $t * \beta$ the annotated tree obtained from t by relabeling all nodes ν in the domain of β to $(t[\nu], \beta(\nu))$ while preserving the labels of all other nodes. Let Q be a query with schema D. We call an annotation β for t *Q-consistent* if all nodes $\nu \in dom(\beta)$ satisfy:

$$\beta(\nu) = 1 \Leftrightarrow \nu \in Q(t).$$



unpruned annotated example | *path-only* pruned example | *path-only*$_{\text{Lib}}$ pruned example

Figure 9: An unpruned annotated example for the two queries from Example 2 and two pruned annotated examples obtained by applying the "path-only" pruning function respectively without and with schema Lib.

In this case, we call $t * \beta$ an *annotated example for* Q. Note that annotations may be partial. Note also that 0-annotations are strict in that all nodes annotated by 0 in some annotated example for Q must be rejected by Q.

We will need a projection operation on annotated trees which deletes all annotations. The Σ-projection of a language L of annotated trees is the language $\Pi_\Sigma(L)$ of pruned trees $t \in \mathcal{T}_\Sigma(X)$ where every t is the Σ-projection of some tree $t * \beta \in L$. For every tree automaton A with signature $\Sigma \cup (\Sigma \times \mathbb{B}) \cup X$, one can compute in linear time a nondeterministic automaton $\Pi_\Sigma(A)$ over $\Sigma \times X$ such that $\mathcal{L}(\Pi_\Sigma(A)) = \Pi_\Sigma(\mathcal{L}(A))$.

4.2 From Pruning Strategies to Pruning Functions

We next show how to lift pruning strategies in order to prune positively annotated trees. There are two main manners to do so, depending on whether one permits to prune trees with many positive annotations or trees with a single positive annotation only.

Given a pruning strategy σ, our first objective is to define a pruning function p_σ that can be applied to all positively annotated trees $t * \beta$ with $t \in \mathcal{L}(D)$, while preserving the critical regions $\sigma(t, \nu)$ of all positively annotated nodes ν with their annotations:

$$p_\sigma(t * \beta) = \left(\sqcup_{\nu \in dom(\beta)} \sigma(t, \nu)\right) * \beta.$$

The least upper bound $\sqcup$ defines the least common instance of the σ relevant regions of all 1-annotated nodes of t. Note that this least upper bound does always exist because t is an upper bound of all $\sigma(t, \nu)$. The above least upper bound thus requires that a node is substituted by a state if and only if it is substituted by some of these prunings and not preserved by any other. Since we assume that β is a positive annotation, note that $\beta(\nu) = 1$ for all ν in the domain of β.

The concrete pruning functions $p_{\textit{path-only}_D}$ and $p_{\textit{path-ext}_D}$ keep all paths (respectively extended paths) to 1-annotated nodes.

Example 3 *We reconsider the library tree from Example 2 (see Figure 8). On the left of Figure 9 we present a positively annotated unpruned example for both queries Q_1 and Q_2 from Example 2. In the middle, we present the result obtained by applying function $p_{\textit{path-only}}$ to this annotated*

example, and on the right the annotated example obtained by applying function $p_{path\text{-}only_D}$. The $p_{path\text{-}only}$ pruned example in the middle contains the minimal information relevant for query Q_1 for first-authors, while the $p_{path\text{-}only_D}$ pruned example on the right contains the minimal information relevant for query Q_2 for first-authors that are not single authors. It should be noticed that negative information will be provided to the learning algorithm independently through negatively annotated examples.

Alternatively, we could permit to prune only trees with a single positive annotation. This is done by the following pruning function p^{can}_σ, which is defined for all $t * [\nu \mapsto 1]$ with $t \in \mathcal{L}(D)$ as below and undefined for all other annotated trees:

$$p^{can}_\sigma(t * [\nu \mapsto 1]) = \sigma(t,\nu) * [\nu \mapsto 1].$$

If using such "canonical" pruning functions in our learning algorithm with a universal schema $D = \mathcal{U}$, we will obtain back the learning algorithm from Lemay et al. (2006) restricted to monadic queries. In our experiments, we will exclusively work with pruning functions p_σ, even though the alternative pruning function p^{can}_σ is highly promising for learning n-ary queries in particular.

In our theoretical framework, we wish to capture both cases. Therefore, we now propose a unifying definition of pruning functions by which our learning algorithm will be parameterized.

Definition 6 *A pruning function with schema D is a partial function p whose domain $dom(p)$ is a subset of positively annotated trees $t * \beta$ with $t \in \mathcal{L}(D)$ such that:*

($\mathbf{P}_1$) *every annotated tree $t * \beta \in dom(p)$ is a D-instance of $p(t * \beta)$, and*

($\mathbf{P}_2$) *for every tree $t \in \mathcal{L}(D)$ with node ν, $p(t * [\nu \mapsto 1])$ is defined and preserves node ν with its label.*

Lemma 7 *For any pruning strategy σ, both p_σ and p^{can}_σ are pruning functions.*

The proof is straightforward. It should also be noticed that pruning functions need to be adapted to unranked trees, before they become suitable for our experiments on XML query induction. Any pruning function on unranked trees can be compiled back to a (more involved) pruning function on ranked trees via a binary encoding of unranked trees (see Section 8).

4.3 Stability for Pruning Functions

We next lift the notion of stability to pruning functions. Let $t * \beta$ be an annotated example for query Q with schema D and p a pruning function with the same schema. We call t_1 a (β, p)*-variant of a tree t* if t_1 is a D- completion of the unique tree t'_1 such that $t'_1 * \beta' = p(t * \beta)$ for some β'. Note that β provides positive annotations only given that p is a pruning function. Furthermore, β' must be the restriction of β to $nodes_\Sigma(t_1)$. The nodes of $dom(\beta')$ are called determined by the (β, p)-variant t_1 of t.

Definition 8 *Let D be a deterministic tree automaton, p be a pruning function and Q be a query both with schema D. We say that Q is p-stable if for any annotated example $t * \beta$ for Q in $dom(p)$, and any node ν determined by any (β, p)-variant t_1 of t satisfies $\nu \in Q(t_1)$.*

Proposition 9 *Let σ be a pruning strategy and Q a query with the same schema D, then:*

$$Q \text{ is } \sigma\text{-stable} \Leftrightarrow Q \text{ is } p_\sigma\text{-stable.}$$

Proof "$\Rightarrow$". Let $t * \beta \in dom(p_\sigma)$ be an annotated example for Q and t' a (β, p_σ)-variant of t that determines ν. Since β is Q-consistent it follows that $\nu \in Q(t)$. We have to show that $\nu \in Q(t')$. By definition of p_σ, $t' = \sqcup_{\nu' \in dom(\beta)} \sigma(t, \nu')$, so that t' is a D-completion of $\sigma(t,\nu)$. The σ-stability of Q then implies that $\nu \in Q(t')$ as required.

"$\Leftarrow$". Let $\nu \in Q(t)$ and t_1 a D-completion of $\sigma(t,\nu)$. We have to show that $\nu \in Q(t_1)$. The annotation $\beta = [\nu \mapsto 1]$ is a Q-consistent for t and satisfies $\sigma(t,\nu) * \beta = p_\sigma(t * \beta)$. Hence, t_1 is a (β, p_σ)-variant of t that determines ν. The p_σ-stability of Q yields that $\nu \in Q(t_1)$. ■

4.4 Stability Characterization

We now characterize p-stable queries by languages of p-pruned positively annotated trees. Let D be a deterministic automaton and p be a pruning function with schema D. For any query Q with schema D, let $\mathcal{L}_Q$

$$\mathcal{L}_Q = \{t * \beta \mid \beta \text{ is a } Q\text{-consistent annotation of } t \in \mathcal{L}(D)\}.$$

be the set of unpruned annotated examples for Q. Note that users of a query induction system will provide elements of $\mathcal{L}_Q$ for the target query Q. Then, the set of pruned annotated examples $p(\mathcal{L}_Q)$ is the set of all p-images of positively annotated examples for Q that belong to $dom(p)$. Note that in our XML information extraction tool, example trees will contain both positive annotations and negative annotations. In a preprocessing step, we collect positively annotated examples to which the pruning function will be applied. We will also collect negatively annotated examples which must remain unpruned in contrast.

Conversely, every language of pruned annotated trees defines a query. Let L be a language of annotated trees, it defines a query with domain $\mathcal{L}(D)$ that we denote by Q_L such that for all trees t in $\mathcal{L}(D)$:

$$Q_L(t) = \{\nu \mid \exists t' * \beta \in L \text{ such that } \beta(\nu) = 1 \text{ and } t \in compl_D(t')\}.$$

The following proposition shows that stable queries can be uniquely identified by their language of pruned positively annotated trees, under the assumption that the schema is known.

Proposition 10 *Any p-stable query Q is defined by the language $L = p(\mathcal{L}_Q)$, that is $Q_L = Q$.*

Proof We assume that Q is p-stable and show that $Q_L = Q$. Both queries have the same domain $\mathcal{L}(D)$, so it is sufficient to show that, for all $t \in \mathcal{L}(D)$ that $Q_L(t) = Q(t)$.

"$\supseteq$" Assume $\nu \in Q(t)$. Then $\beta = [\nu \mapsto 1]$ is a Q-consistent annotation of t, so that by $(\mathtt{P}_2)$, the tree $t * \beta$ can be pruned by p while preserving ν with its label. By definition of $L = p(\mathcal{L}_Q)$, we have $p(t * \beta) \in L$. Let $t' * \beta'$ be equal to $p(t * \beta)$. Since ν is the only node ν in the domain of β and is preserved, we can deduce that $\beta' = \beta$, so that $t' * \beta \in L$. Furthermore, $t \in compl_D(t')$ as a consequence of condition $(\mathtt{P}_1)$ on pruning functions. Therefore, $\nu \in Q_L(t)$ by definition of Q_L.

"$\subseteq$" Assume $\nu \in Q_L(t)$. Then, by definition of Q_L, there exists $t' * \beta \in L$ such that $t \in compl_D(t')$ and $\beta(\nu) = 1$. Now, by definition of $L = p(\mathcal{L}_Q)$, there exists an annotated example $t_1 * \beta_1$ for Q such that $t' * \beta = p(t_1 * \beta_1)$. We have $\beta_1(\nu) = 1$ by condition $(\mathtt{P}_1)$ in the definition of pruning functions, thus $\nu \in Q(t_1)$ by definition of Q-consistency. Finally, notice that t is a (β_1, p)-variant of t_1 that determines ν. Therefore $\nu \in Q(t)$ follows from p-stability. ■

Theorem 11 *Let D be a deterministic tree automaton, p a pruning function and Q a query both with schema D. The following two properties are then equivalent:*

1. *Query Q is p-stable.*
2. *Language $L = p(\mathcal{L}_Q)$ defines query Q in that $Q = Q_L$.*

Proof The implication "1 $\Rightarrow$ 2" was shown by Proposition 10, so it remains to prove "2 $\Rightarrow$ 1". We assume $Q_L = Q$ and show that Q is p-stable. Let $t * \beta$ be a Q-consistently annotated tree in $dom(p)$ and t_1 a (β, p)-variant of t determining ν. Variant t_1 is then a D-completion of t'_1 where $t'_1 * \beta' = p(t * \beta)$ for some β'. The Q-consistency of β on t implies that $t'_1 * \beta' \in L$. Furthermore, it implies that $\nu \in Q(t)$ since β must be all positive by definition of pruning functions. Since variant t_1 determines ν, it follows that $\beta'(\nu) = 1$. Hence $\nu \in Q_L(t_1)$ by definition of Q_L. With our assumption $Q_L = Q$, we can conclude that $\nu \in Q(t_1)$ as required. ■

5. Regularity

From the view point of XML information extraction, XPath queries with DTD schema restrictions are of highest interest for query induction. Modulo binary encoding of unranked trees, DTD schemas can be expressed by deterministic tree automata. Furthermore, since we ignore data values, XPath queries can be defined by tree automata too, that operate on binary encodings of unranked trees. This motivates our study of the class of regular queries for query induction.

In our learning algorithm we will represent regular p-stable queries by tree automata that recognize the language $p(\mathcal{L}_Q)$ of pruned positively annotated examples for Q. The objective of this section is to show that every regular p-stable query can be represented in this manner under the assumption that the pruning function p is regular too, a notion that we will introduce.

5.1 Regular Stable Queries

We recall a definition of regular queries and show how to represent stable regular queries by regular languages of pruned annotated examples.

Definition 12 *A query Q is regular if the set $\mathcal{L}_Q$ of unpruned annotated examples for Q is a regular tree language.*

As before, let D be a fixed deterministic tree automaton with signature Σ and state set X, so that pruned annotated tree have the signature $\Sigma \cup (\Sigma \times \mathbb{B}) \cup X$. Given a tree automaton A that recognizes pruned annotated trees, we define the query Q_A with schema D by $Q_{\mathcal{L}(A)}$. Recall that our notation leaves the dependence of D implicit.

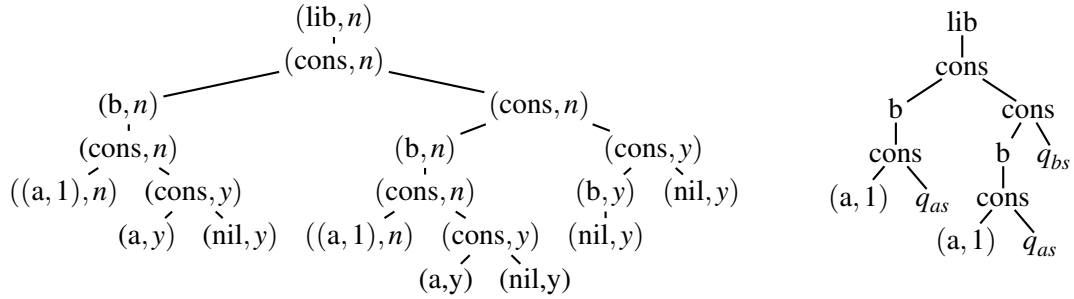


Figure 10: The unpruned annotated tree of Figure 9 with nodes annotated with the symbols y and n according to "path-only" pruning functions is on the left. Its pruning according to pruning function $p_{path\text{-}only_{\text{Lib}}}$ is shown on the right.

Query answering for regular queries is an important task for every interactive query induction system. Fortunately, queries defined by tree automata can be answered efficiently. Indeed, for every tree t in $\mathcal{L}(D)$, the set $Q_A(t)$ of query answers can be computed in time $O(|A|\ |D|\ |t|)$ even without determinism. This result is folklore. It can be shown, for instance, by converting (A,D) into a monadic Datalog program of size $O(|A|\ |D|)$ that defines the same query and applying the algorithms for monadic Datalog from Gottlob and Koch (2002).

Lemma 13 *A query Q with schema D is regular if and only if $Q = Q_L$ for some regular language L of D-pruned annotated trees.*

Proof For the one direction, note that $Q_{\mathcal{L}_Q} = Q$, so that we can choose $L = \mathcal{L}_Q$ if $\mathcal{L}_Q$ is regular. The other direction is more tedious. Given automata A and D we have to construct an automaton recognizing $\mathcal{L}_{Q_A}$ depending on D. How this can be done is shown in Appendix A. ■

It should be noticed that the automaton construction in the above proof requires exponential time in the size of A. Fortunately, this construction will only be used to clarify the expressiveness of our query representation formalism in Theorem 18. It will not be used by our query induction algorithm.

Proposition 14 *A p-stable query Q is regular if its language $p(\mathcal{L}_Q)$ is regular.*

Proof Let Q be a p-stable query and $L = p(\mathcal{L}_Q)$. Proposition 10 shows that $Q = Q_L$. Lemma 13 thus implies that Q_L is regular, under the assumption that L is regular. ■

The converse holds only for regular pruning functions introduced next (see Theorem 18).

5.2 Regular Pruning Functions

We will also need a formalism for specifying pruning functions. Again, we will use tree automata for this purpose, leading to the notion of regular pruning functions.

Example 4 *We reconsider the example on pruned libraries for the pruning function obtained from the path-only$_{Lib}$ strategy. All maximal subtrees in which no node is annotated by* 1 *must be pruned and all nodes above nodes annotated by* 1 *must be preserved. This can be done by a finite state machine. For this, we will annotate nodes that are to be pruned by y and nodes that must be preserved by n. For instance, the annotation of the tree of Figure 9 is shown in Figure 10. Then, it is easy to define a tree automaton which checks whether the definition of a "path-only" pruning function is satisfied. This automaton can also be used to apply the pruning function.*

Formally, let us consider a pruning function p and an annotated tree $t * \beta$ in its domain. We call a node ν of $t * \beta$ unpruned if it belongs to $nodes_{\Sigma \cup \Sigma \times \mathbb{B}}(p(t * \beta))$ and pruned otherwise. Let $t * \beta * p$ be the tree obtained from $t * \beta$ by annotating all nodes that were pruned by p with y and all others by n. This way we can identify p with the language $\mathcal{L}_p = \{t * \beta * p \mid t * \beta \in dom(p)\}$. Note that $\mathcal{L}_p$ contains unpruned trees only. We say that a tree automaton P with signature $(\Sigma \cup (\Sigma \times \mathbb{B})) \times \{y,n\}$ defines a pruning function p with schema D if $\mathcal{L}_p = \{s \in \mathcal{L}(P) \mid \Pi_\Sigma(s) \in \mathcal{L}(D)\}$. Note that schema validation for D is considered as an external issue for P. This pruning function p is then denoted by $\wp_{P,D}$.

Definition 15 *A pruning function p with schema D is called* regular *if it is equal to $\wp_{P,D}$ for some tree automaton P.*

For instance, the pruning function $p_{path\text{-}only_D}$ is regular. It can be defined by the same automaton with 2 states for all D (since schema validation is done externally). This automaton checks whether a node has a 1-annotated descendant (state 1) or not (state 0). Nodes in state 1 must be labeled by n and nodes in state 0 by y. Both states are final. For all symbols $f^{(k)}$ in Σ where $k \geq 0$ we define the following rules, where each occurrence of symbol $*$ stands for either state 0 and 1.

$$(f,y)(0,\ldots,0) \to 0 \quad ((f,1),n)(*,\ldots,*) \to 1 \quad (f,n)(*,\ldots,*,1,*,\ldots*) \to 1.$$

Pruning function $p_{path\text{-}ext_D}$ is also regular. It is sufficient to add a further state to the previous automaton, which checks whether a node is a sibling of a positively annotated node (or not). We leave the precise automaton construction to the reader.

Any pair (P,D) can be transformed in polynomial time into a linear bottom-up tree transducer (Comon et al., 2007) that defines the same pruning function $\wp_{P,D}$. Such transducers, however, may become nondeterministic even if P and D are deterministic, as for instance for $p_{path\text{-}ext_D}$. Furthermore, if we avoid such a conversion, we can indeed evaluate the application of a pruning functions to an annotated tree efficiently, as we show next.

Lemma 16 *Let P and D be deterministic tree automata, $p = \wp_{P,D}$, and $t * \beta$ an unpruned annotated tree. One can decide whether $t * \beta$ belongs to $dom(p)$ and in this case compute $p(t * \beta)$ in time $O(|P|\ |t| + |D| + |t|)$.*

Proof We compute the projection automaton $P' = \Pi_{\Sigma \cup (\Sigma \times \mathbb{B})}(P)$ by "integrating" the labels y and n into the states of P. Note that P' may be nondeterministic despite of the determinism of P. Let $t * \beta$ be an unpruned annotated input tree. Since D is deterministic we can decide in time $O(|D| + |t|)$ whether $t \in \mathcal{L}(D)$. If not, we return that $t * \beta$ does not belong to $dom(p)$. Otherwise, there exists at most one unpruned tree $s \in \mathcal{L}(P)$ such that $\Pi_{\Sigma \cup \Sigma \times \mathbb{B}}(s) = t * \beta$, since p is a partial function

satisfying $\mathcal{L}_p = \{s \in \mathcal{L}(P) \mid \Pi_\Sigma(s) \in \mathcal{L}(D)\}$. Therefore, and since P is deterministic, P' may have at most one successful run on $t * \beta$ and possibly many unsuccessful runs. This unique successful run would contain all information on whether a node is to be pruned or not in the $\{y,n\}$ component of its state. It is thus sufficient to compute the successful run of P' on $t * \beta$ if it exists and to detect is nonexistence otherwise. This can be done by running the nondeterministic automaton P' on t in time $O(|P'|\ |t|)$ in a bottom up phase, then testing whether a final state got reached, and if yes selecting the successful run in a top-down phase. Since P' contains at most twice as many states as P, its size is linear in that that of P. The overall computation time is thus in $O(|P|\ |t| + |D| + |t|)$. ■

Lemma 17 *Let D, A, and P be deterministic tree automata. If $L = \mathcal{L}(A)$ is a language of unpruned annotated trees and $p = \wp_{P,D}$ a pruning function then $p(L)$ can be recognized by a deterministic tree automaton of size $O(|D|\ 2^{|P|}\ |A|)$.*

Proof Let $p = \wp_{P,D}$ and $L = \mathcal{L}(A)$ for deterministic tree automata P, D, A. We rely on a similar algorithm as for computing $p(t * \beta)$ from $t * \beta$ except that we must deal with states from X in pruned trees $p(t * \beta)$. So we use the projection automaton $P' = \Pi_{\Sigma \cup (\Sigma \times \mathbb{B})}(P)$ but we add rules $q \to (r,y)$ for all states $q \in X$ and r of P. Then, we determinize automaton P', lift D to the automaton D' on $\Sigma \cup (\Sigma \times \mathbb{B}) \cup X$ that runs D on the Σ component of any pruned input trees, while adding rules $q \to q$ for all $q \in X$. We note that the determinism of D inherits to D'. We then compute the product $P' \cap D' \cap A$. By running this product on a pruned input tree $t * \beta$, we can test whether there exists a D-completion $t' * \beta'$ such that $t' \in \mathcal{L}(D)$ and $p(t' * \beta') = t * \beta$. The product automaton is deterministic, recognizes $p(L)$, and is of size $O(|D|\ 2^{|P|}\ |A|)$. ■

Theorem 18 *Let Q be a p-stable query for a regular pruning function p. Then:*

$$Q \text{ is regular} \Leftrightarrow p(\mathcal{L}_Q) \text{ is regular.}$$

Proof If Q is regular then $\mathcal{L}_Q$ is regular, and thus $p(\mathcal{L}_Q)$ by Lemma 17. The converse implication was shown in Proposition 14. ■

This shows that every regular p-stable query Q can be represented by a (minimal deterministic) tree automaton that recognizes the language $p(\mathcal{L}_Q)$ of p-pruned positively annotated examples for Q. This is the representation on which we will base our learning algorithm for regular p-stable queries.

5.3 Deciding Stability

For our experimental validation, it will be necessary to decide whether a regular query is stable for a regular pruning function. This can be done in polynomial time:

Theorem 19 *Let D and P be deterministic automata defining a regular pruning function $p = \wp_{P,D}$ and Q a query with domain D. For any tree automaton A recognizing $\mathcal{L}_Q$ we can decide in time $O(|A|^2\ |D|\ |P|)$ whether Q is p-stable.*

The quite technical proof which is based on so called recognizable relations between trees (which is based on the idea of regular language of overlays of tree tuples, see, for example, Comon et al., 2007) is deferred to Appendix A.

6. Algorithms for Consistency Checking

In this section, we will present algorithms for checking consistency properties for regular sets of annotated examples that we will introduce: D-functionality and p-consistency. These consistency checks will be the prime subroutines of our learning algorithm. They help us to avoid the need for further negative examples for the target language, beside the negatively annotated examples for the target query.

6.1 D-Functionality

We consider languages of pruned annotated trees with mixed positive and negative annotations. D-functionality is a consistency notion for such sets which requires that positive and negative annotations are non-contradictory.

Definition 20 *We call a language $L \subseteq \mathcal{T}_{\Sigma\cup(\Sigma\times\mathbb{B})}(X)$ of annotated trees* D-functional *if for any $t*\beta, t'*\beta' \in L$, if t and t' are compatible with respect to D then β and β' coincide on $dom(\beta)\cap dom(\beta')$.*

The language $\{(a,1)(\top),(a,0)(\top)\}$, for instance, is not $\mathcal{U}$-functional. We next relate p-stable queries to D-functional languages.

Proposition 21 *If Q is a p-stable query with schema D, then the language $p(\mathcal{L}_Q)\cup\mathcal{L}_Q$ is D-functional.*

Proof The proof is by contradiction. Let us assume that $p(\mathcal{L}_Q)\cup\mathcal{L}_Q$ is not D-functional. Since domains of pruning functions contain positively annotated trees, there must exist a p-pruned example $t'_1*\beta'_1 \in p(\mathcal{L}_Q)$ for Q and an unpruned example $t_2*\beta_2 \in \mathcal{L}_Q$ for Q such that t'_1 and t_2 are compatible with respect to D, so that there exists a node ν with $\beta'_1(\nu)=1$ and $\beta_2(\nu)=0$. By definition of $p(\mathcal{L}_Q)$, there exists an unpruned example $t_1*\beta_1$ for Q such that $t'_1*\beta'_1 = p(t_1*\beta_1)$. Property $(\mathbb{P}_1)$ of pruning functions implies that $\beta_1(\nu)=\beta'_1(\nu)=1$. Since β_1 is a Q-consistent annotation of t_1, it follows that $\nu\in Q(t_1)$, and since β_2 is a Q-consistent annotation of t_2 that $\nu\notin Q(t_2)$. Compatibility implies that t_2 is a D-completion of t'_1. Hence, t_2 is a (β_1,p)-variant of t_1 that determines ν. The p-stability of Q yields $\nu\in Q(t_2)$. Contradiction. ■

We now show that D-functionality can be tested efficiently for regular queries Q_A, by reduction to testing emptiness of tree automata. Note that determinism does not help to improve emptiness tests, so that it will not help to assume that A is deterministic here.

Theorem 22 *Let D be a deterministic tree automaton with signature Σ and state set X, and A be a tree automaton with signature $\Sigma\cup(\Sigma\times\mathbb{B})\cup X$. Whether $\mathcal{L}(A)$ is D-functional can be decided in time $O(|D|\,|A|^2)$.*

Proof We write $s_1 \circledast s_2$ for the overlay of trees s_1 and s_2, where missing nodes are filled up with a fresh $\perp$-symbol. Let us consider the language L_{contra} of overlays $(t_1*\beta_1)\circledast(t_2*\beta_2)$ such that $t_1*\beta_1$ and $t_2*\beta_2$ are accepted by A, that t_1 and t_2 have a common completion, that is, $compl_D(t_1)\cap compl_D(t_2)\neq\emptyset$, and that there exists a node ν with contradicting annotations $\beta_1(\nu)\neq\beta_2(\nu)$. By definition of D-functionality, $\mathcal{L}(A)$ is D-functional if and only if $L_{contra}=\emptyset$.

The existence of a common D-completion can be checked by running a single copy of D jointly on t_1 and t_2, while checking that the states of D occurring in t_1 and t_2 are chosen appropriately.

More precisely, if a node in $t_1 \circledast t_2$ is labeled by $(f, \perp)$ then D runs on the left component and if it is labeled with $(\perp, f)$ then it is run on the right component. Nodes with labels (f,q), (q,f) are correct, if the state reached by D is equal to q. From this time point on, D continues on both components while testing their equality. Labels (q_1,q_2) with different state contradicts the compatibility of t_1 and t_2 (since D is deterministic), as well as labels (f_1,f_2) with different function symbols. We have also to test that the joined run of D on t_1 and t_2 does reach a final state.

Membership of $t_1 * \beta_1$ and $t_2 * \beta_2$ to $\mathcal{L}(A)$ can be checked by running two copies of A in parallel on their overlay. Last but not least, the existence of a contradiction between annotations can be checked by a very simple automaton with a single state. Thus, L_{contra} can be recognized by a product automaton of size $O(|D|\ |A|^2)$. As emptiness is testable in linear time for tree automata, the theorem follows. ■

Note that if one can assume that the Σ-projections of all trees recognized by A satisfy schema D, as we can do for our learning algorithm, then the test in the proof can be simplified, since never D has to be run in parallel with both copies of A. In this case, D-functionality can be checked in time $O(|D|\ |A| + |A|^2)$.

6.2 p-Consistency

In our learning algorithms, we will need to check whether a language of annotated trees contains only p-pruned trees for a given regular pruning function p.

Definition 23 *We call a language L of annotated trees* p-consistent *if all trees in L are p-pruned, that is, if $L \subseteq p(\mathcal{T}_{\Sigma\cup(\Sigma\times\mathbb{B})})$.*

Proposition 24 *Let $p = \wp_{P,D}$ be a regular pruning function, let $L = \mathcal{L}(A)$ be a regular set of pruned annotated trees defined by a deterministic automaton with signature $\Sigma \cup (\Sigma \times \mathbb{B}) \cup X$. Whether $\mathcal{L}(A)$ is p-consistent can be decided in time $O(|A|\ |D|\ 2^{|P|})$.*

Proof By definition of p-consistency, we must test whether $\mathcal{L}(A) \subseteq p(\mathcal{T}_{\Sigma\cup(\Sigma\times\mathbb{B})})$. By Lemma 13, we can construct a deterministic tree automaton recognizing $p(\mathcal{T}_{\Sigma\cup(\Sigma\times\mathbb{B})})$ in time $O(|D|\ 2^{|P|})$. We can thus decide p-consistency by inclusion checking. According to Proposition 1, inclusion can be checked in time $O(|A|\ |D|\ 2^{|P|})$. ■

The p-consistency test may lead to an exponential blow-up in the size of P due to determinization of the projection of P. In practice this does not raise any problems. The first reason is that the usual determinization procedure does often behaves much better than in the worst case. The second reason is that the automaton P defining the pruning function will usually be very small. For instance, "path-only" pruning functions can be defined by an automaton with 2-states (indeed the same automaton for all schemas D), and "path-extended" pruning functions with 3-states. Pruning functions derived from the unranked case via the binary encoding will be defined with no more than 5-states.

7. Learning Stable Regular Queries

In this section, we present a learning algorithm for stable queries and prove a formal learnability result.

7.1 Learning from p-Pruned Samples

We present a learning algorithm that infers p-stable queries from p-pruned samples and show that it satisfies the learning model from polynomial time and data. The nontrivial aspect is that p-pruned samples are indeed sufficient.

We suppose that the schema is fixed by a deterministic automaton D, and that the pruning function $p = \wp_{P,D}$ is fixed by a deterministic tree automaton P. As shown before, a p-stable regular query Q is uniquely defined by the language $L = p(\mathcal{L}_Q)$ of all p-pruned examples for Q. The idea is therefore to identify the minimal deterministic tree automaton for the language $L = p(\mathcal{L}_Q)$ associated with the p-stable target query Q. As input, it will receive a p-pruned sample for the target query.

Definition 25 *A p-pruned sample is a pair (S^+, S^-) where S^+ is a p-consistent finite set of positively annotated trees and S^- a finite set of negatively annotated trees such that $S^+ \cup S^-$ is D-functional.*

Note that only positively annotated examples can be p-pruned, since pruning functions do not apply to examples with negative annotations. We can now state our main result of the learnability of p-stable queries from p-pruned samples.

Theorem 26 *Let schema D be a deterministic tree automaton and p be a fixed regular pruning function with schema D. Let $\mathcal{A}$ be the class of deterministic tree automata that recognize languages $p(\mathcal{L}_Q)$ of some regular p-stable query Q and $\mathcal{S}$ the class of p-pruned samples of annotated examples. Then, the class of p-stable regular queries represented by automata in class $\mathcal{A}$ is learnable in polynomial time with polynomially many examples from samples in the class $\mathcal{S}$. I.e. any automaton A in $\mathcal{A}$ has a characteristic sample $char(A)$ in $\mathcal{S}$ whose cardinality is polynomial in the size of A and there is an algorithm p-stable-$\mathcal{RPNI}_D$ such that, for any sample S' in $\mathcal{S}$ that subsumes the characteristic sample $char(A)$ of an automaton A recognizing the language $L = p(\mathcal{L}_Q)$ of a p-stable query Q, algorithm p-stable-$\mathcal{RPNI}_D$ outputs in polynomial time in the size of S' the unique minimal deterministic tree automaton recognizing L.*

Proof The proof is by reduction to the problem of learning regular tree languages represented by deterministic tree automata from positive and negative examples (García and Oncina, 1993). Let $\mathcal{A}'$ be the class of deterministic tree automata with signature $\Sigma' = \Sigma \cup (\Sigma \times \mathbb{B}) \cup X$ and $\mathcal{S}'$ the class of samples (S'^+, S'^-) of positive and negative examples $S'^+, S'^- \subseteq \mathcal{T}_{\Sigma'}$ with $S'^+ \cap S'^- = \emptyset$. From the learnability result for deterministic tree automata, we know that every automaton A in $\mathcal{A}'$ has a characteristic sample $char'(A) = (S'^+, S'^-)$ in $\mathcal{S}'$ with $S'^+ \subseteq \mathcal{L}(A)$ and $S'^- \cap \mathcal{L}(A) = \emptyset$ whose cardinality is polynomial in the size of A. Also, there is an algorithm $\mathcal{RPNI}$ such that, for every sample S in $\mathcal{S}'$ that subsumes the characteristic sample $char'(A)$, $\mathcal{RPNI}$ with input S outputs in polynomial time in the size of S the unique minimal deterministic tree automaton recognizing $\mathcal{L}(A)$.

In the first step of the proof, we construct the characteristic sample $char(A) = (S^+, S^-)$ in $\mathcal{S}$ for A in $\mathcal{A}$. We fix a total order on trees over Σ, such that trees with fewer nodes become smaller. Let A be a deterministic tree automaton recognizing the language $L = p(\mathcal{L}_Q)$ of the p-stable target query Q. Since automaton A belongs to class $\mathcal{A}'$, there exists a characteristic sample $char'(A) = (S'^+, S'^-)$

```
fun p-stable-RPNI_D(S+,S−) // input a p–pruned sample (S+,S−)
  A ← init(S+) // deterministic tree automaton recognizing S+ such
               // that at most one tree is recognized per state
  let (q1,...,qn) = sort states of A consistently with order on trees
  Ok ← ∅ // states already merged
  for i=1 to n do
    if qi ∈ Ok then skip else
      for j=1 to i−1 do
        let A' = det-merge(A,qi,qj) // qi becomes a reference to qj
        if L(A') ∪ S− is D–functional // Theorem 22
           and L(A') is p–consistent // Proposition 24
        then
          A ← A'
          update Ok by adding newly merged states from computing A'
          exit // inner for loop
        else
          skip
    end // inner for loop
    add qi to Ok // if qi got merged then it belonged already to Ok
  end // outer for loop
  output A // a deterministic tree automaton defining query QA
```

Figure 11: Learning algorithm for p-stable queries from p-pruned examples.

in $\mathcal{S}'$. We define $char(A)$ in $\mathcal{S}$ from $char'(A)$ as follows. All trees in S'^{+} belong to $L = p(\mathcal{L}_Q)$, so we put them into S^+. For all trees s in S'^{-} we proceed as follows:

- If s does not belong to the image of p then we can safely ignore s, since our algorithm will always check that the language of the target automaton will be p-consistent (recall that p is fixed and known by the algorithm).

- Otherwise, let t be the least tree in the total order fixed above such that $s = p(t * \beta)$ for some β, and fix one of those βs. By definition of pruning functions, β must be a positive annotation. Since $s \in S'^{-}$, however, β cannot be consistent with Q, so there exists a node $\nu \notin Q(t)$ such that $\beta(t) = 1$. We add $t * [\nu \mapsto 0]$ to S^-.

We obtain a sample $char(A) = (S^+, S^-)$ whose cardinality is at most linear in the cardinality of $char'(A)$, and thus polynomial in the size of A. Moreover, it follows from Proposition 21 that $S^+ \cup S^-$ is D-functional since Q is p-stable. Therefore, $char(A)$ is a p-pruned sample in the class $\mathcal{S}$. In Figure 11, we define our learning algorithm p-*stable*-$\mathcal{RPNI}_D$ which receives p-pruned samples (S^+, S^-) in $\mathcal{S}$ as input. It is parameterized by a deterministic tree automaton D and a regular pruning function p with schema D. It remains to show for every sample (S^+, S^-) in $\mathcal{S}$ that subsumes the characteristic sample $char(A)$ of an automaton A recognizing the language $L = p(\mathcal{L}_Q)$ of a p-stable query Q, that p-*stable*-$\mathcal{RPNI}_D(S^+, S^-)$ is the unique minimal deterministic tree automaton recognizing L and that it is computed in polynomial time.

For this, let $A \in \mathcal{A}$ be an automaton. We consider $char(A)$ as defined before from the characteristic sample $char'(A)$. We first show that:

$$p\text{-}stable\text{-}\mathcal{RPNI}_D(char(A)) = \mathcal{RPNI}(char'(A)).$$

Before showing this, we recall the principles of $\mathcal{RPNI}$ and explain the differences to p-*stable*-$\mathcal{RPNI}_D$. Algorithm $\mathcal{RPNI}$ receives as input a finite sample of positive and negative examples for the target language. At the beginning it computes an initial automaton $init(S^+)$ recognizing the positive examples S^+. Each of its states can be associated with the least subtree that is evaluated to this state. Thereby, states become totally ordered, as well as pairs of states. State merge operations preserving determinism are then tried out in this order. A state merge operation is accepted if it preserves consistency in that no negative example can be recognized. Otherwise, it is rejected and must be undone. This procedure is repeated exhaustively. Algorithm p-*stable*-$\mathcal{RPNI}_D$ is similar except that it receives a p-pruned sample as input, and that the consistency test is replaced by a test for D-functionality and p-consistency.

Let $char(A) = (S^+, S^-)$ be the p-pruned sample constructed from $char'(A) = (S'^+, S'^-)$ as described above. By construction, $S'^+ \subseteq S^+$. Therefore, it follows from the learnability result for deterministic tree automata that $\mathcal{RPNI}(S^+, S'^-) = \mathcal{RPNI}(char'(A))$ is the unique minimal deterministic automaton recognizing $\mathcal{L}(A)$. It remains to show that p-*stable*-$\mathcal{RPNI}_D(S^+, S^-) = \mathcal{RPNI}(S^+, S'^-)$. Both algorithms start by constructing the initial automaton $init(S^+)$. Therefore, along their computations, both algorithms will try to perform the same state merge operations in the same order. Successful attempts will be accepted, while all others with be rejected and undone. It remains to show that both algorithms accept the same state merge operations. A merge operation is rejected by $\mathcal{RPNI}$ if the automaton obtained recognizes some tree $s \in S'^-$. This translates either to a lack of p-consistency of the current automaton A reached by evaluating p-*stable*-$\mathcal{RPNI}_D$ or to a lack of D-functionality of $\mathcal{L}(A) \cup S^-$. Conversely, if p-consistency or D-functionality fail for the current automaton of p-*stable*-$\mathcal{RPNI}_D$, then the computation of $\mathcal{RPNI}$ cannot accept this merge operation either, since otherwise its final automaton would accept a larger language than the target language. Thus, p-*stable*-$\mathcal{RPNI}_D(S^+, S^-) = \mathcal{RPNI}(S^+, S'^-)$.

The computation time for $\mathcal{RPNI}$ is polynomial in the size of the input sample. The additional tests for D-functionality and p-consistency performed by p-*stable*-$\mathcal{RPNI}_D$ are in polynomial time (for fixed p) by Theorem 22 and Proposition 24. Thus, the overall computation time of p-*stable*-$\mathcal{RPNI}_D$ is polynomial in the size of the input sample. It can also be proven that if p-*stable*-$\mathcal{RPNI}_D$ receives a superset of $char(A)$ as input, then the output is again the minimal deterministic automaton recognizing the language $L = p(\mathcal{L}_Q)$ of the target p-stable query Q. ■

As shown above, automaton p-*stable*-$\mathcal{RPNI}_D(S^+, S^-)$ can be computed in polynomial time depending on the size of the input sample (S^+, S^-), for fixed regular pruning function $p = \wp_{P,D}$. This may change if moving the deterministic automaton P to the inputs, since then there may be an exponential blow up in the size of P in the worst case (see Proposition 24). Therefore, the choice of the pruning function requires a little care: they should be defined by automata with few states only.

The performance of the inclusion test for checking p-consistency is crucial for practical efficiency, since it is performed repeatedly at every merge attempt of learning algorithm p-*stable*-$\mathcal{RPNI}_D$. Therefore, we have implemented our inclusion test such that it is incremental with respect to the addition of rules to automaton A defining the query hypothesis, see Champavère et al. (2009).

We also designed and implemented an alternative algorithm, where schema consistency is ensured by static state-typing, as in previous inference algorithms for regular languages (Coste et al., 2004; Oncina and Varó, 1996). The experimental results were not convincing though: the algorithm worked well only for queries whose automata share much of their "structure" with the schema,

```
fun σ-stable-learn_D(S)  // input a sample of unpruned annotated trees
                         // that is functional and consistent with schema D
  let S⁺ = p_σ(S^=1)     // remove 0–annotations in S and apply p
  let S⁻ = S^=0          // remove 1–annotations in S
  if S⁺ ∪ S⁻ is not D–functional then raise exception 'unstable query'
     // if no exception is raised then (S⁺,S⁻) is a p–pruned sample
  output p_σ-stable-RPNI_D(S⁺,S⁻)
```

Figure 12: Learning algorithm for σ-stable queries from unpruned examples.

which is rarely the case in our applications. Furthermore, state-typing algorithms are not yet well-founded theoretically, that is, no result on learning in the limit exists.

7.2 Learning from Unpruned Samples

In our experiments, a user annotates unpruned samples for the target query, which may or may not be stable for a given pruning strategy σ. We next discuss how to produce p_σ-pruned samples thereof, in order to obtain a learning algorithm for σ-stable queries from unpruned examples.

Definition 27 *An* unpruned sample for a schema D *is a functional tree language* $L \subseteq \mathcal{T}_{\Sigma\cup(\Sigma\times\mathbb{B})}$ *that is consistent with* D*, that is,* $\Pi_\Sigma(L) \subseteq \mathcal{L}(D)$.[2]

This algorithm which we call σ-*stable-learn*$_D$ is presented in Figure 12. Given an unpruned sample S for schema D, it computes a p_σ-pruned sample (S^+,S^-) if possible. The positive part S^+ is obtained by removing all negative annotations from trees in S and then applying p_σ. The negative part S^- is obtained from S by removing all positive annotations. If S is a sample for a σ-stable query, then $S^+ \cup S^-$ must be D-functional as shown by Propositions 9 and 21, so that (S^+,S^-) is indeed a p-pruned sample. In this case, p_σ-*stable-RPNI*$_D$ can be safely applied. Otherwise, the algorithm raises an exception.[3]

For instance, pruning strategy $\sigma = $ *path-only* with the universal schema $D = \mathcal{U}$, and query Q which selects all a's whose left-sibling is labeled by b. The set $S = \{f(b,(a,1)), f(a,(a,0))\}$ is an unpruned sample for Q. It induces the p_σ-pruned sample (S^+,S^-) where $S^+ = \{f(\top,(a,1))\}$ and $S^- = \{f(a,(a,0))\}$. Clearly, $S^+ \cup S^-$ is not $\mathcal{U}$-functional since Q is not σ-stable, and evaluating σ-*stable-learn*$_D(S)$ will raise exception 'unstable query'.

7.3 Selection of a Pruning Strategy

The choice of a pruning strategy has strong implications on the class of learnable queries. The more aggressive the underlying pruning strategy is, the smaller will be the class of stable queries, and thus the class of learnable queries. For more aggressive pruning strategies, however, our learning algorithm will converge more quickly. So the question is how to find appropriate pruning strategies.

We can approach the problem as follows. First we fix a finite set of pruning strategies, containing for instance *path-only*$_D$, *path-ext*$_D$, *path-only*, and *path-ext* as in our experiments. Given a sample

2. A language of unpruned annotated trees is functional if and only if it is $\mathcal{U}$-functional.
3. The usage of canonical pruning functions p_σ^{can} requires some care since p_σ^{can} might be undefined for some trees in $S^{=1}$. Instead, one can add all trees $p(t * [\nu \mapsto 1])$ to S^+ such that there exists $t * \beta \in S$ with $\beta(\nu) = 1$. This approach, however, requires some algorithmic improvements to become competitive.

S of unpruned trees, the idea is to choose the most aggressive of these pruning strategies, so that the evaluation of p_σ-*stable-learn*$_D(S)$ does not raise exception 'unstable query'. The inferred query that as well as the pruning strategy that is selected may still be unsatisfactory if the sample S is not sufficiently informative. In this case, a larger sample must be provided, so that the pruning strategy can be updated accordingly and the correct query can be found.

8. Stable Regular Queries for Unranked Trees

So far, all notions have been defined for ranked trees. But our goal is to learn XML queries over XML trees. Thus, all notions and results must be lifted to the unranked case and this is the objective of the section.

For this, we use a binary encoding. We choose the bottom-up binary encoding from Carme et al. (2004) that is also known as currying from the lambda-calculus. The advantage of using the bottom-up encoding, rather than the more frequently used top-down encoding based on first-child and next-sibling relations, is that schemas can be defined by bottom-up deterministic tree automata (rather than the less expressive top-down deterministic tree automata). For instance, any deterministic DTD for unranked trees can be transformed in polynomial time into a bottom-up deterministic tree automaton for curried binary encodings. See for instance Champavère et al. (2009) for a precise complexity analysis of the translation and further details.

Example 5 *Let us introduce currying by example. We reconsider the unranked library tree u with three books (b), of which the first two have two authors (a), and the last one none:*

$$u = lib(b(a,a),b(a,a),b).$$

Its curried encoding is as follows where the binary "application" operator @ is written in infix notation, while missing parenthesis are to be added from left to right.

$$curry(u) = lib@(b@a@a)@(b@a@a)@b.$$

This way, function *curry* maps unranked trees over a label set Σ are encoded into binary trees over the ranked alphabet $\Sigma_@$ containing the binary special function symbol $@^{(2)}$ and constants for all symbols of Σ. Function *curry* is one to one and onto, that is, every binary tree over $\Sigma_@$ uniquely defines an unranked tree over Σ. However, the relationship between nodes is a little more intricate. Every node of an unranked tree u corresponds to a unique leaf node of the ranked tree $curry(u)$ and vice versa. Inner nodes of $curry(u)$, that is, those labeled by @, do not correspond to any node of u (but to some of its edges). Node selection queries on unranked trees therefore correspond to leaf selection queries on ranked trees.

We next introduce stable regular queries and pruning strategies for unranked trees, and show how to translate them to ranked trees via a binary encoding. Let Σ be a finite label set. We denote by $\mathcal{U}_\Sigma$ the set of unranked trees over Σ. A schema will be defined as a deterministic tree automaton D with ranked signature $\Sigma_@$ and state set X. Such an automaton evaluates unranked trees by evaluating their binary encoding.

A pruned unranked tree is a unranked tree with label set $\Sigma \cup X$. By $\mathcal{U}_\Sigma(X)$ we denote the set of all pruned unranked trees. As for ranked trees, we can define a subsumption order on $\mathcal{U}_\Sigma(X)$, so that greater trees are obtained by instantiating occurrences of states by unranked trees that can be evaluated to this state by D.

An unranked pruning strategy for a schema D is a function σ that maps unranked trees $u \in \mathcal{U}_\Sigma$ with leaf nodes $\nu \in \mathit{leafs}(u)$ to pruned unranked trees $\sigma(u,\nu) \in \mathcal{U}_\Sigma(X)$, while satisfying literally conditions $(\mathtt{S}_1)$ and $(\mathtt{S}_2)$ from the ranked case (but with u instead of t). There exists a one-to-one and onto mapping between unranked and ranked pruning strategies.

We define pruning strategies $\mathit{path\text{-}only}_D$ and $\mathit{path\text{-}ext}_D$ on unranked trees as before, such that they preserve the path to the input node and respectively the extended path. The next example shows that pruning strategies that correspond on ranked trees are quite different from what one obtains when applying the ranked $\mathit{path\text{-}only}_D$ and $\mathit{path\text{-}ext}_D$ pruning strategies to binary encodings of unranked trees. This means that the automata that defining $\mathit{path\text{-}only}_D$ and $\mathit{path\text{-}ext}_D$ need to be adapted to the unranked case appropriately.

Example 6 *Let u be the unranked library from Example 5. The unranked "path-only" pruning strategy without schema restrictions applied to the first author of the second book yields:*

$$\mathit{path\text{-}ext}(u, 2\cdot 2) = \mathit{lib}(\top, b(a,\top),\top).$$

The correct corresponding path-only pruning of $\mathit{curry}(u)$ at the corresponding leaf $1\cdot 2\cdot 1\cdot 2$ is equal to:

$$\mathit{curry}(\mathit{path\text{-}ext}(u, 2\cdot 2)) = \mathit{lib}@\top@(b@a@\top)@\top.$$

In contrast, the ranked path-ext pruning of $\mathit{curry}(u)$ at this leaf yields something quite different:

$$\mathit{path\text{-}ext}(\mathit{curry}(u), 1\cdot 2\cdot 1\cdot 2) = \top@(\top@a@\top)@\top.$$

This shows that the framework with general ranked pruning strategies is needed for dealing with the unranked case properly.

The definition of σ-stable queries carries over literally to unranked trees. It follows that a node selection query on unranked trees is stable for an unranked pruning strategy if and only if the corresponding leaf selection query on ranked trees is stable for the corresponding ranked pruning strategy. We can also compare unranked pruning strategies for aggressiveness, literally as we did in the ranked case. It follows as before, that query stability inherits to less aggressive pruning strategies

An annotated unranked tree is a tree in $\mathcal{U}_{\Sigma\cup(\Sigma\times\mathbb{B})}(X)$. It should be noticed that annotated ranked trees correspond to annotated unranked trees in which only leafs are annotated, and vice versa. Pruning functions can be defined on unranked trees literally as in the ranked case. Ranked pruning functions then correspond precisely to unranked pruning functions, whose domains are restricted to leaf-only-annotated trees. Furthermore, we can lift pruning strategies to pruning functions in the unranked case in the same manner as in the ranked case. Based on these correspondences, we can also lift to unranked trees our theorem on learning stable queries from pruned examples without any particular difficulties.

9. Experimental Results

We have implemented our learning algorithm and integrated it into Web and XML information extraction systems. In this section, we present experimental results that illustrate the relevance of schema-guided pruning strategies in practice.

We will start with the presentation of an interactive query induction system and how we simulate it in order to evaluate its performance. We then show how we use our learning algorithms in the

system. Then, we compare our system with existing Web information extraction tools in order to illustrate that our algorithms are competitive. Note that no previous system makes use of the DTD of HTML. We then move to XML information extraction, for which no alternative tool exists for the same tasks to the best of our knowledge.

9.1 Interactive Query Induction

Given a set of XML documents, the goal of the user is to find an unknown target query that selects the correct set of nodes in each of them. In order to do so, the user has to play the role of a teacher who provides the learner with annotations that are consistent with the target query.

9.1.1 INTERACTIVE LEARNING PROTOCOL

The interactive learning protocol follows a classical annotate-learn-select-correct loop. We fix a schema D and as pruning strategy σ either $\textit{path-only}_D$ or $\textit{path-ext}_D$. At the beginning, the user chooses an XML document from the collection and annotates some nodes as selected or rejected. This yields an unpruned sample S on which the system runs the learning algorithm $\sigma\text{-}\textit{stable-learn}_D$. Either a non-stability exception is raised or a σ-stable query is inferred. In case of success, the system presents the answer of the induced query on the current document to the user, possibly via a graphical interface, at least in the case of HTML documents. For other kinds XML documents, there might not exist suitable visualizations, but for example as in the introduction, there are some.

If the user agrees with the query's answer on the current document then he can proceed with inspecting the answer of the same query on another document. Otherwise, he must correct a node that is wrongly selected or rejected and provide a correction by adding a negative or a positive annotation respectively. The learner learns a new query but now from a larger collection of annotated examples, and so on and so forth. The process continues until the user accepts the current query.

9.1.2 AUTOMATIC EVALUATION

In order to evaluate our learning algorithm $\sigma\text{-}\textit{stable-learn}_D$, we will simulate the user in the interactive learning protocol. We assume that the target query is known beforehand, so that we can generate annotations simulating the user's behavior automatically.

Given a totally ordered collection of documents and a pruning strategy σ with schema D, the simulated user behaves as follows in order to find the target query. He creates an empty unpruned sample at the beginning, which is enriched incrementally. He then inspects the first document. If the query does not select any node on this document then the sample remains empty. Otherwise, the first answer node of the document in a breadth-first order is annotated by 1, and the thereby annotated document is added to the sample. The learning algorithm $\sigma\text{-}\textit{stable-learn}_D$ is then run with the current sample. It returns a query that is an hypothesis for the target query. If this query returns an incorrect answer set on the current document, then the simulated user corrects the first wrongly selected or rejected node by adding a negative respectively positive annotation, and reruns the learning algorithm with the updated sample. Otherwise, he continues with the next document in the same manner, while always enriching the sample. The simulator has also to decide when the user will stop the interaction loop. We chose to stop once the current query computes correct answers on 30 consecutive documents or until the whole data collection has been processed. In order to reduce the dependency on the particular total order on the document collection, we will

generate 30 different total orders randomly, and report the average results on these 30 repetitions. The quality of the learning algorithm is measured by the following two criteria:

1. the number of annotations to be provided by the simulated user until convergence,
2. the number of XML documents that the simulated user needs to consult until convergence.

These criteria measure the annotation effort of the simulated user until convergence. Its verification effort, on whether the queries proposed by the system are correct, is ignored since considered less relevant.

Compared to a human user, our automatic evaluation procedure with a simulated user is pessimistic in two aspects. First, a human user inspects informative documents eagerly rather than using a random order, and second, he can choose informative annotations and corrections eagerly rather than in breadth first manner.

9.2 Web Information Extraction

We start with a comparison to the Web information extraction tools by Raeymaekers et al. (2008) (see also Raeymaekers, 2008) and Carme et al. (2007). The former is based on learning local tree automata for unranked trees, while the later is based on learning unrestricted deterministic tree automata for ranked trees. No schemas are considered there. Both tools rely on the pruning strategy *path-only* (see the second transformation of Raeymaekers et al., 2008, p. 170), which ignores any schema information. When learning without pruning strategies, both tools yield poor results. We do not present such experiments here but confirm equally poor results for our algorithm. Less aggressive pruning strategies than *path-only* were not tested there. This imposes important restrictions on the class of learnable queries, as we argued here. Furthermore, none of these tools can deal with schema-guided pruning strategies or benefit from the DTD of HTML, in contrast to what we do. Note that we did not try out to work with schemas that got inferred themselves from a collection of XML documents, for instance by the induction algorithm proposed by Bex et al. (2006).

Raeymaekers (2008) provides experimental comparisons to quite some Web information extraction tools based on different methods from machine learning, of which the most efficient are the string-based tools STALKER and BWI (Muslea et al., 2003). Since his tool is shown competitive with those, and ours is doing equally well for the kind of queries tested there, we skip further comparisons. Unfortunately, however, only few of his data sets are available.[4] As a work around, we rely on the data sets from Carme et al. (2007) for some more challenging cases. All of them are available online at the following URL: `http://www.grappa.univ-lille3.fr/~carme/WebWiki/DataSets.html`. Note that we had to use `tidy`[5] in order to make the HTML documents valid w.r.t. a DTD (XHTML 1.0 Transitional in our case). Indeed, this was necessary for all HTML data sets that we tested with, since none of them was consistent with any existing DTD.

The results of our algorithm *p-stable-RPNI*$_{\text{HTML}}$ where $p = p_{\textit{path-only}_{\text{HTML}}}$ are shown in Table 1. They illustrate that our algorithm is fully competitive with both predecessors considered, even though of much larger scope. Indeed, our algorithm often performs best and never needs significantly more annotations that the others. This is probably due to guidance of the pruning strategy by the schema of HTML.

4. This fact got confirmed by Raeymaeker's PhD supervisors by Email contact.
5. Tidy is freely available online at `http://tidy.sourceforge.net`.

Data sets (# of docs.)	Our results		Carme et al. (07)		Raeymaekers (08)	
	# corr.	# docs.	# corr.	# docs.	# corr.	# docs.
Okra-mails (251)	$2.67_{\pm 0.54}$	$1.77_{\pm 0.42}$	3.48	1.6	**2.0**	?
Bigbook-address (234)	**2.17**$_{\pm 0.37}$	$1.17_{\pm 0.37}$	3.02	1	2.3	?
Yahoo (79)	**10.07**$_{\pm 3.00}$	$4.27_{\pm 1.34}$	11.36	6.18	–	
Ebay (34)	**1.87**$_{\pm 0.67}$	$1.20_{\pm 0.40}$	2.62	1.06	–	
NY-Times (22)	$2.47_{\pm 0.50}$	$1.47_{\pm 0.50}$	**1.44**	1.44	–	
Google (33)	**4.47**$_{\pm 0.76}$	$2.57_{\pm 0.56}$	4.78	1.86	–	

Table 1: We compare our algorithm *stable-RPNI* with "path-only" pruning function but guided by the schema of HTML with two previous Web information extraction systems. For each tool and data set, we show the average number of corrections and the average number of documents needed until convergence. For our system, we also show the standard variation over the first 30 experiments. The best result for each target query is highlighted.

Raeymaekers (2008) did not indicate the number of documents necessary to identify the target queries in the two first data sets, and since his tool is no longer available, we cannot obtain them by other means. The four other data sets were designed by Carme et al. (2007). The Yahoo data set is the most difficult. This is because the HTML documents in this collection have heterogeneous formats.

9.3 XML Information Extraction

The queries learned for Web information extraction seem to be stable under the *path-only* pruning strategy, since otherwise they could not be learned thereby. In XML information extraction, however, there is a need for more complex queries, where less aggressive pruning strategies must be considered. We next define data sets for such queries and test our learning algorithm on them.

9.3.1 XPATH QUERIES FOR XML DATA SETS

An XML data set is composed of a collection of XML documents, all valid w.r.t. some DTD, and of a special XML document—called *companion*—that enumerates for each document of the collection all nodes that are selected by some fixed XML query (the target). Companions are also used in order to define input samples.

The data sets we will use in the experiments have been designed upon the XMark benchmark. XMark (Schmidt et al., 2002)[6] is a popular benchmark project in the XML database community. An XMark document stores a set of auctions which contain several data like items, persons, bidders, etc. The main interest of XMark, for our purpose, is that it comes with a rather complex DTD, which defines a set of trees with varied structures.[7]

The target queries we use are based on a set of realistic XPath queries that the authors of XMark, as well as Franceschet (2005), have proposed for testing XQuery or XPath processors. We have chosen XPath queries based only on the structure of XMark documents, that is, queries

6. This benchmark can be found at the following URL: `http://www.ins.cwi.nl/projects/xmark`.
7. This DTD can be found at `http://www.ins.cwi.nl/projects/xmark/Assets/auction.dtd`.

XMark-A1 (50 documents):

$/site/closed\_auctions/closed\_auction$
$/annotation/description/text/keyword$

The target query selects the keywords in the description of the closed auctions. It is the most simple of all queries, since the selection of a node only depends on its path from the root. By definition, all pruning functions capture this condition.

XMark-02 (100 documents):

$/site/open\_auctions/open\_auction/bidder[1]/increase$

The target query selects the increase of the first bidder for all open auctions. It is difficult to learn because the selection of a node depends on its position. That is why the query is not *path-only*-stable.

XMark-17 (100 documents):

$/site/people/person[\texttt{not}(homepage)]/name$

The target query selects the name of the persons that do not have a homepage. The difficulty here is to infer the previous negative condition.

XMark-21 (50 documents):

$/site/open\_auctions/open\_auction[count(bidder) \geq 3]/itemref$

The target query selects the item reference of all open auctions whose numbers of bidders is greater than three. It is hard to learn because the selection of one node depends on information on its siblings. This explains why it is not stable by *path-only*-pruning.

XMark-A8 (100 documents):

$/site/people/person[address\ \texttt{and}\ (phone\ \texttt{or}\ homepage)$
$\texttt{and}\ (creditcard\ \texttt{or}\ profile)]/name$

The target query selects the name of the persons who have filled in several information. The learning algorithm must infer a conjunction of disjunctions, which is a hard task.

XMark-A6 (250 documents):

$/site/people/person[profile/gender\ \texttt{and}\ profile/age]/name$

The target query selects the name of persons who have filled in both their gender and their age in their profile. This is a difficult query because the selected nodes depend on two children of their siblings. Only *path-ext*$_D$-pruning is able to capture this condition.

Figure 13: Description of XML queries used in our experiments.

that do not use tests on data values in their definition. Figure 13 provides the target queries in terms of XPath expressions. All the data sets are available online at the following URL: `http://grappa.univ-lille3.fr/~champavere/Recherche/datasets/`. Table 2 summarizes the pruning strategies for which these queries are stable.

9.3.2 EXPERIMENTAL RESULTS

For each data set, we run our learning algorithm with the pruning functions defined above where D is an automaton for the XMark DTD. Either an exception is raised because the target query is not stable for the pruning strategy. Or, it is stable and then we give the number of interactions done by

Query id.	Stable w.r.t.			
	path-only	$\textit{path-only}_{\text{XMark}}$	*path-ext*	$\textit{path-ext}_{\text{XMark}}$
XMark-A1	yes	yes	yes	yes
XMark-02	no	yes	yes	yes
XMark-17	no	yes	yes	yes
XMark-21	no	yes	yes	yes
XMark-A8	no	yes	yes	yes
XMark-A6	no	no	no	yes

Table 2: Stability of the XML queries used in our experiments and presented in Figure 13 w.r.t. the four pruning strategies on unranked trees.

the simulator, the number of pages which have been visited before convergence, and the running time of the algorithm. The results of our experiments are presented in Table 3.

Let us recall that we compared pruning strategies according to their aggressiveness in Figure 7. The results show that the best pruning strategy for learning a query is always the most aggressive one for which the target query is stable. Furthermore, if the maximally aggressive strategies are $\textit{path-only}_D$ and *path-ext* (recall that they can not be compared), then the results show (with the exception of XMark-17) that $\textit{path-only}_D$ is the better. In other words, when a schema is available, one should use the most aggressive pruning strategy guided by the schema.

10. Conclusion and Future Work

We distinguished classes of stable queries for schema-guided pruning strategies, and proposed new learning algorithms for regular stable queries. Experimental evidence shows that stability is the essential notion for understanding the difficulty of particular queries, in that queries are easier to learn if they remain stable under more aggressive pruning strategies. Furthermore, schema guidance is useful for defining relevant pruning strategies.

Which classes of XPath queries can be learned with what kind of pruning strategies remains to be discussed. That is the question, which classes of XPath queries are stable for which pruning strategies. Simple XPath queries with forward child and descendant axis only, such as `//a/b//c/d`, are stable under the *path-only* pruning strategy. When adding filters that use the child axis once such as `//a[./b]/c[./d]/e` they remain stable for $\textit{path-only}_D$. When permitting a second child axis in filters, such as in `//a[./b/f]/c[./d/g]/e`, we obtain a class of XPath queries that is stable under the $\textit{path-ext}_D$ strategy. Pruning may become useless for queries with descendant axis in filters, such as `/a[.//b]`, where node selection depends on arbitrary *b*-descendants. Whether this happens also depends on the schema. Similar problems are raised when using other recursive axis in filters. When using the "following" axis on the main path, such as in `//a/following::b`, we can use yet another pruning function that keeps all nodes following a selected node in document order (or only the next *b*-node following a selected node). Recursive backward axis, such as in `//a/preceding::*//c` may also quickly lead to non-stability. For more general classes of XPath queries, an interesting problem that we leave open might be to learn suitable pruning strategies. A user could help by annotating nodes that are relevant for selecting others.

XMark-A1	avg. # corrections	avg. # documents	total time (s)
path-only	**1.40** ±0.49	1.40 ±0.49	0.16 ±0.15
path-only$_D$	4.60 ±0.99	3.10 ±0.75	1.17 ±0.70
path-ext	6.03 ±1.64	4.10 ±1.30	4.75 ±2.66
path-ext$_D$	8.87 ±2.08	5.80 ±1.38	14.77 ±5.66

XMark-02

path-only	exception 'unstable query'		
path-only$_D$	**4.43** ±1.12	3.30 ±0.69	0.54 ±0.35
path-ext	13.33 ±7.74	7.43 ±3.22	9.17 ±8.82
path-ext$_D$	18.10 ±6.38	14.07 ±4.09	20.62 ±11.48

XMark-17

path-only	exception 'unstable query'		
path-only$_D$	13.57 ±1.54	4.77 ±1.31	1.09 ±0.36
path-ext	**4.90** ±0.98	2.93 ±0.81	0.42 ±0.21
path-ext$_D$	20.43 ±2.70	6.90 ±1.56	6.32 ±2.04

XMark-21

path-only	exception 'unstable query'		
path-only$_D$	**12.00** ±2.78	7.80 ±1.66	4.17 ±1.74
path-ext	22.97 ±4.69	16.03 ±3.04	47.65 ±21.63
path-ext$_D$	40.47 ±7.08	31.50 ±5.52	83.39 ±29.74

XMark-A8

path-only	exception 'unstable query'		
path-only$_D$	**11.93** ±6.62	5.27 ±1.57	5.95 ±11.86
path-ext	124.87 ±59.34	35.87 ±17.31	872.29 ±1240.39
path-ext$_D$	32.10 ±18.04	14.30 ±6.75	82.79 ±169.44

XMark-A6

path-only	exception 'unstable query'		
path-only$_D$			
path-ext			
path-ext$_D$	**16.03** ±2.74	9.30 ±2.21	12.83 ±5.60

Table 3: Experiments on XML data: average number of corrections and documents, and the total time needed to infer queries with respect to pruning strategies.

Another question is how to deal with XPath queries with tests on data values, such as `//book[./author="Knuth"]`. A practical approach could be to enrich the interactive setting by allowing the user to specify data values of interest. Another approach should be to combine our approach with statistical learning methods dealing with data values. Queries, in which equality of data values can be tested (joins), may be non-regular and thus raise more principal difficulties to

learnability. More generally, an interesting problem is whether classes of path queries for graph databases can be learned and what a pruning strategy could be in this context.

Also, in the future, query induction with schema-guided pruning strategies should be extended to n-ary regular queries (Lemay et al., 2006). This new framework for pruning strategies provided in this paper should be sufficiently general, so that one can define appropriate pruning strategies in the n-ary case (in contrast to previous settings). Other interesting directions would be to study OXPath queries (Sellers et al., 2011b) or tree-to-tree transformations (Lemay et al., 2010).

Acknowledgments

We thank the anonymous reviewers for their excellent work. They read all proofs in great detail while proposing many improvements. In particular, one of them spotted an error in a previous version of Proposition 5 which lead us to the important distinction between pruning strategies and pruning functions. We would also like to thank F. Gire and J.-M. Champarnaud for their valuable comments when reviewing J. Champavère's PhD thesis which preceded the present article. This work was partially supported by the French National Research Agency (ANR) through project CODEX of the program ANR-08-DEFIS-004 (2009-2012).

Appendix A. Remaining Proofs

Lemma 13 (Open Part) *For any tree automaton A recognizing D-pruned annotated trees, the language of unpruned trees $\mathcal{L}_{Q_A}$ of the query Q_A with schema D is regular.*

Proof We have to construct a tree automaton A' recognizing $\mathcal{L}_{Q_A}$. It must check for a tree $t * \beta$ whether all 1-annotations of β are justified by some 1-annotation of a tree $t' * \beta' \in \mathcal{L}(A)$ where $t \in compl_D(t')$, and that no 0-annotation of β is in conflict with some 1-annotation of a tree $t'' * \beta'' \in \mathcal{L}(A)$ with $t \in compl_D(t'')$. We first construct a deterministic automaton A'' that recognizes the language:

$$\mathcal{L}(A'') = \{t * \beta \mid t \in \mathcal{L}(D),\ t \in compl_D(t'),\ t' * \beta \in \mathcal{L}(A)\}.$$

This is straightforward but requires determinization. Given a tree t let γ_t be the function that maps nodes ν of t to the state of A' that contains all states r such that there exists β for which $t[\nu \mapsto r] * \beta$ is recognized by A''. We construct an automaton A''' that recognizes the language:

$$\mathcal{L}(A''') = \{t * \beta * \gamma_t \mid t \in \mathcal{L}(D),\ \beta \text{ is } Q\text{-consistent}\}.$$

Once this will be done, we can define A' by projection $A' = \Pi_{\Sigma \cup (\Sigma \times \mathbb{B})}(A''')$. For trees $t * \beta * \gamma$, it can be tested by an automaton whether $\gamma = \gamma_t$ and also whether $t \in \mathcal{L}(D)$. What remains to construct is an automaton that tests for trees $t * \beta * \gamma$ whether β is Q-consistent. This can be done by an automaton B that we define next. The states of B are the subsets of states of A''. For $f \in \Sigma^{(n)}$, a set of symbols $F \subseteq \{f, (f,0), (f,1)\}$, and states $R_1, \ldots, R_n$ of A'' we define the following set of states:

$$F(R_1, \ldots, R_n) = \{r \mid\ \tilde{f} \in F,\ \tilde{f}(r_1, \ldots, r_n) \to r \text{ in } A'',\ r_i \in R_i\}.$$

These are the states that A'' can reach from some tuple of states in $R_1 \times \ldots \times R_n$ with a symbol from F. The rules of B are inferred as follows where $f \in \Sigma^{(n)}$.

$$\frac{R \cap \{(f,1)\}(R_1, \ldots, R_n) \neq \emptyset \qquad R' = F(R_1, \ldots, R_n) \qquad F = \{(f,0), (f,1), f\}}{((f,1), R)(R_1, \ldots, R_n) \to R' \text{ in } B},$$

$$\frac{R \cap \{(f,1)\}(R_1,\ldots,R_n) = \emptyset \qquad R' = F(R_1,\ldots,R_n) \qquad F = \{(f,0),(f,1),f\}}{((f,0),R)(R_1,\ldots,R_n) \to R' \text{ in } B},$$

$$\frac{R' = F(R_1,\ldots,R_n) \qquad F = \{(f,0),(f,1),f\}}{(f,R)(R_1,\ldots,R_n) \to R' \text{ in } B}.$$

The final states of B are those states R that contain some final state of B. If B goes into a state R on a tree $t * \beta * \gamma$ then all states in R can be reached on some tree $t * \beta' \in \mathcal{L}(A'')$ and all annotations of β are compatible with γ. If $\gamma = \gamma_t$, then β is Q-consistent. ■

Theorem 19 *Let D be a deterministic tree automaton with signature Σ, P a deterministic tree automaton with signature $(\Sigma \cup (\Sigma \times \mathbb{B})) \times \{y,n\}$, $p = \wp_{P,D}$ a pruning function, and Q a query with domain D. Given a tree automaton A with $\mathcal{L}(A) = \mathcal{L}_Q$ we can decide in time $O(|A|^2\,|D|\,|P|)$ whether Q is p-stable.*

Proof We write $t_1 \circledast \ldots \circledast t_n$ for the overlay of trees t_1, …, t_n over possibly different ranked signatures, where missing nodes are filled up with a fresh $\perp$-symbol. We consider the following tree language:

$$\left\{ \begin{array}{l} t * \beta \circledast t' * \beta' \circledast t_1 * \beta_1 \;\mid\; t' * \beta' = p(t * \beta),\ t_1 \in \mathit{compl}_D(t'), \\ \qquad \beta \text{ is } Q\text{-consistent for } t,\ \beta_1 \text{ is } Q\text{-consistent for } t_1, \\ \qquad \exists \nu \in \mathit{nodes}(t').\ \beta(\nu) = 1 \wedge \beta_1(\nu) = 0. \end{array} \right\}$$

By construction, this language is empty and if only if Q is p-stable. This can be decided in linear time in the size of an automaton recognizing the above language:

- By running a single copy of D jointly on $t' \circledast t_1$—as explained in more details in the proof of Theorem 22—one can check whether $t_1 \in \mathit{compl}_D(t')$.
- By running A on the first component in parallel, we can check whether β is a Q-consistent annotation of t, and thus whether $t \in \mathcal{L}(D)$.
- By running $\Pi_{\Sigma \cup (\Sigma \times \mathbb{B})}(P)$ on $t * \beta$, we can check that the second component $t' * \beta'$ is the p-pruning of the first component $t * \beta$.
- By running another copy of A on the third component in parallel, we can check whether β_1 is a Q-consistent annotation of t_1, and thus whether $t_1 \in \mathcal{L}(D)$.
- Testing whether $\exists \nu \in \mathit{nodes}(t').\ \beta(\nu) = 1 \wedge \beta_1(\nu) = 0$ can be done with a 2-state control.

In summary, we have to run the following automata in parallel on different components: twice A, once D and once P, so the overall size of the automaton is in $O(|A|^2\,|D|\,|P|)$. ■

References

Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with lixto. In *28th International Conference on Very Large Data Bases (VLDB)*, pages 119–128, 2001.

Geert J. Bex, Frank Neven, Thomas Schwentick, and Karl Tuyls. Inference of concise DTDs from XML data. In *32nd International Conference on Very Large Data Bases (VLDB)*, pages 115–126, 2006.

Julien Carme, Joachim Niehren, and Marc Tommasi. Querying unranked trees with stepwise tree automata. In *19th International Conference on Rewriting Techniques and Applications (RTA)*, pages 105–118, 2004.

Julien Carme, Michal Ceresna, Oliver Frölich, Georg Gottlob, Tamir Hassan, Marcus Herzog, Wolfgang Holzinger, and Bernhard Krüpl. The Lixto project: exploring new frontiers of web data extraction. In *23rd International Information Systems Conference (BNCOD)*, pages 1–15, 2006.

Julien Carme, Michal Ceresna, and Max Goebel. Query-Based learning of XPath expressions. In *8th International Colloquium on Grammatical Inference (ICGI)*, pages 342–343, 2006.

Julien Carme, Rémi Gilleron, Aurélien Lemay, and Joachim Niehren. Interactive learning of node selecting tree transducers. *Machine Learning*, 66(1):33–67, 2007.

Jérôme Champavère. *Induction de requêtes guidée par schémas*. PhD thesis, Université des Sciences et Technologies de Lille 1, 2010.

Jérôme Champavère, Rémi Gilleron, Aurélien Lemay, and Joachim Niehren. Schema-Guided Induction of Monadic Queries. In *9th International Colloquium on Grammatical Inference (ICGI)*, pages 15–28, 2008.

Jérôme Champavère, Rémi Gilleron, Aurélien Lemay, and Joachim Niehren. Efficient inclusion checking for deterministic tree automata and XML schemas. *Information and Computation*, 207 (11):1181–1208, 2009.

William W. Cohen, Matthew Hurst, and Lee S. Jensen. A flexible learning system for wrapping tables and lists in HTML documents. In *11th International Conference on World Wide Web (WWW)*, pages 232–241, 2002.

Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications. Electronic book available at `http://tata.gforge.inria.fr/`. Revised 2007.

François Coste, Daniel Fredouille, Christopher Kermorvant, and Colin de la Higuera. Introducing domain and typing bias in automata inference. In *7th International Colloquium on Grammatical Inference (ICGI)*, pages 115–126, 2004.

Massimo Franceschet. XPathMark: An XPath benchmark for the XMark generated data. In *3rd International Conference on Database and XML (XSym)*, pages 129–143, 2005.

Dayne Freitag and Andrew K. McCallum. Information extraction with HMMs and shrinkage. In *AAAI Workshop on Machine Learning for Information Extraction*, pages 31–36, 1999.

Pedro García and Jose Oncina. Inference of recognizable tree sets. Technical Report DSIC-II/47/93, Universidad de Alicante, 1993.

Rémi Gilleron, Florent Jousse, Isabelle Tellier, and Marc Tommasi. XML document transformation with conditional random fields. In *5th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, pages 525–539, 2006.

Rémi Gilleron, Patrick Marty, Marc Tommasi, and Fabien Torre. Interactive tuples extraction from semi-structured data. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 997–1004, 2006.

Georg Gottlob and Christoph Koch. Monadic queries over tree-structured data. In *17th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 189–202, 2002.

Georg Gottlob, Erich Grädel, and Helmut Veith. Datalog LITE: a deductive query language with linear time model Checking. *ACM Transactions on Computational Logics*, 3(1):42–79, 2002.

Nicholas Kushmerick. Wrapper induction: efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15–68, 2000.

Aurélien Lemay, Joachim Niehren, and Rémi Gilleron. Learning n-ary node selecting tree transducers from completely annotated examples. In *8th International Colloquium on Grammatical Inference (ICGI)*, pages 253–267, 2006.

Aurélien Lemay, Sebastian Maneth, and Joachim Niehren. A learning algorithm for top-down XML transformations. In *29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 285–296, 2010.

Wolfgang May. Information extraction and integration with FLORID: the MONDIAL case study. Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.

Michel Minoux. LTUR: a simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Information Processing Letters*, 29(1):1–12, 1988.

Ion Muslea, Steve Minton, and Craig Knoblock. Active learning with strong and weak views: a case study on wrapper induction. In *18th International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 415–420, 2003.

Jose Oncina and Miguel A. Varó. Using domain information during the learning of a subsequential transducer. In *3rd International Colloquium on Grammatical Inference (ICGI)*, pages 301–312, 1996.

David Pinto, Andrew McCallum, Xing Lee, and W .Bruce Croft. Table extraction using conditional random fields. In *26th ACM SIGIR*, pages 235–242, 2003.

Stefan Raeymaekers. Information extraction from Web pages based on tree automata induction. PhD thesis, Katholieke Universiteit Leuven, 2008.

Stefan Raeymaekers, Maurice Bruynooghe, and Jan Van den Bussche. Learning (k,l)-contextual tree languages for information extraction from web pages. *Machine Learning*, 71(2-3):155–183, 2008.

Albrecht Schmidt, Florian Waas, Martin Kersten, Michael J. Carey, Ioana Manolescu, and Ralph Busse. XMark: A benchmark for XML data management. In *28th International Conference on Very Large Data Bases (VLDB)*, pages 974–985, 2002.

Andrew J. Sellers, Tim Furche, Georg Gottlob, Giovanni Grasso, and Christian Schallhart. Taking the OXPath down the deep web. In *14th International Conference on Extending Database Technology (EDBT)*, pages 542–545, 2011.

Andrew J. Sellers, Tim Furche, Georg Gottlob, Giovanni Grasso, and Christian Schallhart. OXPath: little language, little memory, great value. In *WWW (Companion Volume)*, pages 261–264, 2011.

Slawomir Staworko and Piotr Wieczorek. Learning twig and path queries. In *15th International Conference on Database Theory (ICDT)*, 2012.

Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2D conditional random fields for web information extraction. In *22nd International Conference on Machine Learning (ICML)*, pages 1044–1051, 2005.

Journal of Machine Learning Research 14 (2013) 965-1003 Submitted 7/12; Revised 1/13; Published 4/13

Bayesian Canonical Correlation Analysis

Arto Klami ARTO.KLAMI@HIIT.FI
Seppo Virtanen SEPPO.J.VIRTANEN@AALTO.FI
Samuel Kaski* SAMUEL.KASKI@AALTO.FI
Helsinki Institute for Information Technology HIIT
Department of Information and Computer Science
PO Box 15600
Aalto University
00076 Aalto, Finland

Editor: Neil Lawrence

Abstract

Canonical correlation analysis (CCA) is a classical method for seeking correlations between two multivariate data sets. During the last ten years, it has received more and more attention in the machine learning community in the form of novel computational formulations and a plethora of applications. We review recent developments in Bayesian models and inference methods for CCA which are attractive for their potential in hierarchical extensions and for coping with the combination of large dimensionalities and small sample sizes. The existing methods have not been particularly successful in fulfilling the promise yet; we introduce a novel efficient solution that imposes group-wise sparsity to estimate the posterior of an extended model which not only extracts the statistical dependencies (correlations) between data sets but also decomposes the data into shared and data set-specific components. In statistics literature the model is known as inter-battery factor analysis (IBFA), for which we now provide a Bayesian treatment.

Keywords: Bayesian modeling, canonical correlation analysis, group-wise sparsity, inter-battery factor analysis, variational Bayesian approximation

1. Introduction

Canonical correlation analysis (CCA), originally introduced by Hotelling (1936), extracts linear components that capture correlations between two multivariate random variables or data sets. During the last decade the model has received a renewed interest in the machine learning community as the standard model for unsupervised multi-view learning settings. In a sense, it is the analogue of principal component analysis (PCA) for two co-occurring observations, or views, retaining the positive properties of closed-form analytical solution and ease of interpretation of its more popular cousin.

A considerable proportion of the work has been on non-linear extensions of CCA, including neural network based solutions (Hsieh, 2000) and kernel-based variants (Bach and Jordan, 2002; Lai and Fyfe, 2000; Melzer et al., 2001). This line of research has been covered in a comprehensive overview by Hardoon et al. (2004), and hence will not be discussed in detail in this article. Instead, we review a more recent trend treating CCA as a generative model, initiated by the work of Bach and

∗. Is also at Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland.

Jordan (2005). Most works in the generative approach retain the linear nature of CCA, but provide inference methods more robust than the classical linear algebraic solution and, more importantly, the approach leads to novel models through simple changes in the generative description or via the basic principles of hierarchical modeling.

The generative modelling interpretation of CCA is essentially equivalent to a special case of a probabilistic interpretation (Browne, 1979) of a model called inter-battery factor analysis (IBFA; Tucker, 1958). While the analysis part of Browne (1979) is limited to the special case of CCA, the generic IBFA model describes not only the correlations between the data sets but provides also components explaining the linear structure within each of the data sets. One way of thinking about IBFA is that it complements CCA by providing a PCA-like component description of all the variation not captured by the correlating components. If the analysis focuses on only the correlating components, or equivalently the latent variables shared by both data sets, the solution becomes equivalent to CCA. However, the extended model provides novel application opportunities not immediately apparent in the more restricted CCA model.

The IBFA model has recently been re-invented in the machine learning community by several authors (Klami and Kaski, 2006, 2008; Ek et al., 2008; Archambeau and Bach, 2009), resulting in probabilistic descriptions identical with that of Browne (1979). The inference has been primarily based on finding the maximum likelihood or maximum a posteriori solution of the model, with practical algorithms based on expectation maximization. Since the terminology of calling these models (probabilistic) CCA has already become established in the machine learning community, we will regard the names CCA and IBFA interchangeable. Using the term CCA emphasizes finding of the correlations and shared components, whereas IBFA emphasizes the decomposition into shared and data source-specific components.

In this paper we extend this IBFA/CCA work to a fully Bayesian treatment, extending our earlier conference paper (Virtanen et al., 2011), and in particular provide two efficient inference algorithms, a variational approximation and a Gibbs sampler, that automatically learn the structure of the model that is, in the general case, unidentifiable. The model is solved as a generic factor analysis (FA) model with a specific group-wise sparsity prior for the factor loadings or projections, and an additional constraint tying the residual variances within each group to be the same. We demonstrate how the model not only finds the IBFA solution, but also provides a CCA solution superior to the earlier Bayesian variants of Klami and Kaski (2007) and Wang (2007).

The technical description of the model and its connection to other models are complemented with demonstrations on practical application scenarios for the IBFA model. The main purpose of the experiments is to show that the tools find the intended solution, and to introduce prototypical application cases.

2. Canonical Correlation Analysis

Before explaining the Bayesian approach for canonical correlation analysis (CCA), we briefly introduce the classical CCA problem. Given two co-occurring random variables with N observations collected as matrices $\mathbf{X}_1 \in \mathbb{R}^{D_1 \times N}$ and $\mathbf{X}_2 \in \mathbb{R}^{D_2 \times N}$, the task is to find linear projections $\mathbf{U} \in \mathbb{R}^{D_1 \times K}$ and $\mathbf{V} \in \mathbb{R}^{D_2 \times K}$ so that the correlation between $\mathbf{u}_k^T \mathbf{X}^{(1)}$ and $\mathbf{v}_k^T \mathbf{X}^{(2)}$ is maximized for the components k, under the constraint that $\mathbf{u}_k^T \mathbf{X}^{(1)}$ and $\mathbf{u}_{k'}^T \mathbf{X}^{(1)}$ are uncorrelated for all $k \neq k'$ (and similarly for the

other view). The solution can be found analytically by solving the eigenvalue problems

$$\begin{aligned} \mathbf{C}_{11}^{-1}\mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\mathbf{u} &= \rho^2\mathbf{u}, \\ \mathbf{C}_{22}^{-1}\mathbf{C}_{21}\mathbf{C}_{11}^{-1}\mathbf{C}_{12}\mathbf{v} &= \rho^2\mathbf{v}, \end{aligned} \tag{1}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix}$$

is the joint covariance matrix of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ and ρ denotes the canonical correlation. In practice all components can be found by solving a single generalized eigenvalue problem. For more detailed discussion on classical CCA, see for instance the review of Hardoon et al. (2004).

3. Model

Our Bayesian approach to CCA is based on latent variable models and linear projections. At the core of the generative process is an unobserved latent variable $\mathbf{z} \in \mathbb{R}^{K\times 1}$, which is transformed via linear mappings to the observation spaces to represent the two multivariate random variables $\mathbf{x}^{(1)} \in \mathbb{R}^{D_1\times 1}$ and $\mathbf{x}^{(2)} \in \mathbb{R}^{D_2\times 1}$. The observed data samples are provided as matrices $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, ..., \mathbf{x}_N^{(m)}] \in \mathbb{R}^{D_m\times N}$ with N observations. To simplify the notation, we denote by $\mathbf{x} = [\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$ and $\mathbf{X} = [\mathbf{X}^{(1)}; \mathbf{X}^{(2)}]$ the feature-wise concatenation of the two random variables. Throughout the paper we use the superscript (m), where m is 1 or 2, to denote the view or data set in question, though for scalar variables (such as D_m) we use the subscript without risk of confusion to streamline the notation. For matrices and vectors the subscripts are used to indicate the individual elements, with $\mathbf{X}_{:,n}$ denoting the whole nth column of $\mathbf{X}$ (also denoted by $\mathbf{x}_n$ to simplify the notation when appropriate) and $\mathbf{X}_{d,:}$ denoting the dth row treated as a column vector. Finally, we use $\mathbf{0}$ and $\mathbf{I}$ to denote zero- and identity matrices of sizes which make sense in the context, without cluttering the notation.

3.1 Inter-battery Factor Analysis

In the latent variable model studied in this work,

$$\begin{aligned} \mathbf{z} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{z}^{(m)} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}^{(m)} &\sim \mathrm{N}(\mathbf{A}^{(m)}\mathbf{z} + \mathbf{B}^{(m)}\mathbf{z}^{(m)}, \boldsymbol{\Sigma}^{(m)}), \end{aligned} \tag{2}$$

following the probabilistic interpretation of inter-battery factor analysis by Browne (1979).[1] The notation $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ corresponds to the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Here the $\boldsymbol{\Sigma}^{(m)} \in \mathbb{R}^{D_m\times D_m}$ are diagonal matrices, indicating independence of the noise over the features. Our practical solutions will further simplify the model by assuming isotropic noise, but could easily be extended to generic diagonal noise covariances as well. A plate diagram of the model is given in Figure 1.

The conceptual meaning of the various terms in the model is as follows. The shared latent variables $\mathbf{z}$ capture the variation common to both data sets, and they are transformed to the observation

1. The original definition by Browne (1979) allows for more relaxed definitions for the various covariance terms, but in practice he resorts to the choices made above in the actual analysis part of his work and makes the same independence assumptions.

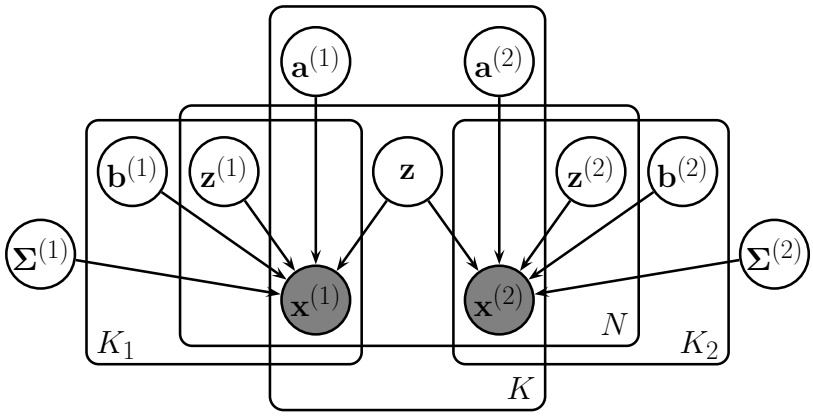


Figure 1: Graphical illustration of the inter-battery factor analysis (IBFA) model as a plate diagram. The shaded nodes $\mathbf{x}^{(m)}$ denote the two observed random variables, and the latent variables $\mathbf{z}$ capture the correlations between them. The variation specific to each view is modeled with view-specific latent variables $\mathbf{z}^{(m)}$. The parameters of the model are the linear projections from the $\mathbf{z}$ to the data ($\mathbf{A}^{(m)}$ with columns $\mathbf{a}^{(m)}$) and from the $\mathbf{z}^{(m)}$ to the data ($\mathbf{B}^{(m)}$ with columns $\mathbf{b}^{(m)}$), complemented by the residual noise covariance of each view denoted by $\boldsymbol{\Sigma}^{(m)}$.

space by the linear mappings $\mathbf{A}^{(m)}\mathbf{z}$, where $\mathbf{A}^{(m)} \in \mathbb{R}^{D_m \times K}$. The remaining variation is modeled by the latent variables $\mathbf{z}^{(m)} \in \mathbb{R}^{K_m \times 1}$ specific to each data set, transformed to the observation space by another linear mapping $\mathbf{B}^{(m)}\mathbf{z}^{(m)}$, where $\mathbf{B}^{(m)} \in \mathbb{R}^{D_m \times K_m}$. The actual observations are then generated by adding up these two terms, followed by addition of noise that is independent over the dimensions. We assume zero-mean data without loss of generality; the model could include a separate mean parameter whose estimate would anyway converge close to the empirical mean which can equivalently be subtracted from the data prior to the analysis. We also assume fully observed data; techniques similar to what Ilin and Raiko (2010) propose for Bayesian PCA could be adopted to handle missing data.

In terms of classical models, the model can be interpreted as CCA complemented by two separate FA models (or PCA models if assuming isotropic noise) factorizing the residuals of the CCA within each data set. This connection will become more apparent in the following sections when the probabilistic interpretation of CCA is introduced. Typically both K and K_m are smaller than the corresponding data dimensionality, implying that the model provides low-rank approximations for the two data matrices.

3.2 Probabilistic Canonical Correlation Analysis

There exists a simple way of converting the IBFA model of (2) into a probabilistic version of CCA (Bach and Jordan, 2005; Browne, 1979; De Bie and De Moor, 2003). The process starts by integrating out the view-specific latent variables $\mathbf{z}^{(m)}$, to reach a model that has explicit components only for the shared variation similarly to how CCA only explains the correlations. Simple algebraic

manipulation gives the model

$$\begin{aligned} \mathbf{z} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}^{(m)} &\sim \mathrm{N}(\mathbf{A}^{(m)}\mathbf{z}, \mathbf{B}^{(m)}\mathbf{B}^{(m)^T} + \mathbf{\Sigma}^{(m)}). \end{aligned}$$

The latent representation of this model is simpler, only containing the $\mathbf{z}$ instead of three separate sets of latent variables, but the diagonal covariance of the IBFA model is replaced with $\mathbf{B}^{(m)}\mathbf{B}^{(m)^T} + \mathbf{\Sigma}^{(m)}$. In effect, the view-specific variation is now modeled only implicitly, in form of correlating noise. If we further assume that the dimensionality of the $\mathbf{z}^{(m)}$ is sufficient for modeling all such variation, the model can be re-parameterized with $\mathbf{\Psi}^{(m)} = \mathbf{B}^{(m)}\mathbf{B}^{(m)^T} + \mathbf{\Sigma}^{(m)}$ without loss of generality. This results in the model

$$\begin{aligned} \mathbf{z} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}^{(m)} &\sim \mathrm{N}(\mathbf{A}^{(m)}\mathbf{z}, \mathbf{\Psi}^{(m)}), \end{aligned} \tag{3}$$

where $\mathbf{\Psi}^{(m)}$ is a generic covariance matrix. This holds even if assuming isotropic noise in (2).

Browne (1979) proved that (3) is equivalent to classical CCA, by showing how the maximum likelihood solution finds the same canonical weights as regular CCA, up to a rotation. Bach and Jordan (2005) proved the same result through a slightly different derivation, whereas De Bie and De Moor (2003) provided a partial proof showing that the CCA solution is a stationary point of the likelihood. The fundamental result of these derivations is that the maximum likelihood estimates $\hat{\mathbf{A}}^{(m)}$ correspond to rank-preserving linear transformations of $\mathbf{U}$ and $\mathbf{V}$, the solutions of (1). While the connection was shown for the case with a generic $\mathbf{\Psi}$, it holds also for the IBFA model as long as the rank of $\mathbf{B}^{(m)}$ is sufficient for modeling all data set-specific variation. This is because the model itself is the same, it just explicitly includes the nuisance parameters $\mathbf{z}^{(m)}$ and $\mathbf{B}^{(m)}$.

Even though the generative formulation is equivalent to classical CCA in the sense that they both find the same subspace, one difference pointed out also by Browne (1979) is worth emphasizing: The generative formulation maintains a single latent variable $\mathbf{z}$ that captures the shared variation, whereas CCA results in two separate but correlating variables obtained by projecting the observed variables into the correlating subspace. It is, however, possible to move between these representations; a single latent variable can be obtained by averaging the canonical scores ($\mathbf{u}_k^T\mathbf{X}^{(1)}$ and $\mathbf{v}_k^T\mathbf{X}^{(2)}$) of regular CCA; two separate latent variables can be produced with the generative formulation by estimating the distribution of $\mathbf{z}$ conditional on having observed only one of the views ($p(\mathbf{z}|\mathbf{x}^{(1)})$ and $p(\mathbf{z}|\mathbf{x}^{(2)})$).

3.3 Identifiability

The model in (2) is in general unidentifiable in two respects. The first is shared with the marginalized version in (3): the models are invariant to rank-preserving linear transformations. For all invertible $\mathbf{R} \in \mathbb{R}^{K\times K}$ we have $\mathbf{A}^{(m)}\mathbf{z} = \mathbf{A}^{(m)}\mathbf{R}\mathbf{R}^{-1}\mathbf{z}$, and hence the solution is defined only up to such transformations. In other words, the model finds the same subspace as the classical CCA would, but extracting the specific components requires further constraints or postprocessing. Browne (1979) resorts to simple identifiability constraints borrowed from regular factor analysis, whereas Archambeau et al. (2006) provide a post-processing step that is close to applying regular CCA to the covariance matrices of the probabilistic solution.

The full IBFA model (2) has additional degrees of freedom in terms of component allocation. The model comes with three separate sets of latent variables with component numbers K, K_1 and K_2. However, individual components can be moved between these sets without influencing the likelihood of the observed data; removal of a shared component can always be compensated by introducing two view-specific components, one for each data set, that have the same latent variables. In practice, all solutions for the full IBFA model hence need to carefully address the choice of model complexity. In the next section we will introduce one such solution, based on Bayesian inference.

3.4 The Role of View-specific Variation

The models (2) and (3) are both very closely related to probabilistic formulation of PCA, FA, and many other simple matrix factorizations. The crucial difference worth pointing out is the definition of the noise. Instead of assuming independent noise over the dimensions the CCA model allows for arbitrary correlations between them. This is done either by explicitly parameterizing the noise through a covariance matrix $\mathbf{\Psi}^{(m)}$ as in (3) or by the separate view-specific components $\mathbf{B}^{(m)}\mathbf{z}^{(m)}$ as in (2).

Modeling the correlations in view-specific noise is crucial for extracting the true correlations between the views. This is easy to illustrate by constructing counter-examples where the correlating dimensions are of smaller scale than some strong view-specific variation. Any joint model assuming independent noise over the dimensions will find the view-specific variation as the most prominent components. It may be possible to identify these components as view-specific in a post-processing step to reach interpretation similar to CCA, but directly modeling the view-specific variation as separate components has obvious advantages.

The importance of modeling the variation within each view in addition to the shared effects is so subtle that even some authors claiming to work with CCA have ignored it. For example, Shon et al. (2006), Fujiwara et al. (2009), and Rai and Daumé III (2009) all describe their models as CCA, but eventually resort to assuming independent noise on top of the shared components. This is a reasonable assumption that simplifies computation dramatically, but it also means that the models do not correspond to CCA but are instead variants of collective matrix factorization (CMF; Singh and Gordon 2008). They are useful tools for multi-view data, but it is important to realize that the simplifying assumption dramatically changes the nature of the model. In particular, such models are likely to misinterpret strong view-specific variation as a shared effect, since they have no means of explaining it otherwise. Our choice of modeling the view-specific variation as a low-rank process results in similar computational performance as ignoring the view-specific variation, but retains the capability of modeling also view-specific variation.

4. Inference

For learning the IBFA model we need to infer both the latent signals $\mathbf{z}$ and $\mathbf{z}^{(m)}$ as well as the linear projections $\mathbf{A}^{(m)}$ and $\mathbf{B}^{(m)}$ from data. For this purpose, we need to estimate the posterior distribution $p(\mathbf{z}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \mathbf{\Sigma}^{(1)}, \mathbf{\Sigma}^{(2)} | \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, and marginalize over the possibly uninteresting variables. In this section we first review the earlier inference solutions for the Bayesian CCA model without presenting the technical details, and then proceed to explaining our solution for the full Bayesian IBFA model.

Before explaining the Bayesian inference solutions, we mention earlier maximum likelihood solutions for completeness. Bach and Jordan (2005) gave an expectation maximization algorithm

for the pure CCA model, whereas Klami and Kaski (2008) extended it for the IBFA model. Both approaches generalize immediately to seeking maximum a posteriori (MAP) estimates, which provides a justified way for adding regularization in the solution. Here, however, we are interested in analysis of the full posterior distribution.

4.1 Bayesian Inference

For Bayesian analysis, the model needs to be complemented with priors for the model parameters. Bayesian treatments of CCA were independently proposed by Klami and Kaski (2007) and Wang (2007). Both formulations use the inverse-Wishart distribution as a prior for the covariance matrices $\mathbf{\Psi}^{(m)}$ in (3) and apply the automatic relevance determination (ARD; Neal, 1996) prior for the linear mappings $\mathbf{A}^{(m)}$.

The ARD is a Normal-Gamma prior for the projection weights. For each component (column) $\mathbf{a}_k^{(m)}$ the prior specifies a precision $\boldsymbol{\alpha}_k^{(m)}$ that controls the scale of the values for that component:[2]

$$\begin{aligned}\text{ARD}(\mathbf{A}^{(m)}|\alpha_0,\beta_0) &= \prod_{k=1}^{K} p(\mathbf{a}_k^{(m)}|\boldsymbol{\alpha}_k^{(m)})p(\boldsymbol{\alpha}_k^{(m)}|\alpha_0,\beta_0),\\ \boldsymbol{\alpha}_k^{(m)} &\sim \text{Gamma}(\alpha_0,\beta_0),\\ \mathbf{a}_k^{(m)} &\sim \text{N}(\mathbf{0},(\boldsymbol{\alpha}_k^{(m)})^{-1}\mathbf{I}).\end{aligned}$$

The hyperpriors α_0,β_0 are set to small values (in our experiments to $\alpha_0=\beta_0=10^{-14}$) to obtain a relatively noninformative prior with wide support.[3] The posterior of the model then becomes one where the number of components is automatically selected by pushing $\boldsymbol{\alpha}_k^{(m)}$ of unnecessary components towards infinity. A justification for this observation is obtained by integrating $\boldsymbol{\alpha}_k^{(m)}$ out in the above prior; we then get a heavy-tailed prior for the elements of $\mathbf{A}^{(m)}$ with considerable posterior mass around zero. The component choice can be made more robust by further assuming $\boldsymbol{\alpha}^{(1)}=\boldsymbol{\alpha}^{(2)}$, which corresponds to placing the ARD prior for $\mathbf{A}=[\mathbf{A}^{(1)};\mathbf{A}^{(2)}]$ (Klami and Kaski, 2007). More data will then be used for determining the activity of each component, but data sets with comparable scale are required. Further insights into the ARD prior are provided by Wipf and Nagarajan (2008).

For the covariance matrices $\mathbf{\Psi}^{(m)}$ a natural choice is to use a conjugate inverse-Wishart prior

$$\mathbf{\Psi}^{(m)} \sim \text{IW}(\mathbf{S}_0,\nu_0)$$

with ν_0 degrees of freedom and scale matrix $\mathbf{S}_0$, which results in positive definite draws as long as the degrees of freedom (which for N samples becomes $N+\nu_0$ in the posterior) is at least equal the data dimensionality. Both Klami and Kaski (2007) and Wang (2007) adopted this choice.

Given the above priors, several inference techniques for the posterior are feasible. Wang (2007) provided a variational mean-field algorithm, whereas Klami and Kaski (2007) used Gibbs sampling. Both of these algorithms are fairly straightforward and easy to derive, since all conditional distributions are conjugate. The former is more efficient in determining the correct model complexity due

2. Note that ARD generates both the matrix $\mathbf{A}^{(m)}$ as well as the scales $\boldsymbol{\alpha}^{(m)}$, and hence notation $\text{ARD}(\mathbf{A}^{(m)},\boldsymbol{\alpha}^{(m)}|\alpha_0,\beta_0)$ would be more accurate. However, since $\boldsymbol{\alpha}^{(m)}$ is irrelevant for the rest of the model, we adopt the more compact notation.

3. The distribution is flat over the positive real line, but slightly favors values near zero.

to the ARD prior updates being more efficient in the variational framework, whereas the latter is easier to extend, as demonstrated by Klami and Kaski (2007) by using the Bayesian CCA as part of a non-parametric hierarchical model. Further extensions of the CCA model, described in Section 6, have used both approaches.

Despite the apparent simplicity of the derivation, it is worth pointing out that inference of the Bayesian CCA model is difficult for large dimensionalities. This is because we need to estimate the posterior distribution over the $D_m \times D_m$ covariance matrices $\mathbf{\Psi}^{(m)}$. The inference algorithms generally need to invert those matrices in every step, resulting in $O(D_m^3)$ complexity. More importantly, providing accurate estimates would require extremely large sample sizes; the covariance matrix has $O(D_m^2)$ parameters, which is often well above the data set size. Hence the direct Bayesian treatment of CCA needs to resort to either using very strong priors (for example, favoring diagonal covariance matrices and hence regularizing the model towards Bayesian PCA), or it will end up doing inference over a very wide posterior. Consequently, all the practical applications of Bayesian CCA in the earlier works have been for relatively low-dimensional data; the original works by Klami and Kaski (2007) and Wang (2007) had at most 8 dimensions in any of their experiments. Later applications have typically used some alternative dimensionality reduction techniques to make Bayesian CCA feasible for otherwise too high-dimensional data (Huopaniemi et al., 2009, 2010).

4.2 Group-wise Sparsity for IBFA

While the above solutions are sufficient for the CCA model, barring the difficulties with high dimensionality, the full IBFA model requires more advanced inference methods. Next we will introduce a novel inference solution extending our earlier conference paper (Virtanen et al., 2011). Besides providing a Bayesian inference technique for the IBFA model, the algorithm is applicable also to the regular CCA case and, as will be shown later, actually is superior to the earlier solutions also for that scenario.

A main challenge in learning the Bayesian IBFA (BIBFA) model, as discussed in Section 3.3, is that it requires learning three separate sets of components and the solution is unidentifiable with respect to allocating components to the three groups. A central element in our solution is to replace these three sets with just one set, and solve the allocation by requiring the projections to be sparse in a specific structured way. This is done in a way that does not change the model itself, but allows automatic complexity selection.

We start with a straightforward re-formatting of the model. We define $\mathbf{y} = [\mathbf{z}; \mathbf{z}^{(1)}; \mathbf{z}^{(2)}] \in \mathbb{R}^{K_c \times 1}$, where $K_c = K + K_1 + K_2$, as the concatenation of the three latent variables and set

$$\mathbf{W} = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & \mathbf{0} \\ \mathbf{A}^{(2)} & \mathbf{0} & \mathbf{B}^{(2)} \end{bmatrix} \tag{4}$$

and

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}^{(2)} \end{bmatrix}.$$

Now we can write (2) equivalently as

$$\begin{aligned} \mathbf{y} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x} &\sim \mathrm{N}(\mathbf{W}\mathbf{y}, \mathbf{\Sigma}). \end{aligned} \tag{5}$$

In other words, we are now analyzing the feature-wise concatenation of the data sources with a single latent variable model with diagonal noise covariance $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$, where $D = D_1 + D_2$, and a projection matrix $\mathbf{W} \in \mathbb{R}^{D \times K_c}$ with the specific structure shown in (4).

Ignoring the structure in $\mathbf{W}$, the model is actually a Bayesian factor analysis model (Ghahramani and Beal, 2000). If $\mathbf{\Sigma}$ is further assumed to be spherical ($\mathbf{\Sigma} = \sigma^2\mathbf{I}$), the model equals the Bayesian PCA (Bishop, 1999) with the specific structure in $\mathbf{W}$. We use $\mathbf{\Sigma}^{(m)} = \sigma_m^2\mathbf{I}$ with a Gamma prior for the noise precisions $\tau_m = \sigma_m^{-2}$. That is, we make the PCA assumption separately for both data sets. However, it is important to remember that the model still allows for dependencies between the features of both views, by modeling them with $\mathbf{B}^{(m)}\mathbf{z}^{(m)}$. Hence, the spherical noise covariance does not restrict the flexibility of the model but decreases the number of parameters. Alternatively, we could allow each dimension to have their own variance parameter as in factor analysis models; this could be useful when the scales of the variables are very different.

Since efficient inference solutions are available for regular factor analysis, the only challenge in learning the BIBFA model is in obtaining the right kind of structure for $\mathbf{W}$. We solve the BIBFA model by doing inference directly for (5) and learn the structure of $\mathbf{W}$ by imposing group-wise sparsity for the components (columns of $\mathbf{W}$), which results in the model automatically converging to a solution that matches (4) (up to an arbitrary re-ordering of the columns). In other words, we do not directly specify the matrices $\mathbf{A}^{(m)}$ and $\mathbf{B}^{(m)}$, but instead learn a single $\mathbf{W}$ matrix. To implement the group-wise sparsity, we divide the variables in $\mathbf{x}$ into two groups corresponding to the two data sets, and construct a prior that encourages sparsity over these groups. For each component $\mathbf{w}_k$ the elements corresponding to one group are either pushed all towards zero, or are all allowed to be active. Recently Jia et al. (2010) introduced a similar sparsity constraint for learning factorized latent spaces; our approach can be seen as a Bayesian realization of the same idea, applied to canonical correlation analysis.

It turns out that the correct form of sparsity can easily be obtained by a simple extension of the ARD prior used for component selection in many Bayesian component models, including the Bayesian CCA described in the previous section. We define the group-wise ARD as

$$p(\mathbf{W}) = \prod_{m=1}^{2} \text{ARD}(\mathbf{W}^{(m)}|\alpha_0, \beta_0),$$

with separate ARD prior for each $\mathbf{W}^{(m)}$. Here $\mathbf{W}^{(1)}$ denotes the first D_1 rows of $\mathbf{W}$ and $\mathbf{W}^{(2)}$ refers to the remaining D_2 rows. Similarly to how ARD has earlier been used to choose the number of components, the group-wise ARD makes unnecessary components $\mathbf{w}_k^{(m)}$ inactive for each of the views separately. The components needed for modeling the shared response will have small $\boldsymbol{\alpha}_k^{(m)}$ (that is, large variance) for both views, whereas the view-specific components will have small $\boldsymbol{\alpha}_k^{(m)}$ for the active view and a large one for the inactive one. Finally, the model still selects automatically the total number of components by making both views inactive for unnecessary components. To our knowledge, Virtanen et al. (2011) is the first to consider this simple extension of ARD into multiple groups. Later Virtanen et al. (2012a) and Damianou et al. (2012) discussed the prior in more detail, presenting also extensions to more than two groups.

In practice, the elements of the inactive $\mathbf{w}_k^{(m)}$ will not become pushed exactly to zero, but instead to very small values. For most applications of BIBFA this is not a problem, since we need not identify the components. For example, the demonstration in Section 7.2 that uses CCA for predicting one view from the other automatically ignores the view-specific components even when $\mathbf{w}_k^{(m)}$ is not

exactly zero. Similarly, the explorative data analysis experiment illustrated in Figures 8 and 9 is invariant to the actual components and only relies on the total amount of contribution each feature has on the shared variation. However, in case the individual components are needed, the structure of (4) can be obtained by thresholding small values to zero, for example based on the amount of relative variance explained, and re-ordering the components. The problem is essentially identical to choosing the threshold for PCA models, and hence the techniques suggested for Bayesian PCA apply directly. The ARD prior efficiently pushes the variance of inactive components towards zero, and hence selecting the threshold is often easy in practice.

We apply variational approximation for inference, using the factorized distribution

$$q(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y}) = \prod_{n=1}^{N} q(\mathbf{y}_n) \prod_{m=1}^{2} \left(q(\tau_m) q(\boldsymbol{\alpha}^{(m)}) \right) \prod_{d=1}^{D_1+D_2} q(\mathbf{W}_{d,:}).$$

Here $\mathbf{W}_{d,:}$ corresponds to the dth row of $\mathbf{W}$, a vector spanning over the K different components. The different terms $q(\cdot)$ in the approximation are updated alternatingly to minimize the Kullback-Leibler divergence $D_{KL}(q,p)$ between $q(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y})$ and $p(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y}|\mathbf{X})$ to obtain an approximation best matching the true posterior. Equivalently, the task is to maximize the lower bound

$$\mathcal{L}(q) = \log p(\mathbf{X}) - D_{KL}(q,p) = \int q(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y}) \log \frac{p(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y}, \mathbf{X})}{q(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y})}$$

for the marginal likelihood, where the integral is over all of the variables in $q(\mathbf{W}, \tau_m, \boldsymbol{\alpha}^{(m)}, \mathbf{Y})$. Since all priors are conjugate, variational optimization over $q(\cdot)$, constrained to be probability densities, automatically specifies the functional form of all of the terms. Furthermore, we get closed-form updates for each of them, conditional on the choices for all other terms, resulting in straightforward update rules for an EM-style algorithm. The details are given in Appendix A.

As mentioned in Section 3.3, the model is unidentifiable with respect to linear transformations of $\mathbf{y}$, and only the prior $p(\mathbf{y})$ is influenced by the allocation of the components into shared and view-specific ones. The above procedure for learning the component complexities via group-wise ARD tremendously helps with the latter issue; even though the likelihood part would be equal for a model that splits a true shared component into two separate ones, the variational lower bound will be considerably better for a choice that does not need to replicate the latent variables. In particular, being able to completely drop a component means that the Kullback-Leibler divergence between $q(\mathbf{y}_k)$ and $p(\mathbf{y}_k)$ becomes essentially zero; this advantage would be lost if a shared component was replicated as two view-specific ones.

Interestingly, the variational approximation solves implicitly also the rotational invariance. Even though the likelihood is invariant with respect to right-multiplication of $\mathbf{W}$ with any invertible matrix $\mathbf{R}$, the variational lower bound is maximal for a specific rotation. Since the $\mathbf{R}$ does not influence the likelihood, it can only improve the lower bound by transforming the approximation into one that best matches the prior distribution. The prior, in turn, assumes independent latent variables, implying that the optimal solution will result in an $\mathbf{R}$ that makes the latent variables of the posterior approximation also maximally independent.[4] The model is hence identified in the same sense as

4. For the current model, the independence corresponds to orthogonality of the latent variables. We prefer to use the phrase independence as it better fits the notion of assuming independent latent variables and may become more precise in extensions; for other priors and inference algorithms independence need not equal orthogonality.

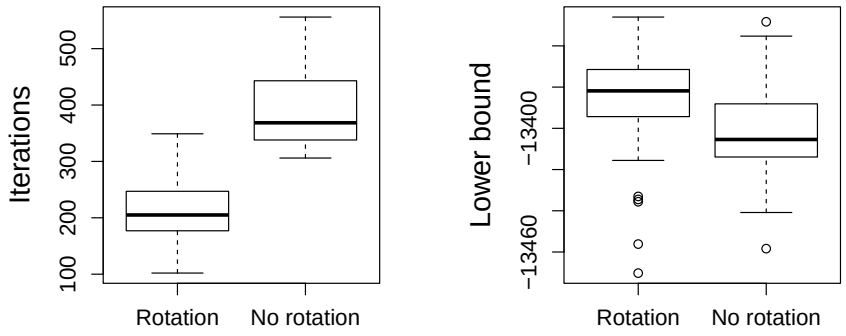


Figure 2: Illustration of the effect of parameter-expanded VB, where the variational lower bound is explicitly optimized with respect to a linear transformation $\mathbf{R}$ to make the updates less correlated. The number of iterations until convergence is reduced clearly when optimizing for the rotation (left), while the lower bound is still on average slightly better (right). However, since $\mathbf{R}$ is optimized numerically the total computation time might still be smaller without the rotation optimization, depending on the data and parameter values (such as K_c). These plots were drawn for the data analyzed in Section 7.1.1 and Figure 9; the boxplots show the results of 50 runs with random initialization. The overall picture is similar for other data sets, but the actual differences vary depending on the complexity of the data. In practice, we recommend trying both with and without the rotation for each data type, and to choose the solution resulting in a better lower bound.

the classical CCA solution is; the latent variables are assumed orthogonal, instead of assuming orthogonal projections (like in PCA).

That property also allows deriving a more efficient algorithm for optimizing the variational approximation, following the idea of parameter-expanded variational Bayes (Qi and Jaakkola, 2007; Luttinen and Ilin, 2010). We introduce explicit parameter $\mathbf{R} \in \mathbb{R}^{K_c \times K_c}$ in the approximation and optimize the lower bound also with respect to it. Transforming the parameters with $\mathbf{R}$ improves the convergence speed dramatically, due to lower correlations between the EM updates for $\mathbf{y}$ and $\mathbf{W}$, and often also results in slightly better lower bound. Both of the properties are illustrated in Figure 2, and the details for how the rotation can be optimized are given in Appendix B.

In summary, the above model formulation with the associated variational approximation provides a fully Bayesian treatment for the IBFA model. It can also be used for solving the CCA problem with a low-rank assumption for the view-specific noise. The model automatically selects the complexity of the three separate component types through a group-wise ARD prior applied for a joint FA model (that additionally shares the noise variances for all variables within a view), and disambiguates between different rotations by maximizing the orthogonality of the latent variables for improved interpretability and computational efficiency. Open-source implementation of the model, written in the R language, is available in CRAN: `http://cran.r-project.org/package=CCAGFA`.

4.3 On the Choice of Group-sparsity Prior

The above derivation uses group-wise ARD for inferring the component activities. This particular choice is, however, not the only possibility. In fact, any reasonable prior that results in group-wise sparse projection matrix $\mathbf{W}$ could be adopted. Here we briefly discuss possible alternatives, and derive one practical implementation that uses sampling-based inference instead of the variational approximation described above.

BIBFA is essentially a linear model for the concatenation of the two sources, made interpretable by the group-wise sparsity. Hence, a sufficient requirement for a model to implement the BIBFA concept is that it can make $\mathbf{W}_{:,k}$ sparse in the specific sense of favouring solutions where all of the elements corresponding to the first D_1 or the last D_2 dimensions (or both) are driven to zero. This can be achieved in two qualitatively different ways, called weak and strong sparsity by Mohamed et al. (2012). The ARD prior is an example of the former, a continuous sparsity-inducing prior that results in elements that are close to zero but not exactly so. Other priors that induce weak sparsity could also be considered, such as the group-wise extensions of the Laplace and scale-mixture priors Archambeau and Bach (2009) and Guan and Dy (2009) used for sparse PCA, but as we will demonstrate in the empirical experiments, already the ARD prior works well. Hence, we use it as a representative of weak sparsity priors. As general properties, such priors allow continuous inference procedures that are often efficient, but it is not always trivial to separate low-activity components from inactive ones for interpretative purposes. This is because the elements are not made exactly zero even for the components deemed inactive, but instead the values are pushed to very small values.

In our applications in Section 7, we do not need to accurately identify the active components, since already near-zero effects become irrelevant for the predictive measures used. In case more precise determination of the active components is needed, it may be better to switch to strong sparsity, using priors that provide exact zeroes in $\mathbf{W}$. For this purpose, we here extend the element-wise sparse factor analysis model of Knowles and Ghahramani (2011) for the BIBFA setup. The original model is based on the spike-and-slab prior, where each element of $\mathbf{W}$ is drawn from a two-component prior. One of the components, the spike, is a delta distribution centered at zero, whereas the other, the slab, is a Gaussian distribution. Hence, each element can either become exactly zero or is drawn from a relatively noninformative distribution. To create a BIBFA method based on this idea, we introduce the group-wise spike-and-slab prior with the prior

$$p(\mathbf{W},\mathbf{H},\alpha_b,\pi|\alpha_0,\beta_0) = p(\pi)p(\mathbf{H}|\pi)\prod_{m=1}^{2} p(\mathbf{W}^{(m)}|\mathbf{H}_{m,:},\boldsymbol{\alpha}^{(m)})p(\boldsymbol{\alpha}^{(m)}|\alpha_0,\beta_0), \tag{6}$$

$$\begin{aligned}
\mathbf{W}^{(m)}_{d,k}|\mathbf{H}_{m,k},\boldsymbol{\alpha}^{(m)}_k &\sim \mathbf{H}_{m,k}\mathrm{N}(0,(\boldsymbol{\alpha}^{(m)}_k)^{-1}) + (1-\mathbf{H}_{m,k})\delta_0,\\
\mathbf{H}_{m,k}|\pi_m &\sim \mathrm{Bernoulli}(\pi_m),\\
\pi_m &\sim \mathrm{Beta}(1,1),\\
\boldsymbol{\alpha}^{(m)}_k|\alpha_0,\beta_0 &\sim \mathrm{Gamma}(\alpha_0,\beta_0),
\end{aligned}$$

where δ_0 denotes a point-density at zero. That is, the view-specific π_m tells the probability for a component to be active, binary $\mathbf{H}_{m,k}$ drawn from the Bernoulli distribution tells whether component k is active in view m, and finally $\mathbf{W}_{:,k}$ is either exactly zero or its elements are all drawn independently from a Gaussian distribution with precision $\boldsymbol{\alpha}^{(m)}_k$ depending on whether $\mathbf{H}_{m,k}$ is zero or one, respectively.

For inference, we use Gibbs sampler by Knowles and Ghahramani (2011) with small modifications. In particular, the elements of $\mathbf{H}$ now depend on D_m features instead of just a single one. This, however, does not make the inference more complicated; the dimensions are independent and hence we get the conditional density by multiplying element-wise terms that still integrate $\mathbf{W}^{(m)}_{d,k}$ out. Another change is motivated by the fact that we only need to estimate a $2 \times K$ matrix $\mathbf{H}$, instead of a $D \times K$ matrix needed for element-wise sparsity. Since we only have two choices for each component, it does not make sense to use the Indian Buffet Process (IBP) prior for $\mathbf{H}$; there cannot be any interesting structure in $\mathbf{H}$. Hence, we simplify the model to merely draw each entry of $\mathbf{H}$ independently. The details of the resulting sampler are presented in Appendix D.

4.4 Model Summary

We will next briefly summarize the Bayesian CCA model and lay out the two alternative inference strategies. These methods will be empirically demonstrated and compared in the following sections.

4.4.1 BAYESIAN CCA WITH LOW-RANK COVARIANCE, OR BAYESIAN IBFA (BIBFA)

The assumption of low-rank covariance results in the IBFA model of (5). Efficient inference is done in the factor analysis model for $\mathbf{x} = [\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$ with group-wise sparsity prior for the projection matrix:

$$\begin{aligned} \mathbf{y} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x} &\sim \mathrm{N}(\mathbf{W}\mathbf{y}, \boldsymbol{\Sigma}), \\ \mathbf{W}^{(m)} &\sim \mathrm{ARD}(\alpha_0, \beta_0). \end{aligned} \tag{7}$$

Here $\mathbf{W}^{(m)}$ denotes the dimensions (rows) of $\mathbf{W}$ corresponding to the mth view, and $\boldsymbol{\Sigma}$ is a diagonal matrix with D_1 copies of τ_1^{-1} and D_2 copies of τ_2^{-1} on its diagonal. The noise precision parameters are given Gamma priors $\tau_m \sim \mathrm{Gamma}(\alpha_0^\tau, \beta_0^\tau)$. Inference for the model is done according to the updates provided in Appendices A and B.

An alternative inference scheme replaces the above ARD prior with the group-wise spike-and-slab prior of (6) and draws samples from the posterior using Gibbs sampling.

4.4.2 BAYESIAN CCA WITH FULL COVARIANCE (BCCA)

The Bayesian CCA as presented by Wang (2007) and Klami and Kaski (2007) models the view-specific variation with a free covariance parameter. The full model is specified as

$$\begin{aligned} \mathbf{z} &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}^{(m)} &\sim \mathrm{N}(\mathbf{A}^{(m)}\mathbf{z}, \boldsymbol{\Psi}^{(m)}), \\ \mathbf{A}^{(m)} &\sim \mathrm{ARD}(\alpha_0, \beta_0), \\ \boldsymbol{\Psi}^{(m)} &\sim \mathrm{IW}(\mathbf{S}_0, \nu_0), \end{aligned} \tag{8}$$

and inference follows the variational updates provided by Wang (2007). When $\mathbf{A} = [\mathbf{A}^{(1)}; \mathbf{A}^{(2)}]$ is drawn from a single ARD prior, the lower bound can analytically be optimized with respect to a rotation $\mathbf{R}$ (see Appendix C), resulting in considerable speedup.

The variational approximations for both BCCA and BIBFA are deterministic and will converge to a local optimum that depends on the initialization. We initialize the model by sampling the latent variables from the prior, and recommend running the algorithm multiple times and choosing the solution with the best variational lower bound. In the experiments we used 10 initializations.

All of the above require pre-specifying the number of components K. However, since the ARD prior (or the spike-and-slab prior for the Gibbs sampler variant) automatically shuts down components that are not needed, the parameter can safely be set large enough; the only drawback of using too large K is in increased computation time. In practice, one can follow a strategy where the model is first run with some reasonable guess for K. In case all components remain active, try again with a larger K.

5. Illustration

In this section we demonstrate the BIBFA model on artificial data, in order to illustrate the factorization into shared and data set-specific components, as well as to show that the inference proceduree converge to the correct solution. Furthermore, we provide empirical experiments demonstrating the importance of making the low-rank assumption for the view-specific noise, in terms of both accuracy and computational speed, by comparing BIBFA (7) with BCCA (8).

The results are illustrated primarily from the point-of-view of the variational inference solution; the variational approximation is easier to visualize and compare with alternative methods. The Gibbs sampler produced virtually identical results for these examples, as demonstrated in Figures 4 and 6.

5.1 Artificial Example

First, we validate the model on artificial data drawn from a model from the same model family, with parameters set up so that it contains all types of components (view-specific and shared components). The latent signals $\mathbf{y}$ were manually constructed to produce components that can be visually matched with the true ones for intuitive assessment of the results. Also the $\boldsymbol{\alpha}^{(m)}$ parameters, controlling the activity of each latent component in both views, were manually specified. The projections $\mathbf{W}$ were then drawn from the prior, and noise with fixed variance was added to the observations.

The left column of Figure 3 illustrates the data generation, showing the four latent components, two of which are shared between the two views. We generated $N = 100$ samples with $D_1 = 50$ and $D_2 = 40$ dimensions, and applied the BIBFA model with $K = 6$ components to show that it learns the correct latent components and automatically discards the excess ones. The results of the variational inference are shown in the middle column of Figure 3; the Gibbs sampler produces virtually indistinguishable results. The learned matrix of $\boldsymbol{\alpha}$-values (and the corresponding elements in $\mathbf{W}$) reveals that the model extracted exactly four components, correctly identifying two of them as shared components and two as view-specific ones (one for each data set). The actual latent components also correspond to the ones used for generating the data. The components are presented in the order returned by the model, which is invariant to the order. We also see how the model is invariant to the sign of $\mathbf{y}$, but that it gives the actual components instead of a linear combination of those, demonstrating that the variational approximation indeed solves the rotational disambiguity that would remain for instance in the maximum likelihood solution.

The BIBFA results are further illustrated in Figure 4. The plot shows the approximate posterior for two of the model parameters, namely the residual noise levels τ_m, demonstrating that the model

has found the true generating parameters. We see that the true parameter values fall nicely within the posterior and both the variational approximation and the Gibbs sampler provide almost the same posterior. We performed similar comparison for all parameters, and the results are also similar, indicating that both variants model this simple data correctly. Furthermore, the results do not here depend notably on the initialization; the model always converges to the right solution. Finally, we also studied that the model is robust with respect to the number of components K. We reran the experiment with multiple values of K upto 30, always getting the same result where only 4 components remain active. This demonstrates that we can safely overestimate the number of components, the only negative side being increased computation time.

5.2 Quantitative Comparison

Next we proceed to quantitatively illustrating that the solution obtained with the BIBFA model is superior to both classical CCA and earlier Bayesian CCA variants not making the low-rank assumption for view-specific noise. We use the same data generation process as above, but explore the two main dimensions of potential applications by varying the number of samples N and the data dimensionality D_m. For the easy case of large N and small D all methods work well since there is enough information for determining the correlations accurately. Classical CCA has an edge in computational efficiency, due to the analytic solution, but also the Bayesian variants are easy to compute since the complexity is only linear in N. Below we will study in more detail the more interesting cases where either D is large or N is small, or both.

We generated data with 4 true correlating components drawn from the prior, drawing N independent samples of $D_1 = D_2$ dimensions, and measure the performance by comparing the average of the four largest correlations ρ_k, normalized by the ground truth correlations. For the Bayesian variants we estimate the correlation between the expectations $\langle \mathbf{Y}_{k,:} | \mathbf{X}^{(1)} \rangle$ and $\langle \mathbf{Y}_{k,:} | \mathbf{X}^{(2)} \rangle$ that are easy to compute by a slight modification of the inference updates. Here $\mathbf{Y}_{k,:}$ contains the kth latent variable for all N samples. Note that $\mathbf{Y}_{k,:} | \mathbf{X}^{(m)}$ follows the prior for the components k that are switched off for the mth view.

We compare the two variants of the Bayesian CCA, denoting by BCCA a model parameterized with full covariance matrices (8) and by BIBFA the fully factorized model (7), with both classical CCA and a regularized CCA (RCCA). For BCCA we used $\nu_0 = D_m$ degrees of freedom and the scale matrix $\mathbf{S}_0 = 0.01 * \mathbf{I}$ to give a reasonably flat prior over the covariances, and for BIBFA we gave a flat prior $\alpha_0^\tau = \beta_0^\tau = 10^{-14}$ for the precisions; the other parameters for BCCA and BIBFA were identical. We followed Gonzales et al. (2008) as the reference implementation of a regularized CCA, but replaced the leave-one-out validation for the two regularization parameters with 20-fold cross-validation instead, after verifying that it does not result in statistically significant differences in accuracy compared to the proposed scheme. With leave-one-out validation the computational complexity of RCCA would be quadratic in N, which would have made it severely too slow. To keep the computational load manageable we further devised a two-level grid for choosing the two regularization parameter values: We first try values in a loose two-dimensional grid of 7×7 values and then search for the optimal value in a dynamically created 7×7 grid around the best values.

Figure 5 illustrates the accuracy of the four methods for various scenarios, showing the relative correlations for both training and test data. The main observation is that BIBFA and RCCA are consistently the best methods, with BIBFA having a slightly better accuracy. Classical CCA without regularization breaks down completely for large D/N ratios, as does BCCA with full covariance

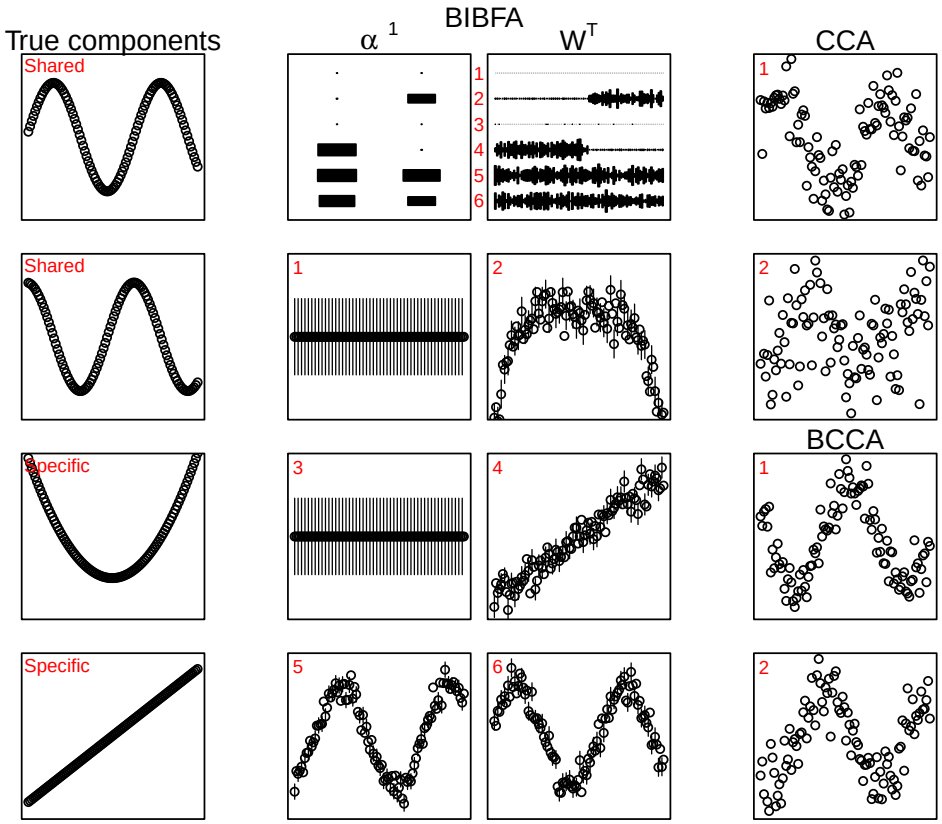


Figure 3: Illustration that Bayesian IBFA (BIBFA) finds the correct underlying latent components. The left column shows four components in the generated data, the first two being shared between the two views and the last two being specific to just one view. BIBFA finds all four components while ignoring the excess ones. The top row of the BIBFA block shows the Hinton-plot (the area of each block indicating the value) of the component variances and the elements of $|\mathbf{W}|$, and the remaining six plots show the estimated latent variables, the small red numbers indicating the link between the latent components and the rows of $\mathbf{W}^T$. Components 5 and 6 are shared, revealed by non-zero variance for both views, components 2 and 4 are the two view-specific components, and the unnecessary components 1 and 3 have been suppressed to the prior in the sense that their mean and variance match those of the prior. The small lines depict one standard deviation, revealing that the model is more confident on its predictions for the shared components, due to more data ($D_1 + D_2$ features compared to just D_1 or D_2) available for inferring them. The classical CCA (top two plots in the right column), which is only applicable for extracting the shared components, finds much noisier versions of the components, and for slightly higher dimensionality would return only noise. Baysian CCA with full covariance matrices (bottom two plots in the right column) does better than classical CCA, but does not capture the components as well as BIBFA. For all methods the latent variables have here been estimated for held-out test data.

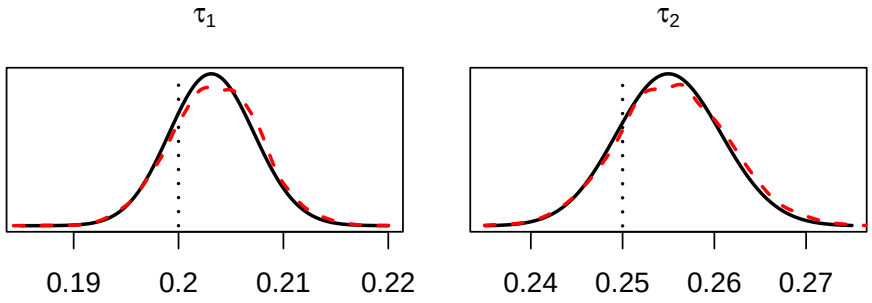


Figure 4: Illustration that Bayesian IBFA (BIBFA) finds also the correct posterior distributions for the model parameters. The plot shows the approximative distributions $q(\tau_m)$ for the variational approximation (solid black line) as well as the posterior obtained with the Gibbs sampling algorithm of the spike-and-slab variant (dashed red line), revealing how both capture the true generating values denoted by dotted black lines. The modes are not exactly at the true value due to the small sample size, but both inference strategies provide the same result.

matrices. Both of these are understandable observations since the methods estimate $D_m \times D_m$ covariance matrices with few or no constraints. To further illustrate the behavior of BIBFA, we plot in Figure 6 the estimated number of shared and view-specific components for both the variational and Gibbs sampling variants. For the purpose of this illustration, we considered a component of the variational inference solution to be active if $\boldsymbol{\alpha}_k^{(m)}$ was below 50 (the true value for active components was 1) and shared if the relative variance of $\boldsymbol{\alpha}_k^{(1)}$ and $\boldsymbol{\alpha}_k^{(2)}$ was below 10, whereas for the Gibbs sampler $\mathbf{H}_{m,k}$ directly reveals the activities. We see that both inference algorithms are conservative in the sense that for very small sample sizes they miss some of the components, using the residual noise term to model the variation that cannot be reliably explained with so limited data. Starting from $N = 64$ (which is still smaller than the data dimensionality for two of the plots) the ranks are estimated correctly.

Another important dimension is the computational time. CCA, RCCA and BCCA all require inverting $D_m \times D_m$ covariance matrices, which results in $O(D_m^3)$ complexity, whereas BIBFA is linear in N and D_m and cubic only with respect to K. The computational times are illustrated in Figure 7, revealing clearly how the lower complexity of BIBFA realizes as faster computation. For very small D the regularized CCA solution is slightly faster than BIBFA, but for large D it becomes impractically slow, even with our faster cross-validation scheme. The overall trend hence is that despite its iterative inference algorithm BIBFA is a much faster solution for high-dimensional CCA problems than regularized CCA solutions that require matrix inversion and cross-validation for tuning the regularization.

The overall summary of these illustrations is that the BIBFA model solves the CCA problem well, even in cases (large dimensionality and/or small sample size) where regular CCA and Bayesian CCA with full covariance matrices do not work at all. Carefully regularized CCA finds the correlations roughly as well as BIBFA, but it is considerably slower for large dimensionalities and lacks interpretable view-specific components, and cannot be extended as easily to directions discussed in the next section. While K was here small, making the gap between BIBFA and the rest of the models

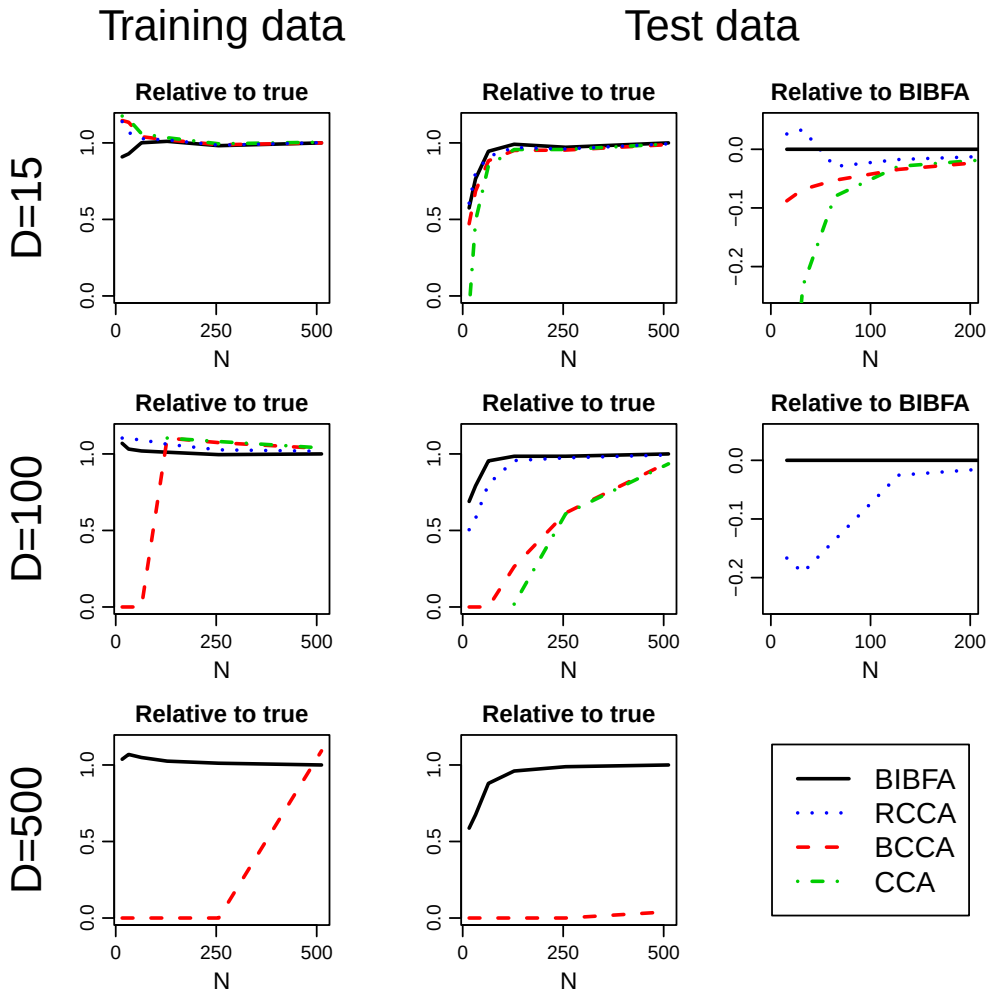


Figure 5: Illustration of the relative performance of the BIBFA model, Bayesian CCA with inverse-Wishart priors (BCCA), regularized CCA (RCCA) and classical CCA for various sample sizes (N; x-axis) and dimensionalities D (row). RCCA is missing for the last row due to too high computational cost, and CCA could not be computed for $D > N$. The first column shows the sum of the first four correlations (the data has four non-zero correlations) on the training data, normalized so that 1 matches the true value (y-axis). All methods but BIBFA overfit for small N and D, whereas BCCA severely underfits for small N and large D, not finding any reliable correlations. The second column shows the same measure for test data, revealing how BIBFA outperforms the other methods for all cases, except RCCA for very small N and D. The third column shows a zoomed inset for the most interesting region, this time normalized so that the result of BIBFA is used as the baseline, revealing more clearly the advantage BIBFA has over RCCA for all but the smallest samples sizes for $D = 15$.

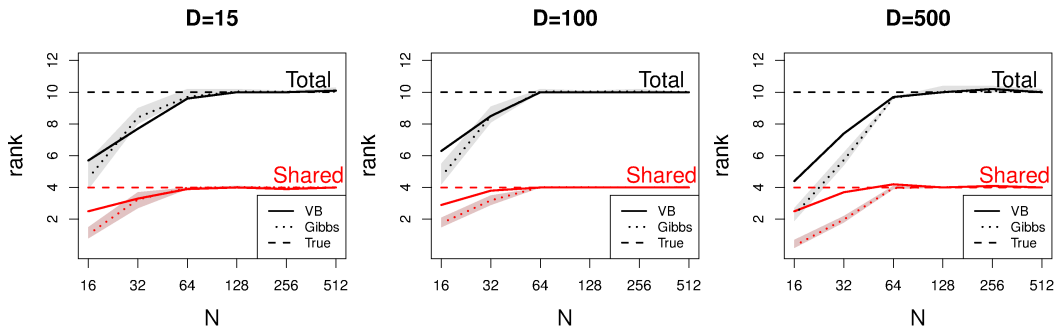


Figure 6: Learning the rank of the data. For reasonable number of samples both the variational approximation (VB) and the spike-and-slab sampler (Gibbs) learn the correct number of both shared components (red lines) and total components (black lines) for all three dimensionalities (subplots). The difference between these two curves corresponds to the sum of residual noise ranks of the two views, which are not shown to avoid cluttering the image. The solid lines correspond to the results of the variational approximation, averaged over 10 random initializations, whereas the dashed line shows the mean of the posterior samples for the Gibbs sampler and the shaded region covers the values between the 5% and 95% quantiles. The two inference algorithms perform roughly as well, and a notable observation is that both methods underestimate the number of components for very small sample sizes, especially for the higher dimensionalities. This is the correct behavior when there is not enough evidence to support the findings.

bigger than in most real applications, the empirical experiments with real-world data in Section 7 reveal that for plenty of practical applications with thousands of dimensions it is sufficient to use values of K in the range of tens. Hence, the computational advantage will hold in real applications as well, making BIBFA a feasible model for scenarios where D would be clearly too large for direct inversion of the covariance matrices.

6. Variants and Extensions

The key advantage of the Bayesian treatment, besides robustness for small sample sizes, is that it enables easy modifications and extensions. In this section we will review a number of extensions presented for the Bayesian CCA model, to provide an overview of the possibilities opened up by the probabilistic treatment of the classical model.

6.1 Modifying the Generative Model

Since the latent variable model is described through a generative process, it is straightforward to change the distributional assumptions in the model to arrive at alternatives designed for specific purposes. Typically these modifications will need to be accompanied by changes in the inference process that are not necessarily trivial, but without the probabilistic formulation extensions like these would be more difficult to keep consistent and justify.

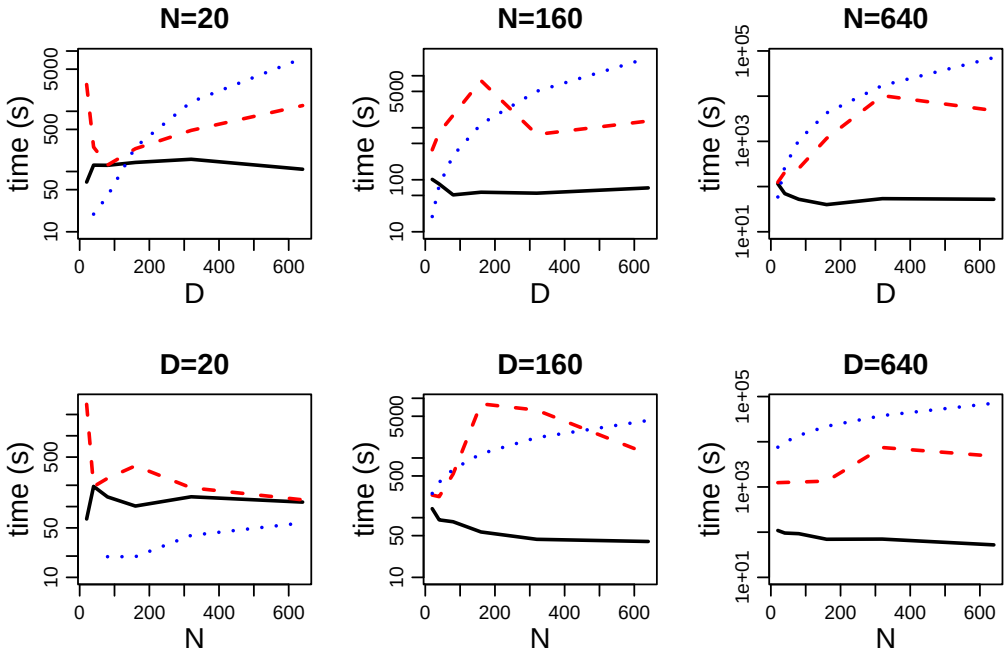


Figure 7: Illustration of the computational time (in seconds) for the BIBFA model (solid black line), Bayesian CCA with inverse-Wishart prior for covariances (BCCA; dashed red line), and regularized CCA (RCCA; dotted blue line). The Bayesian variants assume 10 random restarts, whereas the regularized CCA uses 20-fold cross-validation over a two-stage grid of the two regularization parameters, requiring a total of 20*(49+49) runs. The top row shows how regularized CCA becomes very slow for large dimensionality D, irrespective of the number of samples N. The bottom row shows the linear growth as a function of N for regularized CCA, effectively constant time complexity for BIBFA, and illustrates an interplay of N and D for the BCCA model (it needs more iterations for convergence when D is roughly N). For BIBFA the theoretical complexity is linear in both N and D, but the number of iterations needed for convergence depends on the underlying data in a complex manner and hence the trend is not visible here. Instead, for this data the computational time is almost constant.

The first improvement over the classical CCA brought by the probabilistic interpretation was to replace the Gaussian noise in (3) with the multivariate Student's t distribution (Archambeau et al., 2006). This makes the model more robust for outlier observations, since observations not fitting the general pattern will be better modeled by the noise term. The maximum likelihood solution provided for the robust CCA by Archambeau et al. (2006) was later extended to the Bayesian formulation with a variational approximation by Viinikanoja et al. (2010).

Klami et al. (2010) extended Bayesian CCA by generalizing from the Gaussian noise assumption to noise with any distribution in the exponential family. Using the natural parameter formulation of exponential family distributions, a generic formulation applicable for any choice was de-

rived. The solution was built on top of the Gibbs-sampling scheme of Klami and Kaski (2007), with considerable technical extensions to cope with the fact that conjugate priors are no longer justified.

Recently, Virtanen et al. (2012b) extended the IBFA-type modelling to count data, introducing a multi-view topic model that generates the observed counts similarly to how IBFA generates continuous data. That is, the model automatically learns topics that are shared between the views as well as topics specific to each view, using a hierarchical Dirichlet process (HDP; Teh et al., 2006) formulation.

Another line of extensions changes the prior for the projections $\mathbf{A}^{(m)}$. Archambeau and Bach (2009) presented a range of sparse models based on various prior distributions. They introduced sparsity priors and associated variational approximations for Bayesian PCA and the full IBFA model, but did not provide empirical experiments with the latter. Another sparse variant was provided by Fujiwara et al. (2009), using an element-wise ARD prior to obtain sparsity, though the method is actually not a proper CCA model since it does not model view-specific variation at all. Rai and Daumé III (2009) built similarly motivated sparse CCA models via a non-parametric formulation where an Indian Buffet Process prior (Ghahramani et al., 2007) is used to switch projection weights on and off. The same non-parametric prior also controls the overall complexity of the model. The inference is based on a combination of Gibbs and more general Metropolis-Hastings steps, but again the model lacks the crucial CCA property of separately modeling view-specific variation.

Leen and Fyfe (2006) and Ek et al. (2008) extended the probabilistic formulation to create Gaussian process latent variable models (GP-LVM) for modeling dependencies between two data sets. They integrate out the projections $\mathbf{A}^{(m)}$, giving a representation that enables replacing the outer product with a kernel matrix, resulting in non-linear extensions. Leen and Fyfe (2006) formulated the model as direct generalization of probabilistic CCA, whereas Ek et al. (2008) modeled explicitly also the view-specific variation. Recently, Damianou et al. (2012) extended the approach to a Bayesian multi-view model that uses group-wise sparsity to identify shared and view-specific latent manifolds for a GP-LVM model, using an ARD prior very similar to the one used by Virtanen et al. (2011) and here for BIBFA.

The conceptual idea of CCA has also been extended beyond linear transformation and continuous latent variables. As a practical example, multinomial latent variables provide clustering models. Both Klami and Kaski (2006) and Rogers et al. (2010) presented clustering models that capture the dependencies between two views with the cluster structure while modeling view-specific variation with another set of clusters. Recently, Rey and Roth (2012) followed the same idea, modeling arbitrary view-specific structure within the clusters with copulas. The Bayesian CCA approach has also been extended beyond vectorial data representations; van der Linde (2011) provided a Bayesian CCA model for functional data, building on the variational approximation.

Haghighi et al. (2008) and Tripathi et al. (2011) extended probabilistic CCA beyond the underlying setup of co-occurring data samples. They complement regular CCA learning by a module that infers the relationship between the samples in the two views, by finding close neighbors in the CCA subspace. This enables both computing CCA for setups where the pairing of (some of) the samples is not known but also applications where learning the pairing is the primary task. Recently, Klami (2012) presented a variational Bayesian solution to the same problem, extending BIBFA to include a permutation parameter re-ordering the samples.

Some related methods not described in the terminology of Bayesian CCA are also worth mentioning, due to the close relationship between both the task and the models. Singh and Gordon

(2008) introduced collective matrix factorization (CMF), where the task is to learn simultaneous matrix factorizations of the form $\mathbf{X}^{(m)} = \mathbf{V}^{(m)}\mathbf{Z}$ for multiple (in their application three) views, which is equivalent to the Bayesian CCA formulation. However, the exact definition of the noise additive to the factorization is crucial; BIBFA includes explicit components for modeling view-specific variation (or they are modeled with full covariance matrices as in the earlier Bayesian CCA solutions). CMF, in turn, assumes that all variation is shared, by factorizing the noise over the dimensions. Hence, CMF is more closely related to learning PCA for the concatenated data sources, and is incapable of separating the shared variation from the view-specific one. Recently, Agarwal et al. (2011) extended CMFs to localized factor model (LMF) that allows separate latent variables $\mathbf{z}^{(m)}$ for the views and models them as a linear combination of global latent profiles $\mathbf{u}$. This extended model is capable of implementing the CCA idea by selectively using only some of the global latent profiles for each of the views, though it is not explicitly encouraged and the authors do not discuss the connection. The residual component analysis by Kalaitzis and Lawrence (2012) is also closely related; it is a framework that includes probabilistic CCA as a special case. They assume a model where the data is already partly explained by some components and the rest is explained by a set of factors. By iteratively treating the view-specific and shared components as the explanatory factors they can learn the maximum likelihood solution of IBFA (and hence CCA) through eigen-decompositions, but their general formulation also applies to other data analysis scenarios.

Finally, a number of papers have discussed extensions of probabilistic CCA into more than two views. Already Archambeau and Bach (2009) mention that the generative model directly generalizes to more than two views, but they do not show that their inference solution would provide meaningful results for multiple views. Recently, Virtanen et al. (2012a) presented the first practical multi-view generalization of Bayesian CCA, coining the method group factor analysis (GFA), and Damianou et al. (2012) described a GP-LVM -based solution for multiple views. We do not discuss the multi-view generalizations further in this article, since the extended model cannot be directly interpreted as CCA; the concept of correlation does not directly generalize to multiple views.

6.2 Building Block in Hierarchical Models

The generative formulation of probabilistic models extends naturally to complex hierarchical representations. The Bayesian CCA model itself is already a hierarchical model, but can also be used as a building block in more complex hierarchical models. In essence, most Bayesian models operating on individual data sets can be generalized to work for paired data by incorporating a CCA-type latent variable formulation as a part of the model.

The first practical examples considered the simplest hierarchical constructs. Klami and Kaski (2007) introduced an infinite mixture of Bayesian CCA models, accompanied with a Gibbs sampling scheme. Later Viinikanoja et al. (2010) provided a variational approximation for mixtures of robust CCA models, resulting in a computationally more efficient algorithm for the same problem. These kinds of mixture models can be thought of as locally linear models that partition the data space into clusters and fit a separate CCA model within each. The clustering step is, however, integrated in the solution and is also influenced by the CCA models themselves.

Recently, some authors have used Bayesian CCA as an integral part in more complex hierarchical models. Huopaniemi et al. (2009) integrate a dimensionality reduction step into Bayesian CCA by clustering the original features and applying Bayesian CCA to the latent variables that aggregate features within a cluster, to make BCCA feasible for high-dimensional metabolomics data with

very limited sample size. Huopaniemi et al. (2010) addresses the same application domain, this time combining BCCA with multivariate analysis of variance (ANOVA). Nakano et al. (2011), in turn, created a hierarchical topic trajectory model (HTTM) by using CCA as the observation model in a hidden Markov model (HMM).

7. Applications

In this section we will discuss some of the applications of CCA, covering both general application fields and concrete problem setups. Some of the examples are from fields where the probabilistic variants have not been widely applied yet, but where the need for CCA-type modeling is apparent and the properties of the data suit well the strengths of the Bayesian approach.

We have divided the applications into two broad categories. The first category considers CCA as a tool for exploratory data analysis, seeking to evaluate the amount of correlation or dependency between various information sources or to illustrate which of the dimensions correlate with the other view. The other category uses CCA as a predictive model, building on the observation that CCA is a good predictor for multiple outputs, correctly separating the information useful for prediction from the noise.

7.1 Data Analysis

One of the key strengths of the Bayesian approach is that it enables justified analysis of small samples, providing estimates of the reliability of the results. For the application fields with plenty of data also the classical and kernel-based CCA solutions work well, as has been demonstrated for example in analysis of relationships between text documents and image content (Vinokourov et al., 2003). Hence, we focus here on applications where the amount of data is typically limited.

Life sciences are a prototypical example of a field with limited sample sizes. In many analysis scenarios the samples correspond to individuals, and high cost of measurements prevents collecting large data sets. There are also several application scenarios where the number of samples is restricted for biological reasons, for example when studying rare diseases or effects specific to an individual instead of a population.

CCA has received a lot of attention in analysis of omics data, including genomics measured with microarrays as well as proteomics and metabolomics measured by mass spectrometry. Huopaniemi et al. (2009, 2010) applied extensions of Bayesian CCA to find correlations between concentrations of biomolecules in different tissues and species to build "translational" models. In their studies, the samples correspond to individual mice and humans with a sample size in the order of tens, whereas the features correspond to concentrations of hundreds of lipids. Similar setups but still much more extreme ratios of D_m/N are encountered frequently in microarray analysis, where the dimensions correspond to tens of thousands of genes. Due to the limitations of the earlier models the Bayesian solutions have not yet been used with full strength in such applications.

Another typical application scenario is in brain activity analysis, where the samples typically correspond to time-slices of an experiment and the features span the brain activity measured either through BOLD (blood-oxygen-level-dependent) signal activity in small brain volumes called voxels (in functional magnetic resonance imaging fMRI) or through magnetometers on the scalp (magnetoencephalography MEG). Fujiwara et al. (2009) used sparse Bayesian CCA to predict the visual stimuli from fMRI data, and Koskinen et al. (2012) inferred the identity of short speech segments using mixture of robust Bayesian CCAs applied to MEG. Several authors have also applied classical

CCA or its multiset extensions for fMRI data; Ylipaavalniemi et al. (2009) studied the relationships between brain activity and naturalistic stimuli features, Deleus and Hulle (2011) explored functional connectivity between multiple brain areas, and Rustandi et al. (2009) integrated fMRI data of multiple subjects. Similar tasks could also be solved with the Bayesian variants, in particular with the BIBFA model.

7.1.1 ILLUSTRATION

To demonstrate the use of CCA in an exploratory data analysis scenario, we apply it to the problem of cancer gene prioritization based on co-occurring gene expression and copy number data (Lahti et al., 2012). DNA alterations frequent in cancers, measured by the copy number data, are known to induce changes in the expression levels, and hence cancer-associated genes can be mined by searching for such interactions.

One approach is to proceed over the whole genome in a gene-by-gene fashion, searching for correlations between the gene expression and copy number modification. Lahti et al. (2009) adapted CCA for this task, using it to estimate the amount of correlation inside short continuous windows of the genome. A collection of cancer and control patients are treated as samples, and the features are the genes within a window. For each window they computed so-called similarity-constrained CCA and labeled the genes within that window with the resulting correlation. That is, a gene is assumed to be cancer-related if CCA finds strong correlation within a small chromosomal window around it. This approach is one of the leading solutions for finding cancer-associated genes from integrated copy number and gene expression data, as shown in the recent comparison by Lahti et al. (2012).

For computing the association scores for N genes the above process requires running N separate CCA models, one for each neighborhood. Within each window, the CCA is ran for N' samples (the patients, on the order of 30-50 for typical data sets) and $D_x = D_y$ features (the genes within the window). The authors used window sizes of roughly 10-20, to guarantee that the number of samples exceeds the number of features, satisfying the usual requirement for CCA-style models.

The BIBFA model (7) has been specifically designed to tackle the issue of high dimensionality, and hence it allows a much more direct approach. Instead of measuring the amount of correlation for several small windows, we simply run CCA considering the patients as samples and all of the genes in the whole genome as features. Direct inspection of the weights in the shared components then reveals the cancer-associated genes; a high weight implies an association between the copy number and gene expression, relating the gene to the cancer under study.

We applied BIBFA, using the ARD prior and variational inference, on the two publicly available data sets used in the recent comparison of various integrative cancer-gene mining tools by Lahti et al. (2012), the Pollack and Hyman data sets. We repeated their experimental setup to obtain results directly comparable with their study, and measured the performance by the same measure, the area under curve (AUC) for retrieving known cancer genes (37 out of 4247 genes in Pollack, and 47 out of 7363 genes in Hyman). We ran the BIBFA model for K_c between 5 and 60 components and chose the model with the best variational lower bound, resulting in $K_c = 15$ for Hyman and $K_c = 40$ for Pollack. The full results of BIBFA and the comparison methods are reported in Figure 8, revealing that our method outperforms all of the alternatives for both data sets. The results would be similar for a wide range of values of K_c; for all values we beat the alternative methods. We also applied the Gibbs-sampler variant, which produced very similar results.

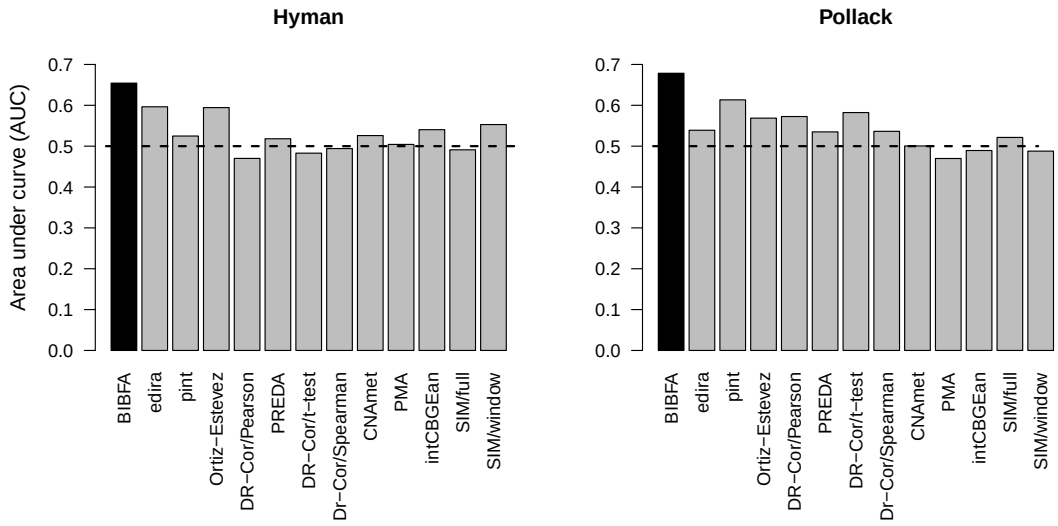


Figure 8: Comparison of AUC scores for the various methods in detecting cancer-related genes in genome-wide data. The BIBFA model ranks the genes based on the weight of that gene in both $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, and finds the cancer genes with better accuracy than any of the methods studied in the recent comparison by Lahti et al. (2012). The accuracies for all the other methods are taken from the publicly available results provided by the authors; see their article for the names and details of the methods. Of these methods, pint is the most closely related to ours. It screens the genome by computing CCA for narrow windows and ranks the genes according to the strength of dependency found within each window.

For ranking the genes we used the measure $s_g = \sum_{k=1}^{K_c} |\langle \mathbf{W}^{(1)}_{g,k} \rangle \langle \mathbf{W}^{(2)}_{g,k} \rangle|$. That is, for each component we multiply the expected projection vectors corresponding to gene expression and copy number change, to emphasize effects seen in both views. We then simply sum the absolute values of these quantities over all components, to reach the measure s_g for each gene g. Note that the view-specific components have no effect on the score, since either $\langle \mathbf{W}^{(1)}_{:,k} \rangle$ or $\langle \mathbf{W}^{(2)}_{:,k} \rangle$ will be zero for all genes. To further illustrate the approach, Figure 9 plots the quantity over one chromosome in the Pollack data (chromosome 17, the one most strongly associated with the breast cancer studied in that data) and compares the result with the activity profile provided by the similarity-constrained CCA model (also called pint, after the name of the public software implementation) of Lahti et al. (2009).

7.2 Multi-label Prediction

Another interesting application for CCA is in prediction. Even though the model is symmetric with respect to $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, it is surprisingly efficient as a predictive model. In a sense, the model can be seen as a combination of purely unsupervised and supervised learning; both of the views can be considered as supervising the other view, yet the model is (here) defined as a generative description of the whole data collection.

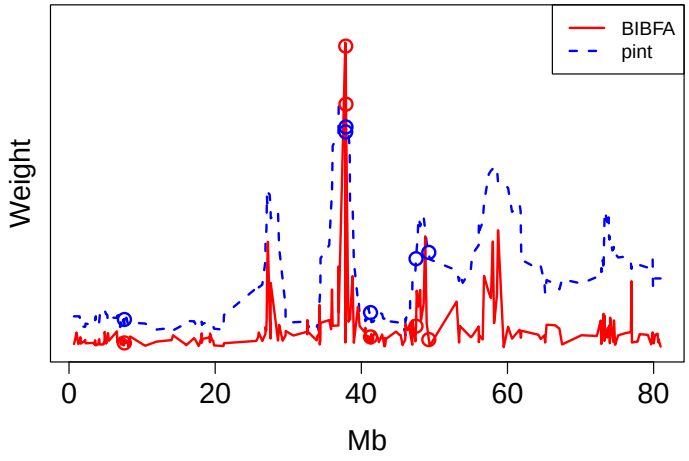


Figure 9: Illustration of the weights s_g (y-axis) learned for the 272 genes in Chromosome 17, depicted in the chromosomal order (x-axis), with BIBFA (red) and the window-based comparison method of pint (blue). Since the gene detection is based on the ranks of the genes, the two weight vectors have here been re-scaled into comparable scales for visual inspection. Both methods reveal a similar overall trend over the chromosome, capturing especially the known cancer-related region around 40Mb, but BIBFA does it in much more direct fashion: The BIBFA profile shown here is the outcome of a single run, whereas 272 applications of similarity-constrained CCA were needed to create the profile for pint. Note that BIBFA gets this result without making any assumptions about mutations in a region causing gene expression changes in the same region, showing the power of the algorithm but resulting in some noise in the result. Pint, in contrast, analyzes local neighborhoods and hence necessarily results in similar values for nearby chromosomal regions. The six known cancer genes are marked with circles. Neither method captures all of them, but BIBFA finds the top two cancer genes at higher ranks. On the other hand, BIBFA seems to here miss some cancer-related genes around the 47Mb region; there is no guarantee that the method would always be more accurate, but the numerical comparisons in Figure 8 show that it on average outperforms pint.

The CCA model is closely related to multiple regression. Breiman and Friedman (1997) showed that CCA is particularly efficient in multiple regression when the response variables are correlated. In recent years, for example Ji et al. (2008), Rai and Daumé III (2009), and Sun et al. (2011) have all demonstrated good predictive performance for CCA-type models in multi-output prediction or multiple regression tasks.

The Bayesian formulation through the BIBFA model helps in understanding why CCA works so well for multiple regression tasks. The predictive distribution of interest is $p(\mathbf{x}^{(1)}|\mathbf{x}^{(2)})$, which cannot be computed in close form, but for which it is easy to obtain expectations from the variational

approximation. The mean prediction is given by

$$\langle \mathbf{x}^{(1)} | \mathbf{x}^{(2)} \rangle = \langle \mathbf{A}^{(1)} \mathbf{z} \rangle_{q(\mathbf{A}^{(1)})q(\mathbf{z}|\mathbf{x}^{(2)})} = \langle \mathbf{A}^{(1)} \rangle \langle \sigma_2^{-2} \rangle \boldsymbol{\Sigma} \langle \mathbf{A}^{(2)^T} \rangle \mathbf{x}^{(2)},$$
$$\text{where} \quad \boldsymbol{\Sigma} = \left(\mathbf{I} + \langle \sigma_2^{-2} \rangle \langle \mathbf{A}^{(2)^T} \mathbf{A}^{(2)} \rangle \right)^{-1}.$$

The notable observation is that neither $\mathbf{B}^{(1)}$ nor $\mathbf{B}^{(2)}$ (or the corresponding latent variables $\mathbf{z}^{(1)}$ or $\mathbf{z}^{(2)}$) appear in the formula. This implies the model correctly neglects variation specific to $\mathbf{X}^{(2)}$ when making the prediction, and additionally does not attempt to predict the variation in $\mathbf{X}^{(1)}$ that cannot be predicted. In other words, the model squeezes the prediction through the shared latent variables $\mathbf{z}$, the lower-dimensional representation capturing all the information flow from one data set to another. Note that in practice the prediction can be directly written in terms of $\mathbf{W}$ and $\mathbf{y}$, without needing to explicitly extract $\mathbf{A}^{(m)}$ and $\mathbf{B}^{(m)}$, since the zeroes induced in $\mathbf{W}$ due to the group-wise sparsity will cancel the unnecessary components out automatically.

Similar observation holds also for the Gibbs sampler variant using the spike-and-slab prior (6); again the mean prediction only depends on the shared components. However, making predictions with that model is considerably more time-consuming, since the predictions need to be averaged over the posterior samples. When making predictions for new samples, we need to store all of the posterior samples and then run the sampler again for each of those estimate the latent variables and the predicted $\mathbf{X}^{(1)}$. Due to this extra computational overhead for the sampler, we will next demonstrate the BIBFA in multi-label prediction tasks using only the variational inference variant.

7.2.1 ILLUSTRATION

To measure the multi-label classification accuracy of the BIBFA model we applied it on 10 benchmark data sets from the Mulan library (Tsoumakas et al., 2010), using the split to train and test samples given in the library. Each of these data sets includes several binary labels that are not mutually exclusive, and we encode them into $\mathbf{X}^{(1)}$ so that each column represents one label. Then the task becomes predicting $\mathbf{X}^{(1)}$ from $\mathbf{X}^{(2)}$. Since the labels are discrete, we feed the predicted values through a simple threshold filter, with the threshold for each class chosen to maximize the accuracy on the training data. The Bayesian formulation would enable integrating also more advanced ways of handling with the binary data (Klami et al., 2010), but here the primary purpose is to demonstrate the application of the basic principle instead of developing a fully-fledged multi-label prediction model.

We compare the BIBFA model (7) with both classical CCA and the standard Bayesian CCA with full covariance matrices (8), using the variational approximation of Wang (2007). For BCCA we set the maximal number of components to $K = \min(D_1, D_2, 50)$, and for classical CCA we chose the number of components by 10-fold cross-validation within the training set. For BIBFA we set K_c to the minimum of 100 and the number of components extracted by Bayesian PCA ran on the concatenation of the two data views. Overall, these choices constitute a fair way of selecting the model complexity for each of the methods. For BIBFA and BCCA we started the optimization from 10 different random initializations and chose the solution that resulted in the best lower bound for the training data, whereas CCA is a deterministic algorithm and always provides the global optimum. The regularized CCA model studied earlier in Section 5.2 was left out due to its immensely high computational cost for most of the data sets, but preliminary studies showed that it did not outperform even classical CCA on the ones with sufficiently few dimensions. Besides showing the

Data Set	D_1	D_2	N_{train}	BIBFA	CCA	BCCA	RML	RAKEL	MLKNN
emotions	6	72	391	0.223	0.232	0.329	0.225	0.223	**0.209**
scene	6	294	1211	0.105	0.332	0.162	0.127	0.115	**0.0953**
yeast	14	103	1500	0.202	0.205	0.211	-	0.233	**0.198**
genbase	27	1186	463	**9.3e-4**	-	**9.3e-4**	-	0.0011	0.0052
medical	45	1449	333	0.0124	-	0.0276	-	**0.0113**	0.0188
enron	53	1001	1123	**0.0465**	-	0.0607	-	0.0509	0.0514
mediamill	101	120	30933	0.0309	0.161	**0.0305**	-	0.0335	0.0314
bibtex	159	1836	4880	**0.0131**	0.0138	-	-	0.0144	0.0140
Corel5-k	374	499	4500	0.0094	0.0099	0.0098	-	0.0096	**0.0093**
delicious	983	500	12920	**0.0182**	0.0183	-	-	0.0185	0.0183

Table 1: Prediction errors (Hamming loss) for 10 benchmark data sets sorted by the increasing number of labels D_1. For each data set the error for the best method has been boldfaced. The proposed Bayesian inter-battery factor analysis model (BIBFA) outperforms the classical CCA and Bayesian CCA with full covariance matrices (BCCA) for almost all data sets. For cases with a large number of labels BIBFA outperforms also designated multi-label prediction models RAKEL and MLKNN, showing that modeling the dependencies between the labels helps more when the number of labels is high. The figures for the reverse multi-label prediction model (RML) were taken from Petterson and Caetano (2010), N_{train} is the number of training samples, and D_2 is the input dimensionality. The values missing for BCCA were excluded due to too long computation time (more than 5 hours per run), and classical CCA was not ran for data sets where the dimensionality of either view is higher than the number of samples (for the enron data, $D_2 > N$ for the cross-validation runs needed for setting the threshold).

relationships between the various CCA-based methods, we also compared BIBFA with three multi-label prediction models with publicly available code or results, RAKEL (Tsoumakas and Vlahavas, 2007) and MLKNN (Zhang and Zhou, 2007) as implemented in the Mulan library, and reverse multi-label prediction model by Petterson and Caetano (2010). For measuring the performance we use the Hamming distance between the predictions and the true labels, penalizing equally much for both false negatives and false positives.

Table 1 shows that BIBFA is the best of the CCA variants on all but one of the data sets. Furthermore, Table 2 demonstrates how it is again considerably faster than BCCA model, even though we analytically optimized for the rotation $\mathbf{R}$ in the Bayesian CCA model. BIBFA also outperforms the other comparison models systematically for the cases with very large number of labels (D_1), with the exception of the *Corel5-k* data set. This demonstrates that CCA-type models are particularly useful for multi-label prediction tasks with an extreme number of labels, most likely because more information can then be extracted from the dependencies between the labels. The improvements presented in this paper are needed especially for that domain, since BIBFA is most useful for analysis of high-dimensional data.

For cases with a low number of labels (below 20 for the first three data sets), MLKNN outperforms IBFA. This is understandable as it is a model specifically designed for multi-label prediction and it explicitly maximizes the prediction accuracy, in contrast to BIBFA that is a generative model

Method	emotions	scene	yeast	genbase	medical
BIBFA	3	13	2	11	14
BCCA	1	8	2	46	79
Method	enron	mediamill	bibtex	Corel5-k	delicious
BIBFA	13	33	14	4	26
BCCA	289	95	>300	104	>300

Table 2: Average computation times for BIBFA and BCCA (in minutes) until convergence (relative change of the lower bound below 10^{-6}). For small dimensionalities the computational demands of the methods are comparable, but for high dimensionality the BCCA model becomes infeasible.

for both data sources. Nevertheless, BIBFA outperforms the two other comparison methods even for data sets with few labels.

8. Discussion

In this paper we have reviewed the works on probabilistic and Bayesian canonical correlation analysis, with particular focus on the extensions made possible by the probabilistic interpretation. While the solutions presented here are linear, as opposed to the possibly nonlinear kernel-based CCA models (Hardoon et al., 2004), the extensions and the ease of including CCA as a sub-model in larger hierarchical models clearly showcase the importance of probabilistic treatment of the problem. Works by Fujiwara et al. (2009) and Huopaniemi et al. (2010) have recently demonstrated how the Bayesian solution has enabled analysis of life science data sets with very low sample sizes.

Besides reviewing the earlier work, we introduced a novel solution that that results in considerably more efficient inference for the Bayesian CCA model, especially for high-dimensional data. The key is to make a low-rank assumption for the noise specific to each data set, which results in re-formulation of CCA as a more complex latent variable model called inter-battery factor analysis (IBFA) in the statistics literature (Tucker, 1958). While the extended model seems more complex due to having more unknown latent variables, it has the advantage of diagonal noise that reduces the risk of overfitting and simplifies the computation to the extent that it is actually much more efficient to learn the Bayesian IBFA (BIBFA) model than it is to directly learn the Bayesian CCA solution.

The computational difficulties stemming from introducing the extra latent variables are solved by clever usage of group-wise sparsity assumption. Instead of explicitly instantiating several latent variables, we re-cast the IBFA model as a straightforward joint factor analysis model with a specific prior driving the component group-wise sparse, showing how the resulting model is equivalent to IBFA. For inference we proposed two alternative solutions with alternative sparsity-inducing priors. One uses parameter expanded variational approximation with automatic relevance determination (ARD) prior, whereas the other uses Gibbs sampling with spike-and-slab prior. Both variants work well in practice, and usually seem to produce very similar results.

Given the efficient inference solution we believe the necessary tools for real-world application of Bayesian CCA are now available, making the approach feasible for scenarios that were previ-

ously not possible to solve. In this work we demonstrated how earlier serial computation of several low-dimensional CCA models could be replaced by single use of BIBFA for the original high-dimensional data in the task of extracting cancer-related genes. Similar conceptual shifts should be possible for other domains as well, in particular in life sciences where the sample sizes are typically in the order of tens while the dimensionality may be thousands or even larger.

Acknowledgments

We acknowledge the support from Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, grant no 251170, and additionally the decision number 133818), the aivoAALTO research project of Aalto University, and PASCAL2 European Network of Excellence (ICT 216886).

Appendix A. Variational Updates for the BIBFA Model

The joint likelihood of the BIBFA model is

$$p(\mathbf{X}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{Y}, \boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \boldsymbol{\tau}^{(1)}, \boldsymbol{\tau}^{(2)}) = \prod_{m=1}^{2} p(\mathbf{W}^{(m)}|\boldsymbol{\alpha}^{(m)}) p(\boldsymbol{\alpha}^{(m)}) p(\tau_m) \prod_{n=1}^{N} p(\mathbf{y}_n) p(\mathbf{x}_n^{(m)}|\mathbf{W}^{(m)}, \mathbf{y}_n, \tau_m),$$

where

$$\begin{aligned}
\mathbf{x}_n^{(m)} &\sim \mathrm{N}(\mathbf{W}^{(m)}\mathbf{y}_n, \tau_m^{-1}\mathbf{I}), \\
\mathbf{W}_{:,k}^{(m)} &\sim \mathrm{N}(\mathbf{0}, (\boldsymbol{\alpha}_k^{(m)})^{-1}\mathbf{I}), \\
\boldsymbol{\alpha}_k^{(m)} &\sim \mathrm{Gamma}(\alpha_0, \beta_0), \\
\tau_m &\sim \mathrm{Gamma}(\alpha_0, \beta_0), \\
\mathbf{y}_n &\sim \mathrm{N}(\mathbf{0}, \mathbf{I}).
\end{aligned}$$

We use mean field variational approximation to approximate the posterior, with the factorization

$$Q(\Theta) = q(\mathbf{Y}) \prod_{m=1}^{2} q(\mathbf{W}^{(m)}) q(\boldsymbol{\alpha}^{(m)}) q(\tau_m),$$

where Θ denotes all of the parameters and latent variables. For the latent variables we further assume column-wise independence (that is, the latent variables of observations are independent) and for projections row-wise independence (that is, each component is independent). The distributions are found by maximizing the lower bound of the marginal log-likelihood

$$\mathcal{L}(Q) = \int q(\Theta) \log \frac{p(\mathbf{X}, \Theta)}{q(\Theta)} d\Theta \leq \log p(\mathbf{X}),$$

and free-form optimization of the factored variables in $Q(\Theta)$ results in analytically tractable distributions due to conjugate priors. The forms of these distributions and the matrix-form updates

rules for efficient computations are given below, after introducing the necessary notation for the expectations.

For normal distribution $\mathbf{x} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the first moment is given as $\langle \mathbf{x} \rangle = \boldsymbol{\mu}$ and the second moment as $\langle \mathbf{x}\mathbf{x}^T \rangle = \langle \mathbf{x} \rangle \langle \mathbf{x} \rangle^T + \boldsymbol{\Sigma}$. We also introduce the notation $\langle \mathbf{X} \rangle = [\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_N]$ and $\langle \mathbf{X}\mathbf{X}^T \rangle = \langle \mathbf{X} \rangle \langle \mathbf{X} \rangle^T + N\boldsymbol{\Sigma}$ to indicate the moments of N independent draws of $\mathbf{x}$ with different $\boldsymbol{\mu}_n$ but the same covariance matrix. For gamma distribution $\alpha \sim \mathrm{Gamma}(\alpha_0, \beta_0)$ the expectation is given by $\langle \alpha \rangle = \frac{\alpha_0}{\beta_0}$.

For the projections we get row-wise independent factorial approximation

$$
\begin{aligned}
q(\mathbf{W}^{(m)}) &= \prod_{d=1}^{D_m} \mathrm{N}(\mathbf{W}^{(m)}_{d,:} | \boldsymbol{\mu}_{\mathbf{W}^{(m)}_{d,:}}, \boldsymbol{\Sigma}_{\mathbf{W}^{(m)}}), \\
\boldsymbol{\Sigma}_{\mathbf{W}^{(m)}} &= (\langle \boldsymbol{\alpha}^{(m)} \rangle^{-1} + \langle \tau^{(m)} \rangle \langle \mathbf{Y}\mathbf{Y}^T \rangle)^{-1}, \\
\langle \mathbf{W}^{(m)} \rangle &= \mathbf{X}^{(m)} \langle \mathbf{Y} \rangle^T \boldsymbol{\Sigma}_{\mathbf{W}^{(m)}} \langle \tau^{(m)} \rangle,
\end{aligned}
$$

where $\langle \mathbf{W}^{(m)} \rangle^T = [\boldsymbol{\mu}_{\mathbf{W}^{(m)}_{1,:}}, ..., \boldsymbol{\mu}_{\mathbf{W}^{(m)}_{D_m,:}}]$, and for the latent variables the update is given by

$$
\begin{aligned}
q(\mathbf{Y}) &= \prod_{n=1}^{N} \mathrm{N}(\mathbf{y}_n | \boldsymbol{\mu}_{\mathbf{y}_n}, \boldsymbol{\Sigma}_{\mathbf{Y}}), \\
\boldsymbol{\Sigma}_{\mathbf{Y}} &= (\mathbf{I} + \sum_{m=1}^{2} \langle \tau^{(m)} \rangle \langle \mathbf{W}^{(m)^T} \mathbf{W}^{(m)} \rangle)^{-1}, \\
\langle \mathbf{Y} \rangle &= \sum_{m=1}^{2} \langle \tau^{(m)} \rangle \boldsymbol{\Sigma}_{\mathbf{Y}} \langle \mathbf{W}^{(m)} \rangle^T \mathbf{X}^{(m)}.
\end{aligned}
$$

For the ARD parameters the updates are

$$
\begin{aligned}
q(\boldsymbol{\alpha}^{(m)}) &= \prod_{k=1}^{K} \mathrm{Gamma}(\boldsymbol{\alpha}^{(m)}_k | a^{(m)}_{\boldsymbol{\alpha}_k}, b^{(m)}_{\boldsymbol{\alpha}_k}), \\
a^{(m)}_{\boldsymbol{\alpha}_k} &= \alpha_0 + D_m/2, \\
b^{(m)}_{\boldsymbol{\alpha}_k} &= \beta_0 + \langle \mathbf{W}^{(m)^T} \mathbf{W}^{(m)} \rangle_{k,k}/2,
\end{aligned}
$$

where $\langle \boldsymbol{\alpha}^{(m)}_k \rangle = \frac{a^{(m)}_{\boldsymbol{\alpha}_k}}{b^{(m)}_{\boldsymbol{\alpha}_k}}$. Finally, for the noise precision parameters we have

$$
\begin{aligned}
q(\tau_m) &= \mathrm{Gamma}(a^{(m)}_\tau, b^{(m)}_\tau), \\
a^{(m)}_\tau &= \alpha_0 + ND_m/2, \\
b^{(m)}_\tau &= \beta_0 + (\sum_{dn} \mathbf{X}^{(m)^2}_{dn} + \mathrm{Trace}[\langle \mathbf{W}^{(m)^T} \mathbf{W}^{(m)} \rangle \langle \mathbf{Y}\mathbf{Y}^T \rangle] \\
&\quad - 2\mathrm{Trace}[\langle \mathbf{W}^{(m)} \rangle \langle \mathbf{Y} \rangle \mathbf{X}^{(m)^T}])/2.
\end{aligned}
$$

The algorithm proceeds by updating the parameters of the above factors sequentially until convergence. In the experiments included in this paper we determined convergence as relative change of $\mathcal{L}(Q)$ falling below 10^{-6}.

Appendix B. Parameter-expanded Variational Bayes for BIBFA

To make the correlations between the mean-field updates of BIBFA smaller, we can optimize $\mathcal{L}(Q)$ also with respect to a likelihood-invariant rotation $\mathbf{R}$

$$\mathbf{x}^{(m)} = \mathbf{W}^{(m)}\mathbf{y} = \mathbf{W}^{(m)}\mathbf{R}\mathbf{R}^{-1}\mathbf{y} = \mathbf{W}^{(m)^*}\mathbf{y}^*,$$

where the asterisk is used to denote the transformed variables. We perform this optimization after updating $\mathbf{W}$ and $\mathbf{Y}$ according to the equations in Appendix A, but before learning the ARD parameters.

Given the new $\mathbf{R}$, the transformed factorial distributions are given as

$$\begin{aligned}
q^*(\mathbf{W}^{(m)}) &= \prod_{d=1}^{D_m} \mathrm{N}(\mathbf{W}^{(m)}_{d,:}|\mathbf{R}^T\boldsymbol{\mu}_{\mathbf{W}^{(m)}_{d,:}}, \mathbf{R}^T\boldsymbol{\Sigma}_{\mathbf{W}^{(m)}}\mathbf{R}),\\
q^*(\mathbf{Y}) &= \prod_{n=1}^{N} \mathrm{N}(\mathbf{y}_n|\mathbf{R}^{-1}\boldsymbol{\mu}_{\mathbf{y}_n}, \mathbf{R}^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}}\mathbf{R}^{-T}),\\
q^*(\boldsymbol{\alpha}^{(m)}) &= \prod_{k=1}^{K} \mathrm{Gamma}(\alpha_0 + D_m/2, \beta_0 + \mathbf{r}_k^T\langle\mathbf{W}^{(m)^T}\mathbf{W}^{(m)}\rangle\mathbf{r}_k/2),
\end{aligned}$$

where $\mathbf{r}_k$ is the kth column of $\mathbf{R}$. The cost function for optimizing $\mathbf{R}$ is

$$\begin{aligned}
\mathcal{L}_{\mathbf{R}} \quad = \quad & \langle\log p(\mathbf{Y})\rangle^* - \langle\log q^*(\mathbf{Y})\rangle^* + \\
& \sum_{m=1}^{2} \langle\log p(\mathbf{W}^{(m)}|\boldsymbol{\alpha}^{(m)})\rangle^* - \langle\log q^*(\mathbf{W}^{(m)})\rangle^* + \\
& \langle\log p(\boldsymbol{\alpha}^{(m)})\rangle^* - \langle\log q^*(\boldsymbol{\alpha}^{(m)})\rangle^*,
\end{aligned}$$

where $\langle\cdot\rangle^*$ denotes the expectation with respect to the transformed distribution. The first four individual terms can be written, omitting constants independent of $\mathbf{R}$, as

$$\begin{aligned}
\langle\log p(\mathbf{Y})\rangle^* &= -\mathrm{Trace}[\mathbf{R}^{-1}\langle\mathbf{Y}\mathbf{Y}^T\rangle\mathbf{R}^{-T}]/2,\\
-\langle\log q(\mathbf{Y})\rangle^* &= -N\log|\mathbf{R}|,\\
\langle p(\mathbf{W}^{(m)}|\boldsymbol{\alpha}^{(m)})\rangle^* &\approx -D_m/2\log\prod_{k=1}^{K}\mathbf{r}_k^T\langle\mathbf{W}^{(m)^T}\mathbf{W}^{(m)}\rangle\mathbf{r}_k,\\
-\langle\log q(\mathbf{W}^{(m)})\rangle^* &= D_m\log|\mathbf{R}|.
\end{aligned}$$

The last two terms can be accurately approximated as constants, since the prior is effectively non-informative ($\alpha_0 = \beta_0 \approx 0$). The same argument has been used to approximate the third term; a part that is effectively constant has been left out. Finally, the cost function to be maximized as a function of $\mathbf{R}$ is

$$\begin{aligned}
\mathcal{L}_{\mathbf{R}} \quad = \quad & -\mathrm{Trace}[\mathbf{R}^{-1}\langle\mathbf{Y}\mathbf{Y}^T\rangle\mathbf{R}^{-T}]/2 + (\sum_{m=1}^{2} D_m - N)\log|\mathbf{R}| \\
& - \sum_{m=1}^{2} D_m/2\log\prod_{k=1}^{K}\mathbf{r}_k\langle\mathbf{W}^{(m)^T}\mathbf{W}^{(m)}\rangle\mathbf{r}_k. \qquad (9)
\end{aligned}$$

For finding the optimum we calculate the gradient of the cost and use standard L-BFGS algorithm for the optimization, with the initial solution of $\mathbf{R} = \mathbf{I}$.

After the optimization converges the expectations can be transformed as

$$\begin{aligned}
\langle \mathbf{Y} \rangle &\leftarrow \mathbf{R}^{-1} \langle \mathbf{Y} \rangle, \\
\boldsymbol{\Sigma}_{\mathbf{Y}} &\leftarrow \mathbf{R}^{-1} \boldsymbol{\Sigma}_{\mathbf{Y}} \mathbf{R}^{-T}, \\
\langle \mathbf{Y}\mathbf{Y}^T \rangle &\leftarrow \langle \mathbf{Y} \rangle \langle \mathbf{Y} \rangle^T + N \boldsymbol{\Sigma}_{\mathbf{Y}}, \\
\langle \mathbf{W}^{(m)} \rangle &\leftarrow \langle \mathbf{W}^{(m)} \rangle \mathbf{R}, \\
\boldsymbol{\Sigma}_{\mathbf{W}} &\leftarrow \mathbf{R}^T \boldsymbol{\Sigma}_{\mathbf{W}} \mathbf{R}, \\
\langle \mathbf{W}^{(m)^T} \mathbf{W}^{(m)} \rangle &\leftarrow \langle \mathbf{W}^{(m)} \rangle^T \langle \mathbf{W}^{(m)} \rangle + D_m \boldsymbol{\Sigma}_{\mathbf{W}}.
\end{aligned}$$

Appendix C. Parameter-expanded Variational Bayes for BCCA

Appendix B explains how to optimize the BIBFA lower bound with respect to the linear transformation. It is also possible to do the same for the BCCA model with full covariance matrices (8). In particular, for the choice of $\boldsymbol{\alpha}^{(1)} = \boldsymbol{\alpha}^{(2)}$ we can solve for optimal $\mathbf{R}$ analytically. In the experiments conducted in this paper we always used this optimization step when computing BCCA.

The derivation follows closely the derivation provided for parameter-expanded factor analysis by Luttinen and Ilin (2010), and hence we only summarize here the main steps. We start with the expression in (9), and note that the last term,

$$\langle p(\mathbf{W}|\boldsymbol{\alpha}) \rangle^* \approx -D/2 \log \prod_{k=1}^{K} \mathbf{r}_k^T \langle \mathbf{W}^T \mathbf{W} \rangle \mathbf{r}_k,$$

is maximized if $\mathbf{R}^T \langle \mathbf{W}^T \mathbf{W} \rangle \mathbf{R}$ is a diagonal matrix. Hence, we can add that as a constraint in the cost function. For a diagonal matrix we can easily compute the trace, and the term simplifies to

$$\langle p(\mathbf{W}|\boldsymbol{\alpha}) \rangle^* \approx -D/2 \log \prod_{k=1}^{K} \mathbf{r}_k^T \langle \mathbf{W}^T \mathbf{W} \rangle \mathbf{r}_k = -D \log |\mathbf{R}|,$$

canceling $\langle \log q(\mathbf{W}) \rangle^* = D \log \mathbf{R}|$ out. By reparameterizing $\mathbf{R}$ through its singular value decomposition we can find the optimum for the remaining terms in (9) by computing the left singular vectors by eigendecomposition of $\langle \mathbf{Z}\mathbf{Z}^T \rangle / N$. The right singular vectors are then chosen to make $\langle \mathbf{W}^T \mathbf{W} \rangle$ diagonal.

Appendix D. BIBFA with Spike-and-slab Prior

For inference with the spike-and-slab prior (6) we use Gibbs sampling, following closely the updates given for element-wise sparse FA model by Knowles and Ghahramani (2011). Here we summarize the necessary changes for adopting their model for group-wise sparsity in BIBFA.

D.1 Sampling H and W

We sample each entry $\mathbf{H}_{m,k}$ independently, based on the relative likelihoods of the two possible values. For $\mathbf{H}_{m,k} = 1$ we integrate out $\mathbf{W}_{:,k}^{(m)}$, which can be done independently for each element.

This results in the relative probability

$$\frac{p(\mathbf{H}_{m,k}=1|\mathbf{X}^{(m)})}{p(\mathbf{H}_{m,k}=0|\mathbf{X}^{(m)})} = \frac{\pi_m \prod_{d=1}^{D_m} \int p(\mathbf{X}^{(m)}|\mathbf{W}^{(m)}_{d,k}) p(\mathbf{W}^{(m)}_{d,k}|0,(\boldsymbol{\alpha}^{(m)}_k)^{-1}) d\mathbf{W}^{(m)}_{d,k}}{(1-\pi_m)\prod_{d=1}^{D_m} p(\mathbf{X}^{(m)}|\mathbf{W}^{(m)}_{d,k}=0)}$$
$$= \frac{\pi_m}{(1-\pi_m)} \left(\frac{(\boldsymbol{\alpha}^{(m)}_k)^{-1}}{\lambda} \right)^{D_m/2} \exp(\frac{1}{2}\lambda \boldsymbol{\mu}^T \boldsymbol{\mu}),$$

where conditioning on the rest of the variables has been dropped for clarity. Here $\lambda = \tau_m \mathbf{Y}^T_{k,:}\mathbf{Y}_{k,:} + \boldsymbol{\alpha}^{(m)}_k$ and $\boldsymbol{\mu} = \frac{\tau_m}{\lambda}(\mathbf{X}^{(m)} - \sum_{j\neq k} \mathbf{W}^{(m)}_{:,j}\mathbf{Y}^T_{j,:})\mathbf{Y}_{k,:}$. Compared to Knowles and Ghahramani (2011), we need to multiply D_m separate terms to reach the final ratio. On the other hand, we need not consider new components since we have replaced the Indian Buffet Process (IBP) prior with a simple Bernoulli prior; IBP would not be useful since we only have two realizations for each component.

While the above step integrates over $\mathbf{W}^{(m)}$, we will still need the projections for sampling other parameters of the model. Hence, we instantiate them by drawing $\mathbf{W}^{(m)}_{:,k} \sim \mathrm{N}(\boldsymbol{\mu}, \lambda^{-1}\mathbf{I})$ if $\mathbf{H}_{m,k} = 1$. Otherwise, we set the whole vector to $\mathbf{0}$ as dictated by the prior.

D.2 Sampling the Rest of the Parameters

The sampling for the rest of the parameters does not depend on the prior used for $\mathbf{W}$, since they depend directly on the current values of $\mathbf{W}$. The conditional distributions are very close to the updates used for the variational approximation, only now they are conditional on the current values instead of the expectations. Below we show the sampling equations for $\mathbf{y}_n$ as an example; the updates for $\boldsymbol{\alpha}^{(m)}$ and $\tau^{(m)}$ can be easily modified from the variational updates given in Appendix A.

$$\mathbf{y}_n \sim \mathrm{N}(\boldsymbol{\mu}_{\mathbf{y}_n}, \boldsymbol{\Sigma}_{\mathbf{Y}}),$$
$$\boldsymbol{\Sigma}_{\mathbf{Y}} = (\mathbf{I} + \sum_{m=1}^{2} \tau^{(m)} \mathbf{W}^{(m)^T} \mathbf{W}^{(m)})^{-1},$$
$$\boldsymbol{\mu}_{\mathbf{y}_n} = \sum_{m=1}^{2} \tau^{(m)} \boldsymbol{\Sigma}_{\mathbf{Y}} \mathbf{W}^{(m)^T} \mathbf{x}^{(m)}_n.$$

Finally, we need to update the variables π_m, drawing them from their Beta-posterior

$$\pi_m \sim \mathrm{Beta}(1 + \sum_k \mathbf{H}_{m,k}, 1 + K - \sum_k \mathbf{H}_{m,k}).$$

References

Deepak Agarwal, Bee-Chung Chen, and Bo Long. Localized factor models for multi-context recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 609–617. ACM, New York, NY, USA, 2011.

Cédric Archambeau and Francis Bach. Sparse probabilistic projections. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 73–80. MIT Press, 2009.

Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. Robust probabilistic projections. In W.W. Cohen and A. Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning*, pages 33–40. ACM, 2006.

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. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

Christopher M. Bishop. Bayesian PCA. In M. S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 382–388. MIT Press, 1999.

Leo Breiman and Jerome H. Friedman. Predicting multivariate responses in multiple linear regression. *Journal of Royal Statistical Society B*, 59(3), 1997.

Michael W. Browne. The maximum-likelihood solution in inter-battery factor analysis. *British Journal of Mathematical and Statistical Psychology*, 32:75–86, 1979.

Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. Manifold relevance determination. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Tijl De Bie and Bart De Moor. On the regularization of canonical correlation analysis. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Source Separation (ICA2003)*, pages 785–790, 2003.

Filip Deleus and Marc M. Van Hulle. Functional connectivity analysis of fMRI data based on regularized multiset canonical correlation analysis. *Journal of Neuroscience methods*, 197(1): 143–157, 2011.

Carl H. Ek, Jon Rihan, Philip H.S. Torr, Grégory Rogez, and Neil D. Lawrence. Ambiquity modelling in latent spaces. In *Proceedings of the International Workshop on Machine Learning for Multimodal Interaction (MLMI'08)*, pages 62–73, 2008.

Yusuke Fujiwara, Yoichi Miyawaki, and Yukiyasu Kamitani. Estimating image bases for visual image reconstruction from human brain activity. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 576–584, 2009.

Zoubin Ghahramani and Matthew J. Beal. Variational inference for Bayesian mixtures of factor analyzers. In S.A. Solla, T.K. Leen, and K-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 449–455. MIT Press, 2000.

Zoubin Ghahramani, Thomas L. Griffiths, and Peter Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics*, 8, 2007.

Ignacio Gonzales, Sebastien Dejean, Pascal G.P. Martin, and Alain Baccini. CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software*, 23(12):1–14, 2008.

Yue Guan and Jennifer G. Dy. Sparse probabilistic principal component analysis. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Volume 5 of JMLR:W&CP*, pages 185–192, 2009.

Aria Haghighi, Percy Liang, Taylor Berh-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio, June 2008. Association for Computational Linguistics.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

William W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13:1095–1105, 2000.

Ilkka Huopaniemi, Tommi Suvitaival, Janne Nikkilä, Matej Orešič, and Samuel Kaski. Two-way analysis of high-dimensional collinear data. *Data Mining and Knowledge Discovery*, 19:261–276, 2009.

Ilkka Huopaniemi, Tommi Suvitaival, Janne Nikkilä, Matej Orešič, and Samuel Kaski. Multivariate multi-way analysis of multi-source data. *Bioinformatics*, 26:i391–i398, 2010. (ISMB 2010).

Alexander Ilin and Tapani Raiko. Practical approaches to principal component analysis in the presence of missing data. *Journal of Machine Learning Research*, 11:1957–2000, 2010.

Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. Extracting shared subspace for multi-label classification. In *Proceedings of thre 14th ACM SIGKDD International Conferece on Knowledge Discovery and Data Mining*, pages 381–389, 2008.

Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 982–990. MIT Press, 2010.

Alfredo A. Kalaitzis and Neil D. Lawrence. Residual Component Analysis: Generalising PCA for more flexible inference in linear-Gaussian models. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, pages 209–216. Omnipress, 2012.

Arto Klami. Variational Bayesian matching. In S. C. H. Hoi and W. Buntine, editors, *Proceedings of the 4th Asian Conference on Machine Learning (ACML), Volume 25 of JMLR:C&WP*, pages 205-220, 2012.

Arto Klami and Samuel Kaski. Generative models that discover dependencies between data sets. In *Proceedings of MLSP'06, IEEE International Workshop on Machine Learning for Signal Processing*, pages 123–128. IEEE, 2006.

Arto Klami and Samuel Kaski. Local dependent components. In Zoubin Ghahramani, editor, *Proceedings of ICML 2007, the 24th International Conference on Machine Learning*, pages 425–432. Omnipress, 2007.

Arto Klami and Samuel Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian exponential family projections for coupled data sources. In P. Grunwald and P. Spirtes, editors, *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (2010)*, pages 286–293. AUAI Press, 2010.

David Knowles and Zoubin Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *Annals of Applied Statistics*, 5:B 1534-1552, 2011.

Miika Koskinen, Jaakko Viinikanoja, Mikko Kurimo, Arto Klami, Samuel Kaski, and Riitta Hari. Identifying fragments of natural speech from the listener's MEG signals. *Human Brain Mapping*, 2012.

Leo Lahti, Samuel Myllykangas, Sakari Knuutila, and Samuel Kaski. Dependency detection with similarity constraints. In *Proceedings of MLSP 2009, IEEE International Workshop on Machine Learning for Signal Processing*, pages 89–94. IEEE, 2009.

Leo Lahti, Martin Schäfer, Hans-Ulrich Klein, Silvio Bicciato, and Martin Dugas. Cancer gene prioritization by integrative analysis of mRNA expression and DNA copy number data: a comparative review. *Briefings in Bioinformatics*, March 2012.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2000.

Gayle Leen and Colin Fyfe. A Gaussian process latent variable model formulation of canonical correlation analysis. In *Proceedings of 14th European Symposium on Artificial Neural Networks*, pages 418–418, 2006.

Jaakko Luttinen and Alexander Ilin. Transformations in variational Bayesian factor analysis to speed up learning. *Neurocomputing*, 73:1093–1102, 2010.

Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *Artificial Neural Networks – ICANN 2001*, pages 353–360. Springer, 2001.

Shakir Mohamed, Katherine A. Heller, and Zoubin Ghahramani. Bayesian and L1 approaches for sparse unsupervised learning. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, pages 751–758. Omnipress, 2012.

Takuho Nakano, Akisato Kimura, Hirokazu Kameoka, Shigeki Miyabe, Shigeki Sagayama, Nobutaka Ono, Kunio Kashino, and Takuya Nishimoto. Automatic video annotation via hierarchical topic trajectory model considering cross-model correlations. In *Proceedings of the IEEE Internatioanl Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2380–2383, 2011.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

James Petterson and Tiberio Caetano. Reverse multi-label learning. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1912–1920. MIT Press, 2010.

Yuan Qi and Tommi S. Jaakkola. Parameter expanded variational Bayesian methods. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1097–1104. MIT Press, 2007.

Piyush Rai and Hal Daumé III. Multi-label prediction via sparse infinite CCA. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1518–1526. MIT Press, 2009.

Mélanie Rey and Volker Roth. Copula mixture model for dependency-seeking clustering. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Simon Rogers, Arto Klami, Janne Sinkkonen, Mark Girolami, and Samuel Kaski. Infinite factorization of multiple non-parametric views. *Machine Learning*, 79(1-2):201–226, 2010.

Indrayana Rustandi, Marcel A. Just, and Tom M. Mitchell. Integrating multiple-study multiple-subject fMRI datasets using canonical correlation analysis. In *Proceedings of the MICCAI 2009 Workshop: Statistical modeling and detection issues in intra- and inter-subject functional MRI data analysis*, 2009.

Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh Rao. Learning shared latent structure for image synthesis and robotic imitation. In Y. Weriss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1233–1240. MIT Press, 2010.

Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 650–658. ACM, New York, NY, USA, 2008.

Liang Sun, Shuiwang Ji, and Jieping Ye. Canonical correlation analysis for multilabel classification: A least-squares formulation, extension, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):194–200, 2011.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Abhishek Tripathi, Arto Klami, Matej Orešič, and Samuel Kaski. Matching samples of multiple views. *Data Mining and Knowledge Discovery*, 23:300–321, 2011.

Grigorios Tsoumakas and Ioannis Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 406–417, 2007.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*. Springer, 2nd edition, 2010.

Ledyard R. Tucker. An inter-battery method of factor analysis. *Psychometrika*, 23:111–136, 1958.

Angelika van der Linde. Reduced rank regression models with latent variables in Bayesian functional data analysis. *Bayesian Analysis*, 6(1):77–126, 2011.

Jaakko Viinikanoja, Arto Klami, and Samuel Kaski. Variational Bayesian mixture of robust CCA models. In A. Gionis J. Luis Balcázar, F. Bonchi and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases. Proceedings of European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010*, volume III, pages 370–385. Springer, 2010.

Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1473–1480. MIT Press, 2003.

Seppo Virtanen, Arto Klami, and Samuel Kaski. Bayesian CCA via group sparsity. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 457–464. ACM, 2011.

Seppo Virtanen, Arto Klami, Suleiman A. Khan, and Samuel Kaski. Bayesian group factor analysis. In N. Lawrence and M. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, volume 22 of JMLR:W&CP*, pages 1269–1277, 2012.

Seppo Virtanen, Jangqing Jia, Arto Klami, and Trevor Darrell. Factorized multi-modal topic model. In N. de Freitas and K. Murphy, editors, *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 843–851. AUAI Press, 2012.

Chong Wang. Variational Bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, 18:905–910, 2007.

David Wipf and Srikantan Nagarajan. A new view on automatic relevance determination. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1625–1632. MIT Press, 2008.

Jarkko Ylipaavalniemi, Eerika Savia, Sanna Malinen, Riitta Hari, Ricardo Vigário, and Samuel Kaski. Dependencies between stimuli and spatially independent fMRI sources: Towards brain correlates of natural stimuli. *NeuroImage*, 48:176–185, 2009.

Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

Journal of Machine Learning Research 14 (2013) 1005-1031 Submitted 9/12; Revised 1/13; Published 4/13

Variational Inference in Nonconjugate Models

Chong Wang CHONGW@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

David M. Blei BLEI@CS.PRINCETON.EDU
Department of Computer Science
Princeton University
Princeton, NJ, 08540, USA

Editor: Neil Lawrence

Abstract

Mean-field variational methods are widely used for approximate posterior inference in many probabilistic models. In a typical application, mean-field methods approximately compute the posterior with a coordinate-ascent optimization algorithm. When the model is conditionally conjugate, the coordinate updates are easily derived and in closed form. However, many models of interest—like the correlated topic model and Bayesian logistic regression—are nonconjugate. In these models, mean-field methods cannot be directly applied and practitioners have had to develop variational algorithms on a case-by-case basis. In this paper, we develop two generic methods for nonconjugate models, Laplace variational inference and delta method variational inference. Our methods have several advantages: they allow for easily derived variational algorithms with a wide class of nonconjugate models; they extend and unify some of the existing algorithms that have been derived for specific models; and they work well on real-world data sets. We studied our methods on the correlated topic model, Bayesian logistic regression, and hierarchical Bayesian logistic regression.

Keywords: variational inference, nonconjugate models, Laplace approximations, the multivariate delta method

1 Introduction

Mean-field variational inference lets us efficiently approximate posterior distributions in complex probabilistic models (Jordan et al., 1999; Wainwright and Jordan, 2008). Applications of variational inference are widespread. As examples, it has been applied to Bayesian mixtures (Attias, 2000; Corduneanu and Bishop, 2001), factorial models (Ghahramani and Jordan, 1997), and probabilistic topic models (Blei et al., 2003).

The basic idea behind mean-field inference is the following. First define a family of distributions over the hidden variables where each variable is assumed independent and governed by its own parameter. Then fit those parameters so that the resulting distribution is close to the conditional distribution of the hidden variables given the observations. Closeness is measured with the Kullback-Leibler divergence. Inference becomes optimization.

In many settings this approach can be used as a "black box" technique. In particular, this is possible when we can easily compute the conditional distribution of each hidden variable given all

of the other variables, both hidden and observed. (This class contains the models mentioned above.) For such models, which are called *conditionally conjugate* models, it is easy to derive a coordinate ascent algorithm that optimizes the parameters of the variational distribution (Beal, 2003; Bishop, 2006). This is the principle behind software tools like VIBES (Bishop et al., 2003) and Infer.NET (Minka et al., 2010), which allow practitioners to define models of their data and immediately approximate the corresponding posterior with variational inference.

Many models of interest, however, do not enjoy the properties required to take advantage of this easily derived algorithm. Such *nonconjugate* models[1] include Bayesian logistic regression (Jaakkola and Jordan, 1997), Bayesian generalized linear models (Wells, 2001), discrete choice models (Braun and McAuliffe, 2010), Bayesian item response models (Clinton et al., 2004; Fox, 2010), and nonconjugate topic models (Blei and Lafferty, 2006, 2007). Using variational inference in these settings requires algorithms tailored to the specific model at hand. Researchers have developed a variety of strategies for a variety of models, including approximations (Braun and McAuliffe, 2010; Ahmed and Xing, 2007), alternative bounds (Jaakkola and Jordan, 1997; Blei and Lafferty, 2006, 2007; Khan et al., 2010), and numerical quadrature (Honkela and Valpola, 2004).

In this paper we develop two approaches to mean-field variational inference for a large class of nonconjugate models. First we develop *Laplace variational inference*. This approach embeds Laplace approximations—an approximation technique for continuous distributions (Tierney et al., 1989; MacKay, 1992)—within a variational optimization algorithm. We then develop *delta method variational inference*. This approach optimizes a Taylor approximation of the variational objective. The details of the algorithm depend on how the approximation is formed. Formed one way, it gives an alternative interpretation of Laplace variational inference. Formed another way, it is equivalent to using a multivariate delta approximation (Bickel and Doksum, 2007) of the variational objective.

Our methods are generic. Given a model, they can be derived nearly as easily as traditional coordinate-ascent inference. Unlike traditional inference, however, they place fewer conditions on the model, conditions that are less restrictive than conditional conjugacy. Our methods significantly expand the class of models for which mean-field variational inference can be easily applied.

We studied our algorithms with three nonconjugate models: Bayesian logistic regression (Jaakkola and Jordan, 1997), hierarchical logistic regression (Gelman and Hill, 2007), and the correlated topic model (Blei and Lafferty, 2007). We found that our methods give better results than those obtained through special-purpose techniques. Further, we found that Laplace variational inference usually outperforms delta method variational inference, both in terms of computation time and the fidelity of the approximate posterior.

Related work. We have described the various approaches that researchers have developed for specific models. There have been other efforts to examine generic variational inference in nonconjugate models. Paisley et al. (2012a) proposed a variational inference approach using stochastic search for nonconjugate models, approximating the intractable integrals with Monte Carlo methods. Gershman et al. (2012) proposed a nonparametric variational inference algorithm, which can be applied to nonconjugate models. Knowles and Minka (2011) presented a message passing algorithm for nonconjugate models, which has been implemented in Infer.NET (Minka et al., 2010); their technique applies to a subset of models described in this paper.[2]

1. Carlin and Polson (1991) coined the term "nonconjugate model" to describe a model that does not enjoy full conditional conjugacy.
2. It may be generalizable to the full set. However, one must determine how to compute the required expectations.

Laplace approximations have been used in approximate inference in more complex models, though not in the context of mean-field variational inference. Smola et al. (2003) used them to approximate the difficult-to-compute moments in expectation propagation (Minka, 2001). Rue et al. (2009) used them for inference in latent Gaussian models. Here we want to use them for variational inference, in a method that can be applied to a wider range of nonconjugate models.

Finally, we note that the delta method was first used in variational inference by Braun and McAuliffe (2010) in the context of the discrete choice model. Our method generalizes their approach.

Organization of this paper. In Section 2 we review mean-field variational inference and define the class of nonconjugate models to which our algorithms apply. In Section 3, we derive Laplace and delta-method variational inference and present our full algorithm for nonconjugate inference. In Section 4, we show how to use our generic method on several example models and in Section 5 we study its performance on these models. In Section 6, we summarize and discuss this work.

2 Variational Inference and a Class of Nonconjugate Models

We consider a generic model with observations x and hidden variables θ and z,

$$p(\theta, z, x) = p(x|z)p(z|\theta)p(\theta). \tag{1}$$

The distinction between the two hidden variables will be made clear below.

The inference problem is to compute the posterior,

$$p(\theta, z|x) = \frac{p(\theta, z, x)}{\int p(\theta, z, x)dzd\theta}.$$

This is intractable for many models because the denominator is difficult to compute; we must approximate the distribution. In variational inference, we approximate the posterior by positing a simple family of distributions over the latent variables $q(\theta, z)$ and then finding the member of that family which minimizes the Kullback-Leibler (KL) divergence to the true posterior (Jordan et al., 1999; Wainwright and Jordan, 2008).[3]

In this section we review variational inference and discuss mean-field variational inference for the class of conditionally conjugate models. We then define a wider class of nonconjugate models for which mean-field variational inference is not as easily applied. In the next section, we derive algorithms for performing mean-field variational inference in this larger class of models.

2.1 Mean-field Variational Inference

Mean-field variational inference is simplest and most widely used variational inference method. In mean-field variational inference we posit a fully factorized variational family,

$$q(\theta, z) = q(\theta)q(z). \tag{2}$$

3. In this paper, we focus on mean-field variational inference where we minimize the KL divergence to the posterior. We note that there are other kinds of variational inference, with more structured variational distributions or with alternative objective functions (Wainwright and Jordan, 2008; Barber, 2012). In this paper, we use "variational inference" to indicate mean-field variational inference that minimizes the KL divergence.

In this family of distributions the variables are independent and each is governed by its own distribution. This family usually does not contain the posterior, where θ and z are dependent. However, it is very flexible—it can capture any set of marginals of the hidden variables.

Under the standard variational theory, minimizing the KL divergence between $q(\theta, z)$ and the posterior $p(\theta, z|x)$ is equivalent to maximizing a lower bound of the log marginal likelihood of the observed data x. We obtain this bound with Jensen's inequality,

$$\begin{aligned}\log p(x) &= \log \int p(\theta, z, x) \mathrm{d}z \mathrm{d}\theta \\ &\geq \mathbb{E}_q [\log p(\theta, z, x)] - \mathbb{E}_q [\log q(\theta, z)] \\ &\triangleq \mathcal{L}(q), \end{aligned} \tag{3}$$

where $\mathbb{E}_q[\cdot]$ is the expectation taken with respect to q and note the second term is the entropy of q. We call $\mathcal{L}(q)$ the variational objective.

Setting $\partial \mathcal{L}(q)/\partial q = 0$ shows that the optimal solution satisfies the following,

$$q^*(\theta) \propto \exp\left\{\mathbb{E}_{q(z)} [\log p(z|\theta) p(\theta)]\right\}, \tag{4}$$

$$q^*(z) \propto \exp\left\{\mathbb{E}_{q(\theta)} [\log p(x|z) p(z|\theta)]\right\}. \tag{5}$$

Here we have combined the optimal conditions from Bishop (2006) with the particular factorization of Equation 1. Note that the variational objective usually contains many local optima.

These conditions lead to the traditional coordinate ascent algorithm for variational inference. It iterates between holding $q(z)$ fixed to update $q(\theta)$ from Equation 4 and holding $q(\theta)$ fixed to update $q(z)$ from Equation 5. This converges to a local optimum of the variational objective (Bishop, 2006).

When all the nodes in a model are *conditionally conjugate*, the coordinate updates of Equation 4 and Equation 5 are available in closed form. A node is conditionally conjugate when its conditional distribution given its Markov blanket (i.e., the set of random variables that it is dependent on in the posterior) is in the same family as its conditional distribution given its parents (i.e., its factor in the joint distribution). For example, in Equation 1 suppose the factor $p(\theta)$ is a Dirichlet and both factors $p(z|\theta)$ and $p(x|z)$ are multinomials. This means that the conditional $p(\theta|z)$ is also a Dirichlet and the conditional $p(z|x, \theta)$ is also a multinomial. This model, which is latent Dirichlet allocation (Blei et al., 2003), is conditionally conjugate. Many applications of variational inference have been developed for this type of model (Bishop, 1999; Attias, 2000; Beal, 2003).

However, if there exists any node in the model that is not conditionally conjugate then this coordinate ascent algorithm is not available. That setting arises in many practical models and does not permit closed-form updates or easy calculation of the variational objective. We will develop generic variational inference algorithms for a wide class of nonconjugate models. First, we define that class.

2.2 A Class of Nonconjugate Models

We present a wide class of nonconjugate models, still assuming the factorization of Equation 1.

1. We assume that θ is real-valued and the distribution $p(\theta)$ is twice differentiable with respect to θ. If we require $\theta > \theta_0$ (θ_0 is a constant), we may define a distribution over $\log(\theta - \theta_0)$. These assumptions cover exponential families, such as the Gaussian, Poisson and gamma, as well as more complex distributions, such as a student-t.

2. We assume the distribution $p(z \mid \theta)$ is in the exponential family (Brown, 1986),

$$p(z \mid \theta) = h(z) \exp\left\{\eta(\theta)^\top t(z) - a(\eta(\theta))\right\}, \tag{6}$$

where $h(z)$ is a function of z; $t(z)$ is the sufficient statistic; $\eta(\theta)$ is the natural parameter, which is a function of the conditioning variables; and $a(\eta(\theta))$ is the log partition function. We also assume that $\eta(\theta)$ is twice differentiable; since θ is real-valued, this is satisfied in most statistical models. Unlike in conjugate models, these assumptions do not restrict $p(\theta)$ and $p(z \mid \theta)$ to be a conjugate pair; the conditional distribution $p(\theta \mid z)$ is not necessarily in the same family as the prior $p(\theta)$.

3. The distribution $p(x \mid z)$ is in the exponential family,

$$p(x \mid z) = h(x) \exp\left\{t(z)^\top \langle t(x), 1\rangle\right\}. \tag{7}$$

We set up this exponential family so that the natural parameter for x is all but the last component of $t(z)$ and the last component is the negative log normalizer $-a(\cdot)$. Thus, the distribution of z is conjugate to the conditional distribution of x; the conditional $p(z \mid \theta, x)$ is in the same family as $p(z \mid \theta)$ (Bernardo and Smith, 1994).

Our terminology follows these assumptions: θ is the *nonconjugate variable*, z is the *conjugate variable*, and x is the *observation*.

This class of models is larger than the class of conditionally conjugate models. Our expanded class also includes nonconjugate models like the correlated topic model (Blei and Lafferty, 2007), dynamic topic model (Blei and Lafferty, 2006), Bayesian logistic regression (Jaakkola and Jordan, 1997; Gelman and Hill, 2007), discrete choice models (Braun and McAuliffe, 2010), Bayesian ideal point models (Clinton et al., 2004), and many others. Further, the methods we develop below are easily adapted more complicated graphical models, those that contain conjugate and nonconjugate variables whose dependencies are encoded in a directed acyclic graph. Appendix A outlines how to adapt our algorithms to this more general case.

Example: Hierarchical language modeling. We introduce the hierarchical language model, a simple example of a nonconjugate model to help ground our derivation of the general algorithms. Consider the problem of unigram language modeling. We are given a collection of documents $\mathcal{D} = x_{1:D}$ where each document x_d is a vector of word counts, observations from a discrete vocabulary of length V. We model each document with its own distribution over words and place a Dirichlet prior on that distribution. This model is used, for example, in the language modeling approach to information retrieval (Croft and Lafferty, 2003).

We want to place a prior on the Dirichlet parameters, a positive V-vector, that govern each document's distribution over terms. In theory, every exponential family distribution has a conjugate prior (Bernardo and Smith, 1994) and the prior to the Dirichlet is the multi-gamma distribution (Kotz et al., 2000). However, the multi-gamma is difficult to work with because its log normalizer is not easy to compute. As an alternative, we place a log normal distribution on the Dirichlet parameters. This is not the conjugate prior.

The full generative process is as follows:

1. Draw log Dirichlet parameters $\theta \sim \mathcal{N}(0, I)$.
2. For each document d, $1 \le d \le D$:

(a) Draw multinomial parameter $z_d \mid \theta \sim \text{Dirichlet}(\exp\{\theta\})$.
(b) Draw word counts $x_d \sim \text{Multinomial}(N, z_d)$.

Given a collection of documents, our goal is to compute the posterior distribution $p(\theta, z_{1:D} \mid x_{1:D})$. Traditional variational or Gibbs sampling methods cannot be easily used because the normal prior on the parameters θ is not conjugate to the Dirichlet$(\exp\{\theta\})$ likelihood.

This language model fits into our model class. In the notation of the joint distribution of Equation 1, $\theta = \theta$, $z = z_{1:D}$, and $x = x_{1:D}$. The per-document multinomial parameters z and word counts x are conditionally independent given the Dirichlet parameters θ,

$$p(z \mid \theta) = \textstyle\prod_d p(z_d \mid \theta),$$
$$p(x \mid z) = \textstyle\prod_d \prod_n p(x_{dn} \mid z_d).$$

In this case, the natural parameter $\eta(\theta) = \exp\{\theta\}$. This model satisfies the assumptions: the log normal $p(\exp\{\theta\})$ is not conjugate to the Dirichlet $p(z_d \mid \exp\{\theta\})$ but is twice differentiable; the Dirichlet is conjugate to the multinomial $p(x_d \mid z_d)$ and the multinomial is in the exponential family.

Below we will use various components of the exponential family form of the Dirichlet:

$$h(z_d) = \textstyle\prod_i z_{di}^{-1}; \;\; t(z_d) = \log z_d; \;\; a_d(\eta(\theta)) = \sum_i \log\Gamma(\exp\{\theta_i\} - \log\Gamma(\sum_i \exp\{\theta_i\}). \tag{8}$$

We will return to this model as a simple running example.

3 Laplace and Delta Method Variational Inference

We have defined a class of nonconjugate models. Variational inference is difficult to derive for these models because $p(\theta)$ is not conjugate to $p(z \mid \theta)$. Specifically, the update in Equation 4 does not necessarily have the form of an exponential family we can work with and it is difficult to use $\mathbb{E}_{q(\theta)}[\log p(z \mid \theta)]$ in the update of Equation 5.

We will develop two variational inference algorithms for this class: *Laplace variational inference* and *delta method variational inference*. Both use coordinate ascent to optimize the variational parameters, iterating between updating $q(\theta)$ and $q(z)$. They differ in how they update the variational distribution of the nonconjugate variable $q(\theta)$. In Laplace variational inference, we use Laplace approximations (MacKay, 1992; Tierney et al., 1989) within the coordinate ascent updates of Equation 4 and Equation 5. In delta method variational inference, we apply Taylor approximations to approximate the variational objective in Equation 3 and then derive the corresponding updates. Different ways of taking the Taylor approximation lead to different algorithms. Formed one way, this recovers the Laplace approximation. Formed another way, it is equivalent to using a multivariate delta approximation (Bickel and Doksum, 2007) of the variational objective function.

In both variants, the variational distribution is the mean-field family in Equation 2. The variational distribution of the nonconjugate variable $q(\theta)$ is a Gaussian; the variational distribution of the conjugate variable $q(z)$ is in the same family as $p(z \mid \eta(\theta))$. In Laplace inference, these forms emerge from the derivation. In delta method inference, they are assumed. The complete variational family is,

$$q(\theta, z) = q(\theta \mid \mu, \Sigma) q(z \mid \phi).$$

where (μ, Σ) are the parameters for a Gaussian distribution and ϕ is a natural parameter for z. For example, in the hierarchical language model of Section 2.2, ϕ is a collection of D Dirichlet parameters. We will sometimes suppress the parameters, writing $q(\theta)$ for $q(\theta \mid \mu, \Sigma)$.

Our algorithms are coordinate ascent algorithms, where we iterate between updating the nonconjugate variational distribution $q(\theta)$ and updating the conjugate variational distribution $q(z)$. In the subsections below, we derive the update for $q(\theta)$ in each algorithm. Then, for both algorithms, we derive the update for $q(z)$. The full procedure is described in Section 3.4 and Figure 1.

3.1 Laplace Variational Inference

We first review the Laplace approximation. Then we show how to use it in variational inference.

3.1.1 The Laplace Approximation

Laplace approximations use a Gaussian to approximate an intractable density. Consider approximating an intractable posterior $p(\theta \mid x)$. (There is no hidden variable z in this set up.) Assume the joint distribution $p(x, \theta) = p(x \mid \theta) p(\theta)$ is easy to compute. Laplace approximations use a Taylor approximation around the maximum a posterior (MAP) point to construct a Gaussian proxy for the posterior. They are used for continuous distributions.

First, notice the posterior is proportional to the exponentiated log joint

$$p(\theta \mid x) = \exp\{\log p(\theta \mid x)\} \propto \exp\{\log p(\theta, x)\}.$$

Let $\hat{\theta}$ be the MAP of $p(\theta \mid x)$, found by maximizing $\log p(\theta, x)$. A Taylor expansion around $\hat{\theta}$ gives

$$\log p(\theta \mid x) \approx \log p(\hat{\theta} \mid x) + \tfrac{1}{2}(\theta - \hat{\theta})^\top H(\hat{\theta})(\theta - \hat{\theta}). \tag{9}$$

The term $H(\hat{\theta})$ is the Hessian of $\log p(\theta \mid x)$ evaluated at $\hat{\theta}$, $H(\hat{\theta}) \triangleq \nabla^2 \log p(\theta \mid x)|_{\theta=\hat{\theta}}$.

In the Taylor expansion of Equation 9, the first-order term $(\theta - \hat{\theta})^\top \nabla \log p(\theta \mid x)|_{\theta=\hat{\theta}}$ equals zero. The reason is that $\hat{\theta}$ is the maximum of $\log p(\theta \mid x)$ and so its gradient $\nabla \log p(\theta \mid x)|_{\theta=\hat{\theta}}$ is zero. Exponentiating Equation 9 gives the approximate Gaussian posterior

$$p(\theta \mid x) \approx \tfrac{1}{C} \exp\left\{-\tfrac{1}{2}(\theta - \hat{\theta})^\top \left(-H(\hat{\theta})\right)(\theta - \hat{\theta})\right\},$$

where C is a normalizing constant. In other words, $p(\theta \mid x)$ can be approximated by

$$p(\theta \mid x) \approx \mathcal{N}(\hat{\theta}, -H(\hat{\theta})^{-1}).$$

This is the Laplace approximation. While powerful, it is difficult to use in multivariate settings, for example, when there are discrete hidden variables. Now we describe how we use Laplace approximations as part of a variational inference algorithm for more complex models.

3.1.2 Laplace Updates in Variational Inference

We adapt the idea behind Laplace approximations to update the variational distribution $q(\theta)$. First, we combine the coordinate update in Equation 4 with the exponential family assumption in Equation 6,

$$q(\theta) \propto \exp\left\{\eta(\theta)^\top \mathbb{E}_{q(z)}[t(z)] - a(\eta(\theta)) + \log p(\theta)\right\}. \tag{10}$$

Define the function $f(\theta)$ to contain the terms inside the exponent of the update,

$$f(\theta) \triangleq \eta(\theta)^\top \mathbb{E}_{q(z)}[t(z)] - a(\eta(\theta)) + \log p(\theta). \tag{11}$$

The terms of $f(\theta)$ come from the model and involve $q(z)$ or θ. Recall that $q(z)$ is in the same exponential family as $p(z \mid \theta)$, $t(z)$ are its sufficient statistics, and ϕ is the variational parameter. We can compute $\mathbb{E}_{q(z)}[t(z)]$ from a basic property of the exponential family (Brown, 1986),

$$\mathbb{E}_{q(z)}[t(z)] = \nabla a(\phi).$$

Seen another way, $f(\theta) = \mathbb{E}_{q(z)}[\log p(\theta, z)]$. This function will be important in both Laplace and delta method inference.

The problem with nonconjugate models is that we cannot update $q(\theta)$ exactly using Equation 10 because $q(\theta) \propto \exp\{f(\theta)\}$ cannot be normalized in closed form. We approximate the update by taking a second-order Taylor approximation of $f(\theta)$ around its maximum, following the same logic as from the original Laplace approximation in Equation 9. The Taylor approximation for $f(\theta)$ around $\hat{\theta}$ is

$$f(\theta) \approx f(\hat{\theta}) + \nabla f(\hat{\theta})(\theta - \hat{\theta}) + \tfrac{1}{2}(\theta - \hat{\theta})^\top \nabla^2 f(\hat{\theta})(\theta - \hat{\theta}), \tag{12}$$

where $\nabla^2 f(\hat{\theta})$ is the Hessian matrix evaluated at $\hat{\theta}$. Now let $\hat{\theta}$ be the value that maximizes $f(\theta)$. This implies that $\nabla f(\hat{\theta}) = 0$ and Equation 12 simplifies to

$$q(\theta) \propto \exp\{f(\theta)\} \approx \exp\left\{f(\hat{\theta}) + \tfrac{1}{2}(\theta - \hat{\theta})^\top \nabla^2 f(\hat{\theta})(\theta - \hat{\theta})\right\}.$$

Thus the approximate update for $q(\theta)$ is to set it to

$$q(\theta) \approx \mathcal{N}\left(\hat{\theta}, -\nabla^2 f(\hat{\theta})^{-1}\right). \tag{13}$$

Note we did not assume $q(\theta)$ is Gaussian. Its Gaussian form stems from the Taylor approximation.

The update in Equation 13 can be used in a coordinate ascent algorithm for a nonconjugate model. We iterate between holding $q(z)$ fixed while updating $q(\theta)$ from Equation 13, and holding $q(\theta)$ fixed while updating $q(z)$. (We derive the second update in Section 3.3.) Each time we update $q(\theta)$ we must use numerical optimization to obtain $\hat{\theta}$, the optimal value of $f(\theta)$.

We return to the hierarchical language model of Section 2.2, where θ are the log of the parameters to the Dirichlet distribution. Implementing the algorithm to update $q(\theta)$ involves forming $f(\theta)$ for the model at hand and deriving an algorithm to optimize it.

With the model equations in Equation 8, we have

$$f(\theta) = \exp(\theta)^\top \mathbb{E}_{q(z)}[t(z)] - D\left(\textstyle\sum_i \log\Gamma(\exp(\theta_i) - \log\Gamma\left(\sum_i \exp(\theta_i)\right)\right) - (1/2)\theta^\top\theta. \tag{14}$$

The expected sufficient statistics of the conjugate variable are

$$\mathbb{E}_{q(z)}[t(z)] = \textstyle\sum_d \mathbb{E}_{q(z_d)}[t(z_d)] = \sum_d \Psi(\phi_d) - \Psi\left(\sum_i \phi_{di}\right),$$

where $\Psi(\cdot)$ is the digamma function, the first derivative of $\log\Gamma(\cdot)$. (This function will also arise in the gradient.) It is straightforward to use numerical methods, such as conjugate gradient (Bertsekas, 1999), to optimize Equation 14. We can then use Equation 13 to update the nonconjugate variable.

3.2 Delta Method Variational Inference

In Laplace variational inference, the variational distribution $q(\theta)$ Equation 13 is solely a function of $\hat{\theta}$, the maximum of $f(\theta)$ in Equation 11. A natural question is, would other values of θ be suitable as well? To consider such alternatives, we describe a different technique for variational inference. We approximate the variational objective $\mathcal{L}$ in Equation 3 and then optimize that approximation.

Again we focus on updating $q(\theta)$ in a coordinate ascent algorithm and postpone the discussion of updating $q(z)$. We set the variational distribution $q(\theta)$ to be a Gaussian $\mathcal{N}(\mu,\Sigma)$, where the parameters are free variational parameters fit to optimize the variational objective. (Note that in Laplace inference, this Gaussian family came out of the derivation.) We isolate the terms of the objective in Equation 3 related to $q(\theta)$, and we substitute the exponential family form of $p(z\,|\,\theta)$ in Equation 6,

$$\mathcal{L}(q(\theta)) = \mathbb{E}_{q(\theta)}\left[\eta(\theta)^\top \mathbb{E}_{q(z)}[t(z)] - a(\eta(\theta)) + \log p(\theta)\right] + \tfrac{1}{2}\log|\Sigma|.$$

The second term comes from the entropy of the Gaussian,

$$-\mathbb{E}_{q(\theta)}[\log q(\theta)] = \frac{1}{2}\log|\Sigma| + C,$$

where C is a constant and is excluded from the objective. The first term is $\mathbb{E}_{q(\theta)}[f(\theta)]$, where $f(\cdot)$ is the same as defined for Laplace inference in Equation 11. Thus,

$$\mathcal{L}(q(\theta)) = \mathbb{E}_{q(\theta)}[f(\theta)] + \tfrac{1}{2}\log|\Sigma|.$$

We cannot easily compute the expectation in the first term. So we use a Taylor approximation of $f(\theta)$ around a chosen value $\hat{\theta}$ (Equation 12) and then take the expectation,

$$\mathcal{L}(q(\theta)) \approx f(\hat{\theta}) + \nabla f(\hat{\theta})^\top(\mu-\hat{\theta}) + \tfrac{1}{2}(\mu-\hat{\theta})^\top \nabla^2 f(\hat{\theta})(\mu-\hat{\theta})]] + \tfrac{1}{2}\left(\mathrm{Tr}\left\{\nabla^2 f(\hat{\theta})\Sigma\right\} + \log|\Sigma|\right), \quad (15)$$

where $\mathrm{Tr}(\cdot)$ is the Trace operator. In the coordinate update of $q(\theta)$, this is the function we optimize with respect to its variational parameters $\{\mu,\Sigma\}$.

To fully specify the algorithm we must choose $\hat{\theta}$, the point around which to approximate $f(\theta)$. We will discuss three choices. The first is to set $\hat{\theta}$ to be the maximum of $f(\theta)$. With this choice, maximizing the approximation in Equation 15 gives $\mu = \hat{\theta}$ and $\Sigma = -\nabla^2 f(\hat{\theta})^{-1}$. Notice this is the update derived in Section 3.1. We have given a different derivation of Laplace variational inference.

The second choice is to set $\hat{\theta}$ as the mean of the variational distribution from the previous iteration of coordinate ascent. If the prior $p(\theta)$ is Gaussian, this recovers the updates derived in Ahmed and Xing (2007) for the correlated topic model.[4] In our study, we found this algorithm did not work well. It did not always converge, possibly due to the difficulty of choosing an appropriate initial $\hat{\theta}$.

The third choice is to set $\hat{\theta} = \mu$, that is, the mean of the variational distribution $q(\theta)$. With this choice, the variable around which we center the Taylor approximation becomes part of the optimization problem. The objective is

$$\mathcal{L}(q(\theta)) \approx f(\mu) + \tfrac{1}{2}\mathrm{Tr}\left\{\nabla^2 f(\mu)\Sigma\right\} + \tfrac{1}{2}\log|\Sigma|. \quad (16)$$

4. This is an alternative derivation of their algorithm. They derived these updates from the perspective of generalized mean-field theory (Xing et al., 2003).

This is the multivariate delta method for evaluating $\mathbb{E}_{q(\theta)}[f(\theta)]$ (Bickel and Doksum, 2007). *Delta method variational inference* optimizes this objective in the coordinate update of $q(\theta)$.

In more detail, we first optimize μ with gradient methods and then optimize Σ in closed form $\Sigma = -\nabla^2 f(\mu)^{-1}$. Note this is more expensive than Laplace variational inference because optimizing Equation 16 requires the third derivative $\nabla^3 f(\theta)$. Braun and McAuliffe (2010) were the first to use the delta method in a variational inference algorithm, developing this technique for the discrete choice model. If we assume the prior $p(\theta)$ is Gaussian then we recover their algorithm. With the ideas presented here, we can now use this strategy in many models.

We return briefly to the unigram language model. The delta method update for $q(\theta)$ optimizes Equation 16, using the specific $f(\cdot)$ found in Equation 14. While Laplace inference required the digamma function and $\log\Gamma$ function, delta method inference will further require the trigamma function.

3.3 Updating the Conjugate Variable

We derived variational updates for $q(\theta)$ using two methods. We now turn to the update for the variational distribution of the conjugate variable $q(z)$. We show that both Laplace inference (Section 3.1) and delta method inference (Section 3.2) lead to the same update. Further, we have implicitly assumed that $\mathbb{E}_{q(z)}[t(z)]$ in Equation 11 is easy to compute. We will confirm this as well.

We first derive the update for $q(z)$ when using Laplace inference. We apply the exponential family form in Equation 6 to the exact update of Equation 5,

$$\log q(z) = \log p(x\,|\,z) + \log h(z) + \mathbb{E}_{q(\theta)}[\eta(\theta)]^\top t(z) + C,$$

where C is a constant not depending on z. Now we use $p(x\,|\,z)$ from Equation 7 to obtain

$$q(z) \propto h(z)\exp\left\{\left(\mathbb{E}_{q(\theta)}[\eta(\theta)] + t(x)\right)^\top t(z)\right\}, \tag{17}$$

which is in the same family as $p(z\,|\,\theta)$ in Equation 6. This is the update for $q(z)$.

Recall that $\eta(\theta)$ maps the nonconjugate variable θ to the natural parameter of the conjugate variable z. The update for $q(z)$ requires computing $\mathbb{E}_{q(\theta)}[\eta(\theta)]$. For some models, this expectation is computable. If not, we can take a Taylor approximation of $\eta(\theta)$ around the variational parameter μ,

$$\eta_i(\theta) \approx \eta_i(\mu) + \nabla\eta(\mu)_i^\top(\theta-\mu) + \tfrac{1}{2}(\theta-\mu)^\top\nabla^2\eta_i(\mu)(\theta-\mu),$$

where $\eta(\theta)$ is a vector and i indexes the ith component. This requires $\eta(\theta)$ is twice differentiable, which is satisfied in most models. Since $q(\theta) = \mathcal{N}(\mu,\Sigma)$, this means that

$$\mathbb{E}_{q(\theta)}[\eta_i(\theta)] \approx \eta_i(\mu) + \tfrac{1}{2}\mathrm{Tr}\left\{\nabla^2\eta_i(\mu)\Sigma\right\}. \tag{18}$$

(Note that the linear term $\mathbb{E}_{q(\theta)}\left[\nabla\eta_i(\mu)^T(\theta-\mu)\right] = 0$.)

Using delta method variational inference to update $q(\theta)$, the update for $q(z)$ is identical to that in Laplace variational inference. We isolate the relevant terms in Equation 3,

$$\mathcal{L}(q(z)) = \mathbb{E}_{q(z)}\left[\log p(x\,|\,z) + \log h(z) + \mathbb{E}_{q(\theta)}[\eta(\theta)]^\top t(z)\right] - \mathbb{E}_{q(z)}[\log q(z)].$$

1: Initialize variational distributions $q(\theta \mid \mu, \Sigma)$ and $q(z \mid \phi)$.
2: **repeat**
3: Compute the statistics $\mathbb{E}_{q(z)}[t(z)]$.
4: Update $q(\theta)$ using one of the methods:

a) For Laplace inference, compute Equation 13.
b) For delta method inference, optimize Equation 16.

5: Compute $\mathbb{E}_{q(\theta)}[\eta(\theta)]$ or approximate it as in Equation 18.
6: Update $q(z)$ from Equation 17.
7: **until** convergence. (See Section 3.4 for the criterion.)
8: **return** $q(\theta)$ and $q(z)$.

Figure 1: Nonconjugate variational inference

Setting the partial gradient $\partial \mathcal{L}(q(z))/\partial q(z) = 0$ gives the same optimal $q(z)$ of Equation 5. Computing this update reduces to the approach for Laplace variational inference in Equation 17.

We return again to the unigram language model with log normal priors on the Dirichlet parameters. In this model, we can compute $\mathbb{E}_{q(\theta)}[\eta(\theta)]$ exactly by using properties of the log normal,

$$\mathbb{E}_{q(\theta)}[\eta(\theta)] = \mathbb{E}_{q(\theta)}[\exp\{\theta\}] = \exp\{\mu + \mathrm{diag}(\Sigma)/2\}.$$

Recall that x_d are the word counts for document d and note that it is its own sufficient statistic in a multinomial count model. Given the calculation of $\mathbb{E}_{q(\theta)}[\eta(\theta)]$ and the model-specific calculations in Equation 8, the update for $q(z_d)$ is

$$q(z_d) = \mathrm{Dirichlet}\left(\exp(\mu + \mathrm{diag}(\Sigma)/2) + x_d\right).$$

This completes our derivation in the example model. To implement nonconjugate inference we need this update for $q(z)$ and the definition of $f(\cdot)$ in Equation 14.

3.4 Nonconjugate Variational Inference

We now present the full algorithm for nonconjugate variational inference. In this section, we will be explicit about the variational parameters. Recall that the variational distribution of the nonconjugate variable is a Gaussian $q(\theta \mid \mu, \Sigma)$; the variational distribution of the conjugate variable is $q(z \mid \phi)$, where ϕ is a natural parameter in the same family as $p(z \mid \eta(\theta))$.

The algorithm is as follows. Begin by initializing the variational parameters. Iterate between updating $q(\theta)$ and updating $q(z)$ until convergence. Update $q(\theta)$ by either Equation 13 (Laplace inference) or optimizing Equation 16 (Delta method inference). Update $q(z)$ from Equation 17. Assess convergence by measuring the L_2 norm of the mean of the nonconjugate variable, $\mathbb{E}_q[\theta]$.

This algorithm is summarized in Figure 1. In either Laplace or delta method inference, we have reduced deriving variational updates for complicated nonconjugate models to mechanical work—calculating derivatives and calling a numerical optimization library. We note that Laplace inference is simpler to derive because it only requires second derivatives of the function in Equation 11;

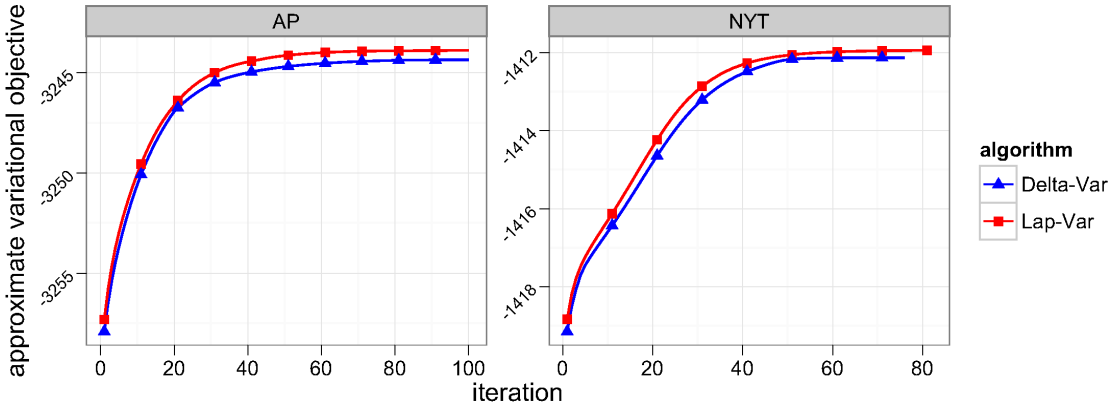


Figure 2: The approximate variational objective from Equation 19 goes up as a function of the iteration. This is for document-level inference in the correlated topic model. The left plot is for a collection from the *Associated Press*; the right plot is for a collection from the *New York Times*. (See Section 4.1 and Section 5.1 for details about the model and data.)

delta method inference requires third derivatives. We study the empirical difference between these methods in Section 5.

Our algorithm (in either setting) is based on approximately optimal coordinate updates for the variational objective, but we cannot compute that objective. However, we can compute an approximate objective at each iteration with the same Taylor approximation used in the coordinate steps, and this can be monitored as a proxy. The approximate objective is

$$\mathcal{L} \approx f(\hat{\theta}) + \nabla f(\hat{\theta})^\top (\mu - \hat{\theta}) + \tfrac{1}{2}(\mu - \hat{\theta})^\top \nabla^2 f(\hat{\theta})(\mu - \hat{\theta}) + \tfrac{1}{2}\left(\mathrm{Tr}\left\{\nabla^2 f(\hat{\theta})\Sigma\right\} + \log|\Sigma|\right) - \mathbb{E}_{q(z)}\left[\log q(z)\right] \quad (19)$$

where $f(\theta)$ is defined in Equation 11 and $\hat{\theta}$ is defined as for Laplace or delta method inference.[5]

Figure 2 shows this score at each iteration for two runs of inference in the correlated topic model. (See Section 4.1 for details about the model.) The approximate objective increases as the algorithm proceeds, and these plots were typical. In practice, as did Braun and McAuliffe (2010) in their setting, we found that this is a good score to monitor.

4 Example Models

We have described a generic algorithm for approximate posterior inference in nonconjugate models. In this section we derive this algorithm for several nonconjugate models from the research literature: the correlated topic model (Blei and Lafferty, 2007), Bayesian logistic regression (Jaakkola and Jordan, 1997), and hierarchical Bayesian logistic regression (Gelman and Hill, 2007). For each

5. We note again that Equation 19 is not the function we are optimizing. Even the simpler Laplace approximation is not clearly minimizing a well-defined distance function between the approximate Gaussian and true posterior (MacKay, 1992). Thus, while this approach is an approximate coordinate ascent algorithm, clearly characterizing the corresponding objective function is an open problem.

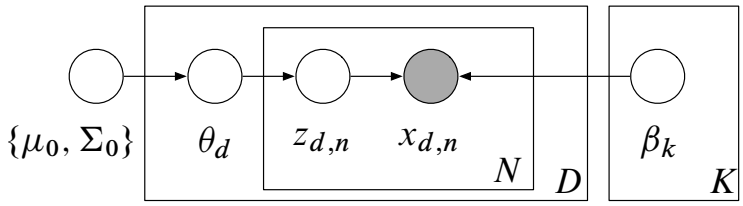


Figure 3: The graphical representation of the correlated topic model (CTM). The nonconjugate variable is θ; the conjugate variable is the collection $z = z_{1:N}$; the observation is the collection of words $x = x_{1:N}$.

model, we identify the variables—the nonconjugate variable θ, conjugate variable z, and observations x—and we calculate $f(\theta)$ from Equation 11. (The calculations of $f(\theta)$ are in the appendices.) In the next section, we study how our algorithms perform when analyzing data under these models.[6]

4.1 The Correlated Topic Model

Probabilistic topic models are models of document collections. Each document is treated as a group of observed words that are drawn from a mixture model. The mixture components, called "topics," are distributions over terms that are shared for the whole collection; each document exhibits them with individualized proportions.

Conditioned on a corpus of documents, the posterior topics place high probabilities on words that are associated under a single theme; for example, one topic may contain words like "bat," "ball," and "pitcher." The posterior topic proportions reflect how each document exhibits those themes; for example, a document may combine the topics of *sports* and *health*. This posterior decomposition of a collection can be used for summarization, visualization, or forming predictions about a document. See Blei (2012) for a review of topic modeling.

The per-document topic proportions are a latent variable. In latent Dirichlet allocation (LDA) (Blei et al., 2003)—which is the simplest topic model—these are given a Dirichlet prior, which makes the model conditionally conjugate. Here we will study the correlated topic model (CTM) (Blei and Lafferty, 2007). The CTM extends LDA by replacing the Dirichlet prior on the topic proportions with a logistic normal prior (Aitchison, 1982). This is a richer prior that can capture correlations between occurrences of the components. For example, a document about *sports* is more likely to also be about *health*. The CTM is not conditionally conjugate. But it is a more expressive model: it gives a better fit to texts and provides new kinds of exploratory structure.

Suppose there are K topic parameters $\beta_{1:K}$, each of which is a distribution over V terms. Let $\pi(\theta)$ denote the multinomial logistic function, which maps a real-valued vector to a point on the simplex with the same dimension, $\pi(\theta) \propto \exp\{\theta\}$. The CTM assumes a document is drawn as follows:

1. Draw log topic proportions $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$.
2. For each word n:
   (a) Draw topic assignment $z_n \,|\, \theta \sim \text{Mult}(\pi(\theta))$.

6. Python implementations of our algorithms are available at `http://www.cs.cmu.edu/~chongw/software/nonconjugate_inference.tar.gz`.

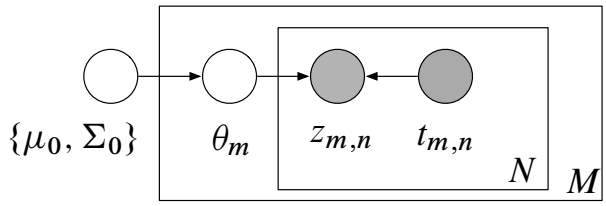


Figure 4: The graphical representation of hierarchical logistic regression. (When $M = 1$, this is standard Bayesian logistic regression.)The nonconjugate variable is the vector of coefficients θ_m, the conjugate variable is the collection of observed classes for each data point, $z_m = z_{m,1:N}$. (In this case there is no additional observation x downstream.)

(b) Draw word $x_n \mid z_n, \beta \sim \text{Mult}(\beta_{z_n})$.

Figure 3 shows the graphical model. The topic proportions $\pi(\theta)$ are drawn from a logistic normal distribution; their correlation structure is captured in its covariance matrix Σ_0. The topic assignment variable z_n indicates from which topic the nth word is drawn.

Holding the topics $\beta_{1:K}$ fixed, the main inference problem in the CTM is to infer the conditional distribution of the document-level hidden variables $p(\theta, z_{1:N} \mid x_{1:N}, \beta_{1:K})$. This calculation is important in two contexts: it is used when forming predictions about new data; and it is used as a subroutine in the variational expectation maximization algorithm for fitting the topics and logistic normal parameters (mean μ_0 and covariance Σ_0) with maximum likelihood. The corresponding per-document inference problem is straightforward to solve in LDA, thanks to conditional conjugacy. In the CTM, however, it is difficult because the logistic normal on θ is not conjugate to the multinomial on z. Blei and Lafferty (2007) used a Taylor approximation designed specifically for this model. Here we apply the generic algorithm from Section 3.

In terms of the earlier notation, the nonconjugate variable is the topic proportions θ, the conjugate variable is the collection of topic assignments $z = z_{1:N}$, and the observation is the collection of words $x = x_{1:N}$. The variational distribution for the topic proportions θ is Gaussian, $q(\theta) = \mathcal{N}(\mu, \Sigma)$; the variational distribution for the topic assignments is discrete, $q(z) = \prod_n q(z_n \mid \phi_n)$ where each ϕ_n is a distribution over K elements. In delta method inference, as in Braun and McAuliffe (2010), we restrict the variational covariance Σ to be diagonal to simplify the derivative of Equation 16. Laplace variational inference does not require this simplification. Appendix B gives the detailed derivations of the algorithm.

Besides the CTM, this approach can be adapted to a variety of nonconjugate topic models, including the topic evolution model (Xing, 2005), Dirichlet-multinomial regression (Mimno and McCallum, 2008), dynamic topic models (Blei and Lafferty, 2006; Wang et al., 2008), and the discrete infinite logistic normal distribution (Paisley et al., 2012b).

4.2 Bayesian Logistic Regression

Bayesian logistic regression is a well-studied model for binary classification (Jaakkola and Jordan, 1997). It places a Gaussian prior on a set of coefficients and draws class labels, conditioned on covariates, from the corresponding logistic. Let t_n is be a p-dimensional observed covariate vector for the nth sample and z_n be its class label (an indicator vector of length two). Let θ be the real-valued

coefficients in $\mathbb{R}^p$; there is a coefficient for each feature. Bayesian logistic regression assumes the following conditional process:

1. Draw coefficients $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$.
2. For each data point n and its covariates t_n, draw its class label from

$$z_n \,|\, \theta, t_n \sim \text{Bernoulli}\left(\sigma(\theta^\top t_n)^{z_{n,1}} \sigma(-\theta^\top t_n)^{z_{n,2}}\right),$$

where $\sigma(y) \triangleq 1/(1+\exp(-y))$ is the logistic function.

Figure 4 shows the graphical model. Given a data set of labeled feature vectors, the posterior inference problem is to compute the conditional distribution of the coefficients $p(\theta \,|\, z_{1:N}, t_{1:N})$. The issue is that the Gaussian prior on the coefficients is not conjugate to the conditional likelihood of the label.

This is a subset of the model class in Section 2.2. The nonconjugate variable θ is identical and the variable z is the collection of observed classes of each data point, $z_{1:N}$. Note there is no additional observed variable x downstream. The variational distribution need only be defined for the coefficients, $q(\theta) = \mathcal{N}(\mu, \Sigma)$. Using Laplace variational inference, our approach recovers the standard Laplace approximation for Bayesian logistic regression (Bishop, 2006). This gives a connection between standard Laplace approximation and variational inference. Delta method variational inference provides an alternative. Appendix C gives the detailed derivations.

An important extension of Bayesian logistic regression is hierarchical Bayesian logistic regression (Gelman and Hill, 2007). It simultaneously models related logistic regression problems, and estimates the hyperparameters of the shared prior on the coefficients. With M related problems, we construct the following hierarchical model:

1. Draw the global hyperparameters,

$$\Sigma_0^{-1} \sim \text{Wishart}(\nu, \Phi_0), \tag{20}$$
$$\mu_0 \sim \mathcal{N}(0, \Phi_1). \tag{21}$$

2. For each problem m:
   (a) Draw coefficients $\theta_m \sim \mathcal{N}(\mu_0, \Sigma_0)$.
   (b) For each data point n and its covariates t_{mn}, draw its class label,

$$z_{mn} \,|\, \theta_m, t_{mn} \sim \text{Bernoulli}(\sigma(\theta_m^\top t_{mn})^{z_{mn,1}} \sigma(-\theta_m^\top t_{mn})^{z_{mn,2}}).$$

As for the CTM, we use nonconjugate inference as a subroutine in a variational EM algorithm (where the M step is regularized). We construct $f(\theta_m)$ in Equation 11 separately for each problem m, and fit the hyperparameters μ_0 and Σ_0 from their approximate expected sufficient statistics (Bishop, 2006). This amounts to MAP estimation with priors as specified above. See Appendix C for the complete derivation.

Finally, we note that logistic regression is a generalized linear model with a binary response and canonical link function (McCullagh and Nelder, 1989). It is straightforward to use our algorithms with other Bayesian generalized linear models (and their hierarchical forms).

5 Empirical Study

We studied nonconjugate variational inference with correlated topic models and Bayesian logistic regression. We found that nonconjugate inference is more accurate than the existing methods tailored to specific models. Between the two nonconjugate inference algorithms, we found that Laplace inference is faster and more accurate than delta method inference.

5.1 The Correlated Topic Model

We studied Laplace inference and delta method inference in the CTM. We compared it to the original inference algorithm of Blei and Lafferty (2007).

We analyzed two collections of documents. The *Associated Press* (AP) collection contains 2,246 documents from the *Associated Press*. We used a vocabulary of 10,473 terms, which gave a total of 436K observed words. The *New York Times* (NYT) collection contains 9,238 documents from the *New York Times*. We used a vocabulary of 10,760 terms, which gave a total of 2.3 million observed words. For each corpus we used 80% of the documents to fit models and reserved 20% to test them.

We fitted the models with variational EM. At each iteration, the algorithm has a set of topics $\beta_{1:K}$ and parameters to the logistic normal $\{\mu_0, \Sigma_0\}$. In the E-step we perform approximate posterior inference with each document, estimating its topic proportions and topic assignments. In the M-step, we re-estimate the topics and logistic normal parameters. We fit models with different kinds of E-steps, using both of the nonconjugate inference methods from Section 3 and the original approach of Blei and Lafferty (2007). To initialize nonconjugate inference we set the variational mean parameter $\mu = 0$ for log topic proportions θ and computed the corresponding updates for the topic assignments z. We initialize the topics in variational EM to random draws from a uniform Dirichlet.

With nonconjugate inference in the E-step, variational EM approximately optimizes a bound on the marginal probability of the observed data. We can calculate an approximation of this bound with Equation 19 summed over all documents. We monitor this quantity as we run variational EM.

To test our fitted models, we measured predictive performance on held-out data with predictive distributions derived from the posterior approximations. We follow the testing framework of Asuncion et al. (2009) and Blei and Lafferty (2007). We fix fitted topics and logistic normal parameters $M = \{\beta_{1:K}, \mu_0, \Sigma_0\}$. We split each held-out document in to two halves $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and form the approximate posterior log topic proportions $q_{\boldsymbol{w}_1}(\theta)$ using one of the approximate inference algorithms and the first half of the document $\boldsymbol{w}_1$. We use this to form an approximate predictive distribution,

$$p(w \mid \boldsymbol{w}_1, M) \approx \int_\theta \sum_z p(w \mid z, \beta_{1:K}) q_{\boldsymbol{w}_1}(\theta) d\theta \approx \sum_{k=1}^K \beta_{kw} \pi_k,$$

where $\pi_k \propto \exp\{\mathbb{E}_q[\theta_k]\}$. Finally, we evaluate the log probability of the second half of the document using that predictive distribution; this is the *held out log likelihood*. A better model and inference method will give higher predictive probabilities of the unseen words. Note that this testing framework puts the approximate posterior distributions on the same playing field. The quantities are comparable regardless of how the approximate posterior is formed.

Figure 5 shows the per-word approximate bound and the per-word held out likelihood as functions of the number of topics. Figure 5 (a) indicates that the approximate bounds from nonconjugate inference generally go up as the number of topics increases. This is a property of a good approximation because the marginal certainly goes up as the number of parameters increases. In contrast,

Blei and Lafferty's (2007) objective (which is a true bound on the marginal of the data) behaves erratically. This is illustrated for the *New York Times* corpus; on the *Associated Press* corpus, it does not come close to the approximate bound and is not plotted.

Figure 5 (b) shows that on held out data, Blei and Lafferty's approach, tailored for this model, performed worse than both of our algorithms. Our conjecture is that while this method gives a strict lower bound on the marginal, it might be a loose bound and give poor predictive distributions. Our methods use an approximation which, while not a bound, might be closer to the objective and give better predictive distributions. The held out likelihood plots also show that when the number of topics increases the algorithms eventually overfit the data. Finally, note that Laplace variational inference was always better than both other algorithms.

Finally, Figure 6 shows the approximate bound and the held out log likelihood as functions of running time.[7] From Figure 6 (a), we see that even though variational EM is not formally optimizing this approximate objective (see Equation 19), the increase at each iteration suggests that the marginal probability is also increasing. The plot also shows that Laplace inference converges faster than delta method inference. Figure 6 (b) confirms that Laplace inference is both faster and gives better predictive performance.

5.2 Bayesian Logistic Regression

We studied our algorithms on Bayesian logistic regression in both standard and hierarchical settings. In the standard setting, we analyzed two data sets. With the *Yeast* data (Elisseeff and Weston, 2001), we form a predictor of gene functional classes from features composed of micro-array expression data and phylogenetic profiles. The data set has 1,500 genes in the training set and 917 genes in the test set. For each gene there are 103 covariates and up to 14 different gene functional classes (14 labels). This corresponds to 14 independent binary classification problems. With the *Scene* data (Boutell et al., 2004), we form a predictor of scene labels from image features. It contains 1,211 images in the training set and 1,196 images in the test set. There are 294 images features and up to 6 scene labels per image. This corresponds to 6 independent binary classification problems.[8]

We used two performance measures. First we measured accuracy, which is the proportion of test-case examples correctly labeled. Second, we measured average log predictive likelihood. Given a test-case input t with label z, we compute the log predictive likelihood,

$$\log p(z \,|\, \mu, t) = z_1 \log \sigma(\mu^\top t) + z_2 \log \sigma(-\mu^\top t),$$

where μ is the mean of variational distribution $q(\theta) = \mathcal{N}(\mu, \Sigma)$. Higher likelihoods indicate a better fit. For both accuracy and predictive likelihood, we used cross validation to estimate the generalization performance of each inference algorithm. We set the priors $\mu_0 = 0$ and $\Sigma_0 = I$.

We compared Laplace inference (Section 3.1), delta method inference (Section 3.2), and the method of Jaakkola and Jordan (1997). Jaakkola and Jordan's (1996) method preserves a lower bound on the marginal likelihood with a first-order Taylor approximation and was developed specifically for Bayesian logistic regression. (We note that Blei and Lafferty's bound-preserving method for the CTM was built on this technique.)

7. We did not formally compare the running time of Blei and Lafferty's (2007) method because we used the authors' C implementation, while ours is in Python. We observed that their method took more than five times longer than ours.
8. The *Yeast* and *Scene* data are at `http://mulan.sourceforge.net/datasets.html`.

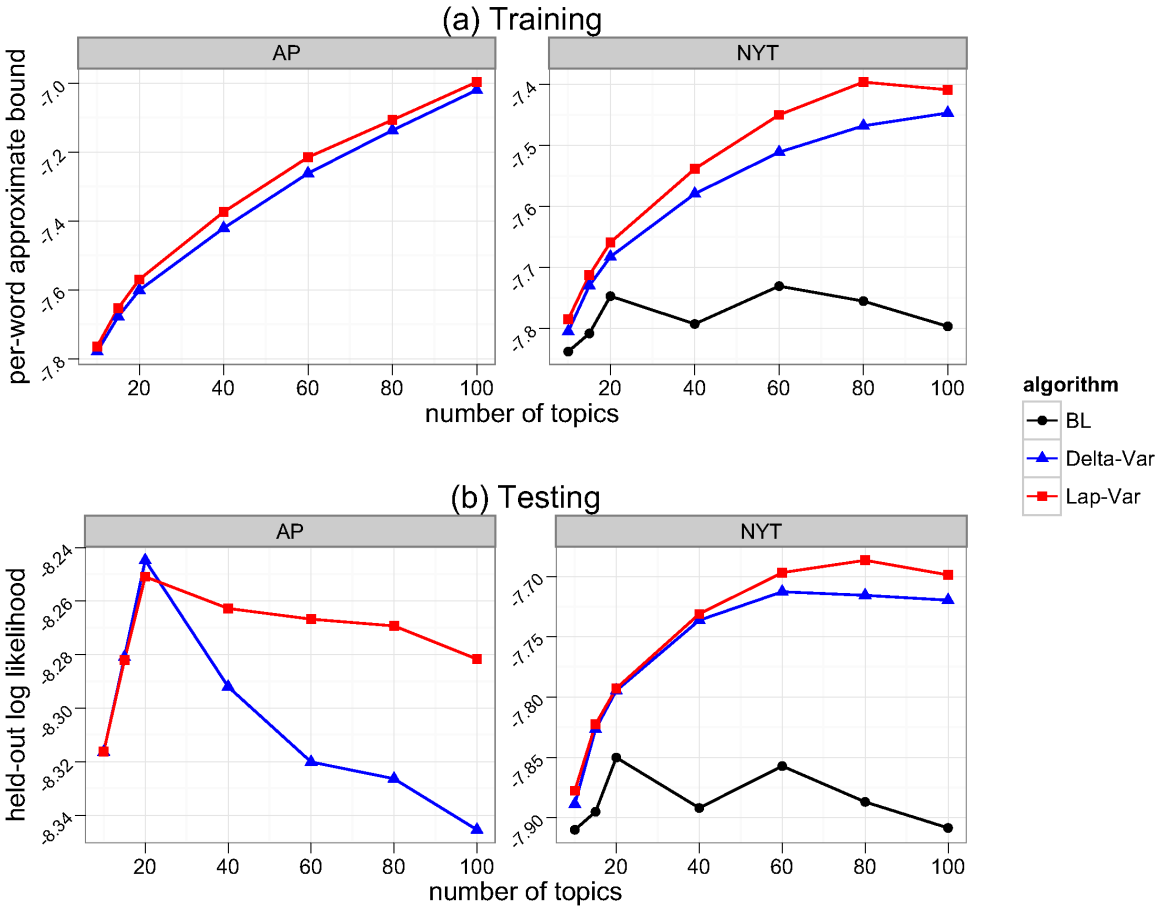


Figure 5: Laplace variational inference is "Lap-Var"; delta method variational inference is "Delta-Var"; Blei and Lafferty's method is "BL." (a) Approximate per-word lower bound against the number of topics. A good approximation will go up as the number of topics increases, but not necessarily indicate a better predictive performance on the held out data. (b) Per-word held-out log likelihood against the number of topics. Higher numbers are better. Both nonconjugate methods perform better than Blei and Lafferty's method. Laplace inference performs best. Blei and Lafferty's method was erratic in both collections. (It is not plotted for the AP collection.)

Table 1 gives the results. To compare methods we compute the difference in score (accuracy or log likelihood) on the independent binary classification problems, and then perform a standard t-test (at level 0.05) to test if the mean of the differences is larger than 0. Laplace inference and delta method inference gave slightly better accuracy than Jaakkola and Jordan's method, and much

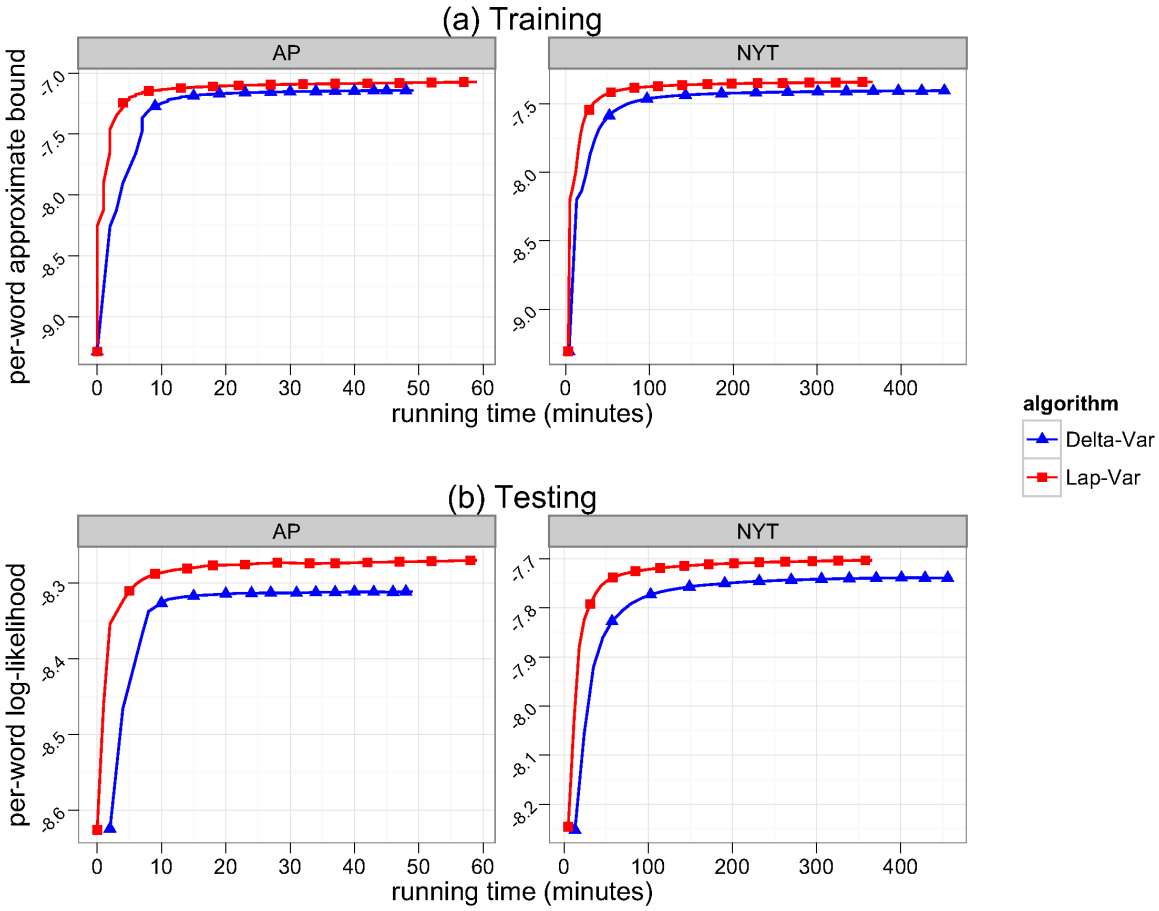


Figure 6: In this figure, we set the number of topics as $K = 60$. (Others are similar.) (a) The per-word approximate bound during model fitting with variational EM. Though it is an approximation of the variational EM objective, it converges in practice. (b) The per-word held out likelihood during the model fitting with variational EM. Laplace inference performs best in terms of speed and predictive performance.

better log predictive likelihood.[9] The t-test showed that both Laplace and delta method inference are better than Jaakkola and Jordan's method.

We next examined a data set of student performance in a collection of schools. With the *School* data, our goal is to use various features of a student to predict whether he or she will perform above or below the median on a standardized exam.[10] The data came from the Inner London Education Authority. It contains examination records from 139 secondary schools for the years 1985, 1986 and 1987. It is a random 50% sample with 15,362 students. The students' features contain four student-dependent features and school-dependent features. The student dependent features are

9. Previous literature, for example, Xue et al. (2007) and Archambeau et al. (2011) treat *Yeast* and *Scene* as multi-task problems. In our study, we found that our standard Bayesian logistic regression algorithms performed the same as the algorithms developed in these papers.
10. The data is available at `http://multilevel.ioe.ac.uk/intro/datasets.html`.

	Yeast		*Scene*	
	Accuracy	Log Likelihood	Accuracy	Log Likelihood
Jaakkola and Jordan (1996)	79.7%	-0.678	87.4%	-0.670
Laplace inference	**80.1%**	**-0.449**	**89.4%**	**-0.259**
Delta method inference	**80.2%**	**-0.450**	**89.5%**	-0.265

Table 1: Comparison of the different methods for Bayesian logistic regression using accuracy and averaged log predictive likelihood. Higher numbers are better. These results are averaged from five random starts. (The variance is too small to report.) Bold results indicate significantly better performance using a standard t-test. Laplace and delta method inference perform best.

the year of the exam, gender, VR band (individual prior attainment data), and ethnic group; the school-dependent features are the percentage of students eligible for free school meals, percentage of students in VR band 1, school gender, and school denomination. We coded the binary indicator of whether each was below the median ("bad") or above ("good"). We use the same 10 random splits of the data as Argyriou et al. (2008).

In this data, we can either treat each school as a separate classification problem, pool all the schools together as a single classification problem, or analyze them with hierarchical logistic regression (Section 4.2). The hierarchical model allows the predictors for each school to deviate from each other, but shares statistical strength across them. Let p be the number of covariates. We set the prior on the hyperparameters to the coefficients to $\nu = p + 100$, $\Phi_0 = 0.01I$, and $\Phi_1 = 0.01I$ (see Equation 20 and Equation 21) to favor sparsity. We initialized the variational distributions to $q(\theta) = \mathcal{N}(0, I)$.

Table 2 gives the results. A standard t-test (at level 0.05) showed that the hierarchical models are better than the non-hierarchical models both in terms of accuracy and predictive likelihood. With predictive likelihood, Laplace variational inference in the hierarchical model is significantly better than all other approaches.

6 Discussion

We developed Laplace and delta method variational inference, two strategies for variational inference in a large class of nonconjugate models. These methods approximate the variational objective function with a Taylor approximation, each in a different way. We studied them in two nonconjugate models and showed that they work well in practice, forming approximate posteriors that lead to good predictions. In the examples we analyzed, our methods worked better than methods tailored for the specific models at hand. Between the two, Laplace inference was better and faster than delta method inference. These methods expand the scope of variational inference.

		Accuracy	Log Likelihood
Separate			
	Jaakkola and Jordan (1996)	70.5%	-0.684
	Laplace inference	70.8%	-0.569
	Delta inference	70.8%	-0.571
Pooled			
	Jaakkola and Jordan (1996)	71.2%	-0.685
	Laplace inference	71.3%	-0.557
	Delta inference	71.3%	-0.557
Hierarchical			
	Jaakkola and Jordan (1996)	71.3%	-0.685
	Laplace inference	**71.9%**	**-0.549**
	Delta inference	**71.9%**	-0.559

Table 2: Comparison of the different methods on the *School* data using accuracy and averaged log predictive likelihood. Results are averaged from 10 random splits. (The variance is too small to report.) We compared Laplace inference, delta inference and Jaakkola and Jordan's (1996) method in three settings: separate logistic regression models for each school, a pooled logistic regression model for all schools, and the hierarchical logistic regression model in Section 4.2. Bold indicates significantly better performance by a standard t-test (at level 0.05). The hierarchical model performs best.

Acknowledgments

We thank Jon McAuliffe and the anonymous reviewers for their valuable comments. Chong Wang was supported by Google Ph.D. and Siebel Scholar Fellowships. David M. Blei is supported by NSF IIS-0745520, NSF IIS-1247664, NSF IIS-1009542, ONR N00014-11-1-0651, and the Alfred P. Sloan foundation.

Appendix A. Generalization to Complex Models

We describe how we can generalize our approaches to more complex models. Suppose we have a directed probabilistic model with latent variables $\theta = \theta_{1:m}$ and observations x. (We will not differentiate notation between conjugate and nonconjugate variables.) The log joint likelihood of all latent and observed variables is

$$\log p(\theta, x) = \sum_{i=1}^{m} \log p(\theta_i \,|\, \theta_{\pi_i}) + \log p(x \,|\, \theta),$$

where π_i are the indices of the parents of θ_i, the variables it depends on.

Our goal is to approximate the posterior distribution $p(\theta \,|\, x)$. Similar to the main paper, we use mean-field variational inference (Jordan et al., 1999). We posit a fully-factorized variational family

$$q(\theta) = \prod_{i=1}^{m} q(\theta_i),$$

and optimize ach factor $q(\theta_i)$ to find the member closest in KL-divergence to the posterior.

As in the main paper, we solve this optimization problem with coordinate ascent, iteratively optimizing each variational factor while holding the others fixed. Recall that Bishop (2006) shows that this leads to the following update

$$q(\theta_i) \propto \exp\left\{\mathrm{E}_{-i}\left[\log p(\theta, x)\right]\right\}, \tag{22}$$

where $\mathrm{E}_{-i}[\cdot]$ denotes the expectation with respect to $\prod_{j, j \neq i} q(\theta_j)$.

Many of the terms of the log joint will be constant with respect to θ_i and absorbed into the constant of proportionality. This allows us to simplify the update in Equation 22 to be $q(\theta_i) \propto \exp\{f(\theta_i)\}$ where

$$f(\theta_i) = \mathrm{E}_{-i}\left[\log p(\theta_i \,|\, \theta_{\pi_i})\right] + \sum_{\{j : i \in \pi_j\}} \mathrm{E}_{-i}\left[\log p(\theta_j \,|\, \theta_{\pi_j})\right] + \mathrm{E}_{-i}\left[\log p(x \,|\, \theta)\right]. \tag{23}$$

As in the main paper, this update is not tractable in general. We use Laplace variational inference (Section 3.1) to approximate it, although delta method variational inference (Section 3.2) is also applicable. In Laplace variational inference, we take a Taylor approximation of $f(\theta_i)$ around its maximum $\hat{\theta}_i$. This naturally leads to $q(\theta_i)$ as a Gaussian factor,

$$q^*(\theta_i) \approx \mathcal{N}(\hat{\theta}_i, -\nabla^2 f(\hat{\theta}_i)^{-1}).$$

The main paper considers the case where θ is a single random variable and updates its variational distribution. In the more general coordinate ascent setting considered here, we need to compute or approximate the expected log probabilities (and their derivatives) in Equation 23.

Now suppose each factor is in the exponential family. (This is weaker than the conjugacy assumption, and describes most graphical models from the literature.) The log joint likelihood becomes

$$\log p(\theta, x) = \sum_{i=1}^{m} \left(\eta(\theta_{\pi_i})^\top t(\theta_i) - a(\eta(\theta_{\pi_i}))\right) + \log p(x \mid \theta),$$

where $\eta(\cdot)$ are natural parameters, $t(\cdot)$ are sufficient statistics, and $a(\eta(\cdot))$ are log normalizers. (All are overloaded.) Substituting the exponential family assumptions into $f(\theta_i)$ gives

$$\begin{aligned} f(\theta_i) &= \mathrm{E}_{-i}\left[\eta(\theta_{\pi_i})\right]^\top t(\theta_i) \\ &\quad + \sum_{\{j:i\in\pi_j\}} \left(\mathrm{E}_{-i}\left[\eta(\theta_{\pi_j})\right]^\top \mathrm{E}_{-i}\left[t(\theta_j)\right] - \mathrm{E}_{-i}\left[a(\eta(\theta_{\pi_j}))\right]\right) \\ &\quad + \mathrm{E}_{-i}\left[t(\theta)\right]^\top t(x) - \mathrm{E}_{-i}\left[a(\eta(\theta))\right]. \end{aligned}$$

Here we can use further Taylor approximations of the natural parameters $\eta(\cdot)$, sufficient statistics $t(\cdot)$, and log normalizers $a(\cdot)$ in order to easily take their expectations.

Finally, for some variables we may be able to exactly compute $f(\theta_i)$ and form the $q^*(\theta_i)$ without further approximations. (These are conjugate variables for which the complete conditional $p(\theta_i \mid \theta_{-i}, x)$ is available in closed form.) These variables were separated out in the main paper; here we note that they can be updated exactly in the coordinate ascent algorithm.

Appendix B. The Correlated Topic Model

The correlated topic model is described in Section 4.1. We identify the quantities from Equation 6 and Equation 7 that we need to compute $f(\theta)$ in Equation 11,

$$\begin{aligned} h(z) = 1, \;\; t(z) &= \textstyle\sum_n z_n, \\ \eta(\theta) &= \theta - \log\left\{\textstyle\sum_k \exp\{\theta_k\}\right\}, \\ a(\eta(\theta)) &= 0. \end{aligned}$$

With this notation,

$$f(\theta) = \eta(\theta)^\top \mathbb{E}_{q(z)}\left[t(z)\right] - \tfrac{1}{2}(\theta - \mu_0)^\top \Sigma_0^{-1}(\theta - \mu_0),$$

where $\mathbb{E}_{q(z)}\left[t(z)\right]$ is the expected word counts of each topic under the variational distribution $q(z)$.

Let $\pi \propto \exp\{\eta(\theta)\}$ be the topic proportions. Using $\partial\pi_i/\partial\theta_j = \pi_i(1_{[i=j]} - \pi_j)$, we obtain the gradient and Hessian of the function $f(\theta)$ in the CTM,

$$\begin{aligned} \nabla f(\theta) &= \mathbb{E}_{q(z)}\left[t(z)\right] - \pi \textstyle\sum_{k=1}^{K} \left[\mathbb{E}_{q(z)}\left[t(z)\right]\right]_k - \Sigma_0^{-1}(\theta - \mu_0), \\ \nabla^2 f(\theta)_{ij} &= (-\pi_i 1_{[i=j]} + \pi_i\pi_j) \textstyle\sum_{k=1}^{K} \left[\mathbb{E}_{q(z)}\left[t(z)\right]\right]_k - (\Sigma_0^{-1})_{ij}. \end{aligned}$$

where $1_{[i=j]} = 1$ if $i = j$ and 0 otherwise. Note that $\nabla f(\theta)$ is all we need for Laplace inference.

In delta method variational inference, we also need to compute the gradient of

$$\mathrm{Trace}\left\{\nabla^2 f(\theta)\Sigma\right\} = \left(-\textstyle\sum_{k=1}^{K} \pi_k \Sigma_{kk} + \pi^T \Sigma \pi\right) \textstyle\sum_{k=1}^{K} \left[\mathbb{E}_{q(z)}\left[t(z)\right]\right]_k - \mathrm{Trace}(\Sigma_0^{-1}\Sigma).$$

Following Braun and McAuliffe (2010), we assume Σ is diagonal in the delta method. (In Laplace inference, we do not need this assumption.) This gives

$$\frac{\partial \mathrm{Trace}\left\{\nabla^2 f(\theta)\Sigma\right\}}{\partial \theta_i} = \pi_i(1 - 2\pi_i)(\textstyle\sum_k \pi_k \Sigma_{kk} - 1).$$

These quantities let us implement the algorithm in Figure 1 to infer the per-document posterior of the CTM hidden variables.

As we discussed Section 4.1, we use this algorithm in variational EM for finding maximum likelihood estimates of the model parameters. The E-step runs posterior inference on each document. Since the variational family is the same, the M-step is as described in Blei and Lafferty (2007).

Appendix C. Bayesian Logistic Regression

Bayesian logistic regression is described in Section 4.2.

The distribution of the observations $z_{1:N}$ fit into the exponential family as follows,

$$\begin{aligned} h(z) &= 1, \;\; t(z) = [z_1, \ldots, z_N], \\ \eta(\theta) &= [\log \sigma(\theta^\top t_n), \log \sigma(-\theta^\top t_n)]_{n=1}^N, \\ a(\eta(\theta)) &= 0. \end{aligned}$$

In this set up, $t(z)$ represents the whole set of labels. Since z is observed, its "expectation" is just itself. With this notation, $f(\theta)$ from Equation 11 is

$$f(\theta) = \eta(\theta)^\top t(z) - \tfrac{1}{2}(\theta - \mu_0)^\top \Sigma_0^{-1}(\theta - \mu_0).$$

The gradient and Hessian of $f(\theta)$ are

$$\begin{aligned} \nabla f(\theta) &= \textstyle\sum_{n=1}^N t_n \left(z_{n,1} - \sigma(\theta^T t_n)\right) - \Sigma_0^{-1}(\theta - \mu_0), \\ \nabla^2 f(\theta) &= -\textstyle\sum_{n=1}^N \sigma(\theta^T t_n)\sigma(-\theta^T t_n) t_n t_n^T - \Sigma_0^{-1}. \end{aligned} \tag{24}$$

This is the standard Laplace approximation to Bayesian logistic regression (Bishop, 2006).

For delta variational inference, we also need the gradient for Trace $\left\{\nabla^2 f(\theta)\Sigma\right\}$. It is

$$\frac{\partial \text{Trace}\left\{\nabla^2 f(\theta)\Sigma\right\}}{\partial \theta_i} = -\sum_{n=1}^N \sigma(\theta^T t_n)\sigma(-\theta^T t_n)(1 - 2\sigma(\theta^T t_n)) t_n t_n^T \Sigma t_n.$$

Here we do not need to assume Σ is diagonal, since the special structure of the Hessian in Equations 24 makes the computation of Trace $\left\{\nabla^2 f(\theta)\Sigma\right\}$ fairly simple.

C.1 Hierarchical Logistic Regression

Here we describe how we update the global hyperparameters (μ_0, Σ_0) (Equations 20 and 21) in hierarchical logistic regression. At each iteration, we first compute the variational distribution of coefficients θ_m for each problem $m = 1, ..., M$,

$$q(\theta_m) = \mathcal{N}(\mu_m, \Sigma_m).$$

We then estimate the global hyperparameters (μ_0, Σ_0) using the MAP estimate. These come from the following update equations,

$$\begin{aligned} \mu_0 &= \left(\frac{\Sigma_0 \Phi_1^{-1}}{M} + I_p\right)^{-1} \frac{\sum_{m=1}^M \mu_m}{M}, \\ \Sigma_0 &= \frac{\Phi_0^{-1} + \sum_{m=1}^M (\mu_m - \mu_0)(\mu_m - \mu_0)^\top}{M + \nu - p - 1}, \end{aligned}$$

where p is the dimension of coefficients θ_m.

References

A. Ahmed and E. Xing. On tight approximate inference of the logistic normal topic admixture model. In *Workshop on Artificial Intelligence and Statistics*, 2007.

J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society, Series B*, 44(2):139–177, 1982.

C. Archambeau, S. Guo, and O. Zoeter. Sparse Bayesian multi-task learning. In *Advances in Neural Information Processing Systems*, 2011.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Maching Learning*, 73:243–272, December 2008.

A. Asuncion, M. Welling, P. Smyth, and Y. Teh. On smoothing and inference for topic models. In *Uncertainty in Artificial Intelligence*, 2009.

H. Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*, 2000.

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

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

J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley & Sons Ltd., Chichester, 1994.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

P. Bickel and K. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*, volume 1. Pearson Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2007.

C. Bishop. Variational principal components. In *International Conference on Artificial Neural Networks*, volume 1, pages 509–514. IET, 1999.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer New York., 2006.

C. Bishop, D. Spiegelhalter, and J. Winn. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems*, 2003.

D. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

D. Blei and J. Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, 2006.

D. Blei and J. Lafferty. A correlated topic model of Science. *Annals of Applied Statistics*, 1(1): 17–35, 2007.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

M. Braun and J. McAuliffe. Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 2010.

L. Brown. *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, Hayward, CA, 1986.

B. Carlin and N. Polson. Inference for nonconjugate Bayesian models using the Gibbs sampler. *Canadian Journal of Statistics*, 19(4):399–405, 1991.

J. Clinton, S. Jackman, and D. Rivers. The statistical analysis of roll call data. *American Political Science Review*, 98(2):355–370, 2004.

A. Corduneanu and C. Bishop. Variational Bayesian model selection for mixture distributions. In *International Conference on Artifical Intelligence and Statistics*, 2001.

W. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, 2001.

J. Fox. *Bayesian Item Response Modeling: Theory and Applications*. Springer Verlag, 2010.

A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

S. Gershman, M. Hoffman, and D. Blei. Nonparametric variational inference. In *International Conference on Machine Learning*, 2012.

Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 31(1), 1997.

A. Honkela and H. Valpola. Unsupervised variational Bayesian learning of nonlinear models. In *Advances in Neural Information Processing Systems*, 2004.

T. Jaakkola and M. Jordan. Bayesian logistic regression: A variational approach. In *Artificial Intelligence and Statistics*, 1997.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

M. Khan, B. Marlin, G. Bouchard, and K. Murphy. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems*, 2010.

D. Knowles and T. Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Neural Information Processing Systems*, 2011.

S. Kotz, N. Balakrishnan, and N. Johnson. *Continuous Multivariate Distributions, Models and Applications*, volume 334. Wiley-Interscience, 2000.

D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

P. McCullagh and J. Nelder. *Generalized Linear Models*. London: Chapman and Hall, 1989.

D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, 2008.

T. Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*, 2001.

T. Minka, J. Winn, J. Guiver, and D. Knowles. Infer.NET 2.4, 2010. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

J. Paisley, D. Blei, and M. Jordan. Stick-breaking beta processes and the Poisson process. In *Artificial Intelligence and Statistics*, 2012a.

J. Paisley, C. Wang, and D. Blei. The discrete infinite logistic normal distribution. *Bayesian Analysis*, 7(2):235–272, 2012b.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society, Series B (Methodological)*, 71(2):319–392, 2009.

A. Smola, V. Vishwanathan, and E. Eskin. Laplace propagation. In *Advances in Neural Information Processing Systems*, 2003.

L. Tierney, R. Kass, and J. Kadane. Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of American Statistical Association*, 84(407), 1989.

M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Uncertainty in Artificial Intelligence*, 2008.

M. Wells. Generalized linear models: A Bayesian perspective. *Journal of American Statistical Association*, 96(453):339–355, 2001.

E. Xing. On topic evolution. *CMU-ML TR-05-115*, 2005.

E. Xing, M. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence*, 2003.

Y. Xue, D. Dunson, and L. Carin. The matrix stick-breaking process for flexible multi-task learning. In *International Conference on Machine Learning*, 2007.

Journal of Machine Learning Research 14 (2013) 1033-1090 Submitted 8/12; Revised 1/13; Published 4/13

Beyond Fano's Inequality: Bounds on the Optimal F-Score, BER, and Cost-Sensitive Risk and Their Implications

Ming-Jie Zhao* MZHAO@CS.MANCHESTER.AC.UK
Narayanan Edakunni EDAKUNNI@CS.MANCHESTER.AC.UK
Adam Pocock ADAM.POCOCK@CS.MANCHESTER.AC.UK
Gavin Brown GAVIN.BROWN@CS.MANCHESTER.AC.UK
School of Computer Science
University of Manchester
Manchester M13 9PL, UK

Editor: Charles Elkan

Abstract

Fano's inequality lower bounds the probability of transmission error through a communication channel. Applied to classification problems, it provides a lower bound on the Bayes error rate and motivates the widely used Infomax principle. In modern machine learning, we are often interested in more than just the error rate. In medical diagnosis, different errors incur different cost; hence, the overall risk is cost-sensitive. Two other popular criteria are balanced error rate (BER) and F-score. In this work, we focus on the two-class problem and use a general definition of conditional entropy (including Shannon's as a special case) to derive upper/lower bounds on the optimal F-score, BER and cost-sensitive risk, extending Fano's result. As a consequence, we show that *Infomax is not suitable for optimizing F-score or cost-sensitive risk*, in that it can potentially lead to low F-score and high risk. For cost-sensitive risk, we propose a new conditional entropy formulation which avoids this inconsistency. In addition, we consider the common practice of using a threshold on the posterior probability to tune performance of a classifier. As is widely known, a threshold of 0.5, where the posteriors cross, minimizes error rate—we derive similar optimal thresholds for F-score and BER.

Keywords: balanced error rate, F-score (F_β-measure), cost-sensitive risk, conditional entropy, lower/upper bound

1. Introduction

In the information theory literature, Fano's inequality (Fano, 1961) is a well known result linking the transmission error probability of a noisy communication channel to standard information theoretic quantities such as conditional entropy and mutual information (Shannon, 1948). From a machine learning perspective, we can treat a classification *problem* as a noisy channel; then the inequality provides us with a lower bound on the *Bayes error rate*, that is, the minimum error rate attainable by any classifier, for that problem. A few years later, several upper bounds were also reported, of which the simplest one is as follows: the Bayes error rate of a multi-class problem cannot exceed half of the Shannon conditional entropy (of the class label given the feature vector). This relationship was first obtained by Tebbe and Dwyer III (1968)[1]—see Equation (7) therein, and later by Hellman

∗. The corresponding author
1. We thank an anonymous reviewer for bringing this to our attention.

and Raviv (1970) using a different argument from Tebbe's. It will be nevertheless referred to as Hellman's bound or Hellman's inequality in the paper, as Tebbe's result is actually stronger than the one we have just stated. See Appendix A (Figure 10) for more detail.

In practice, information measures are often easier than the error probability to evaluate and manipulate (Kailath, 1967). Consequently, both Fano's and Hellman's bounds are useful since they give, from the respective side, some indication of the minimum achievable error rate for a given classification task. More importantly, as shown by Figure 1, the two bounds are both increasing functions of the conditional entropy. Therefore, minimizing the conditional entropy of a system is roughly equivalent to minimizing its probability of error or Bayes error rate. This justifies a general learning principle proposed in the late 1980's, called the *Infomax* or *maximum information preservation* principle:[2]

> **The Infomax Principle** (Linsker, 1989, p. 186): The principle applies to a layer L of cells that provides input to a next layer M. The mapping of the input signal vector L onto an output signal vector M, $f : L \to M$, is characterized by a conditional probability density function ("pdf") $P(M|L)$. The set S of allowed mappings f is specified. The input pdf $P_L(L)$ is also given. The infomax principle states that a mapping f should be chosen for which the Shannon information rate [*the authors: that is, the mutual information* $I(L;M)$] is a maximum (over all f in the set S).
>
> **The Infomax Principle** (Linsker, 1988, p. 486): An equivalent statement of this principle is: The L-to-M transformation is chosen so as to minimize the amount of information that would be conveyed by the input values L to someone who already knows the output values M. [*The authors: that is, the Shannon conditional entropy* $H(M|L)$ *is the quantity to be minimized.*]

As an optimization principle, Infomax has been employed to devise learning algorithms for a wide range of applications. For instance, Linsker (1989) used it to identify independent input signals fed into a linear system from the system's output. His work was later extended by Bell and Sejnowski (1995) to nonlinear systems, yielding an *independent component analysis* algorithm that is capable of successfully separating unknown mixtures of up to ten speakers.

Another important example is the family of information theoretic filtering methods for feature selection and extraction (Guyon and Elisseeff, 2003; Torkkola, 2003; Duch, 2006). In feature selection (for classification problems), the input signal could be any subset of features, X_θ, where, following Brown et al. (2012), θ is a binary vector with a 1 indicating the corresponding feature is selected and a 0 indicating it is discarded. The output signal is the class label Y. The Infomax principle in this context can thus be stated as:

> *A subset of features* X_θ *should be chosen so that the mutual information* $I(X_\theta;Y)$ *is maximized, or, equivalently, the conditional entropy* $H(Y|X_\theta)$ *is minimized.*

Indeed, this is well justified by Fano's inequality and the monotonically increasing relationship between Fano's bound on error probability and conditional entropy (see Figure 1). As shown by Brown et al. (2012), most of the mutual-information-based feature selection filters in the literature are in fact heuristic approximations of the above Infomax principle, under different independence

2. While Linsker directly introduced it as a heuristic principle, we highlight the close relationship between Infomax and the error rate minimization principle; and regard the former as a "derived" principle of the latter.

assumptions on features. It seems that people have been taking Infomax for granted: many believe that choosing those features sharing the maximum mutual information with the class label will best facilitate the subsequent classification procedure. In this paper, however, we will show that *this is* **not** *necessarily the case when F-score or cost-sensitive risk is concerned*, via both analytical analysis and numerical examples.

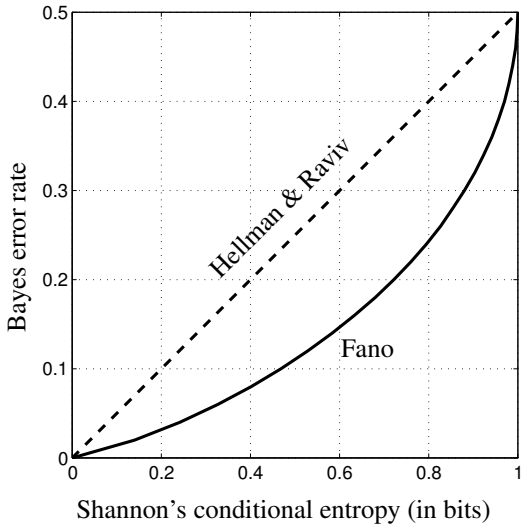


Figure 1: The lower (Fano) and upper (Hellman & Raviv) bounds on the Bayes error rate in terms of Shannon's conditional entropy, for the two-class problem. As both bounds are increasing functions of the conditional entropy, minimizing the conditional entropy would implicitly minimize the Bayes error rate.

Inspired by Fano's result and its widespread utility in machine learning, over the past several decades many researchers have focused on deriving new lower/upper bounds of the Bayes error rate, using various definitions of conditional entropy. We have already mentioned the works of Tebbe and Dwyer III (1968) and Hellman and Raviv (1970). Besides that, Ben-Bassat (1978) derived the lower and upper bounds by means of f-entropies, following the lines originally proposed by Kovalevsky (1968). Using the same method, Golic (1987) discussed the lower and upper bounds based on what he called *concave measures* and *information measures*. Later, Feder and Merhav (1994) re-derived the same upper bound in terms of Shannon's conditional entropy. More recently, Erdogmus and Principe (2004) proposed a family of lower/upper bounds in terms of the Rényi entropy (Rényi, 1961). These bounds are all *increasing* functions of the concerned entropy. This extends our understanding of the Infomax principle, since there are dozens of definitions of entropy in the literature (Taneja, 2001) of which most can be used as the objective function. For instance, Hild II et al. (2006) proposed a mutual information measure based on Rényi's quadratic entropy; and use it to perform feature extraction in the Infomax framework.

All the above bounds are on the Bayes error rate; and to date no *analytical* investigation has been reported on the relationship between conditional entropy and other performance criteria of

classifiers such as F-score and balanced error rate.[3] On the other hand, both balanced error rate and F-score are widely employed in practice; and under certain circumstances they are of more interest than the error rate. Indeed, F-score is used widely in the field of *information retrieval* (Manning et al., 2008); whereas balanced error rate is suitable for situations where the distribution of objects is biased among classes. Another situation for which the error rate alone is of little interest is that when different decision errors incur different penalties. In this case, the cost difference between different kinds of errors should be taken into account; the resulting performance measure is called *cost-sensitive risk* in this paper.

As we have discussed above, it is the monotonicity of Fano's and Hellman's bounds that justifies Infomax as an optimization principle for minimizing the error rate. An important question arises: "*is it still a rational principle when the ultimate goal is to minimize the balanced error rate, to maximize the F-score, or to minimize the cost-sensitive risk?*" In this work, we provide an answer to this question, by first deriving the *tight* lower/upper bounds on the *minimum* balanced error rate, the *maximum* F-score and the *minimum* cost-sensitive risk, as functions of conditional entropy; and then examining the monotonicity of these bounds.

1.1 Paper Outline

For binary classification problems Fano's and Hellman's inequalities provide respectively the tight lower and upper bounds on the Bayes error rate (the *minimal* achievable error rate), in terms of the Shannon conditional entropy. Analogously, in this paper we concentrate on the two-class problem and aim to derive the *tight* lower and upper bounds on the *minimum* balanced error rate, on the *maximum* F-score, and on the *minimum* cost-sensitive risk. We however shall do this using a general definition of conditional entropy that includes Shannon's as a special case, in three steps:

1. Derive the analytical expressions of balanced error rate, F-score and cost-sensitive risk for a given classifier applied to a given classification task. These three quantities will be denoted as BER, FSC and CSR, respectively. See Table 1 for a list of the notations consistently employed in this paper.

2. Compute the optimum values of BER, FSC and CSR over *all classifiers*. The resulting quantities are denoted as $\underline{\text{BER}}$, $\overline{\text{FSC}}$ and $\underline{\text{CSR}}$, respectively. Note here that we use the underline (overline) to indicate that a quantity has been minimized (maximized).

3. Derive the tight lower and upper bounds on $\underline{\text{BER}}$, on $\overline{\text{FSC}}$ and on $\underline{\text{CSR}}$, by means of the conditional entropy (of the considered problem) as given by Definition 2 (page 1043).

Notice that while the values of BER, FSC and CSR depend on both the given task and the concerned classifier, their optimum values $\underline{\text{BER}}$, $\overline{\text{FSC}}$ and $\underline{\text{CSR}}$ are classifier-independent. In other words, here we emphasize that the paper is mainly concerned with *problems*, rather than *classifiers* or *algorithms*. More precisely, one main target of this paper is to establish the *universal* relationship between the conditional entropy and one of the three optimum performance measures: $\underline{\text{BER}}$, $\overline{\text{FSC}}$ and $\underline{\text{CSR}}$. Here the word "universal" refers to that our results hold for *any* classification task instead of a particular one. To make this point even clearer, a formal expression unifying the main results of this paper will be highlighted at the end of Section 3, after we have set forth the necessary notions.

3. That being said, we should point out that *empirical* analysis and comparison of different performance criteria does exist in the literature. See Caruana and Niculescu-Mizil (2004) for example.

Symbol	Meaning
$\mathcal{X}$	space of feature vectors (or objects)
x, y; x, y	feature vector of an object and its *true* class label, the sans-serif font is used when they are treated as random variables
$\hat{y}(\cdot)$; $\hat{y}(x)$, $\hat{y}(\mathsf{x})$	classifiers, or the *predicted* class label for a given object
$\mathcal{X}_0$, $\mathcal{X}_1$	decision region corresponding to class 0 and class 1, see Equation (1)
μ	marginal distribution of the feature vector x, see Equation (2)
$\eta(x)$, $\eta(\mathsf{x})$	posterior probability of class 1 given the object x, see Equation (2); the symbol $\eta(\mathsf{x})$ is used when it is seen as a random variable
(μ,η)	the pair (μ,η) is called a (*classification*) *task*, see page 1038
$\tilde{t}$ for $t \in [0,1]$	shorthand of $1-t$, for example, $\tilde{\eta}(x) = \Pr\{\mathsf{y}=0 \mid \mathsf{x}=x\}$ (cf. Equation (2))
t^+ for $t \in \mathbb{R}$	shorthand of $\max\{0,t\}$, used in Equation (47) and thereafter
π, $\tilde{\pi}$	prior probability of class 1 and 0, see Equations (4) and (5)
TP, FP, TN, FN	*proportion* of true positive, false positive, true negative, false negative; see also Table 2
c_0, c_1	the cost of false positive and false negative, see Equation (15)
PREC, REC, SPEC	precision, recall and specificity of classifiers, see page 1039
ERR, BER, CSR, FSC	error rate, balanced error rate, F-score and cost-sensitive risk of a given classifier $\hat{y}(\cdot)$; see Equations (11), (13), (15) and (17)
$\underline{\text{ERR}}$, $\underline{\text{BER}}$, $\underline{\text{CSR}}$, $\overline{\text{FSC}}$	the optimum value of ERR, BER, CSR or FSC in a given task
$h_{\text{bin}}(\eta)$, $\eta \in [0,1]$	binary entropy function, see Equation (19) for its definition

Table 1: List of symbols consistently used in the paper and their meaning

The rest of the paper is organized as follows. Section 2 explains some terminologies and notations to be used in this paper; these include the asymptotic expressions of (balanced) error rate (Section 2.2), cost-sensitive risk (Section 2.3) and F-score (Section 2.4). In Section 3, after briefly introducing Fano's and Hellman's inequalities, we present a novel geometric derivation of the two for the case where the conditional entropy is defined by a concave function. We then derive the analytical expression of the minimum cost-sensitive risk, as well as its tight lower and upper bounds in Section 4. The expression of minimum balanced error rate and its lower/upper bounds are given in Section 5. While Section 6 is devoted to computing the maximum F-score, in Section 7 we examine the relationship between the maximum F-score and conditional entropy. In Section 8, we show that *minimizing conditional entropy does not necessarily maximize the F-score or minimize the cost-sensitive risk*. Consequently, standard mutual information is *not* a proper criterion for learning if the final target is to minimize the cost-sensitive risk or maximize the F-score of the subsequent classification process. A proper information measure for cost-sensitive risk, called *cost-sensitive conditional entropy*, is proposed in Section 8.2. Finally, Section 9 concludes the paper with a summary of the main contributions and some possible extensions of this work.

2. Background

In this section we introduce the necessary background and establish the appropriate formal notions to frame the contributions of the paper.

2.1 Classification Tasks and Binary Classifiers

In this paper, we denote by $\mathcal{X}$ the space of feature vectors; and identify each object with its feature vector $x \in \mathcal{X}$. In the binary classification problem, each object x is assumed to belong to one of two classes which are labeled as $y = 0$ (negative) and $y = 1$ (positive), respectively. A classifier can then be described as a binary-valued function, $\hat{y} : \mathcal{X} \to \{0,1\}$, that maps each object $x \in \mathcal{X}$ to its predicted class label $\hat{y}(x)$.[4] Each such classifier $\hat{y}(\cdot)$ induces naturally a partition of the feature space $\mathcal{X}$ into two *decision regions*, $\mathcal{X}_0$ and $\mathcal{X}_1$, as defined respectively by

$$\mathcal{X}_0 = \{x \in \mathcal{X} \mid \hat{y}(x) = 0\}, \qquad \mathcal{X}_1 = \{x \in \mathcal{X} \mid \hat{y}(x) = 1\}. \tag{1}$$

By definition, it is obvious that $\mathcal{X}_0 \cup \mathcal{X}_1 = \mathcal{X}$ and $\mathcal{X}_0 \cap \mathcal{X}_1 = \varnothing$ for any classifier. Conversely, any pair $(\mathcal{X}_0, \mathcal{X}_1)$ satisfying the two conditions defines a binary classifier $\hat{y}(x)$ that takes the value 0 for $x \in \mathcal{X}_0$ and 1 for $x \in \mathcal{X}_1$. In this paper we shall use the two representations of classifiers interchangeably.

In the traditional probabilistic framework, both the feature vector and the class label are seen as random variables. For the sake of clarity, we shall use the sans-serif font for random variables; so $\mathsf{x} \in \mathcal{X}$ represents the feature vector of an object and $\mathsf{y} \in \{0,1\}$ the corresponding class label. To specify the joint distribution of x and y, we denote by μ the (marginal) distribution of x and by $\eta(x) \in [0,1]$ the conditional probability of class 1 given that $\mathsf{x} = x$—the two symbols are borrowed from Devroye et al. (1996, Chapter 2). Formally, for any measurable subset A of $\mathcal{X}$ and any feature vector $x \in \mathcal{X}$, we write

$$\mu(A) := \Pr\{\mathsf{x} \in A\}, \qquad \eta(x) := \Pr\{\mathsf{y} = 1 \mid \mathsf{x} = x\}. \tag{2}$$

Furthermore, for any $t \in [0,1]$, we define $\tilde{t} := 1 - t$. Then $\tilde{\eta}(x) = \Pr\{\mathsf{y} = 0 \mid \mathsf{x} = x\}$ for any $x \in \mathcal{X}$; and the joint distribution of (x, y) can be written as

$$\Pr\{\mathsf{x} \in A, \mathsf{y} = 1\} = \int_A \eta(x) \mathrm{d}\mu, \qquad \Pr\{\mathsf{x} \in A, \mathsf{y} = 0\} = \int_A \tilde{\eta}(x) \mathrm{d}\mu. \tag{3}$$

We shall call (μ, η) a *classification task*, or simply a *task*, as it completely describes the problem in the sense that other quantities can all be computed from the pair. For instance, putting $A = \mathcal{X}$ in the two equations of Equation (3), we get the (marginal) probability of the two classes, which will be denoted as π and $\tilde{\pi}$, respectively:

$$\pi := \Pr\{\mathsf{y} = 1\} = \int_{\mathcal{X}} \eta(x) \mathrm{d}\mu, \tag{4}$$

$$\tilde{\pi} = \Pr\{\mathsf{y} = 0\} = \int_{\mathcal{X}} \tilde{\eta}(x) \mathrm{d}\mu. \tag{5}$$

4. Such classifiers are sometimes called *deterministic* in the literature; the other type being *probabilisitc*, which produce a vector of estimated class probabilities instead of a class label for each given object (Garg and Roth, 2001). A more general variant of the latter is a *discriminant function*, which outputs vectors of continuous scores (often bearing no probabilistic interpretations). See Steinwart (2007) and Tewari and Bartlett (2007) for instance. In this paper we consider only deterministic classifiers.

2.2 Error Rate and Balanced Error Rate of a Classifier

The *error rate* of a classifier is the proportion of misclassified examples in a test data set; and the *balanced error rate* is the arithmetic mean of the misclassification rate in each class. So the value of (balanced) error rate depends not only on the classifier, but also on the test data set selected. To remove finite sample effects, we consider a data set of infinite size and hence the *asymptotic* expressions of error rate and balanced error rate. In particular, for the two-class problem, these can be defined based on the notions of *true positive*, *true negative*, *false positive* and *false negative*.

Let $\hat{y}(\cdot)$ be a classifier applied to the task (μ, η); and $\{(x_i, y_i)\}_{i=1}^n$ a set of test examples independently drawn from the distribution (3). According to the value of true class labels y_i and their predictions $\hat{y}_i = \hat{y}(x_i)$, $i = 1, \ldots, n$, the n examples fall into four categories, as shown in Table 2. Denote by TP, FP, FN and TN the *proportion*[5] of examples in the four types, then, as these are also the frequency of the respective events, when $n \to \infty$ they tend to

$$\text{TP} \to \Pr\{\hat{y}(\mathsf{x}) = 1, \mathsf{y} = 1\} = \Pr\{\mathsf{x} \in \mathcal{X}_1, \mathsf{y} = 1\} = \textstyle\int_{\mathcal{X}_1} \eta(x) \mathrm{d}\mu, \tag{6}$$

$$\text{FP} \to \Pr\{\hat{y}(\mathsf{x}) = 1, \mathsf{y} = 0\} = \Pr\{\mathsf{x} \in \mathcal{X}_1, \mathsf{y} = 0\} = \textstyle\int_{\mathcal{X}_1} \tilde{\eta}(x) \mathrm{d}\mu, \tag{7}$$

$$\text{FN} \to \Pr\{\hat{y}(\mathsf{x}) = 0, \mathsf{y} = 1\} = \Pr\{\mathsf{x} \in \mathcal{X}_0, \mathsf{y} = 1\} = \textstyle\int_{\mathcal{X}_0} \eta(x) \mathrm{d}\mu, \tag{8}$$

$$\text{TN} \to \Pr\{\hat{y}(\mathsf{x}) = 0, \mathsf{y} = 0\} = \Pr\{\mathsf{x} \in \mathcal{X}_0, \mathsf{y} = 0\} = \textstyle\int_{\mathcal{X}_0} \tilde{\eta}(x) \mathrm{d}\mu, \tag{9}$$

respectively, where the subsets $\mathcal{X}_0$ and $\mathcal{X}_1$ are defined by Equation (1); and the last equality in each equation follows from Equation (3).

	$y = 1$	$y = 0$
$\hat{y} = 1$	true positive (TP); cost: c_{11}	false positive (FP); cost: c_{10}
$\hat{y} = 0$	false negative (FN); cost: c_{01}	true negative (TN); cost: c_{00}

Table 2: Confusion matrix for two possible outcomes and the associated cost matrix

We now define some commonly known performance criteria of binary classifiers for later use. As shown in Nguyen et al. (2009), these can all be written as functions of the above four quantities.

- The *error rate* of a classifier is denoted as ERR in this paper, which is the proportion of misclassified objects, that is, $\text{ERR} := \Pr\{\hat{y}(\mathsf{x}) \neq \mathsf{y}\} = \text{FN} + \text{FP}$.
- The *precision*, PREC, is the proportion of predicted positives ($\hat{y} = 1$) which are actual positive ($\hat{y} = y = 1$), that is, $\text{PREC} := \Pr\{\mathsf{y} = 1 \mid \hat{y}(\mathsf{x}) = 1\} = \text{TP}/(\text{TP} + \text{FP})$.
- The *recall*, denoted REC, is the proportion of actual positives ($y = 1$) which are predicted positive ($y = \hat{y} = 1$), that is, $\text{REC} := \Pr\{\hat{y}(\mathsf{x}) = 1 \mid \mathsf{y} = 1\} = \text{TP}/(\text{TP} + \text{FN})$.
- Finally, the *balanced error rate* is defined as the arithmetic mean of the error rate within the two classes 0 and 1, that is,

$$\begin{aligned} \text{BER} :=& \tfrac{1}{2} \Pr\{\hat{y}(\mathsf{x}) = 1 \mid \mathsf{y} = 0\} + \tfrac{1}{2} \Pr\{\hat{y}(\mathsf{x}) = 0 \mid \mathsf{y} = 1\} \\ =& \tfrac{1}{2} \left\{ \text{FP}/(\text{TN} + \text{FP}) + \text{FN}/(\text{TP} + \text{FN}) \right\}. \end{aligned} \tag{10}$$

5. Typically, the four quantities refer to the *number* of examples; by "rescaling" them to the *proportion* we are able to discuss the case where the test set contains infinitely many examples, that is, $n \to \infty$.

We now derive the analytical expressions of error rate and balanced error rate. By the asymptotic expressions of FP and FN, Equations (7) and (8), we immediately obtain

$$\text{ERR} = \text{FN} + \text{FP} = \int_{\mathcal{X}_0} \eta(x)\mathrm{d}\mu + \int_{\mathcal{X}_1} \tilde{\eta}(x)\mathrm{d}\mu. \tag{11}$$

Furthermore, by Equations (6)–(9) and the facts that $\mathcal{X}_1 \cap \mathcal{X}_0 = \varnothing$ and $\mathcal{X}_1 \cup \mathcal{X}_0 = \mathcal{X}$, we know

$$\text{TP} + \text{FN} = \int_{\mathcal{X}} \eta(x)\mathrm{d}\mu = \pi, \qquad \text{TN} + \text{FP} = \int_{\mathcal{X}} \tilde{\eta}(x)\mathrm{d}\mu = \tilde{\pi}. \tag{12}$$

It then follows that

$$\frac{\text{FN}}{\text{TP}+\text{FN}} = \pi^{-1} \cdot \int_{\mathcal{X}_0} \eta(x)\mathrm{d}\mu, \qquad \frac{\text{FP}}{\text{TN}+\text{FP}} = \tilde{\pi}^{-1} \cdot \int_{\mathcal{X}_1} \tilde{\eta}(x)\mathrm{d}\mu.$$

Therefore, by Equation (10),

$$\text{BER} = \tfrac{1}{2}\left(\pi^{-1}\int_{\mathcal{X}_0} \eta(x)\mathrm{d}\mu + \tilde{\pi}^{-1}\int_{\mathcal{X}_1} \tilde{\eta}(x)\mathrm{d}\mu\right). \tag{13}$$

2.3 Cost-Sensitive Risk

According to Table 2, when an object gets misclassified, it can be either a false positive or a false negative. In the criterion of error rate, the two types of errors are treated equally. In some applications, however, the two kinds of errors may have significantly different consequences. In medical testing, for instance, a false negative (i.e., a mistaken diagnosis that a disease is absent, when it is actually present) is typically more serious than a false positive.

One common way to capture the different effects of false positive and false negative is to assign a (different) *cost* to each of the four outcomes in Table 2. Following the convention of Elkan (2001), we denote by $c_{\hat{y}y}$ the cost of classifying an object to the class $\hat{y}$, when it is actually from the class y. For the two-class problem, this gives rise to a 2×2 matrix called the *cost matrix*, which is presented also in Table 2. The expected cost of a given classifier $\hat{y}(\cdot)$ is called the *cost-sensitive risk* and denoted CSR in the paper—here the modifier "cost-sensitive" is borrowed from Elkan (2001). From Table 2, we see that

$$\text{CSR} = c_{11} \cdot \text{TP} + c_{10} \cdot \text{FP} + c_{01} \cdot \text{FN} + c_{00} \cdot \text{TN}. \tag{14}$$

As has been pointed out by Elkan (2001), for a "reasonable" cost matrix, the cost of labeling an example incorrectly should always be greater than the cost of labeling it correctly. In our notation, this is equivalent to requiring that $c_{10} > c_{00}$ and $c_{01} > c_{11}$. In this paper, we further assume that $c_{11} = c_{00} = 0$;[6] and, to simplify our notations, write $c_0 = c_{10}$ and $c_1 = c_{01}$—that is, the first subscript (which is $\hat{y}$) is dropped; so c_y ($y = 0, 1$) is the cost incurred when an object in the class y is misclassified. Using these notations and Equations (7), (8), we can rewrite Equation (14) as

$$\text{CSR} = c_{01} \cdot \text{FN} + c_{10} \cdot \text{FP} = c_1 \cdot \int_{\mathcal{X}_0} \eta(x)\mathrm{d}\mu + c_0 \cdot \int_{\mathcal{X}_1} \tilde{\eta}(x)\mathrm{d}\mu. \tag{15}$$

Obviously, the above expression degenerates into Equation (11) when $c_1 = c_0 = 1$. This confirms that the error rate ERR is in fact a special case of the family of cost-sensitive risks.

6. This condition can actually be weakened to $c_{11} = c_{00}$; in other words, the cost of labeling an object correctly is a constant, regardless of the true class of that object. In this case, we have $\text{CSR} = c_{00} + (c_{10} - c_{00}) \cdot \text{FP} + (c_{01} - c_{00}) \cdot \text{FN}$; so by subtracting the constant c_{00} from CSR, we obtain essentially the same expression as in Equation (15).

The relationship between BER and CSR is little more subtle. At first glance one may think of BER also as a special case of CSR, since we can get Equation (13) by putting

$$c_1 = \tfrac{1}{2}\pi^{-1}, \qquad c_0 = \tfrac{1}{2}\tilde{\pi}^{-1} \tag{16}$$

in Equation (15). But a closer look at the expressions of BER and CSR reveals that they are both functionals of the task (μ, η) and the classifier (X_0, X_1) under consideration. Moreover, the value of CSR depends on the two costs c_0 and c_1, whereas BER does not—the two coefficients in Equation (16) are computed from (μ, η). Hence the two quantities should be written, in a more formal way, as $\mathrm{BER}(\mu, \eta, \hat{y})$ and $\mathrm{CSR}(\mu, \eta, \hat{y}; c_0, c_1)$, respectively. It is now clear that in general we cannot treat BER as a special CSR, because there is no uniform setting of c_0 and c_1 such that $\mathrm{BER}(\mu, \eta, \hat{y}) = \mathrm{CSR}(\mu, \eta, \hat{y}; c_0, c_1)$. On the other hand, most machine learning papers are about learning algorithms, with the underlying distribution (μ, η) assumed to be fixed. In that case, or, more generally, as far as only the tasks (μ, η) with fixed priors π and $\tilde{\pi}$ are concerned, BER can be regarded as the cost-sensitive risk as defined by Equations (15) and (16). We will discuss this problem further in Section 5 when we derive bounds on the minimum BER.

2.4 Information Retrieval and F-Score

Manning et al. (2008, p. 1) defines information retrieval as:

> Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

As an illustrative example, let us consider a typical document retrieval system which accepts a query from the user and returns a subset of "matched" documents retrieved from a huge collection. To evaluate the performance of the system, we assume that each document is known to be either relevant or non-relevant to a particular query. This has formulated the process as a two-class problem in which the positive class consists of those relevant documents; and the negative class corresponds to the set of irrelevant ones. Accordingly, the retrieval system acts as a classifier: the retrieved documents are (seen as) predicted positive. Therefore, we can rewrite, for example,

$$\text{precision as:} \qquad \mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|},$$

$$\text{recall as:} \qquad \mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}.$$

From the above two expressions, we see that precision can be seen as the probability that a retrieved document is truly relevant to the query. Therefore, a high value of precision can be obtained by only returning those documents that are relevant with high confidence. In this way, however, we probably will miss lots of relevant documents. Similarly, the recall can be viewed as the probability that a relevant document is retrieved for the query. So it is trivial to achieve recall of 100% by returning all documents in response to any query. In conclusion, neither precision nor recall alone is enough to serve as a performance measure of information retrieval systems; and we need to take the two into account simultaneously.

A well-known criterion in the community of information retrieval is *F-score*, defined as the harmonic mean of precision and recall,

$$\text{FSC} := \frac{2 \times \text{PREC} \times \text{REC}}{\text{PREC} + \text{REC}} = \frac{2 \times \text{TP}}{(\text{TP} + \text{FN}) + (\text{TP} + \text{FP})} = \frac{2 \cdot \int_{\mathcal{X}_1} \eta(x) \mathrm{d}\mu}{\pi + \mu(\mathcal{X}_1)} . \tag{17}$$

In the above computation, we have used the first equality of Equation (12) and the identity

$$\text{TP} + \text{FP} = \int_{\mathcal{X}_1} \eta(x) \mathrm{d}\mu + \int_{\mathcal{X}_1} \tilde{\eta}(x) \mathrm{d}\mu = \int_{\mathcal{X}_1} 1 \, \mathrm{d}\mu = \mu(\mathcal{X}_1) .$$

F-score is also known as F_1 measure. It is a member of a broader family of performance measures called F_β, where β varies the emphasis on precision versus recall. In this paper we shall focus on the case of $\beta = 1$, and consistently use the term "F-score".

3. Extending Fano's and Hellman's Bounds for the Two-Class Problem

For a given classification problem, the minimum achievable error rate by any classifier is called its *Bayes error rate*, and denoted as $\underline{\text{ERR}}$ in this paper. In the introduction section, we have already mentioned the main results in the literature that are related to our work. They are all about bounding the quantity $\underline{\text{ERR}}$ by means of different conditional entropies, to which a unifying introduction will be given shortly. Although these known bounds hold for the multi-class problem in general, here we shall review only the binary case, leaving a brief introduction to the multi-class case to Appendix A. More precisely, in this section we will present a novel geometric derivation of Fano's and Hellman's inequalities for the two-class case and extend it to a broad family of conditional entropies. We do this because the same technique will be used throughout the paper to derive bounds on other optimal performance criteria.

For the two-class problem, it is well known that the classifier which predicts all objects x with the posterior $\eta(x) = \Pr\{\mathsf{y} = 1 \mid \mathsf{x} = x\} > 0.5$ as positive (and others as negative) minimizes the error rate; and the minimum error rate is[7]

$$\underline{\text{ERR}} = \int_{\mathcal{X}} \min\{\eta(x), \tilde{\eta}(x)\} \mathrm{d}\mu . \tag{18}$$

It is also well known that for a binary random variable with the distribution $(\eta, 1-\eta) = (\eta, \tilde{\eta})$, its Shannon entropy is defined by the *binary entropy function*

$$h_{\text{bin}}(\eta) := -\eta \cdot \log \eta - \tilde{\eta} \cdot \log \tilde{\eta} , \qquad \eta \in [0, 1] . \tag{19}$$

For binary classification, the value of η depends on the input object $x \in \mathcal{X}$, as given by Equation (2). The expectation of the above function with respect to the object distribution, $\mathsf{x} \sim \mu$, is the *Shannon conditional entropy* (of the class y given the object x):

$$H_{\text{s}}(\mathsf{y}|\mathsf{x}) := \mathbb{E}_{\mathsf{x} \sim \mu}[h_{\text{bin}}(\eta(\mathsf{x}))] = \int_{\mathcal{X}} h_{\text{bin}}(\eta(x)) \mathrm{d}\mu , \tag{20}$$

where the subscript $_{\text{s}}$ stands for "Shannon".

Fano's inequality connects the Shannon conditional entropy, $H_{\text{s}}(\mathsf{y}|\mathsf{x})$, to the Bayes error rate, $\underline{\text{ERR}}$, by $h_{\text{bin}}(\underline{\text{ERR}}) \geqslant H_{\text{s}}(\mathsf{y}|\mathsf{x})$. As $\underline{\text{ERR}} \leqslant 0.5$ and the function $h_{\text{bin}}(\eta)$ is monotonically increasing

7. These facts will become clear after we have derived the expression of the minimum cost-sensitive risk in Section 4.

for $0 \leqslant \eta \leqslant 0.5$, this actually provides a lower bound on $\underline{\mathrm{ERR}}$ in terms of $H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})$. The upper bound is defined by Hellman's inequality, which can be written in our notation as $\underline{\mathrm{ERR}} \leqslant \frac{1}{2}H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})$. The two bounds had been graphically shown in Figure 1.

In the literature, the two inequalities were proven using different methods; see, for example, Cover and Thomas (2006, Section 2.10) and Hellman and Raviv (1970). Here we propose a novel geometric proof that they can be obtained simultaneously, based upon an "obvious" fact which we state as a theorem (because of its fundamental importance in the paper).

Theorem 1 *The expectation of a random vector (assume it exists) in the Euclidean space $\mathbb{R}^m$ lies in the convex hull of the range of that random vector.*

This proposition, probably well known and intuitively clear—since the expectation of a random vector is essentially the convex combination of the vectors in its range, is in fact nontrivial. To the best of our knowledge (and to our surprise), there is no proof to Theorem 1 in the literature (we thought it should be in some textbooks on probability theory, but we cannot find one). We hence provide one of ourselves in Appendix B.4.

Theorem 1 gives rise to a *general geometric strategy* for deriving/proving inequalities like Fano's, as outlined in Scheme 3, where the derivation of the lower and upper bounds on $H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})$ has been used as a demonstration. One should have no difficulty to see that this geometric method can be extended, straightforwardly, to the family of concave and symmetric functions $h(\eta)$, instead of the particular function $h_{\mathrm{bin}}(\eta)$. In fact, we even can go one step further, by dropping the requirement that $h(\eta)$ be symmetric. We hence introduce the following general definition of conditional entropy.

Definition 2 *Let $h(\eta)$ with $\eta \in [0,1]$ be a concave function satisfying*[8] *$h(0) = h(1) = 0$. The conditional entropy of a given classification task (μ, η) is defined as*

$$H(\mathsf{y}|\mathsf{x}) := \mathbb{E}_{\mathsf{x}\sim\mu}[h(\eta(\mathsf{x}))] = \textstyle\int_{\mathcal{X}} h(\eta(x))\mathrm{d}\mu. \tag{21}$$

This definition of conditional entropy is general enough to include most of entropies in the literature. For example, the Shannon entropy is obtained by setting $h(\eta) = h_{\mathrm{bin}}(\eta)$; and letting $h(\eta) = -\log(\eta^2 + \tilde{\eta}^2)$, we get Rényi's quadratic entropy (Principe and Xu, 1999). Another example is *weighted entropy* (Guiasu, 1971), which for the binary case is defined by the function $h(\eta) = -w_1\eta \cdot \log\eta - w_0\tilde{\eta} \cdot \log\tilde{\eta}$, where $w_0, w_1 > 0$ are two weights. Note that this function is asymmetric when $w_0 \neq w_1$.

Scheme 3 A general geometric strategy for deriving tight lower and upper bounds on one performance/information measure in terms of another measure

---

Assume we want to derive the tight lower/upper bounds on $H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})$ in terms of $\underline{\mathrm{ERR}}$:

1. The first step is to find a random vector with expectation $[\underline{\mathrm{ERR}}, H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})]$ (the vector comprising the concerned quantities). In fact, by Equations (20) and (18) we easily see that $[\underline{\mathrm{ERR}}, H_{\mathrm{s}}(\mathsf{y}|\mathsf{x})] = \mathbb{E}_{\mathsf{x}\sim\mu}[e(\eta(\mathsf{x})), h_{\mathrm{bin}}(\eta(\mathsf{x}))]$, where the function $e(\cdot)$ is defined by

$$e(\eta) := \min\{\eta, \tilde{\eta}\}, \qquad \eta \in [0,1]. \tag{22}$$

   So the random vector $[e(\eta(\mathsf{x})), h_{\mathrm{bin}}(\eta(\mathsf{x}))]$ is what we want.

8. As subtracting a linear function (of η) from a concave function still gives a concave function, imposing the condition $h(0) = h(1) = 0$ on the definition will result in no loss of generality.

2. Next, we need to find the range of $[e(\eta(\mathsf{x})), h_{\text{bin}}(\eta(\mathsf{x}))]$, the random vector obtained in Step 1. Apparently, this is the curve $\ell := \{[e(\eta), h_{\text{bin}}(\eta)] \mid \eta \in [0,1]\}$.[9] But $h_{\text{bin}}(\eta)$ is a symmetric function, that is, $h_{\text{bin}}(\eta) = h_{\text{bin}}(\tilde{\eta})$, by the definition of $e(\eta)$ we know the curve ℓ is in fact the left half of $h_{\text{bin}}(\eta)$, as depicted in Figure 2-a.

3. We then construct the convex hull of the curve ℓ, which for this example is the bow shape OABCO bounded by the curve OCB (i.e., $h = h(e)$) from above and by the line segment OAB (i.e., $h = 2e$) from below—see Appendix B.2 and B.3 for a rigorous discussion on the convex hull of a given curve or subset.

4. Now Theorem 1 shows the point $[\underline{\text{ERR}}, H_{\text{s}}(\mathsf{y}|\mathsf{x})]$ is in the area OABCO. We can thus *directly "read"*, for any given value of $\underline{\text{ERR}}$, the lower and upper bounds of $H_{\text{s}}(\mathsf{y}|\mathsf{x})$ from the convex hull of ℓ, in an obvious way and *simultaneously*. The correctness of the bounds so obtained is *guaranteed* by Theorem 1. For this example, the two bounds are $2e|_{e=\underline{\text{ERR}}} \leqslant H_{\text{s}}(\mathsf{y}|\mathsf{x}) \leqslant h(e)|_{e=\underline{\text{ERR}}}$, that is, $2\underline{\text{ERR}} \leqslant H_{\text{s}}(\mathsf{y}|\mathsf{x}) \leqslant h(\underline{\text{ERR}})$, which are exactly Fano's and Hellman's results.

5. Last but not least, it is easy to show that each point in the convex hull of ℓ can be attained by some classification task (see the proof to Theorem 5). Thus, the bounds obtained as above are tight.

To illustrate the generality of the proposed geometric scheme, we apply it to a general concave function $h(\eta)$ which might be *asymmetric*. For this, only the second and third steps in Scheme 3 need to be adapted slightly, as follows.

2. For an asymmetric function $h(\eta)$, the curve $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$ consists of two parts which can be expressed as $h = h(e)$—as $e(\eta) = \eta$ for $\eta \in [0, 0.5]$, and $h = h(1-e)$—as $e(\eta) = \tilde{\eta} = 1 - \eta$ for $\eta \in [0.5, 1]$. Graphically, this means that ℓ is the left half of the curve $h = h(\eta)$ plus its right half flipped along the vertical line $\eta = 0.5$, as is shown in Figure 2-b.

3. The convex hull of ℓ, denoted $\text{co}\,\ell$, can then be expressed as[10] (recall that $\tilde{e} = 1 - e$)

$$\text{co}\,\ell = \{(e,h) \mid e \in [0, 0.5],\ [\min\{h(e), h(\tilde{e})\}]_{\smile} \leqslant h \leqslant [\max\{h(e), h(\tilde{e})\}]^{\frown}\}, \qquad (23)$$

   where, for any real-valued function $f(\cdot)$ defined on a convex set, $f_{\smile}$ denotes the *convex hull* of f, that is, the greatest convex function with the same domain as f that does not exceed f; and $f^{\frown}$ is the *concave hull* of f, the smallest concave function that is larger than or equal to f at each point in the domain of f.

We are now ready to "read" the lower and upper bounds of $H(\mathsf{y}|\mathsf{x})$ from the set $\text{co}\,\ell$, as Theorem 1 has already told us that $[\underline{\text{ERR}}, H(\mathsf{y}|\mathsf{x})] \in \text{co}\,\ell$. But before that, we would first simplify the two bounds $[\ldots]_{\smile}$ and $[\ldots]^{\frown}$ in Equation (23), to get a cleaner result. The function $h(\cdot)$ is concave, so is

9. Strictly speaking, this should be $\ell := \{[e(\eta(x)), h_{\text{bin}}(\eta(x))] \mid x \in \mathcal{X}\}$. But as we are investigating the universal relationship, the "wildest" case where the range of $\eta(x)$ is $[0,1]$ should be considered.

10. See Appendix B.3 for a proof for this.

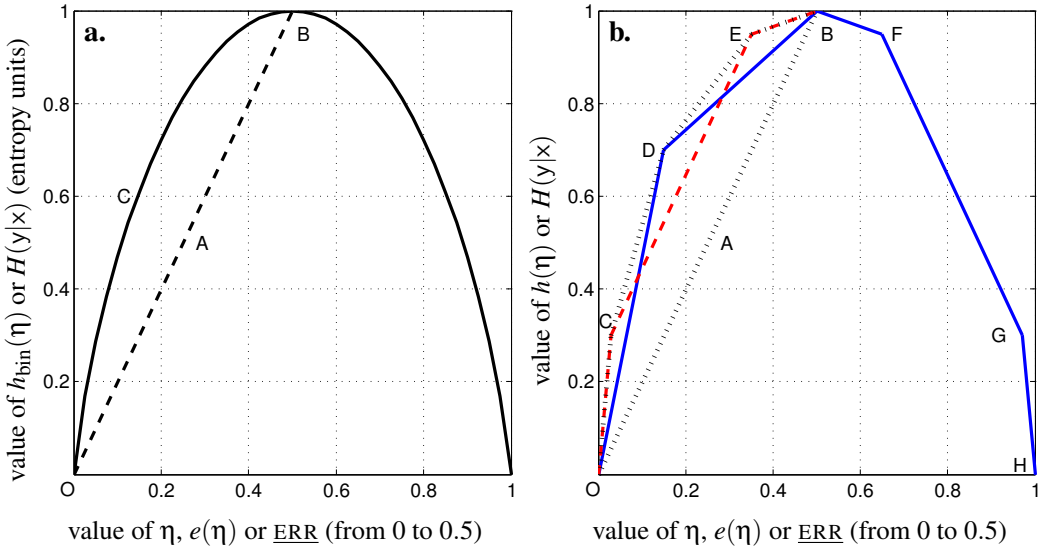


Figure 2: **a.** The graph of the function $h_{\text{bin}}(\eta)$ (solid line), with η as the x-axis and h_{bin} as the y-axis. As this is a symmetric function, its left part BCO represents the curve $\ell = \{[e(\eta), h_{\text{bin}}(\eta)] \mid \eta \in [0,1]\}$—now the x-axis stands for $e(\eta)$. The convex hull of ℓ is hence the region bounded by the solid curve (left half) and the dashed line OAB—now we have ERR for the x-axis and $H(y|x)$ for the y-axis. By flipping this bow shape along the diagonal line through the points $[0,0]$ and $[1,1]$, we get exactly Figure 1, that is, Fano's and Hellman's bounds.

b. The graph of an asymmetric function $h(\eta)$—the broken line ODBFGH, and the curve $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$, which consists of the broken lines ODB and OCEB (obtained from HGFB through a right-to-left flipping). The convex hull of ℓ is then the polygon OBEDCO (dotted line). As in the symmetric case, here the upper/lower bounds on ERR can be obtained by flipping this polygon along the diagonal line.

$\min\{h(e), h(1-e)\}$ as a function of $e \in [0, 0.5]$, as it is the minimum of two concave functions. But the convex hull of a concave function is an affine function through its two endpoints. Therefore,

$$[\min\{h(e), h(1-e)\}]_{\smile} = 2 \cdot h(0.5) \cdot e\,.$$

Here we have used the assumption that $h(0) = h(1) = 0$. Moreover, if $h(\cdot)$ is symmetric, that is, $h(e) = h(\tilde{e})$, then $\max\{h(e), h(\tilde{e})\} = h(e)$ is a concave function; and its concave hull is itself: $[\max\{h(e), h(\tilde{e})\}]^{\frown} = h(e)$.

Putting the above discussion together, we obtain the following theorem that extends Fano's and Hellman's results to that using an arbitrary concave function $h : [0,1] \to \mathbb{R}$ in the definition of the conditional entropy.

Theorem 4 (extension of Fano's and Hellman's inequalities) *Let $h : [0,1] \to \mathbb{R}$ be a concave function with $h(0) = h(1) = 0$. Then for any classification task (μ, η) we have*

$$2 \cdot h(0.5) \cdot \text{ERR} \leqslant H(y|x) \leqslant [\max\{h(\text{ERR}), h(1 - \text{ERR})\}]^{\frown}\,. \tag{24}$$

In particular, for symmetric functions $h(\cdot)$ the above inequality can be simplified to

$$2 \cdot h(0.5) \cdot \underline{\mathrm{ERR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(\underline{\mathrm{ERR}}) .$$

In next section we shall extend the above theorem further to a relationship between the conditional entropy $H(\mathsf{y}|\mathsf{x})$ and the minimum cost-sensitive risk $\underline{\mathrm{CSR}}$—of which $\underline{\mathrm{ERR}}$ is a special case.

Furthermore, from Theorem 1 and the definition of the convex hull of a set, we can easily see the two bounds of $H(\mathsf{y}|\mathsf{x})$ given by Equation (24) are tight, in the sense that for *any concave function* $h(\cdot)$ and *any given value of* $\underline{\mathrm{ERR}}$, both bounds are reachable by some task (μ, η). In fact, we have an even stronger result, for which a short proof is presented as the "template" for other similar tightness proofs in the paper.

Theorem 5 *For any concave function $h : [0,1] \to \mathbb{R}$ with $h(0) = h(1) = 0$, and any point $[e_0, h_0]$ inside the convex set* $\mathrm{co}\,\ell$ *as given by Equation* (23)*, there exists a task (μ, η) for which it holds that $\underline{\mathrm{ERR}} = e_0$ and $H(\mathsf{y}|\mathsf{x}) = h_0$.*

Proof Since the point $[e_0, h_0]$ lies in the convex hull of $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$, there are n points on the curve ℓ, say $\{[e(\eta_i), h(\eta_i)]\}_{i=1,\ldots,n}$, such that $[e_0, h_0]$ is the convex combination of the n points with the coefficients $\{\beta_i\}_{i=1,\ldots,n}$, that is,

$$e_0 = \textstyle\sum_{i=1}^n \beta_i e(\eta_i) , \qquad h_0 = \sum_{i=1}^n \beta_i h(\eta_i) ,$$

where $\beta_i \geqslant 0$ satisfy $\sum_{i=1}^n \beta_i = 1$. We can thus construct a classification task in which the feature space $\mathcal{X}$ consists exactly of n points, $\{x^{(1)}, \ldots, x^{(n)}\}$, with the probability mass $\mu(x^{(i)})$ and the posterior $\eta(x^{(i)})$ given by

$$\mu(x^{(i)}) = \Pr\{\mathsf{x} = x^{(i)}\} = \beta_i , \qquad \eta(x^{(i)}) = \Pr\{\mathsf{y} = 1 \mid \mathsf{x} = x^{(i)}\} = \eta_i .$$

Clearly, for this task (μ, η) we have $\underline{\mathrm{ERR}} = e_0$ and $H(\mathsf{y}|\mathsf{x}) = h_0$. ■

Corollary 6 *In Theorem 4 (Equation* (24)*), the two bounds on $H(\mathsf{y}|\mathsf{x})$ are tight. That is, given any concave function $h : [0,1] \to \mathbb{R}$ with $h(0) = h(1) = 1$ and any value of $\underline{\mathrm{ERR}} \in [0, 0.5]$, there are two (different) tasks for which the two inequalities in Equation* (24) *become equalities, respectively.*

Proof Apply Theorem 5 to the point $[e_0, h_0] = [\underline{\mathrm{ERR}},\ 2 \cdot h(0.5) \cdot \underline{\mathrm{ERR}}]$ and to the point $[e_0, h_0] = [\underline{\mathrm{ERR}},\ [\max\{h(\underline{\mathrm{ERR}}), h(1 - \underline{\mathrm{ERR}})\}]^{\frown}]$. ■

To summarize, in this fundamental section we proposed a general geometric approach to deriving/proving inequalities that links the conditional entropy of classification tasks with an optimal performance measure, for example, the Bayes error rate $\underline{\mathrm{ERR}}$. By Theorem 1, Theorem 5 and Corollary 6, the inequalities obtained in this way are guaranteed to be *correct* and *sharp*. They are also *universal* in that Theorem 4 holds for any task (μ, η).

Following the discussion at the end of Section 2.3, here we would emphasize again that the two quantities in the inequality, $\underline{\mathrm{ERR}}$ and $H(\mathsf{x}|\mathsf{y})$, are actually functionals of tasks; and should be written respectively as $\underline{\mathrm{ERR}}(\mu, \eta)$ and $H_h(\mu, \eta)$ in a more formal way. Here the subscript $_h$ is used to stress

the role of the function $h(\cdot)$ in the definition of conditional entropy. In accordance, Equation (24) should be written as

$$2 \cdot h(0.5) \cdot \underline{\mathrm{ERR}}(\mu,\eta) \leqslant H_h(\mu,\eta) \leqslant [\max\{h(\underline{\mathrm{ERR}}(\mu,\eta)), h(1-\underline{\mathrm{ERR}}(\mu,\eta))\}]^{\frown};$$

and it holds for *any* concave function $h : [0,1] \to \mathbb{R}$ satisfying $h(0) = h(1) = 0$ and *any* classification task (μ,η), as has been asserted by Theorem 4.

In the next four sections we will derive the similar inequalities for the quantities $\underline{\mathrm{CSR}}$, $\underline{\mathrm{BER}}$ and $\overline{\mathrm{FSC}}$, which have the following uniform form:

$$f(\mathrm{XX}(\mu,\eta)) \leqslant H_h(\mu,\eta) \leqslant g(\mathrm{XX}(\mu,\eta)), \qquad \mathrm{XX} \text{ stands for } \underline{\mathrm{CSR}}, \underline{\mathrm{BER}} \text{ or } \overline{\mathrm{FSC}}, \tag{25}$$

where $f(\cdot)$ is a proper *convex* function and $g(\cdot)$ a proper *concave* function. Like Equation (24), for $\underline{\mathrm{CSR}}$ and $\overline{\mathrm{FSC}}$ the corresponding inequality holds for *any* task (μ,η). The quantity $\underline{\mathrm{BER}}$ is special, for which the two "bounding" functions $f(\cdot)$ and $g(\cdot)$ involve an extra parameter: the positive prior π, which presents also in the expression of $\underline{\mathrm{BER}}$—see Equation (32) in page 1052. Consequently, the result holds only for the tasks (μ,η) with *fixed* class priors, namely, $\Pr\{\mathsf{y}=1\} = \pi$ and $\Pr\{\mathsf{y}=0\} = \tilde{\pi}$. But when π is also seen as a functional of (μ,η), then the inequality (25)—which now links $\underline{\mathrm{BER}}$, H_h and π, becomes universal.

4. Bounds on the Minimum Cost-Sensitive Risk

We now study the relationship between the conditional entropy $H(\mathsf{y}|\mathsf{x})$ and the minimum cost-sensitive risk $\underline{\mathrm{CSR}}$, using the same geometric strategy as given in the preceeding section. To this end, we need first to derive the expression of $\underline{\mathrm{CSR}}$.

We have already derived in Section 2.3 the analytical expression of the cost-sensitive risk for a given classifier, which, as shown in Equation (15), is the sum of two integrals of the functions $c_1\eta(x)$ and $c_0\tilde{\eta}(x)$ over the disjoint subsets $\mathcal{X}_0$ and $\mathcal{X}_1$ of the space $\mathcal{X}$, respectively. To obtain its minimum (over all possible classifiers), we use that both $c_1\eta(x)$ and $c_0\tilde{\eta}(x)$ are larger than or equal to the minimum of the two. It thus follows that

$$\begin{aligned}\mathrm{CSR} &= \textstyle\int_{\mathcal{X}_0} c_1\eta(x)\mathrm{d}\mu + \int_{\mathcal{X}_1} c_0\tilde{\eta}(x)\mathrm{d}\mu \\ &\geqslant \textstyle\int_{\mathcal{X}_0} \min\{c_1\eta(x), c_0\tilde{\eta}(x)\}\mathrm{d}\mu + \int_{\mathcal{X}_1} \min\{c_1\eta(x), c_0\tilde{\eta}(x)\}\mathrm{d}\mu.\end{aligned}$$

But as $\mathcal{X}_0 \cap \mathcal{X}_1 = \varnothing$ and $\mathcal{X}_0 \cup \mathcal{X}_1 = \mathcal{X}$, the above two integrals $\int_{\mathcal{X}_i} \min\{\ldots\}\mathrm{d}\mu$, $i = 0,1$, can be combined into one (over the whole space $\mathcal{X}$), yielding $\mathrm{CSR} \geqslant \int_{\mathcal{X}} \min\{c_1\eta(x), c_0\tilde{\eta}(x)\}\mathrm{d}\mu$. Moreover, this inequality becomes equality when (and only when) the condition:

$$\begin{aligned} c_1\eta(x) &= \min\{c_1\eta(x), c_0\tilde{\eta}(x)\} \quad \text{on } \mathcal{X}_0\,; \text{ and} \\ c_0\tilde{\eta}(x) &= \min\{c_1\eta(x), c_0\tilde{\eta}(x)\} \quad \text{on } \mathcal{X}_1 \end{aligned}$$

is fulfilled. This is equivalent to requiring that $c_1\eta(x) \leqslant c_0\tilde{\eta}(x)$, that is, $\eta(x) \leqslant \frac{c_0}{c_0+c_1}$ for (and only for) all $x \in \mathcal{X}_0$. Therefore, the minimum cost-sensitive risk is

$$\underline{\mathrm{CSR}} = \textstyle\int_{\mathcal{X}} \min\{c_1\eta(x), c_0\tilde{\eta}(x)\}\mathrm{d}\mu; \tag{26}$$

and this minimum is achieved by the classifier $\hat{y}(x) = [\![\eta(x) > \frac{c_0}{c_0+c_1}]\!]$, where $[\![\cdot]\!]$ denotes the indicator function which takes value 1 if the bracketed statement is true and 0 otherwise. This result is well known in Bayesian decision theory; see, for example, Duda et al. (2001, page 26).

Note that the error rate can be seen as a special cost-sensitive risk with $c_0 = c_1 = 1$, so its minimum can be obtained from Equation (26) by putting $c_0 = c_1 = 1$. This gives us exactly the expression Equation (32), and the corresponding optimal classifier is $\hat{y}(x) = [\![\eta(x) > \frac{c_0}{c_0+c_1}]\!] = [\![\eta(x) > 0.5]\!]$, which have been stated in Section 3 as well established in the literature.

We now derive, in terms of $\underline{\text{CSR}}$, the lower and upper bounds on the conditional entropy $H(\mathsf{y}|\mathsf{x})$ as given by Definition 2. Following the geometric lines in Scheme 3, we define the function $e(\eta)$ as (again, this is reduced to Equation (22) for $c_0 = c_1 = 1$)

$$e(\eta) := \min\{c_1\eta, c_0\tilde{\eta}\}, \qquad \eta \in [0,1]. \tag{27}$$

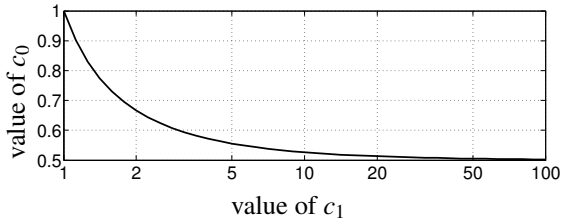
Then, Equations (21) and (26) can rewritten as the mathematical expectations of $h(\eta(\mathsf{x}))$ and $e(\eta(\mathsf{x}))$, respectively:

$$[\underline{\text{CSR}}, H(\mathsf{y}|\mathsf{x})] = \mathbb{E}_{\mathsf{x}\sim\mu}[e(\eta(\mathsf{x})), h(\eta(\mathsf{x}))]. \tag{28}$$

We thus have accomplished the first step in Scheme 3. By Theorem 1, in the *e-h* plane the point $[\underline{\text{CSR}}, H(\mathsf{y}|\mathsf{x})]$ lies in the convex hull of the curve $\ell := \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$. The problem then amounts to finding the convex hull of ℓ which we shall discuss shortly.

By the definition of $e(\eta)$, Equation (27), one easily sees that $e = c_1\eta$ when $\eta \leqslant \frac{c_0}{c_0+c_1}$ and that $e = c_0\tilde{\eta} = c_0 - c_0\eta$ when $\eta \geqslant \frac{c_0}{c_0+c_1}$. It then follows that $0 \leqslant e(\eta) \leqslant \frac{c_0c_1}{c_0+c_1}$, with the minimum value 0 attained at $\eta = 0$ or $\eta = 1$; and the maximum $\frac{c_0c_1}{c_0+c_1}$ obtained at $\eta = \frac{c_0}{c_0+c_1}$. At this point, we find it most convenient to normalize the two costs c_0 and c_1 (by multiplying them by a common factor) to such that $\frac{c_0c_1}{c_0+c_1} = 0.5$, that is, $c_0^{-1} + c_1^{-1} = 2$. Then the range of e is always $[0, 0.5]$.

To simplify the derivation procedure we further assume, without loss of generality, that $c_1 \geqslant c_0$. In Section 5, this assumption will be used to obtain bounds on $\underline{\text{BER}}$ from that on $\underline{\text{CSR}}$, see the proof to Corollary 8. The inequality $c_1 \geqslant c_0$ is equivalent to $c_1^{-1} \leqslant c_0^{-1}$, which together with $c_0^{-1} + c_1^{-1} = 2$ implies that $1 \leqslant c_0^{-1} < 2$ and $0 < c_1^{-1} \leqslant 1$. We thus get $c_0 \in (0.5, 1]$, $c_1 \in [1, \infty)$ and $\frac{c_0}{c_0+c_1} = \frac{1}{2c_1} \leqslant \frac{1}{2}$. Furthermore, from the equality $c_0^{-1} + c_1^{-1} = 2$ we know $c_0 = \frac{c_1}{2c_1-1}$. So the cost matrix is now characterized by a single parameter $c_1 \in [1, \infty)$, as shown in Figure 3.



	$y=1$	$y=0$
$\hat{y}=1$	0	$c_0 = \frac{c_1}{2c_1-1}$
$\hat{y}=0$	c_1	0

Figure 3: The relationship between the values of c_0 and c_1 (left) and the cost-matrix as characterized by the cost c_1 (right).

We now study the curve $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$ (Step 2 in Scheme 3). Based on the above assumptions, we see that when η changes from 0 to $\frac{1}{2c_1}$, $e = c_1\eta$ changes from 0 to 0.5; and when η changes from $\frac{1}{2c_1}$ *further* to 1, $e = c_0 - c_0\eta$ changes from 0.5 *back* to 0, both in a linear manner. Therefore, the curve $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$ consists of two parts, namely $h = h(c_1^{-1}e)$, $e \in [0, 0.5]$ (corresponding to $\eta \in [0, \frac{1}{2c_1}]$) and $h = h(1 - c_0^{-1}e)$, $e \in [0, 0.5]$ (corresponding to $\eta \in [\frac{1}{2c_1}, 1]$). The first part $h = h(c_1^{-1}e)$ is obtained from the graph of $h(\eta)$, $\eta \in [0, \frac{1}{2c_1}]$ by linearly lengthening it from the interval $[0, \frac{1}{2c_1}]$ to that on the interval $[0, 0.5]$. The second part

$h = h(1 - c_0^{-1}e)$ is obtained from the graph of $h(\eta)$, $\eta \in [\frac{1}{2c_1}, 1]$ by first linearly shrinking it from the interval $[\frac{1}{2c_1}, 1]$ to over the interval $[\frac{1}{2}, 1]$, and then flipping the resulting curve along the vertical line at $\eta = 0.5$.

The above dynamical procedure is demonstrated in Figure 4-a for the settings $c_1 = 2.5$, $c_0 = 0.625$ and $h(\eta) = -\eta \cdot \log \eta - (1-\eta)\log(1-\eta)$ (Shannon). In Figure 4-a, we start with the graph of $h = h(\eta)$, the curve OABF. This curve is divided into two parts by the point A whose coordinate is $(\frac{1}{2c_1}, h(\frac{1}{2c_1}))$. To obtain the curve ℓ, we first move horizontally A to the point C which has the coordinate of $(0.5, h(\frac{1}{2c_1}))$. The other points on the curve OABF are moved linearly, with the two endpoints O and F being fixed. This gives us the curve OCDF, whose left part OC represents the function $h = h(c_1^{-1}e)$, $e \in [0, 0.5]$; and its right half CDF is described by the function $h = h(1 - c_0^{-1}(1-e))$, $e \in [0.5, 1]$. Next, we flip the right part CDF along the vertical line at $\eta = 0.5$, yielding the curve OHEC which is the graph of $h = h(1 - c_0^{-1}e)$, $e \in [0, 0.5]$. The curve ℓ is then the union of the curve OHEC and the curve OC, that is, the closed curve OHECO.

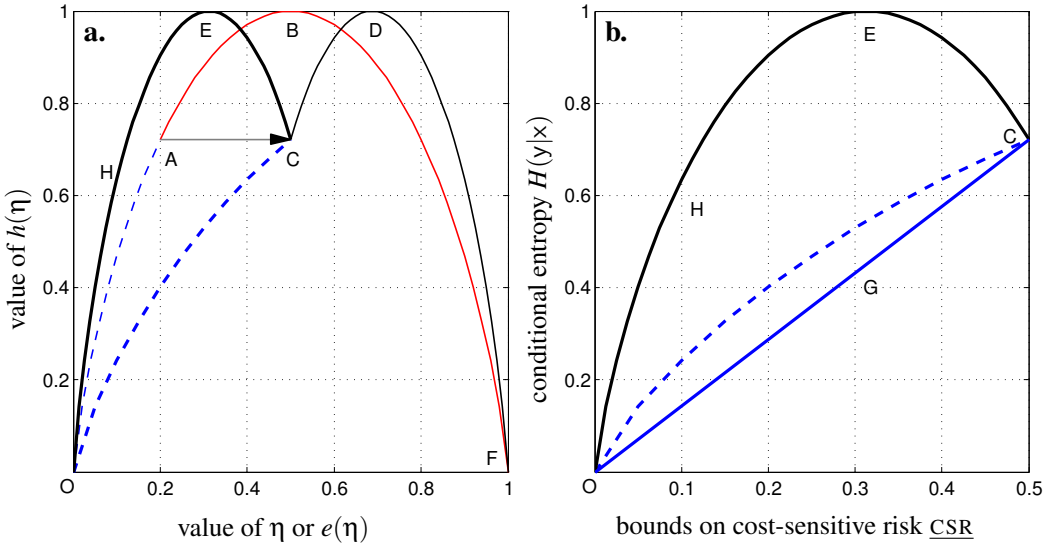


Figure 4: **a.** The procedure for getting the curve $\ell = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}$ (OHECO) from the graph of $h(\eta)$ (the curve OABF), for the settings of $c_0 = 0.625$, $c_1 = 2.5$ and $h(\eta) = -\eta \log \eta - \tilde{\eta} \log \tilde{\eta}$. See the text for more explanation.
b. The tight lower (the line OGC) and upper (the curve OHEG) bounds on $H(y|x)$ as functions of CSR. Stated in the other way, if the conditional entropy $H(y|x)$ is known, then the upper bound of CSR is determined by the curve OGCE; and its lower bound is given by the curve OHE.

The next step is to find the convex hull of ℓ. By Definition 2, $h(\eta)$ is a concave function. So both $h = h(c_1^{-1}e)$ and $h = h(1 - c_0^{-1}e)$ are concave functions (of e). Moreover, for any symmetric function $h(\eta)$, from Figure 4-a we see that the curve OHEC is above the curve OC. Mathematically, this can be expressed as $h(c_1^{-1}e) \leqslant h(1 - c_0^{-1}e)$, which is true for all *symmetric* concave functions

$h(\eta)$.[11] Therefore, the convex hull of the curve OHECO can be obtained by simply connecting the points O and C with a straight line. This is plotted in Figure 4-b, where the *region* OHECGO is the convex hull of the curve OHECO; it also represents the reachable region of the point $[\underline{\mathrm{CSR}}, H(\mathsf{y}|\mathsf{x})]$.

It is now clear that the value of $H(\mathsf{y}|\mathsf{x})$ is lower bounded by the straight line OC, which is the graph of the function $h = 2h(\frac{1}{2c_1})e$, $e \in [0,0.5]$, and upper bounded by the curve OHEC, which is described by the function $h = h(1-c_0^{-1}e) = h(c_0^{-1}e)$, $e \in [0,0.5]$—since $h(\cdot)$ has been assumed to be symmetric here. We thus obtain

$$2 \cdot h(\tfrac{1}{2c_1}) \cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(c_0^{-1} \cdot \underline{\mathrm{CSR}}) \,. \tag{29}$$

By a similar discussion to that above Theorem 4, we can extend Equation (29) to asymmetric functions $h(\cdot)$. In this case it is not necessarily that $h(c_1^{-1}e) \leqslant h(1-c_0^{-1}e)$ for $e \in [0,0.5]$. In Figure 4-b, this means that the dashed curve OC, that is, $h = h(c_1^{-1}e)$, could intersect with the curve OHEC, that is, $h = h(1-c_0^{-1}e)$, at points other than O and C. Consequently, the right hand side of Equation (29) should now be replaced by the concave hull function of the maximum of $h(c_1^{-1}e)$ and $h(1-c_0^{-1}e)$. That is,

$$2 \cdot h(\tfrac{1}{2c_1}) \cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant [\max\{h(c_1^{-1} \cdot \underline{\mathrm{CSR}}),\, h(1-c_0^{-1} \cdot \underline{\mathrm{CSR}})\}]^{\frown} \,.$$

Moreover, analogous to Theorem 5 and Corollary 6, we can prove the above two bounds on $H(\mathsf{y}|\mathsf{x})$ are both tight. These results are summarized in the following theorem.

Theorem 7 (tight bounds on $H(\mathsf{y}|\mathsf{x})$ as functions of $\underline{\mathrm{CSR}}$) *Let $h : [0,1] \to \mathbb{R}$ be a concave function that satisfies $h(0) = h(1) = 0$. Then for any binary classification problem (μ, η) we have*

$$2 \cdot h(\tfrac{1}{2c_1}) \cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant [\max\{h(c_1^{-1} \cdot \underline{\mathrm{CSR}}),\, h(1-c_0^{-1} \cdot \underline{\mathrm{CSR}})\}]^{\frown} , \tag{30}$$

where $H(\mathsf{y}|\mathsf{x})$ is defined as in Definition 2, and $\underline{\mathrm{CSR}}$ given by Equation (26)*. In particular, when $h(\cdot)$ is a symmetric function, it holds that $2 \cdot h(\frac{1}{2c_1}) \cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(c_0^{-1} \cdot \underline{\mathrm{CSR}})$.*

One observes that when $c_0 = c_1 = 1$ the above theorem is reduced to Theorem 4, so it is the extension of Fano's and Hellman's results to the cost-sensitive case.

To simplify the discussion, let us return to the symmetric case. To get the lower/upper bounds on $\underline{\mathrm{CSR}}$ in terms of $H(\mathsf{y}|\mathsf{x})$ from Theorem 7, we notice that the function $h(\eta)$ is symmetric on the interval $[0,1]$ and hence monotonically non-decreasing on $[0,0.5]$. It thus follows from the inequality $H(\mathsf{y}|\mathsf{x}) \leqslant h(c_0^{-1} \cdot \underline{\mathrm{CSR}})$ that

$$h_{[0,0.5]}^{-1}(H(\mathsf{y}|\mathsf{x})) \leqslant c_0^{-1} \cdot \underline{\mathrm{CSR}} \leqslant 1 - h_{[0,0.5]}^{-1}(H(\mathsf{y}|\mathsf{x})) \,,$$

where $h_{[0,0.5]}^{-1}$ denotes the inverse of the function $h(\eta)$ restricted on $[0,0.5]$. This inequality together with $2h(\frac{1}{2c_1}) \cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x})$ implies

$$c_0 \cdot h_{[0,0.5]}^{-1}(H(\mathsf{y}|\mathsf{x})) \leqslant \underline{\mathrm{CSR}} \leqslant \min\left\{ c_0 - c_0 \cdot h_{[0,0.5]}^{-1}(H(\mathsf{y}|\mathsf{x})), \tfrac{1}{2} \cdot [h(\tfrac{1}{2c_1})]^{-1} \cdot H(\mathsf{y}|\mathsf{x}) \right\} . \tag{31}$$

In Figure 4-b, the above lower bound $c_0 \cdot h_{[0,0.5]}^{-1}(H(\mathsf{y}|\mathsf{x}))$ corresponds to the curve OHE; and the upper bound $\min\{\ldots\}$ corresponds to the curve OGCE (see the description of Figure 4-b).

11. Here is a short proof. As $h(\eta)$ is symmetric and concave, it attains its maximum at $\eta = 0.5$; and it is monotonically non-decreasing on the interval $[0,0.5]$ and monotonically non-increasing on $[0.5,1]$. So to prove $h(c_1^{-1}e) \leqslant h(1-c_0^{-1}e)$ it suffices to show that $c_1^{-1}e \leqslant 1-c_0^{-1}e \leqslant 1-c_1^{-1}e$, of which the first inequality follows from the facts $e \in [0,0.5]$ and $c_0^{-1}+c_1^{-1} = 2$; and the second inequality is clear from the assumption $c_0 \leqslant c_1$.

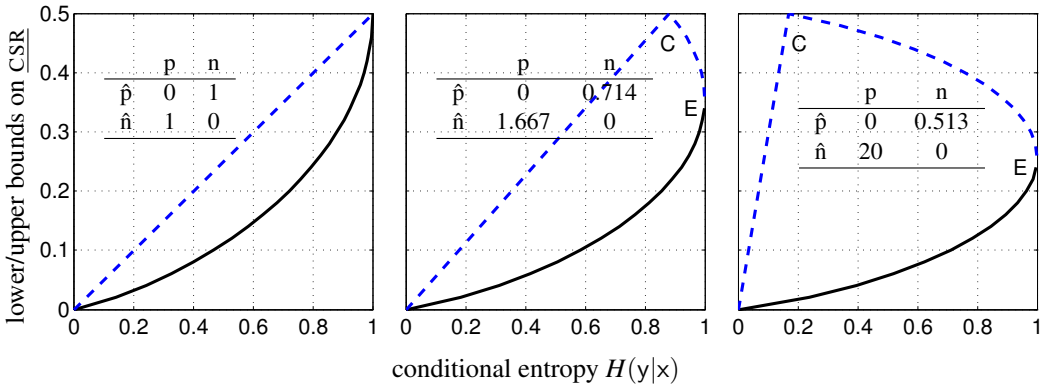


Figure 5: The lower (solid line) and upper (dashed line) bounds on the minimum cost-sensitive risk, CSR, in terms of Shannon's conditional entropy. From left to right the cost of false negative is set to be $c_1 = 1, \frac{5}{3}$ and 20, respectively; and the cost of false positive is determined by the condition $c_0^{-1} + c_1^{-1} = 2$. Attached with each graph is the corresponding cost matrix, where p (n) refers to the real positive (negative); and $\hat{\mathsf{p}}$ ($\hat{\mathsf{n}}$) is the predicted positive (negative). Note that the left figure reproduces Fano's and Hellman's bounds (see Figure 1).

To give the reader an intuitive feeling about how the two bounds on the minimum cost-sensitive risk as shown in Equation (31) vary in accordance with different settings of c_0 and c_1, we plotted in Figure 5 curves of these lower/upper bounds for $c_1 = 1$, $c_1 = \frac{5}{3}$ and $c_1 = 20$, with the corresponding cost matrix attached for each subfigure. From the figure we see that when the parameter c_1 increases from 1 to ∞, the peak point C (at which CSR takes the maximum value 0.5) moves left from the top right corner $[1, 0.5]$ to the top left corner $[0, 0.5]$; whereas another extreme point E (at which $H(\mathsf{y}|\mathsf{x})$ takes the maximum value 1) moves down from the point $[1, 0.5]$ to the point $[1, 0.25]$.

A fact one should notice is that here the (tight) upper bound on CSR is no longer monotonically increasing with $H(\mathsf{y}|\mathsf{x})$, especially when c_1 is large, that is, the positive class is regarded as much more important than the negative class. This implies a non-intuitive situation. Usually with classification problems, as we decrease the conditional entropy, we would expect the worst case classification scenario to improve. With cost-sensitive risk, however, as we decrease entropy, in the worst case the cost-sensitive risk could possibly become larger—compare the points C and E in Figure 5. This important observation was *not* noted before in the literature. We will discuss it in more detail in Section 8.

5. Bounding the Minimum Balanced Error Rate by Conditional Entropy

The main goal in this section is to derive the *tight* lower and upper bounds on minimum balanced error rate, BER, for binary classification problems with given conditional entropy $H(\mathsf{y}|\mathsf{x})$ and (prior) class probabilities, $\pi = \Pr\{\mathsf{y} = 1\}$ and $\tilde{\pi} = \Pr\{\mathsf{y} = 0\}$. To do so, we need first to derive the expression of BER.

As we have already mentioned at the end of Section 2.3, for any given task (μ,η) the balanced error rate BER as a functional of classifiers $\hat{y}(\cdot)$ can be seen as a special cost-sensitive risk with $c_1 = \frac{1}{2}\pi^{-1}$ and $c_0 = \frac{1}{2}\tilde{\pi}^{-1}$. This observation allows us to re-use the analysis presented in Section 4 to obtain the expression of the minimum balanced error rate, $\underline{\text{BER}}$. In fact, putting $c_1 = \frac{1}{2}\pi^{-1}$ and $c_0 = \frac{1}{2}\tilde{\pi}^{-1}$ in Equation (26)—note that the derivation of Equation (26) does not rely on the conditions $c_0^{-1} + c_1^{-1} = 2$ and $c_0 \leqslant c_1$, we at once get

$$\underline{\text{BER}} = \tfrac{1}{2}\int_{\mathcal{X}} \min\{\pi^{-1}\eta(x), \tilde{\pi}^{-1}\tilde{\eta}(x)\}\mathrm{d}\mu; \tag{32}$$

and the corresponding optimal classifier is $\hat{y}(x) = [\![\eta(x) > \frac{c_0}{c_0+c_1}]\!] = [\![\eta(x) > \pi]\!]$.

We have also pointed out that, as far as only those tasks (μ,η) with constant class probabilities $\pi = \Pr\{\mathsf{y}=1\}$ and $\tilde{\pi} = \Pr\{\mathsf{y}=0\}$ are concerned, the quantity BER can still be regarded as a special case of CSR. Accordingly, $\underline{\text{BER}}$ as a functional of tasks (μ,η) is a special case of $\underline{\text{CSR}}$, that is, $\underline{\text{BER}}(\mu,\eta) = \underline{\text{CSR}}(\mu,\eta;c_0,c_1)$ with $c_0 = \frac{1}{2}\tilde{\pi}^{-1}$ and $c_1 = \frac{1}{2}\pi^{-1}$. (Conceptually, however, the two are totally different, as we will see soon.) We thus get from Theorem 7 the following corollary.

Corollary 8 (bounds on $H(\mathsf{y}|\mathsf{x})$ as functions of $\underline{\text{BER}}$) *Let $h(\eta)$, $\eta \in [0,1]$, be a concave function satisfying $h(0) = h(1) = 0$ and $h(\eta) = h(1-\eta)$ (symmetric). Then for any binary classification task (μ,η) with $\Pr\{\mathsf{y}=1\} = \pi$ it holds that*

$$2\cdot h(\pi)\cdot\underline{\text{BER}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant \begin{cases} h(2\tilde{\pi}\cdot\underline{\text{BER}}) & \text{if } \pi \leqslant 0.5 \\ h(2\pi\cdot\underline{\text{BER}}) & \text{if } \pi > 0.5 \end{cases}, \tag{33}$$

where $H(\mathsf{y}|\mathsf{x})$ is defined as in Definition 2, and $\underline{\text{BER}}$ is given by Equation (32)*.*

Proof If $\pi \leqslant \frac{1}{2}$, the settings $c_1 = \frac{1}{2}\pi^{-1}$ and $c_0 = \frac{1}{2}\tilde{\pi}^{-1}$ satisfy the two conditions $c_0 \leqslant c_1$ and $c_0^{-1} + c_1^{-1} = 2$. So from Equation (29) we get $h(\pi)\cdot 2\underline{\text{BER}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(2\tilde{\pi}\cdot\underline{\text{BER}})$. For the case of $\pi \geqslant \frac{1}{2}$, we still set $c_1 = \frac{1}{2}\pi^{-1}$ and $c_0 = \frac{1}{2}\tilde{\pi}^{-1}$ but interchange the roles of c_0 and c_1 in Equation (29). This gives us $h(\pi)\cdot 2\underline{\text{BER}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(2\pi\cdot\underline{\text{BER}})$. Here we have used the fact that $h(\pi) = h(\tilde{\pi})$—notice that $h(\cdot)$ is a symmetric function. ■

As we have just said, balanced error rate and cost-sensitive risk are two different concepts. The main difference is that in the expression of CSR, Equation (15), the two coefficients c_0 and c_1 depend on neither μ nor η; whereas for BER, Equation (13), the values of c_0 and c_1 depend on the concerned problem (μ,η). This difference results in that the two bounds on $H(\mathsf{y}|\mathsf{x})$ as claimed by Corollary 8 are *not* necessarily tight any more. The reason is that, in terms of $\underline{\text{CSR}}$, the lower/upper bounds on $H(\mathsf{y}|\mathsf{x})$ are obtained over all tasks with a fixed value of $\underline{\text{CSR}}$; whereas in terms of $\underline{\text{BER}}$, we implicitly imposed an additional condition on the tasks, namely, the class probabilities should also be fixed as π and $\tilde{\pi}$. Consequently, the development presented in Section 4 can not be directly applied here to obtain the *tight* bounds.

We now use Scheme 3 to derive the tight bounds on $H(\mathsf{y}|\mathsf{x})$ in terms of $\underline{\text{BER}}$ and π. The resulting bounds are presented in Theorem 11.—This is only for theoretical convenience: in practice we are often more interested in using $H(\mathsf{y}|\mathsf{x})$ to bound $\underline{\text{BER}}$, which can be obtained from Theorem 11 by simply interchanging the axes of $H(\mathsf{y}|\mathsf{x})$ and $\underline{\text{BER}}$, as has been done in Figure 6-b. In fact, we have already used this technique to study the bounds on $\underline{\text{CSR}}$ in terms of $H(\mathsf{y}|\mathsf{x})$, see the derivation of Equation (31) in page 1050.

In Equation (21), we have already expressed $H(\mathsf{y}|\mathsf{x})$ as the mathematical expectation of some function of $\eta(\mathsf{x})$. The other two quantities involved, π and $\underline{\mathrm{BER}}$, can also be written in this way. In fact, by Equations (32) and (4), one easily sees that $\underline{\mathrm{BER}} = \frac{1}{2}\mathbb{E}_{\mathsf{x}\sim\mu}[r(\eta(\mathsf{x}))]$ and $\pi = \mathbb{E}_{\mathsf{x}\sim\mu}[\eta(\mathsf{x})]$, where the function $r(\eta)$ is defined as

$$r(\eta) := \min\{\pi^{-1}\eta, \tilde{\pi}^{-1}\tilde{\eta}\}\,, \qquad \eta \in [0,1]\,.$$

So Equations (4), (32) and (21) can be rewritten together in vector notation as

$$[\pi, 2\underline{\mathrm{BER}}, H(\mathsf{y}|\mathsf{x})] = \mathbb{E}_{\mathsf{x}\sim\mu}[\eta(\mathsf{x}), r(\eta(\mathsf{x})), h(\eta(\mathsf{x}))]\,. \tag{34}$$

It then follows from Theorem 1 that the point $[\pi, 2\underline{\mathrm{BER}}, H(\mathsf{y}|\mathsf{x})]$ is in the convex hull of the curve $\ell = \{[\eta, r(\eta), h(\eta)] \mid \eta \in [0,1]\}$ in the *three* dimensional η-r-h space.

Next we would implement the second and third steps in Scheme 3, for which we use the graph of ℓ and its convex hull with $\pi = 0.3$ and $h(\eta) = -\eta\log\eta - \tilde{\eta}\log\tilde{\eta}$ (Shannon) as the example (see Figure 6-a). For any given value of π, the function $r(\eta)$ is piecewise linear: it equals to $\pi^{-1}\eta$ if $\eta \leqslant \pi$ and $\tilde{\pi}^{-1}\tilde{\eta}$ otherwise. The curve ℓ is hence divided into two parts by the point $[\pi, 1, h(\pi)]$—the point A in Figure 6-a. The part with $\eta \leqslant \pi$ is in the plane $\eta = \pi r$ (plane AOD); and the part with $\eta \geqslant \pi$ is in the plane $\tilde{\eta} = \tilde{\pi} r$ (plane ABD). For such a curve simple geometry tells us that its convex hull is bounded by the triangle OAB, the two bow shapes OAO and ABA, and the minimal "concave" curved surface OAB bordered by the curve ℓ and the line segment OB—see Appendix B.3 for more detail.

Finally, we want to compute the *tight* lower and upper bounds on $H(\mathsf{y}|\mathsf{x})$ from the convex hull of ℓ. For any fixed value of π, this is equivalent to seeking the intersection of the plane $\eta = \pi$ and the convex hull of ℓ. From Figure 6-a, it is easy to see that $H(\mathsf{y}|\mathsf{x})$ is lower bounded by the line segment AC, the intersection of the planes ADC (the plain $\eta = \pi$) and OAB (the "tight lower bound" of the convex hull of ℓ). It is also obvious that the two endpoints of AC have the coordinates $\mathrm{A}(\pi, 1, h(\pi))$ and $\mathrm{C}(\pi, 0, 0)$. Therefore,

$$h(\pi) \cdot 2\underline{\mathrm{BER}} \leqslant H(\mathsf{y}|\mathsf{x})\,. \tag{35}$$

This inequality is same as the first inequality in Equation (33); they are nevertheless obtained by different methods. The most important difference is that here we can safely claim the sharpness of the inequality (by an argument similar to that in Theorem 5 and Corollary 6), which is not clear from Corollary 8.

Analogously, the tight upper bound on $H(\mathsf{y}|\mathsf{x})$ is determined by the intersection curve of plane $\eta = \pi$ (plane ADC) and the aforementioned curved surface OAB. Therefore, to compute this tight upper bound we need to find the maximal value of h such that the point $[\pi, 2\underline{\mathrm{BER}}, h]$ is in the convex hull of ℓ. By the definition of convex hull, this can be done as follows. Pick any two points, say M and N (not plotted in Figure 6-a), on the curve ℓ or the line segment OB, so that the line segment MN meets the vertical line defined by $\eta = \pi$ and $r = 2\underline{\mathrm{BER}}$, say the line EF in the figure, at some point K. By definition, we know point K is in the convex hull. So its h-coordinate, K_h, is no more than the maximal value of $H(\mathsf{y}|\mathsf{x})$; and the maximum of K_h (over all possible pairs M and N) is exactly the *tight* upper bound of $H(\mathsf{y}|\mathsf{x})$.

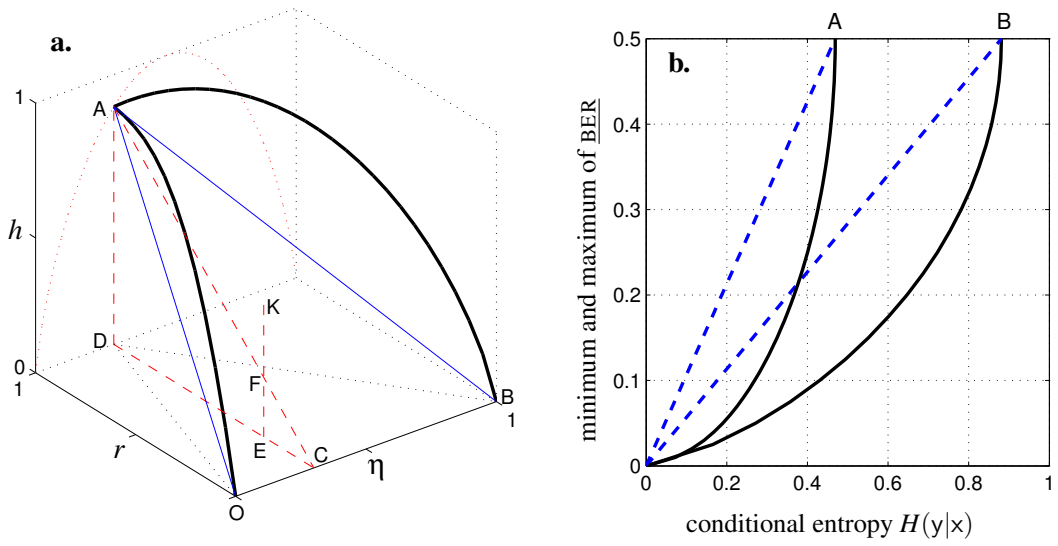


Figure 6: **a.** The curve $\ell = \{[\eta, r(\eta), h(\eta)] \mid \eta \in [0,1]\}$ and its convex hull in the η-r-h space. **b.** The tight lower (solid lines) and upper (dashed lines) bounds on BER versus the Shannon conditional entropy for $\pi = 0.1$ (A) and $\pi = 0.3$ (B). Note the difference between the bow shape (B) here and that in the middle graph of Figure 5—the two use the same parameters: $\pi = 0.3$, that is, $c_1 = \frac{5}{3}$.

To compute the maximal value of K_h, let M, N be as above and write $\rho := 2\mathrm{BER}$. Then, as K is on the line segment MN, there exists a unique $t \in (0,1)$ such that

$$\mathrm{K}_\eta = \tilde{t} \cdot \mathrm{M}_\eta + t \cdot \mathrm{N}_\eta = \pi\,, \tag{36}$$

$$\mathrm{K}_r = \tilde{t} \cdot \mathrm{M}_r + t \cdot \mathrm{N}_r = \rho\,, \tag{37}$$

$$\mathrm{K}_h = \tilde{t} \cdot \mathrm{M}_h + t \cdot \mathrm{N}_h\,, \tag{38}$$

where the subscript η, r or h denotes the corresponding coordinate of the concerned point. We shall discuss two different cases separately.

Case 1: One of M *and* N*, say* M*, is on line segment* OB. That is, $\mathrm{M}_r = \mathrm{M}_h = 0$ and $0 \leqslant \mathrm{M}_\eta \leqslant 1$. If $0 \leqslant \mathrm{M}_\eta \leqslant \pi$, by Equation (36), $\mathrm{N}_\eta \geqslant \pi$ and so $\mathrm{N}_r = \tilde{\pi}^{-1}(1 - \mathrm{N}_\eta)$. This equation, together with Equations (36) and (37), implies that $\mathrm{N}_\eta = 1 - t^{-1}\tilde{\pi}\rho$ and $\mathrm{M}_\eta = 1 - \tilde{t}^{-1}\tilde{\pi}\tilde{\rho}$. So from Equation (38) we know $\mathrm{K}_h = t \cdot h(1 - t^{-1}\tilde{\pi}\rho)$, where the range of t is determined by the condition $0 \leqslant \mathrm{M}_\eta \leqslant \pi$, from which we obtain $t \in [\rho, \rho + \pi\tilde{\rho}]$. Similarly, for $\pi \leqslant \mathrm{M}_\eta \leqslant 1$ it holds that $\mathrm{N}_\eta \leqslant \pi$ and $\mathrm{N}_r = \pi^{-1}\mathrm{N}_\eta$; and by solving the three equations (36)–(38) we get $\mathrm{N}_\eta = t^{-1}\pi\rho$, $\mathrm{M}_\eta = \tilde{t}^{-1}\pi\tilde{\rho}$, and $\mathrm{K}_h = t \cdot h(t^{-1}\pi\rho)$, with $t \in [\rho, \rho + \tilde{\pi}\tilde{\rho}]$.

Case 2: Both M *and* N *are on the curve* ℓ. Without loss of generality, assume $\mathrm{M}_\eta \leqslant \mathrm{N}_\eta$. Then by Equation (36) we know $\mathrm{M}_\eta \leqslant \pi \leqslant \mathrm{N}_\eta$; and hence $\mathrm{M}_r = \pi^{-1}\mathrm{M}_\eta$ and $\mathrm{N}_r = \tilde{\pi}^{-1}(1 - \mathrm{N}_\eta)$. Substituting the two equations into Equation (37) and solving the resulting linear equations (36) and (37), we

arrive at $\mathrm{M}_\eta = \pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}$ and $\mathrm{N}_\eta = \pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}$. It then follows from Equation (38) that $\mathrm{K}_h = \tilde{t}\cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t\cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho})$, where $t \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]$, as is determined by the conditions $\mathrm{M}_\eta \geqslant 0$ and $\mathrm{N}_\eta \leqslant 1$.

Summing up the above discussion, we conclude that the tight upper bound on $H(\mathsf{y}|\mathsf{x})$ is the maximum of the three maxima:

$$\mathrm{K}_h^{(1)} = \max\{f_1(t) := t\cdot h(1 - t^{-1}\tilde{\pi}\rho) \mid t \in [\rho, \rho + \pi\tilde{\rho}]\}, \tag{39}$$

$$\mathrm{K}_h^{(2)} = \max\{f_2(t) := t\cdot h(t^{-1}\pi\rho) \mid t \in [\rho, \rho + \tilde{\pi}\tilde{\rho}]\}, \tag{40}$$

$$\mathrm{K}_h^{(3)} = \max\{f_3(t) := \tilde{t}\cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t\cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) \mid t \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]\}. \tag{41}$$

Lemma 9 *Let $h : [0,1] \to \mathbb{R}$ be a concave function. Let $u, v, w \in [0,1]$ and $\alpha, \beta, \gamma \geqslant 0$ be such that $\alpha + \beta = \gamma$ and $\alpha u + \beta v = \gamma w$. Then*

$$\alpha\cdot h(u) + \beta\cdot h(v) \leqslant \gamma\cdot h(w). \tag{42}$$

Proof The case of $\alpha = \beta = \gamma = 0$ is trivial. If at least one of the three is nonzero, then $\gamma > 0$ since $\gamma = \alpha + \beta$. Equation (42) is then just a reformulation of the characterizing (defining) inequality of concave functions, $t\cdot h(u) + \tilde{t}\cdot h(v) \leqslant h(t\cdot u + \tilde{t}\cdot v)$, with $t = \gamma^{-1}\alpha$. ■

Lemma 10 *In Equations* (39)–(41)*,* (a) *the functions $f_1(t)$ and $f_2(t)$ are monotonically non-decreasing, so $\mathrm{K}_h^{(1)} = f_1(\rho + \pi\tilde{\rho})$ and $\mathrm{K}_h^{(2)} = f_2(\rho + \tilde{\pi}\tilde{\rho})$;* (b) *the function $f_3(t)$ is concave, and its value at the two endpoints are $f_3(\rho + \pi\tilde{\rho}) = \mathrm{K}_h^{(1)}$ and $f_3(\pi\tilde{\rho}) = \mathrm{K}_h^{(2)}$, respectively.*

Proof (a) For $t_1, t_2 \in [\rho, \rho + \pi\tilde{\rho}]$ satisfying $t_1 \leqslant t_2$, we need to show that $f_1(t_1) \leqslant f_1(t_2)$, that is, $t_1\cdot h(1 - t_1^{-1}\tilde{\pi}\rho) \leqslant t_2\cdot h(1 - t_2^{-1}\tilde{\pi}\rho)$. This can be obtained by substituting $(\alpha, \beta, \gamma) = (t_1, t_2 - t_1, t_2)$ and $(u, v, w) = (1 - t_1^{-1}\tilde{\pi}\rho, 1, 1 - t_2^{-1}\tilde{\pi}\rho)$ into Equation (42) and using the fact that $h(1) = 0$.

Similarly, the inequality $f_2(t_1) \leqslant f_2(t_2)$, that is, $t_1\cdot h(t_1^{-1}\pi\rho) \leqslant t_2\cdot h(t_2^{-1}\pi\rho)$ can be proven using the settings $(\alpha, \beta, \gamma) = (t_1, t_2 - t_1, t_2)$ and $(u, v, w) = (t_1^{-1}\pi\rho, 0, t_2^{-1}\pi\rho)$ for Equation (42), as well as the fact that $h(0) = 0$.

(b) For any $t_1, t_2 \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]$ and $\alpha \in (0,1)$, write $t = \alpha\cdot t_1 + \tilde{\alpha}\cdot t_2$. We want to prove that $f_3(t) \geqslant \alpha\cdot f_3(t_1) + \tilde{\alpha}\cdot f_3(t_2)$. But Lemma 9 implies that

$$\tilde{t}\cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) \geqslant \alpha\tilde{t}_1\cdot h(\pi - \tilde{t}_1^{-1}\pi\tilde{\pi}\tilde{\rho}) + \tilde{\alpha}\tilde{t}_2\cdot h(\pi - \tilde{t}_2^{-1}\pi\tilde{\pi}\tilde{\rho}),$$
$$t\cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) \geqslant \alpha t_1\cdot h(\pi + t_1^{-1}\pi\tilde{\pi}\tilde{\rho}) + \tilde{\alpha}t_2\cdot h(\pi + t_2^{-1}\pi\tilde{\pi}\tilde{\rho}).$$

The sum of the above two inequalities is exactly what we want: $f_3(t) \geqslant \alpha\cdot f_3(t_1) + \tilde{\alpha}\cdot f_3(t_2)$. Finally, the two identities $f_3(\rho + \pi\tilde{\rho}) = f_1(\rho + \pi\tilde{\rho}) = \mathrm{K}_h^{(1)}$ and $f_3(\pi\tilde{\rho}) = f_2(\rho + \tilde{\pi}\tilde{\rho}) = \mathrm{K}_h^{(2)}$ can be verified by direct computation. ■

As a consequence of the above lemma, we see that $H(\mathsf{y}|\mathsf{x})$ is actually upper bounded by the quantity $\mathrm{K}_h^{(3)}$ as defined in Equation (41), that is,

$$H(\mathsf{y}|\mathsf{x}) \leqslant \max\left\{f_3(t) = \tilde{t}\cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t\cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) \mid t \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]\right\}, \tag{43}$$

where $\rho = 2\underline{\text{BER}}$ is two times of the minimum balanced error rate and $\pi = \Pr\{y = 1\}$ the prior probability of the positive class. Furthermore, using an argument similar to that for Theorem 5 and Corollary 6, we can prove the above upper bound on $H(y|x)$ is tight.

Combining Equation (35) and Equation (43), we get

Theorem 11 (tight bounds on $H(y|x)$ in terms of $\underline{\text{BER}}$ and π) *Let $h(\eta)$, $\eta \in [0,1]$, be a concave function satisfying $h(0) = h(1) = 0$. Then for any binary classification task (μ, η) with $\Pr\{y = 1\} = \pi$ it holds that*

$$2 \cdot h(\pi) \cdot \underline{\text{BER}} \leqslant H(y|x) \leqslant \max\{f_3(t) \mid t \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]\} = \text{K}_h^{(3)}, \tag{44}$$

where $\rho = 2\underline{\text{BER}}$ and the function $f_3(t)$ is defined by Equation (43)*.*

Notice that Theorem 11 does not require that $h(\cdot)$ be symmetric. Furthermore, as has been pointed out earlier, there are two ways to understand this theorem. The first way is to see π as a given parameter, then Equation (44) describes the relationship between the functionals $H(y|x)$ and $\underline{\text{BER}}$; and it holds for any task with $\Pr\{y = 1\} = \pi$. We can also regard π as a functional of tasks, then Equation (44) connects the three quantities: π, $\underline{\text{BER}}$ and $H(y|x)$; and holds for any classification task (μ, η).

In Theorem 11, the tight upper bound on $H(y|x)$ has been written as the maximum of a concave function $f_3(t) = \tilde{t} \cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t \cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho})$ over the interval $[\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]$. This maximum has *no* closed-form expression in general, we therefore resort to numerical methods. If the function $h(\cdot)$ is differentiable, so is $f_3(t)$—the derivative of $f_3(t)$ is

$$\begin{aligned} f_3'(t) = &- h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho} \cdot h'(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) \\ &+ h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) - t^{-1}\pi\tilde{\pi}\tilde{\rho} \cdot h'(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}). \end{aligned}$$

In this case, the maximum of $f_3(t)$ can be obtained by checking the values of its derivative $f_3'(t)$. First at the two endpoints $\pi\tilde{\rho}$ and $\rho + \pi\tilde{\rho}$: if $f_3'(\pi\tilde{\rho}) \leqslant 0$, then $\text{K}_h^{(3)} = f_3(\pi\tilde{\rho})$; if $f_3'(\rho + \pi\tilde{\rho}) \geqslant 0$, then $\text{K}_h^{(3)} = f_3(\rho + \pi\tilde{\rho})$. Otherwise we need to calculate the unique solution $t_0 \in (\pi\tilde{\rho}, \rho + \pi\tilde{\rho})$ to the equation $f_3'(t) = 0$ and obtain $\text{K}_h^{(3)} = f_3(t_0)$. This can be done very efficiently by simple numerical methods such as bisection, since $f_3'(t)$ is a non-increasing function of t. If $h(\cdot)$ is not differentiable, one still can use simple numerical methods such as the Fibonacci search and the golden section search (Brent, 1973, p. 68) to locate the maximum of $f_3(t)$, since $f_3(t)$ is a unimodal function.

For the Shannon conditional entropy, $h(\eta) = -\eta \log p - \tilde{\eta} \log \tilde{\eta}$ and the value of $f_3'(t)$ at the two endpoints are $f_3'(\rho + \pi\tilde{\rho}) = -\infty$ and $f_3'(\pi\tilde{\rho}) = \infty$, respectively. The problem is thus reduced to solving the equation $f_3'(t) = 0$, which can be simplified to

$$\pi \log(1 - \tilde{t}^{-1}\tilde{\pi}\tilde{\rho}) + \tilde{\pi} \log(1 + \tilde{t}^{-1}\pi\tilde{\rho}) = \pi \log(1 + t^{-1}\tilde{\pi}\tilde{\rho}) + \tilde{\pi} \log(1 - t^{-1}\pi\tilde{\rho}).$$

Its solution t_0 is then substituted into the expression of $f_3(t)$, yielding the tight upper bound of $H(y|x)$. In Figure 6-b the lower and upper bounds on $H(y|x)$ are plotted versus minimum balanced error rate $\underline{\text{BER}} = \frac{1}{2}\rho$ for $\pi = 0.1$ and 0.3—corresponding to $c_1 = \frac{1}{2}\pi^{-1} = 5$ and $\frac{5}{3}$. The graph has used the x-axis for $H(y|x)$ and the y-axis for $\underline{\text{BER}}$, so that one can easily check the bounds on $\underline{\text{BER}}$ for given values of $H(y|x)$. One may compare the middle graph of Figure 5 with Figure 6-b, to confirm that the upper bound on $H(y|x)$ stated by Corollary 8 is indeed untight (for those tasks with $0 < \underline{\text{BER}} < 0.5$). In Appendix C, we will show that *the upper bound of $H(y|x)$ given by Theorem 11 is never looser than that in Corollary 8.*

To conclude, we have used the proposed geometric method to derive the *tight* lower and upper bounds on $H(\mathsf{y}|\mathsf{x})$ in terms of $\underline{\mathrm{BER}}$ and π. By flipping the curve of these bounds along the diagonal line, we can also get the *tight* lower and upper bounds on $\underline{\mathrm{BER}}$ as functions of $H(\mathsf{y}|\mathsf{x})$ and π, as we have done for the quantity $\underline{\mathrm{CSR}}$—see Equation (31). As shown by Corollary 8, these tight bounds are not obtained by simply taking the balanced error rate as a special cost-sensitive risk, even though here the value of π is assumed to be a constant. This confirms that balanced error rate and cost-sensitive risk are two *essentially* different performance measures.

6. Maximum F-score in the Binary Classification Problem

We now consider the relationship between F-score and conditional entropy. As before, we shall first derive the maximum value of F-score for a given classification problem (μ,η). By Equation (17), this amounts to maximizing the set function

$$\Gamma(\mathcal{X}_1) := \tfrac{1}{2}\mathrm{FSC} = [\pi+\mu(\mathcal{X}_1)]^{-1}\cdot\textstyle\int_{\mathcal{X}_1}\eta(x)\mathrm{d}\mu, \tag{45}$$

under the assumption that the object distribution μ and the conditional class probability $\eta(x)$ are constants (so the class probability π is also a constant).

Lemma 12 *Let the set function $\Gamma(\mathcal{X}_1)$ be as in Equation* (45)*. For any measurable subset $\mathcal{X}_1$ of $\mathcal{X}$, let $\mathcal{X}_1' = \{x\in\mathcal{X} \mid \eta(x) > \Gamma(\mathcal{X}_1)\}$. Then $\Gamma(\mathcal{X}_1') \geqslant \Gamma(\mathcal{X}_1)$.*

Proof Write $\theta = \Gamma(\mathcal{X}_1)$. Let $A = \{x\in\mathcal{X}_1 \mid \eta(x)\leqslant\theta\}$ and $B = \{x\notin\mathcal{X}_1 \mid \eta(x)>\theta\}$. Then $A\subseteq\mathcal{X}_1$, $B\cap\mathcal{X}_1 = \varnothing$ and $\mathcal{X}_1' = (\mathcal{X}_1\setminus A)\cup B$. Thus, by Equation (45),

$$\Gamma(\mathcal{X}_1') = \frac{\int_{\mathcal{X}_1'}\eta(x)\mathrm{d}\mu}{\pi+\mu(\mathcal{X}_1')} = \frac{\int_{\mathcal{X}_1}\eta(x)\mathrm{d}\mu+\int_B\eta(x)\mathrm{d}\mu-\int_A\eta(x)\mathrm{d}\mu}{\pi+\mu(\mathcal{X}_1)+\mu(B)-\mu(A)}.$$

As $\eta(x)>\theta$ on B, $\int_B\eta(x)\mathrm{d}\mu\geqslant\theta\mu(B)$. Similarly, $\int_A\eta(x)\mathrm{d}\mu\leqslant\theta\mu(A)$. Furthermore, by the definition of $\Gamma(\mathcal{X}_1)$, we have $\int_{\mathcal{X}_1}\eta(x)\mathrm{d}\mu = \theta[\pi+\mu(\mathcal{X}_1)]$. All these three facts together imply that $\Gamma(\mathcal{X}_1')\geqslant\theta=\Gamma(\mathcal{X}_1)$. ■

This theorem allows us to consider only classifiers of the form $\hat{y}(x) = [\![\eta(x)>\theta]\!]$ when maximizing the F-score, where $\theta\in[0,1]$ is a threshold. To determine the optimal threshold θ so that the F-score, or, equivalently, the function $\Gamma(\mathcal{X}_1)$ is maximized, where the set $\mathcal{X}_1$ is defined via θ as $\mathcal{X}_1(\theta) = \{x\in\mathcal{X}\mid\eta(x)>\theta\}$, we rewrite Equation (45) as a function of θ:

$$\Gamma(\theta) = \frac{\int_{\mathcal{X}_1}\eta(x)\mathrm{d}\mu}{\pi+\mu(\mathcal{X}_1)} = \frac{\theta\mu(\mathcal{X}_1)+\int_{\mathcal{X}_1}[\eta(x)-\theta]\mathrm{d}\mu}{\pi+\mu(\mathcal{X}_1)} = \theta+\frac{\int_{\mathcal{X}_1}[\eta(x)-\theta]\mathrm{d}\mu-\pi\theta}{\pi+\mu(\mathcal{X}_1)}. \tag{46}$$

For any $r\in\mathbb{R}$, write $r^+ := \max\{0,r\}$. By the definition of $\mathcal{X}_1$, $\eta(x)-\theta>0$ iff $x\in\mathcal{X}_1$. So for $x\in\mathcal{X}_1$, $[\eta(x)-\theta]^+ = \eta(x)-\theta$; and for $x\notin\mathcal{X}_1$, $[\eta(x)-\theta]^+ = 0$. It therefore follows that $\int_{\mathcal{X}_1}[\eta(x)-\theta]\mathrm{d}\mu = \int_{\mathcal{X}}[\eta(x)-\theta]^+\mathrm{d}\mu$. Substituting this into Equation (46), we obtain

$$\Gamma(\theta) = \frac{\theta\mu(\mathcal{X}_1)+\int_{\mathcal{X}}[\eta(x)-\theta]^+\mathrm{d}\mu}{\pi+\mu(\mathcal{X}_1)} = \theta+\frac{\int_{\mathcal{X}}[\eta(x)-\theta]^+\mathrm{d}\mu-\pi\theta}{\pi+\mu(\mathcal{X}_1)}. \tag{47}$$

Lemma 13 *The function* $g(\theta) := \int_{\mathcal{X}} [\eta(x) - \theta]^+ d\mu - \pi\theta$*, where* $\theta \in [0,1]$*, is continuous and strictly decreasing, with* $g(0) = \pi$ *and* $g(1) = -\pi$*.*

Proof Since $|(\eta - \theta_1)^+ - (\eta - \theta_2)^+| \leqslant |\theta_1 - \theta_2|$, we have $|g(\theta_1) - g(\theta_2)| \leqslant (1+\pi)|\theta_1 - \theta_2|$, so $g(\theta)$ is continuous. It is strictly decreasing because its first term is non-increasing (with respect to θ) and its second term, $-\pi\theta$, is strictly decreasing. Finally, as $0 \leqslant \eta(x) \leqslant 1$, we know $[\eta(x)]^+ = \eta(x)$ and $[\eta(x) - 1]^+ = 0$; so $g(0) = \pi$ and $g(1) = -\pi$. ■

By this lemma, we know there exists a unique $\theta^* \in (0,1)$ such that $g(\theta^*) = 0$, that is,

$$\int_{\mathcal{X}} [\eta(x) - \theta^*]^+ d\mu - \pi\theta^* = 0\,. \tag{48}$$

We now prove it is this θ^* that maximizes the function $\Gamma(\theta)$; and the maximum value is $\Gamma(\theta^*) = \theta^*$, as can be easily seen from Equation (47) and Equation (48).

Lemma 14 *The function* $\Gamma(\theta)$ *as given by Equation* (46) *is maximized at* θ^**.*

Proof We shall use the first expression of $\Gamma(\theta)$ from Equation (46), in which the subset $\mathcal{X}_1$ is defined as $\mathcal{X}_1 = \{x \in \mathcal{X} \mid \eta(x) > \theta\}$. If $\theta < \theta^*$, define $A := \{x \in \mathcal{X} \mid \eta(x) > \theta^*\}$ and $B := \{x \in \mathcal{X} \mid \theta^* \geqslant \eta(x) > \theta\}$. Then it is clear that $A \cap B = \varnothing$ and $A \cup B = \mathcal{X}_1$. Thus,

$$\Gamma(\theta) = \frac{\int_A \eta(x) d\mu + \int_B \eta(x) d\mu}{\pi + \mu(A) + \mu(B)}\,.$$

Now, as $\theta^* = \Gamma(\theta^*) = [\pi + \mu(A)]^{-1} \cdot \int_A \eta(x) d\mu$, we have $\int_A \eta(x) d\mu = \theta^* \cdot [\pi + \mu(A)]$. Moreover, $\int_B \eta(x) d\mu \leqslant \theta^* \mu(B)$ since $\eta(x) \leqslant \theta^*$ on B. Therefore, $\Gamma(\theta) \leqslant \theta^* = \Gamma(\theta^*)$.

If $\theta > \theta^*$, define A as before and $B := \{x \in \mathcal{X} \mid \theta \geqslant \eta(x) > \theta^*\}$. Then $B \subseteq A$ and $\mathcal{X}_1 = A \setminus B$. Thus,

$$\Gamma(\theta) = \frac{\int_A \eta(x) d\mu - \int_B \eta(x) d\mu}{\pi + \mu(A) - \mu(B)}\,.$$

Since $\eta(x) > \theta^*$ for $x \in B$, it holds that $\int_B \eta(x) d\mu \geqslant \theta^* \mu(B)$; whereas the equality $\int_A \eta(x) d\mu = \theta^* \cdot [\pi + \mu(A)]$ remains true. So, again, we obtain $\Gamma(\theta) \leqslant \theta^* = \Gamma(\theta^*)$. ■

In summary, to determine the maximum F-score for a given classification problem, one needs only to find the unique solution θ^* to the equation (48). The maximum F-score is then $\overline{\text{FSC}} = 2 \cdot \Gamma(\theta^*) = 2\theta^*$; and the corresponding optimal classifier is $\hat{y}(x) = [\![\eta(x) > \theta^*]\!]$. An interesting implication of the equality $\overline{\text{FSC}} = 2\theta^*$ is that $\theta^* \leqslant \frac{1}{2}$ (as $\overline{\text{FSC}} \leqslant 1$). That is, for F-score the optimal threshold is always less than or equal to 0.5.

7. Bounds on the Maximum F-score in Terms of Conditional Entropy

In this section, we derive bounds on maximum F-score, $\overline{\text{FSC}}$, in terms of the conditional entropy $H(\mathsf{y}|\mathsf{x})$ as defined by Equation (21). As before, we shall first examine the range of $H(\mathsf{y}|\mathsf{x})$ for any given value of $\overline{\text{FSC}}$.

In the preceding section we have proved that $\theta^* := \frac{1}{2}\overline{\mathrm{FSC}} \in [0, 0.5]$ (as $\overline{\mathrm{FSC}} \leqslant 1$) is the unique solution to the equation (48), which, by Equation (4), can be rewritten as

$$\int_{\mathcal{X}} \{\theta^* \cdot \eta(x) - [\eta(x) - \theta^*]^+\} \, \mathrm{d}\mu = 0; \quad \text{i.e.,} \quad \mathbb{E}_{\mathsf{x}\sim\mu}[u(\eta(\mathsf{x}))] = 0,$$

where $u(\eta) := \theta^*\eta - (\eta - \theta^*)^+$ is a function on $[0, 1]$. For any fixed value of θ^*, we know from the above equation and Equations (4) and (21) that

$$[\pi, 0, H(\mathsf{y}|\mathsf{x})] = \mathbb{E}_{\mathsf{x}\sim\mu}[\eta(\mathsf{x}), u(\eta(\mathsf{x})), h(\eta(\mathsf{x}))]. \tag{49}$$

By Theorem 1, in the η-u-h space $[\pi, 0, H(\mathsf{y}|\mathsf{x})]$ is a point in the convex hull of the curve $\ell = \{[\eta, u(\eta), h(\eta)] \mid \eta \in [0, 1]\}$.—*We have completed (a variant of) the first step in Scheme 3.*

In Figure 7-a the graph of the curve ℓ is plotted for $\theta^* = 0.3$ and the Shannon conditional entropy. By the definition of $u(\eta)$, we have $u(\eta) = \theta^*\eta$ for $\eta \leqslant \theta^*$ and $u(\eta) = \theta^* - \tilde{\theta}^*\eta$ for $\eta \geqslant \theta^*$. Thus, as in the case of balanced error rate, here the curve ℓ consists also of two parts each of which is in a plane. Consequently, its convex hull is bounded by three flat facets and one curved surface OAB (O is the origin) which is the minimum concave surface with line segment OB and curve ℓ as its boundary.—*These are the second and third steps in Scheme 3.*

As its second coordinate is a constant 0, the point $[\pi, 0, H(\mathsf{y}|\mathsf{x})]$ lies in the intersection of the plane $u = 0$ and the convex hull of ℓ. Therefore, as shown by Figure 7-a, $H(\mathsf{y}|\mathsf{x})$ is lower bounded by the line OD; and upper bounded by the curve OE that is the intersection of the plane $u = 0$ and the curved surface OAB we just mentioned. Notice here that, π is not fixed, but may take values between the points O and C (C is the intersection point of line $u = h = 0$ and plane $u = \theta^* - \tilde{\theta}^*\eta$). That is, $\pi \in [0, \theta^*/\tilde{\theta}^*]$. Thus, the lower bound of $H(\mathsf{y}|\mathsf{x})$ is given by the minimum h-coordinate of points on the line segment OD. Obviously, this equals to 0, the h-coordinate of the origin point O, which happens when π tends to zero. That means, for any given value of θ^*, the *tight* lower bound of $H(\mathsf{y}|\mathsf{x})$ is always 0.

Similarly, the maximum h-coordinate of points on the curve OE is the upper bound of $H(\mathsf{y}|\mathsf{x})$. For *symmetric* functions $h(\cdot)$, we shall soon prove that the endpoint E has the maximum h-coordinate (over all points on the curve OE). Furthermore, from Figure 7-a we know the η-coordinate of E is $\mathrm{E}_\eta = \theta^*/\tilde{\theta}^*$, so its h-coordinate $\mathrm{E}_h = h(\theta^*/\tilde{\theta}^*) = h\left(\frac{\overline{\mathrm{FSC}}}{2-\overline{\mathrm{FSC}}}\right)$.

From the above discussion we obtain the tight lower and upper bounds on $H(\mathsf{y}|\mathsf{x})$, as follows (*the fourth step in Scheme 3*).

Theorem 15 (tight bounds on $H(\mathsf{y}|\mathsf{x})$ in terms of $\overline{\mathrm{FSC}}$) *Let $H(\mathsf{y}|\mathsf{x})$ be the conditional entropy defined by a symmetric concave function $h : [0, 1] \to \mathbb{R}$. Then for any two-class problem (μ, η), it holds that*

$$0 \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h\left(\frac{\overline{\mathrm{FSC}}}{2-\overline{\mathrm{FSC}}}\right), \tag{50}$$

and the two inequalities are sharp.

A remark on Theorem 15: As we have used a variant of Scheme 3 to *derive* the inequality (50), the two general theorems 1 and 5 cannot be directly applied here and we need to *prove* it separately. In fact, in the above we have established the one-to-one correspondence between $\overline{\mathrm{FSC}}$ and the equation $\mathbb{E}_{\mathsf{x}\sim\mu}[u(\eta(\mathsf{x}))] = 0$ with $u(\eta) = \theta^*\eta - (\eta - \theta^*)^+$ and $\theta^* = \frac{1}{2}\overline{\mathrm{FSC}}$. That is, a task with the optimum F-score $\overline{\mathrm{FSC}}$ must satisfy the condition $\mathbb{E}_{\mathsf{x}\sim\mu}[u(\eta(\mathsf{x}))] = 0$; and conversly, a task satisfying this condition must have the optimum F-score $\overline{\mathrm{FSC}}$. Therefore, although we cannot find a

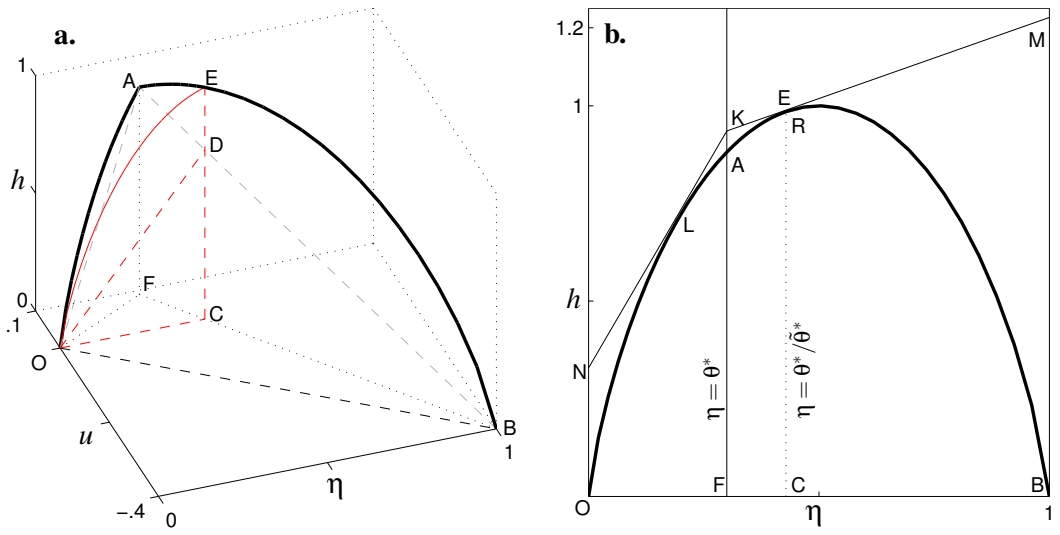


Figure 7: **a.** The curve $\ell = \{[\eta, u(\eta), h(\eta)] \mid \eta \in [0,1]\}$ and its convex hull in the η-u-h space; ODE is the intersection of the plane $u = 0$ and the convex hull, in which we are interested.
b. An geometric interpretation of Lemma 18: draw from a point K on the line $\eta = \theta \leqslant \frac{1}{2}$ two tangent lines of $h(\eta)$, the height of the left tangent point L is less than that of the right one R, provided that $h(\eta)$ is symmetric. To prove the second inequality in Equation (50), here the right tangent point R is set to be on the line $\eta = \tilde{\theta}^{-1}\theta$; so it corresponds to the point E in Figure a.

random variable with expectation $\overline{\text{FSC}}$, we still can apply Theorem 1 to the auxiliary random variable $u(\eta(\mathsf{x}))$ in which $\theta^* = \frac{1}{2}\overline{\text{FSC}}$ serves as a parameter. The resulting tight bounds on $H(\mathsf{y}|\mathsf{x})$ are in fact $0 \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant h(\theta^*/\tilde{\theta}^*)$, which can be rewritten as Equation (50).—*So here we see an implicit use of Scheme 3.*

Analogous to the analysis in page 1050, from Theorem 15 we can easily derive the lower and upper bounds on the maximum F-score by means of conditional entropy. This is best illustrated by Figure 8, where the Shannon conditional entropy is used for $H(\mathsf{y}|\mathsf{x})$. In general, as the function $h(\cdot)$ is symmetric and concave, for any given value of $H(\mathsf{y}|\mathsf{x})$ in the range of $h(\cdot)$, there exists a unique $\beta \in [0, 0.5]$ such that $h(\beta) = h(1-\beta) = H(\mathsf{y}|\mathsf{x})$. So from Figure 8 we know the value of $\overline{\text{FSC}}$ must satisfy $\beta \leqslant \frac{\overline{\text{FSC}}}{2-\overline{\text{FSC}}} \leqslant 1-\beta$, that is, $\frac{2\beta}{1+\beta} \leqslant \overline{\text{FSC}} \leqslant \frac{2-2\beta}{2-\beta}$. An interesting observation of this inequality is that $\overline{\text{FSC}}$ can only assume the value $\frac{2}{3}$ when $\beta = \frac{1}{2}$, that is, when $H(\mathsf{y}|\mathsf{x}) = 1$. This can be explained as follows.

For the Shannon entropy it holds that $1 \geqslant H(\mathsf{y}) \geqslant H(\mathsf{y}|\mathsf{x})$; so $H(\mathsf{y}|\mathsf{x}) = 1$ would imply that $H(\mathsf{y}) = 1$ and $I(\mathsf{x};\mathsf{y}) = H(\mathsf{y}) - H(\mathsf{y}|\mathsf{x}) = 0$, where $I(\mathsf{x};\mathsf{y})$ is the mutual information between x and y. From $I(\mathsf{x};\mathsf{y}) = 0$ we know x and y are independent; and from $H(\mathsf{y}) = 1$, $\Pr\{\mathsf{y} = 0\} = \Pr\{\mathsf{y} = 1\} = \frac{1}{2}$. In other words, essentially there is only one classification task whose conditional entropy is 1; and it

is actually the most "uncertain" one. It is also the most difficult problem in that the feature vector is completely uninformative and the class label totally random. For such a task, the error probability on any object is always 0.5, regardless of which class is predicted. Hence, $\text{TP} = \text{FP}$ and $\text{TN} = \text{FN}$. It then follows from the second expression of FSC in Equation (17) that $\text{FSC} = \frac{2\times\text{TP}}{3\times\text{TP}+\text{FN}}$. Now, as $\text{FN} \geqslant 0$, the F-score has the maximum value $\frac{2}{3}$; and this happens when $\text{FN} = 0$, which requires that all objects be regarded as positive.

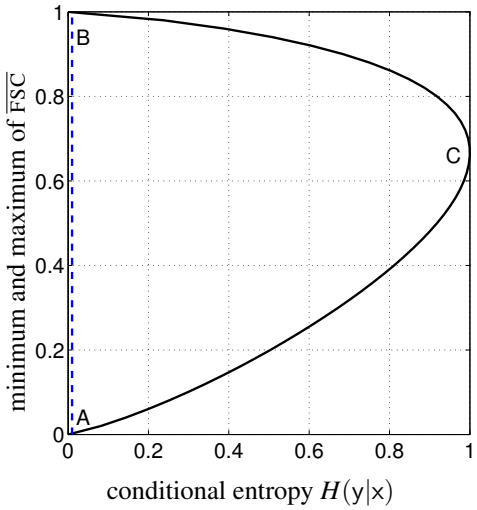


Figure 8: The relationship between $\overline{\text{FSC}}$ and $H(\text{y}|\text{x})$. For given $\overline{\text{FSC}}$, the dashed line AB is the lower bound on $H(\text{y}|\text{x})$; and the solid line ACB the upper bound. Therefore, for given value of $H(\text{y}|\text{x})$, the curve BC is the upper bound of $\overline{\text{FSC}}$; and the curve AC is the lower bound.

For the remainder of the section we will complete the derivation/proof of Theorem 15 by showing that E is the "highest" point on the curve OE, under the assumption that the concave function $h(\cdot)$ is *symmetric*, that is, $h(\eta) = h(\tilde{\eta})$ for any $\eta \in [0,1]$. To simplify the proof, we further assume that $h(\cdot)$ is a differentiable function.[12] The following two lemmas are easy to see. They are there only because they will be referenced several times and so help to shorten the argument that follows.

Lemma 16 *Let $h : [0,1] \to \mathbb{R}$ be a symmetric, differentiable and concave function. Then for $\eta \in [0, \frac{1}{2}]$, $h'(\eta) \geqslant 0$, that is, $h(\eta)$ is monotonically non-decreasing; and for $\eta \in [\frac{1}{2}, 1]$, $h(\eta)$ is monotonically non-increasing. Moreover, $h'(1-\eta) = -h'(\eta)$ for $\eta \in [0,1]$.*

Lemma 17 *Let $h : [0,1] \to \mathbb{R}$ be a differentiable concave function and $a \in [0,1]$. Then $f(t) := h(t) + h'(t) \cdot (a-t)$ is non-increasing on $[0,a]$; and non-decreasing on $[a,1]$.*

Proof For $s,t \in [0,a]$ satisfying $s < t$, the mean value theorem implies that, for some $u \in (s,t)$, $h(s) - h(t) = h'(u) \cdot (s-t)$. Since h is concave, its derivative h' is monotonically non-increasing. In

12. This assumption is in fact unnecessary: if $h(\cdot)$ is non-differentiable at some point η_0, we can use any number between its *right derivative* $h'(\eta_0+)$ and *left derivative* $h'(\eta_0-)$ to replace $h'(\eta_0)$.

particular, we have $h'(s) \geqslant h'(u) \geqslant h'(t)$. It thus follows that

$$\begin{aligned} f(s) &= h(s) + h'(s)\cdot(a-s) \\ &= h(t) + h'(u)\cdot(s-t) + h'(s)\cdot(a-s) \\ &\geqslant h(t) + h'(u)\cdot(s-t) + h'(u)\cdot(a-s) \\ &\geqslant h(t) + h'(t)\cdot(a-t) \;=\; f(t)\,. \end{aligned}$$

So $f(t)$ is a non-increasing function on $[0,a]$. Following a similar line, one can prove that $f(t)$ is non-decreasing on the interval $[a,1]$. ■

Lemma 18 *Let $h(\eta)$ be as in Lemma 16. For $\theta \in [0,\frac{1}{2}]$, let $a \leqslant \theta$ and $b \geqslant \theta$ be such that $h(a) + h'(a)(\theta - a) = h(b) - h'(b)(b-\theta)$. Then $h(a) \leqslant h(b)$.*

Proof If $b \leqslant \frac{1}{2}$, by Lemma 16 we at once get $h(a) \leqslant h(b)$. Assume now $b > \frac{1}{2}$, by Lemma 16 we know $h'(a) \geqslant 0$ and $h'(b) \leqslant 0$. Since $\theta \leqslant \frac{1}{2}$, the assumed equality implies

$$\begin{aligned} h(b) - h'(b)(b-\tfrac{1}{2}) &\leqslant h(b) - h'(b)(b-\theta) && \text{as } \theta \leqslant \tfrac{1}{2} \text{ and } h'(b) \leqslant 0 \\ &= h(a) + h'(a)(\theta - a) && \text{the assumed equality} \\ &\leqslant h(a) + h'(a)(\tfrac{1}{2} - a) && \text{as } \theta \leqslant \tfrac{1}{2} \text{ and } h'(a) \geqslant 0 \\ &= h(\tilde{a}) - h'(\tilde{a})(\tilde{a} - \tfrac{1}{2})\,. && \text{by Lemma 16} \end{aligned}$$

By $a \leqslant \theta \leqslant \frac{1}{2}$, we have $\tilde{a} \geqslant \frac{1}{2}$. By Lemma 17, $h(b) - h'(b)(b-\frac{1}{2})$ is non-decreasing with respect to $b \in [\frac{1}{2}, 1]$. So the above inequality implies $b \leqslant \tilde{a}$. As $b > \frac{1}{2}$, by Lemma 16 we know $h(b) \geqslant h(\tilde{a}) = h(a)$. ■

In geometry (see Figure 7-b), $h(a) + h'(a)(\theta - a)$ represents the "height" of the intersection point of the vertical line $\eta = \theta$ and the tangent line of $h(\eta)$ at $\eta = a$. With this in mind, we see that the assumed equality in Lemma 18 means the two tangent lines are drawn from one point on the line $\eta = \theta$. Thus, for a symmetric function $h(\eta)$ and $\theta = \frac{1}{2}$, this would give us the tangent points a and $b = \tilde{a}$ (by symmetry); and so $h(a) = h(b)$. When the line $\eta = \theta$ moves left, that is, $\theta < \frac{1}{2}$, the tangent points a and b also move left. This would result in $h(a) \leqslant h(b)$.

In Figure 7-a, we draw in the plane AFB (i.e., $u = \theta^* - \tilde{\theta}^*\eta$) the tangent line of the curve ℓ at point E, intersecting with line AF at K. From this point K we draw the tangent line of ℓ in the plane OAF ($u = \theta^*\eta$). These are represented in Figure 7-b as their projection on the plane $u = 0$. Assume M and N are the intersection points of the two tangent lines with the vertical lines at B and O, respectively. Then, by Lemma 16, the slope of KN is larger than zero; so the h-coordinate of N, N_h, is less than that of the left tangent point L, which, by Lemma 18, is further less than that of the right tangent point E. We thus get $\mathrm{N}_h \leqslant \mathrm{E}_h$.

We now "transfer" the graph in Figure 7-b back to Figure 7-a. Intuitively, one can imagine that Figure 7-b is folded along the line FA; and then put on the broken-line OFB in Figure 7-a (after the obvious lengthening operation). As h is concave, in the η-u-h space the broken line NKM is obviously "above" curve ℓ, that is, the curve OAEB. So the plane KMN is above the convex hull of ℓ. It follows that line NE, the intersection line of the two planes KMN and $u = 0$, is above the curve OE (the one in Figure 7-a), the intersection of plane $u = 0$ and the convex hull of ℓ. Therefore, the maximum h-coordinate of points on line NE is larger than that of the curve OE. But we have already shown that $\mathrm{N}_h \leqslant \mathrm{E}_h$, so $\mathrm{E}_h = h(\theta^*/\tilde{\theta}^*)$ is larger than the maximum h-coordinate of points on the curve OE. The second inequality in Equation (50) now gets proved.

8. Infomax Is Not Proper for Optimizing Cost-Sensitive Risk or F-Score

In the introduction section, we pointed out that the Infomax principle is consistent with the learning target of minimizing the error rate. The reason is that both Fano's bound and Hellman's bound are monotonically increasing with the conditional entropy; so minimizing the conditional entropy normally results in lower error rate. The same phenomenon is also observed between conditional entropy and balanced error rate (see Figure 6-b). In this sense, Infomax is suitable also for minimizing the balanced error rate.

As for F-score, however, the lower bound on the maximum F-score, $\overline{\mathrm{FSC}}$, is an *increasing* function of conditional entropy, as is depected in Figure 8. This implies a counterintuitive situation. Usually with classification problems, as we decrease the conditional entropy, we can expect the worst case (measured by the maximum F-score) classification scenario to improve. With F-score, however, as we decrease entropy, the worst case F-score gets even worse, decreasing to zero when $H(\mathsf{y}|\mathsf{x})$ tends to zero. As we have briefly mentioned at the end of Section 4, the same non-intuitve scenario is observed for the upper bound on the minimum cost-sensitive risk—see, Figure 4-b. Moreover, from Figure 8 we see that $\overline{\mathrm{FSC}}$ may take any value between 0 and 1 when $H(\mathsf{y}|\mathsf{x})$ tends to zero. This also seems non-intuitive as $H(\mathsf{y}|\mathsf{x}) = 0$ means y is a deterministic function of x, for which the best classifier should have F-score 1.

In this section, we discuss the possible reasons of these inconsistencies through some simple examples. The first example is constructed to illustrate that for any given value of $\overline{\mathrm{FSC}}$, there are classification problems whose maximum F-score is $\overline{\mathrm{FSC}}$; but the conditional entropy $H(\mathsf{y}|\mathsf{x})$ can be arbitrary small.—If the maximum F-score of a problem is small, we would think of it as a difficult task, since no classifier would perform well (as measured by F-score) on it. Intuitively, this means the relationship between x and y is quite uncertain, hence the conditional entropy $H(\mathsf{y}|\mathsf{x})$ should be large. However, our first example shows that is is not necessarily the case.

Example 1 *Here the feature space consists only of two distinct vectors, say,* $\mathcal{X} = \{x^{(1)}, x^{(2)}\}$. *The joint distribution of* x *and* y *is given by*

$$[\Pr\{x^{(1)},0\},\ \Pr\{x^{(2)},0\},\ \Pr\{x^{(1)},1\},\ \Pr\{x^{(2)},1\}] = [a,\ b,\ 0,\ \pi]\,,$$

where a, b, π *are positive numbers with sum* 1.

For this task there are four different classifiers which can be encoded naturally as 00, 01, 10 and 11, according to the predicted label on $x^{(1)}$ and $x^{(2)}$. The F-score of these classifiers are $\mathrm{FSC}(00) = \mathrm{FSC}(10) = 0$, $\mathrm{FSC}(01) = \frac{2\pi}{2\pi+b}$ and $\mathrm{FSC}(11) = \frac{2\pi}{2\pi+a+b}$. The computation procedure is detailed in Table 3, where the top-left corner is the joint distribution of x and y. It thus follows that $\overline{\mathrm{FSC}} = \mathrm{FSC}(01) = \frac{2\pi}{2\pi+b}$. The (Shannon) conditional entropy of this task is calculated as

$$H(\mathsf{y}|\mathsf{x}) = H(\mathsf{xy}) - H(\mathsf{x}) = (b+\pi)\log(b+\pi) - b\log b - \pi\log\pi.$$

Now let $\lambda = \frac{b}{\pi}$, then the above $\overline{\mathrm{FSC}}$ and $H(\mathsf{y}|\mathsf{x})$ can be written respectively as

$$\overline{\mathrm{FSC}} = \tfrac{2}{2+\lambda}\,, \quad H(\mathsf{y}|\mathsf{x}) = \pi\cdot[(\lambda+1)\log(\lambda+1) - \lambda\log\lambda] =: \pi\cdot f(\lambda)\,.$$

In the above computation, we have factorized $H(\mathsf{y}|\mathsf{x})$ as the product of two terms. The first term π describes the imbalance between the two classes; the second term $f(\lambda)$ is an increasing function

$\mathsf{y}=$	0	1	$\hat{y}(\mathsf{x})$	$\hat{y}(\mathsf{x})$	$\hat{y}(\mathsf{x})$	$\hat{y}(\mathsf{x})$
$\mathsf{x}=x^{(1)}$	a	0	0	0	1	1
$\mathsf{x}=x^{(2)}$	b	π	0	1	0	1
		TP	0	π	0	π
		TN	$a+b$	a	b	0
		FN	π	0	π	0
		FP	0	b	a	$a+b$
		FSC	0	$\frac{2\pi}{2\pi+b}$	0	$\frac{2\pi}{2\pi+a+b}$

Table 3: Computing the maximum F-score for Example 1

of the ratio $\lambda = \frac{b}{\pi} = \frac{\Pr(0|x_2)}{\Pr(1|x_2)}$, which reflects the uncertainty of the task. In particular, for any fixed $\overline{\text{FSC}}$ or λ, $H(\mathsf{y}|\mathsf{x})$ can be arbitrarily small if we let $\pi \to 0$. This certainly does not hint the involved problem is deterministic.

Next, we will see a more instructive example. It can be seen as the "dual" of Example 1, in that now the value of $\overline{\text{FSC}}$ is shown to be variable (by tuning a free parameter) with the conditional entropy being fixed.

Example 2 *In this example, the feature space consists of three vectors,* $\mathcal{X} = \{x^{(1)}, x^{(2)}, x^{(3)}\}$. *The marginal distribution of* $\mathsf{x} \in \mathcal{X}$ *(i.e., the probability measure* μ*, see Table 1) is given by*

$$[\Pr\{x^{(1)}\},\ \Pr\{x^{(2)}\},\ \Pr\{x^{(3)}\}] = [a,\ b,\ c]\,,$$

where a, b, c *are positive numbers with sum* 1. *The conditional probability of class* 1 *is denoted as* $\eta_i = \Pr\{\mathsf{y}=1 \mid \mathsf{x}=x^{(i)}\}$ *for* $i = 1,2,3$*; and set to be* $[\eta_1, \eta_2, \eta_3] = [0.5, 0, 1]$.

According to Definition 2, the conditional entropy of the above task can be written as

$$H(\mathsf{y}|\mathsf{x}) = a \cdot h(\eta_1) + b \cdot h(\eta_2) + c \cdot h(\eta_3) = a\,,$$

since for the binary entropy function it holds that $h(0.5) = 1$ and $h(0) = h(1) = 0$. There are eight different classifiers though, we actually need only to compute the F-score for two of them to get the maximum F-score. This is because the object $x^{(2)}$ should be classified as negative and $x^{(3)}$ as positive for sure, by any F-score-maximizing classifier—see, the results of Section 6. Only the classification of $x^{(1)}$ is unclear; so we calculate the F-score of the two classifiers 001 and 101. Letting $\eta_1 = 0.5$ in Table 4 we get $\text{FSC}(001) = \frac{2c}{2c+0.5a}$ and $\text{FSC}(101) = \frac{2c+a}{2c+1.5a}$. It is clear that $\text{FSC}(001) \leqslant \text{FSC}(101)$, therefore

$$\overline{\text{FSC}} = \text{FSC}(101) = \frac{2c+a}{2c+1.5a}\,.$$

Although the example is very simple, it does reveal quite a few insights into the notions of conditional entropy and F-score. First of all, for any given value of $H(\mathsf{y}|\mathsf{x}) = a$, one can freely adjust the value of the maximum F-score by tuning the parameter c. In more detail, $\overline{\text{FSC}}$ is an increasing function of c. As $0 \leqslant c \leqslant 1-a$, it is easy to see that $\overline{\text{FSC}}$ ranges from $\frac{2}{3}$ (at $c=0$) to $\frac{2-a}{2-0.5a}$ (at $c = 1-a$) for the particular problem considered here.

Secondly, the quantity b does not present in the expressions of $H(\mathsf{y}|\mathsf{x})$ and $\overline{\text{FSC}}$. In general, based on the conditional probability $\eta(x) = \Pr\{\mathsf{y}=1 \mid \mathsf{x}=x\}$, we can classify the objects (feature

$\mathsf{y}=$	0	1	$\hat{y}(\mathsf{x})$	$\hat{y}(\mathsf{x})$
$\mathsf{x}=x^{(1)}$	$\tilde{\eta}_1 a$	$\eta_1 a$	0	1
$\mathsf{x}=x^{(2)}$	b	0	0	0
$\mathsf{x}=x^{(3)}$	0	c	1	1
		TP	c	$\eta_1 a + c$
		TN	$\tilde{\eta}_1 a + b$	b
		FN	$\eta_1 a$	0
		FP	0	$\tilde{\eta}_1 a$
		FSC	$\frac{2c}{2c+\eta_1 a}$	$\frac{2c+2\eta_1 a}{2c+(1+\eta_1)a}$

Table 4: Computing the maximum F-score for Example 2

vectors) in a given task into three catrgories, namely, those belong surely to the positive ($\eta(x)=1$) or the negative ($\eta(x)=0$) class and those might be in either class ($0<\eta(x)<1$). The proportion of the three types are denoted here as c, b, and a, respectively. Then from the example we see that the conditional entropy $H(\mathsf{y}|\mathsf{x})$ is independent of the "certain" objects (due to the fact that $h(0)=h(1)=0$). In other words, it measures purely the amount of "uncertainty" for a classification task, which includes two factors, a and $h(\eta_1)$. The former factor represents the "population" of uncertain objects; and the latter represents the (average) degree of uncertainty of these objects.

On the other hand, the maximum F-score depends on the uncertain objects and the positive objects; but not on the negative objects. This is because the definition of F-score, Equation (17), does not take the true negative term, TN, into account. Consequently, classifiers aiming to maximize the F-score would intend to classify objects as positive, as this will increase the true positive and so increase the F-score—it will decrease the true negative at the same time, which however is not captured by F-score. This phenomenon is also reflected in the expression of the optimal classifier, $\hat{y}(x)=[\![\eta(x)>\theta^*]\!]$ (cf. the last paragraph of Section 6). Here the threshold θ^* is determined by the condition $\mathrm{FSC}(\theta^*)=2\theta^*$, which, as has been explained at the end of Section 6, is below 0.5. So an object would be regarded as positive even if the conditional probability of the positive class is less than half.

Finally, in this example the minimum value of $\overline{\mathrm{FSC}}$ is $\frac{2}{3}$ (the horizontal line through the point C in Figure 8), far from the lower bound. This is due to that we have set $\eta_1=0.5$; by using a lower value for η_1, we can in principle hit the lower bound curve AC in Figure 8. For instance, in the next example, we will see a setup with $\overline{\mathrm{FSC}}=0.625<\frac{2}{3}$.

8.1 On Information-Theoretic Feature Filtering Methods

Feature selection is a key step when dealing with high-dimensional data; it aims to find useful features and discard others, hence reduces the dimensionality. There are three major categories of feature selection techniques (Guyon and Elisseeff, 2003). *Embedded* methods (Lal et al., 2006) exploit the structure of specific classes of classifiers to guide the feature selection process. *Wrapper* methods (Kohavi and John, 1997) search the space of feature subsets, using the training/validation performance of a particular classifier to measure the utility of a candidate subset. These two are classifier-dependent, with the disadvantage of a considerable computational load, and may produce subsets that are overly specific to the classifiers used. In contrast, *filter* methods (Duch, 2006) separate the classification and feature selection components, and select features using a heuristic

scoring criterion that measures how potentially useful a feature or feature subset may be when used in a classifier.

Information-theoretic feature filters use an information measure (usually the mutual information between the selected features and class label) as the scoring criterion. The idea behind is that features showing maximum mutual information with class label are usually most useful for predicting the class label. This is well justified when the (balanced) error rate is concerned, as we have argued earlier. In this section, however, we will illustrate, using a simple example, that feature selection methods based on mutual information may fail to choose the optimal features when the classification performance is measured by F-score or cost-sensitive risk. Here we assume the perfect classifier, that is, the classifier with maximum F-score or minimum cost-sensitive risk, can be derived once the feature subset is determined.

Example 3 *In this example, we assume the objects are described by two features,* x_1 *and* x_2*, both of which take three distinct values. That is,* $\mathsf{x}_i \in \{x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\}$ *for* $i = 1, 2$*. The joint distribution of* x_1*,* x_2*, and* y *are set to be*

$\mathsf{y}=0$	$\mathsf{x}_2 = x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$
$\mathsf{x}_1 = x_1^{(1)}$	0.33	0	0
$x_1^{(2)}$	0.174	0.1	0
$x_1^{(3)}$	0	0	0

$\mathsf{y}=1$	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$
$x_1^{(1)}$	0.15	0	0.18
$x_1^{(2)}$	0	0	0
$x_1^{(3)}$	0.066	0	0

The target here is to select one feature to predict the class label.

As we can see here, by selecting either feature we are actually comparing two different problems that are described respectively by the distribution of the pairs $(\mathsf{x}_1, \mathsf{y})$ and $(\mathsf{x}_2, \mathsf{y})$. So we compute the two distributions from the given joint distribution of $(\mathsf{x}_1, \mathsf{x}_2, \mathsf{y})$, which gives us

$$\Pr\{\mathsf{x}_1, \mathsf{y}\} = \begin{bmatrix} 0.33 & 0.33 \\ 0.274 & 0 \\ 0 & 0.066 \end{bmatrix}; \quad \Pr\{\mathsf{x}_2, \mathsf{y}\} = \begin{bmatrix} 0.504 & 0.216 \\ 0.1 & 0 \\ 0 & 0.18 \end{bmatrix}. \tag{51}$$

Both $\Pr\{\mathsf{x}_1, \mathsf{y}\}$ and $\Pr\{\mathsf{x}_2, \mathsf{y}\}$ are structurally similar to the one in Example 2, with the parameters $[a, b, c; \eta_1] = [0.66, 0.274, 0.066; 0.5]$ and $[0.72, 0.1, 0.18; 0.3]$ respectively. So we can reuse the computation there to obtain the Shannon conditional entropy

$$\begin{aligned} H(\mathsf{y}|\mathsf{x}_1) &= a \cdot h(\eta_1) = 0.66 \cdot h(0.5) = 0.66, \\ H(\mathsf{y}|\mathsf{x}_2) &= a \cdot h(\eta_1) = 0.72 \cdot h(0.3) = 0.6345. \end{aligned}$$

Thus, according to the Infomax principle, the second feature x_2 should be selected as the class label predictor.

However, the maximum F-score of the two problems tells us a different story. For $(\mathsf{x}_1, \mathsf{y})$, we already have (see Example 2)

$$\overline{\mathrm{FSC}} = \mathrm{FSC}(101) = \frac{2c + a}{2c + 1.5a} = \frac{2 \times 0.066 + 0.66}{2 \times 0.066 + 1.5 \times 0.66} = 0.7059. \tag{52}$$

For $(\mathsf{x}_2, \mathsf{y})$, we need to compare the F-score of the classifiers 001 and 101. It follows from Table 4 that

$$\begin{aligned}\mathrm{FSC}(001) &= \frac{2c}{2c+\eta_1 a} = \frac{2\times 0.18}{2\times 0.18+0.3\times 0.72} = 0.625\,,\\ \mathrm{FSC}(101) &= \frac{2c+2\eta_1 a}{2c+(1+\eta_1)a} = \frac{2\times 0.18+0.6\times 0.72}{2\times 0.18+1.3\times 0.72} = 0.6111\,.\end{aligned}$$

Thus, $\overline{\mathrm{FSC}} = \mathrm{FSC}(001) = 0.625$, which is less than that of $(\mathsf{x}_1, \mathsf{y})$ at 0.7059 in Equation (52). This reveals that while the feature x_2 is selected by Infomax, it is in fact possible to design a better classifier (as measured by F-score) using the first feature x_1. The constructed problem shows that to minimize error rate and balanced error rate we should pick feature x_2; whereas to minimize cost-sensitive risk we should pick a different feature, x_1.

We now examine the minimum cost-sensitive risk of the two problems $(\mathsf{x}_1, \mathsf{y})$ and $(\mathsf{x}_2, \mathsf{y})$. Assume the cost of a false negative is $c_1 = 2.5$ and that of a false positive is $c_0 = 0.625$. If the feature x_1 is used, then, by Equation (51), the optimal classifier is 101 (which produces a false positive on $x^{(1)}$ with probability 0.33); and the corresponding minimum cost-sensitive risk is $\underline{\mathrm{CSR}} = \mathrm{CSR}(101) = 0.625\times 0.33 = 0.2063$. When the feature x_2 is selected, we compute $\mathrm{CSR}(001) = 2.5\times 0.216 = 0.54$ and $\mathrm{CSR}(101) = 0.625\times 0.504 = 0.315$. Thus, $\underline{\mathrm{CSR}} = \mathrm{CSR}(101) = 0.315$. It thus follows that choosing the feature x_1 would (potentially) obtain a lower cost-sensitive risk 0.2063, contradicting the selection suggested by Infomax.

On the other hand, the minimum (balanced) error rate of the problem $(\mathsf{x}_1, \mathsf{y})$ is

$$\underline{\mathrm{ERR}} = 0.33\,, \quad \underline{\mathrm{BER}} = \tfrac{0.33}{2\times(0.33+0.274)} = 0.2732\,.$$

For both criteria, the optimal classifier is 101. For the problem $(\mathsf{x}_2, \mathsf{y})$, we have

$$\underline{\mathrm{ERR}} = 0.216\,, \quad \underline{\mathrm{BER}} = \tfrac{0.216}{2\times(0.216+0.18)} = 0.2727\,,$$

with both minima obtained at the classifier 001. Therefore, selecting x_2 will do better than x_1 as to minimize the (balanced) error rate, in agreement with Infomax.

8.2 Towards Proper Information Measures for Cost-Sensitive Risk

In the preceding section, we constructed an example demonstrating that Shannon's mutual information is generally not a proper criterion for feature selection when the cost-sensitive risk or F-score is concerned. A natural question one would immediately raise is *what is the proper information measure for the two criteria then?* So far, this problem is not completely solved; and we only have partial solution.

In Section 4 we derived the tight lower and upper bounds on the conditional entropy $H(\mathsf{y}|\mathsf{x})$ in terms of the minimum cost-sensitive risk $\underline{\mathrm{CSR}}$ (see Figure 4-b); and noticed that the upper bound is not an increasing function of $\underline{\mathrm{CSR}}$. On the other hand, as we have emphasized several times, it is the monotonicity of Fano's and Hellman's bounds that justifies the Infomax principle. This motivates us to construct concave functions $h(\eta)$ such that the conditional entropy $H(\mathsf{y}|\mathsf{x})$ as defined in Definition 2 has lower and upper bounds that are monotonically increasing with respect to $\underline{\mathrm{CSR}}$.

As we can see in Figure 4, the curve OCDF plays an important role in determining the lower and upper bounds on $H(\mathsf{y}|\mathsf{x})$. It is obtained from the graph of the function $h(\eta)$ by a simple piecewise

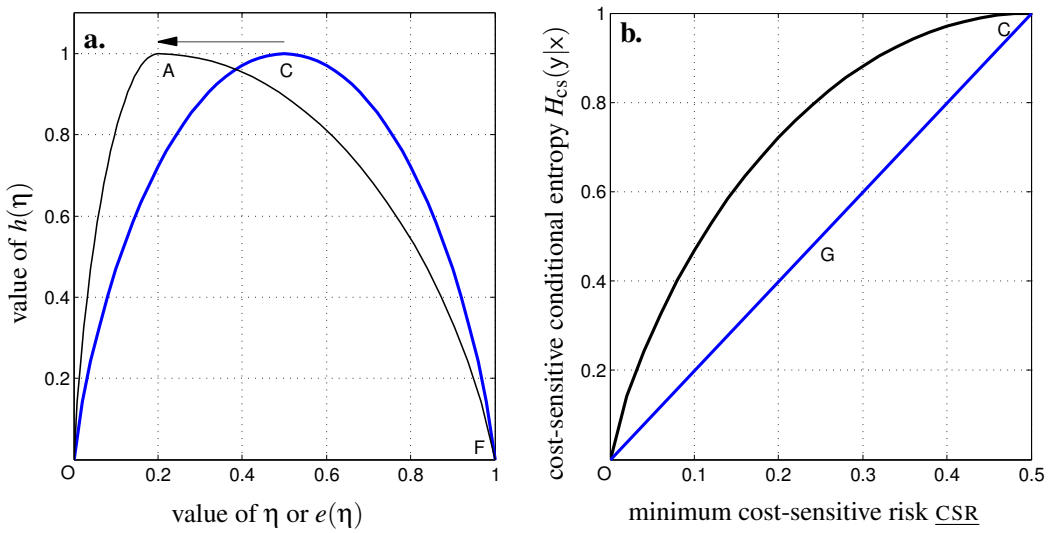


Figure 9: **a.** To get a symmetric concave curve OCF after the transform as indicated by the arrow AC in Figure 4-a, we apply an inverse transform to the target curve OCF. That is, we move the peak point C back to the point A with the η-coordinate $\frac{1}{2c_1}$, yielding the curve OAF that represents the function $h(\eta)$ in the definition of conditional entropy—see Equation (21). **b.** The lower (the line OGC) and upper (the curve OC) bounds on $H_{cs}(y|x)$ are obtained from the curve OCF in Figure 9-a using the same transform as in Figure 4. Both bounds are now monotonically increasing functions of CSR. This can be contrasted with the standard conditional entropy bounds in Figure 4-b.

linear transformation on the input η. Its left part OC determines the lower bound; and its right part CDF (after flipping along the central vertical line) corresponds to the upper bound. Thus, if we can construct a concave function $h(\eta)$ that makes the curve OCDF symmetric (for example, coincide with the curve OABF), then the lower and upper bounds on $H(y|x)$ would be very similar to Fano's and Hellman's, respectively. As shown in Figure 9, this can be done by applying a piecewise linear transform to the input variable of a symmetric concave function, so that the peak point C on its graph is moved left to the point A with the first coordinate $\frac{1}{2c_1}$.

Denote by $g(\eta)$, $\eta \in [0,1]$, the function corresponding to the curve OCF in Figure 9-a. Then the curve OAF is described by the function[13]

$$h(\eta) = \begin{cases} g(c_1\eta) & \text{if } \eta \in [0, \frac{1}{2c_1}), \\ g(1-c_0\tilde{\eta}) & \text{if } \eta \in [\frac{1}{2c_1}, 1], \end{cases} \tag{53}$$

13. This can be easily verified by checking the value of $h(\eta)$ at $\eta = 0, \frac{1}{2c_1}, 1$, which should be $g(0)$, $g(\frac{1}{2})$ and $g(1)$, respectively.

where the costs c_0 and c_1 satisfy the conditions $c_1 \geqslant c_0$ and $c_0^{-1} + c_1^{-1} = 2$. In particular, when $g(\eta) = h_{\text{bin}}(\eta) = -\eta\log\eta - (1-\eta)\log(1-\eta)$ (Shannon), we have

$$h(\eta) = \begin{cases} -(c_1\eta)\log(c_1\eta) - (1-c_1\eta)\log(1-c_1\eta) = h_{\text{bin}}(c_1\eta) & \text{if } \eta \in [0, \frac{1}{2c_1}), \\ -(c_0\tilde{\eta})\log(c_0\tilde{\eta}) - (1-c_0\tilde{\eta})\log(1-c_0\tilde{\eta}) = h_{\text{bin}}(c_0\tilde{\eta}) & \text{if } \eta \in [\frac{1}{2c_1}, 1]. \end{cases} \tag{54}$$

Substituting the above expression into Equation (21), we get a new definition of conditional entropy which we call the *cost-sensitive conditional entropy* and denote as $H_{\text{cs}}(\mathsf{y}|\mathsf{x})$. That is, $H_{\text{cs}}(\mathsf{y}|\mathsf{x}) := \mathbb{E}_{\mathsf{x}\sim\mu}[h(\eta(\mathsf{x}))]$, with $h(\eta)$ defined as above.

We now compute the value of $H_{\text{cs}}(\mathsf{y}|\mathsf{x})$ for the two classification tasks $(\mathsf{x}_1, \mathsf{y})$ and $(\mathsf{x}_2, \mathsf{y})$ as defined in Example 3, under the settings of $c_0 = 0.625$, $c_1 = 0.25$. By Equation (51), we know that the marginal distribution of x_1 is $\Pr\{\mathsf{x}_1 = x_1^{(1,2,3)}\} = [0.66, 0.274, 0.066]$; the conditional probability of the positive class is $\eta_{1,2,3} = \Pr\{\mathsf{y} = 1 \mid \mathsf{x}_1 = x_1^{(1,2,3)}\} = [0.5, 0, 1]$. By Equation (54), we have $h(\eta_1) = h_{\text{bin}}(c_0\tilde{\eta}_1) = h_{\text{bin}}(0.625 \cdot 0.5) = 0.6211$; $h(\eta_2) = h_{\text{bin}}(c_1\eta_2) = h_{\text{bin}}(0) = 0$; and $h(\eta_3) = h_{\text{bin}}(c_0\tilde{\eta}_3) = h_{\text{bin}}(0) = 0$. Thus,

$$H_{\text{cs}}(\mathsf{y}|\mathsf{x}_1) = \Pr\{\mathsf{x}_1 = x_1^{(1)}\} \cdot h(\eta_1) = 0.66 \times 0.6211 = 0.4099.$$

Similarly, for the feature x_2, we have $\eta_{1,2,3} = \Pr\{\mathsf{y} = 1 \mid \mathsf{x}_2 = x_2^{(1,2,3)}\} = [0.3, 0, 1]$ and $\Pr\{\mathsf{x}_2 = x_2^{(1,2,3)}\} = [0.72, 0.1, 0.18]$. Thus, $h(\eta_1) = h_{\text{bin}}(c_0\tilde{\eta}_1) = h_{\text{bin}}(0.625 \cdot 0.7) = 0.6853$ and $h(\eta_2) = h(\eta_3) = 0$. We thus get

$$H_{\text{cs}}(\mathsf{y}|\mathsf{x}_2) = \Pr\{\mathsf{x}_2 = x_2^{(1)}\} \cdot h(\eta_1) = 0.72 \times 0.6853 = 0.4934.$$

Since $H_{\text{cs}}(\mathsf{y}|\mathsf{x}_1) < H_{\text{cs}}(\mathsf{y}|\mathsf{x}_2)$, the feature x_1 would be selected according to the cost-sensitive conditional entropy. This coincides with the decision we previously obtained by directly comparing the value of $\underline{\text{CSR}}$.

In conclusion, Shannon's mutual information or conditional entropy is not a proper surrogate learning objective in dealing with a cost-sensitive situation or when the subsequent classification process is assessed by the metric of F-score. Conversely, we have proven the positive result that Shannon's information is appropriate for balanced error rate. For cost-sensitive risk minimization problems, we suggest to use a cost-sensitive variant of normal symmetric conditional entropies as defined by Equation (53). As far as the authors know, this definition of conditional entropy has not been studied in the context of feature selection. The work by Elkan (2001) might be the closest to ours, where he investigated the possibility of adapting a given learning algorithm to the cost-sensitive situation by simply adjusting the prior probability of each class (whereas here we intend to change the posterior probabilities). For F-score maximization, we have not found a proper information measure so far.

9. Conclusion and Future Work

Inspired by the widespread use of Fano's inequality in machine learning—in particular, in feature selection, the paper has extended Fano's and Hellman's bounds (on error probability) to the bounds on other commonly used criteria including balanced error rate, F-score and cost-sensitive risk. To this end, we developed a general geometric method which enables us to derive the tight bounds on the above mentioned criteria using a general definition of conditional entropy (see Definition

2), in a uniform way. These bounds are presented in three main theorems of the paper: Theorems 7, 11 and 15. Our work extends previous knowledge on the relationship between classification performance criteria and conditional entropy (Ben-Bassat, 1978; Golic, 1987; Feder and Merhav, 1994; Erdogmus and Principe, 2004).

The advantage of the proposed geometric approach is clear: it provides a visible and intuitive insight into the relationship between the concerned criteria and information measures. Moreover, defining the conditional entropy through a general concave function $h(\eta)$ in fact gives us much more than what we have stated so far. For example, let $h(\eta) = \min\{\eta, \tilde{\eta}\}$ in Theorem 15, we immediately get the bounds on the Bayes error rate in terms of maximum F-score: $0 \leqslant \underline{\mathrm{ERR}} \leqslant \min\{\frac{\overline{\mathrm{FSC}}}{2-\overline{\mathrm{FSC}}}, 1 - \frac{\overline{\mathrm{FSC}}}{2-\overline{\mathrm{FSC}}}\}$.

When deriving the bounds on the maximum F-score and the minimum cost-sensitive risk, some new findings were noticed, which, interestingly, might be of more interest than the bounds themselves. Firstly, as a by-product of the bounds on the maximum F-score, $\overline{\mathrm{FSC}}$, in Section 6 we proved that the optimal classifiers for maximizing the F-score have the form $\hat{y}(x) = [\![\eta(x) > \theta]\!]$. This property is called *the probability thresholding principle for binary classifications* by Lewis (1995); and has been proved by Lewis (1995) and Jansche (2007) independently for finite input spaces $\mathcal{X}$. Here we presented a proof for the general case where $\mathcal{X}$ is an arbitrary set, which, to the best of our knowledge, is novel.

The most important new finding in the paper is that the Infomax principle based on standard information measures could be misleading when F-score or cost-sensitive risk is used as the performance measure. We illustrated this by analytical argument and a simple example in the field of feature selection. For cost-sensitive risk, we proposed an alternative information measure, whose usefulness is justified by the same example (and by the monotonicity of the resulting bounds, see Figure 9-b). To summarize,

> *Shannon's conditional entropy is* ***not*** *a proper criterion for feature selection when the subsequent classification process is measured by F-score or cost-sensitive risk. Instead, we suggest to use a cost-sensitive variant as defined by Equation* (53).

A corresponding measure for F-score is left as an open problem for further research. This is a challenging question due to the fact that F-score is defined on the *whole* object space, whereas information measures are usually defined through the conditional probabilities on *single* objects, $\Pr\{\mathsf{y} = 1 \mid \mathsf{x} = x\}$. To find a proper information measure for the F-score maximization problem is a research topic in our group to be pursued in the future. As the presented bounds hold only for binary problems, extending them to the multi-class problem is also a topic of interest in the group.

We finish the paper with an important remark. The paper is theory-oriented; it is concerned with *problems*, not with *classifiers* or *algorithms*. More precisely, while the performance of a classifier could be measured by error rate, balanced error rate, F-score or cost-sensitive risk, their optimum value over *all* classifiers can be seen as different difficulty measures of the concerned problem. On the other hand, the conditional entropy $H(\mathsf{y}|\mathsf{x})$ measures the amount of uncertainty about the class label remaining after we have observed the object. Thus, it can also be seen as a difficulty measure of classification tasks. From this perspective, in this paper we are examining the relationship between two different types of difficulty measures of classification problems. Our main finding is that Shannon's conditional entropy as a difficulty measure is inconsistent with the maximum F-score and the minimum cost-sensitive risk. This fact has serious implications in the field of feature selection, as we have discussed in Section 8.

Acknowledgments

This research was conducted with support from the UK Engineering and Physical Sciences Research Council. Ming-Jie Zhao and Narayanan Edakunni were supported by grants EP/F023855/1, EP/G000662/1, and EP/G013500/1, and Adam Pocock was supported on an EPSRC studentship from EP/G000662/1.

Appendix A. Bounds on the Bayes Error Rate: the Multi-Class Case

We have already mentioned in the introduction section the main work in the literature that are related to ours. These are all about bounding the Bayes error rate by means of different conditional entropies. This section briefly introduces a unifying derivation of these bounds, based on the work of Tebbe and Dwyer III (1968), Ben-Bassat (1978) and Golic (1987). As Fano's bound and others' actually hold for the *multi-class* problem, we need to extend the notations introduced in Section 2 to catch up with the multi-class case.[14] These new notations are used only in this section and not listed in Table 1.

Assume there are m classes which are labeled by the integers 1 to m. Then a classifier can be written as a function $\hat{y}(x)$ on $\mathcal{X}$ that takes values in the set $\{1,\dots,m\}$. Similar to the binary case, we decompose the joint distribution of (x,y) as the product of the marginal distribution of x and the conditional distribution of y given x. As such, the definition of $\mu(A)$ is unchanged, see Equation (2). But the quantity $\eta(x)$ is now replaced by an m-dimensional vector $\boldsymbol{\eta}(x) = [\eta_1(x),\dots,\eta_m(x)]$, with

$$\eta_y(x) := \Pr\{\mathsf{y} = y \mid \mathsf{x} = x\}, \qquad \forall x \in \mathcal{X},\ \forall y \in \{1,\dots,m\}.$$

By the above definition we see that the elements of $\boldsymbol{\eta}(x)$ are non-negative and sum to 1. Such vectors are called *probability vectors* in statistics. We shall denote by $\mathcal{P}_m$ the set of probability vectors of dimension m, which is also known as the *probability simplex* in $\mathbb{R}^m$:

$$\mathcal{P}_m := \{\boldsymbol{\eta} \in \mathbb{R}^m \mid \eta_y \geqslant 0 \text{ for all } y = 1,\dots,m;\ \text{and } \textstyle\sum_{y=1}^m \eta_y = 1\}.$$

In terms of μ and $\boldsymbol{\eta}$, the joint distribution of (x,y) can be written as

$$\Pr\{\mathsf{x} \in A, \mathsf{y} = y\} = \textstyle\int_A \eta_y(x)\mathrm{d}\mu, \quad \forall A \subseteq \mathcal{X} \text{ measurable}, \forall y \in \{1,\dots,m\}.$$

Letting $A = \mathcal{X}$ in the above formula, we get the (prior) probability of each class,

$$\pi_y := \Pr\{\mathsf{y} = y\} = \textstyle\int_{\mathcal{X}} \eta_y(x)\mathrm{d}\mu, \qquad \forall y \in \{1,\dots,m\}.$$

Note that the vector of class probabilities, $\boldsymbol{\pi} := [\pi_1,\dots,\pi_m]$, is also a probability vector.

As before, we call the pair $(\mu,\boldsymbol{\eta})$ a (classification) task, whose conditional entropy is defined as follows.

Definition 19 *Let $h : \mathcal{P}_m \to \mathbb{R}$ be a symmetric concave function—the word "symmetric" refers to that, for any $\boldsymbol{\eta} = [\eta_1,\dots,\eta_m] \in \mathcal{P}_m$ and any permutation $(i_1,\dots,i_m)$ of $\{1,\dots,m\}$, it holds that $h(\eta_1,\dots,\eta_m) = h(\eta_{i_1},\dots,\eta_{i_m})$. The conditional entropy of a task $(\mu,\boldsymbol{\eta})$ is*

$$H(\mathsf{y}|\mathsf{x}) := \textstyle\int_{\mathcal{X}} h(\boldsymbol{\eta}(x))\mathrm{d}\mu = \mathbb{E}_{\mathsf{x}\sim\mu}[h(\boldsymbol{\eta}(\mathsf{x}))].$$

14. Only in this section we discuss the multi-class problem; the rest of the paper is devoted to the binary case.

In particular, letting $h(\boldsymbol{\eta}) = -\sum_{i=1}^{m} \eta_i \log \eta_i$, we get the Shannon conditional entropy.

For any classification task $(\mu, \boldsymbol{\eta})$, the (expected) error rate of a given classifier $\hat{y} : \mathcal{X} \to \{1, \ldots, m\}$ can be computed as

$$\begin{aligned} \text{ERR} &= \Pr\{\mathsf{y} \neq \hat{y}(\mathsf{x})\} \\ &= 1 - \Pr\{\mathsf{y} = \hat{y}(\mathsf{x})\} \\ &= 1 - \textstyle\sum_{y=1}^{m} \Pr\{\mathsf{y} = y, \hat{y}(\mathsf{x}) = y\} \\ &= 1 - \textstyle\sum_{y=1}^{m} \int_{\mathcal{X}_y} \eta_y(x) \mathrm{d}\mu, \end{aligned}$$

where $\mathcal{X}_y$ are subsets of the feature space $\mathcal{X}$ determined by the classifier $\hat{y}(x)$ via $\mathcal{X}_y := \{x \in \mathcal{X} \mid \hat{y}(x) = y\}$, for $y = 1, \ldots, m$. Therefore, the error rate is minimized when $\eta_y(x)$ is the maximum element in the whole vector $\boldsymbol{\eta}(x)$ on the set $\mathcal{X}_y$. That is,

$$\underline{\text{ERR}} = 1 - \textstyle\sum_{y=1}^{m} \int_{\mathcal{X}_y} \max\{\boldsymbol{\eta}(x)\} \mathrm{d}\mu = 1 - \int_{\mathcal{X}} \max\{\boldsymbol{\eta}(x)\} \mathrm{d}\mu = \mathbb{E}_{\mathsf{x}\sim\mu}[1 - \max\{\boldsymbol{\eta}(\mathsf{x})\}],$$

where, for any vector $\boldsymbol{\eta}$, $\max\{\boldsymbol{\eta}\}$ denotes its maximum entry.

Following the line presented in Section 3, here we need to examine the range of the point $[\underline{\text{ERR}}, H(\mathsf{y}|\mathsf{x})] = \mathbb{E}_{\mathsf{x}\sim\mu}[e(\boldsymbol{\eta}(\mathsf{x})), h(\boldsymbol{\eta}(\mathsf{x}))]$ in the error rate versus conditional entropy plane. Here the function $e(\boldsymbol{\eta})$ is defined as $e(\boldsymbol{\eta}) = 1 - \max\{\boldsymbol{\eta}\}$; for the binary case, this becomes $e(\eta) = \min\{\eta, \tilde{\eta}\}$. So the problem is now reduced to finding the convex hull of the set $\{[e(\boldsymbol{\eta}), h(\boldsymbol{\eta})] \mid \boldsymbol{\eta} \in \mathcal{P}_m\}$, which further amounts to computing the extreme values of $h(\boldsymbol{\eta})$ given that $e(\boldsymbol{\eta})$ is fixed.

Lemma 20 *Let $h : \mathcal{P}_m \to \mathbb{R}$ be a symmetric concave function. Let $\boldsymbol{\eta} \in \mathcal{P}_m$ be such that $e(\boldsymbol{\eta}) = r$. Then $h(\boldsymbol{\eta})$ is maximized when one element of $\boldsymbol{\eta}$ equals to $1 - r$ and the others are all $\frac{r}{m-1}$; and it is minimized when all entries of $\boldsymbol{\eta}$ are either $1 - r$ or 0, except one whose value is determined by the condition that $\boldsymbol{\eta}$ has element sum 1.*

In particular, for the function $h(\boldsymbol{\eta}) = -\sum_{i=1}^{m} \eta_i \log \eta_i$, we have

$$\begin{aligned} h_{\max}(r) &:= \max_{\boldsymbol{\eta} \in \mathcal{P}_m, e(\boldsymbol{\eta}) = r} h(\boldsymbol{\eta}) = -(1-r) \cdot \log(1-r) - r \cdot \log\left(\tfrac{r}{m-1}\right), \\ h_{\min}(r) &:= \min_{\boldsymbol{\eta} \in \mathcal{P}_m, e(\boldsymbol{\eta}) = r} h(\boldsymbol{\eta}) = -k \cdot (1-r) \cdot \log(1-r) - \beta \cdot \log \beta, \end{aligned}$$

where k is the maximum integer such that $k \cdot (1-r) \leqslant 1$ and $\beta = 1 - k \cdot (1-r)$.

The graphs of $h_{\max}(r)$ and $h_{\min}(r)$ are plotted in Figure 10 for the case of $m = 5$ classes. Notice that for $m = 5$, $\max\{\boldsymbol{\eta}\} \geqslant 0.2$, so the range of $r = e(\boldsymbol{\eta}) = 1 - \max\{\boldsymbol{\eta}\}$ is $[0, 0.8]$. From the figure we see that while $h_{\max}(r)$ is a smooth function, the curve of $h_{\min}(r)$ consists of $m - 1 = 4$ segments, connected by the endpoints A, B, and C. The r-coordinate of these endpoints are determined by the condition $k \cdot (1-r) = 1$, for $k = 1, \ldots, m$ corresponding to the points O, A, …, D, respectively.

By definition, the region between the curves of $h_{\max}(r)$ and $h_{\min}(r)$ is exactly the set $\{[e(\boldsymbol{\eta}), h(\boldsymbol{\eta})] \mid \boldsymbol{\eta} \in \mathcal{P}_m\}$. Moreover, it can be proven that $h_{\max}(r)$ is concave and $h_{\min}(r)$ concave within each segment; their graph also shows this. Therefore, the convex hull of $\{[e(\boldsymbol{\eta}), h(\boldsymbol{\eta})] \mid \boldsymbol{\eta} \in \mathcal{P}_m\}$ is bounded by the curve OD and the broken line OABCD, which represent the *tight* lower (Fano) and upper (Tebbe) bounds on the Bayes error rate, $\underline{\text{ERR}}$, in terms of the conditional entropy, $H(\mathsf{y}|\mathsf{x})$. Furthermore, the broken line OABCD forms a convex function, so it is lower bounded by its most left segment OA. This actually gives us the Hellman inequality, $\underline{\text{ERR}} \leqslant \frac{1}{2} H(\mathsf{y}|\mathsf{x})$.

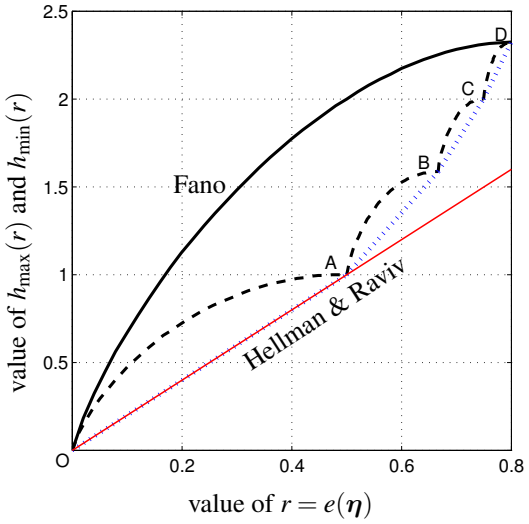


Figure 10: Graphs of the functions $h_{\max}(r)$ (solid line) and $h_{\min}(r)$ (dashed line) for the $m = 5$ class problem. From the two curves we obtain the tight lower bound (Fano), the tight upper bound (Tebbe, the dotted broken line OABCD) on the Bayes error rate, and the Hellman bound.

The above derivation is simple and elegant. However, it can not be directly applied to the case of balanced error rate (for multi-class problems). The main difficulty is that now we have an extra condition on the posterior probabilities $\boldsymbol{\eta}(x)$, namely, its integral over the space $\mathcal{X}$ should be equal to the class probability $\boldsymbol{\pi}$. So the problem of bounding balanced error rate actually amounts to

$$\begin{array}{ll} \min \text{ or } \max & H(\mathsf{y}|\mathsf{x}) = \int_{\mathcal{X}} h(\boldsymbol{\eta}(x))\mathrm{d}\mu \\ \text{subject to} & \int_{\mathcal{X}} \boldsymbol{\eta}(x)\mathrm{d}\mu = \boldsymbol{\pi}\,, \\ & \frac{1}{m}\int_{\mathcal{X}} \min_{y=1,\ldots,m}\{\pi_y^{-1}\eta_y(x)\}\mathrm{d}\mu = \underline{\mathrm{BER}}\,, \end{array}$$

which is a very difficult optimization problem, even for the case of $m = 2$.

In this paper, we restrict ourselves to the binary case (so the vector $\boldsymbol{\eta}(x)$ can be represented by a scalar $\eta(x)$, as is shown in the paper) and consider the reachable region of the 3-dimensional point $[\eta, r(\eta), h(\eta)]$—where $r(\eta) = \min\{\pi^{-1}\eta, \tilde{\pi}^{-1}\tilde{\eta}\}$, rather than the 2-dimensional point $[r(\eta), h(\eta)]$. As such, we avoid solving the above optimization problem, which is extremely hard as we can tell.

Finally, we should also mention that extending our method to the multi-class case is also difficult, if not impossible. Because that will involve the reachable region of the high $(m+2)$ dimensional point $[\boldsymbol{\eta}, r(\boldsymbol{\eta}), h(\boldsymbol{\eta})]$ with $r(\boldsymbol{\eta}) := \min_{y=1,\ldots,m}\{\pi_y^{-1}\eta_y\}$.

Appendix B. Mathematical Foundation

In this section we discuss three fundamental points which support the content of the paper and provide a mathematically more rigorous foundation.[15]

15. We thank two anonymous reviewers for pointing out these imperfection to us.

1. The concept of a *true label* can be very messy in the commonly used setup for pattern classification which involves only the jointly-distributed random variables (x, y).

2. The convex hull of the curves ℓ in the paper are obtained heuristically, a more rigorous treatement is needed.

3. The geometric arguments in this paper are based on the proposition: *the expectation of a random vector* $\mathsf{u} \in \mathbb{R}^n$ *lies in the convex hull of its range*. While this is correct in intuition, it needs a mathematical proof.

As the above three problems make sense not only for the specific topic studied here, but also from a broader viewpoint of pattern classification and probability theory, we place the discussion of them in a separate section.

B.1 A Mathematical Definition of True Class Labels

To describe the classification problem in a mathematical framework, many textbooks on machine learning start off with a joint distribution of the feature vector x and the class label y. For example, in Devroye et al. (1996, Chapter 1) the authors wrote:

> ...More formally, an *observation* is a d-dimensional vector x. The unknown nature of the observation is called a *class*. It is denoted by y ...

and in Chapter 2 they continued with:

> ...The random pair (x, y) may be described in a variety of ways: for example, it is defined by the pair (μ, η), where μ is the probability measure for x and η is the regression of y on x. ...

While this treatment has the advantages of simplicity and ease to understand, it fails to capture some natural notions rised in real applications such as the *true label* of an object. It is also not a uniform framework in the sense that, whenever a new feature is added, we have to extend the vector x by one component and redefine the joint distribution.

We now introduce an alternative framework that allows for a clear definition of true class labels. The key idea is to distinguish between an object and the features describing it. For this, we denote by Ω the set of all objects ω[16] in the considered problem. In medical diagnosis, for example, this could be (the set of) all people in a country or an area. We can then define a σ-algebra $\mathscr{F}$ of subsets of Ω and a probability measure P on $\mathscr{F}$, yielding a probability space $(\Omega, \mathscr{F}, P)$. Note that, here $(\Omega, \mathscr{F}, P)$ serves only as a uniform base for discussion; and the concrete definition of $\mathscr{F}$ and P are not important.

According to the problem at hand, the set Ω is often naturally divided into several (measurable) subsets that are *pairwise disjoint*, say $\Omega = \Omega_0 \cup \Omega_1$. For example, Ω_1 may represent those people who are affected by a certain disease; and Ω_0 is the set of the others. This can be conveniently described by a function $\mathsf{y} : \Omega \to \{0, 1\}$ that sends each $\omega \in \Omega_i$ to the value i ($i = 0, 1$). Since the subsets Ω_i are pairwise disjoint and their union equals to Ω, the function y is well defined. *We can now define the value of* y *at a particular* ω *as the* true class label *of the object* ω.

16. More rigorously speaking, here ω is in fact the "name" of the object it represented.

One main goal in pattern classification is to predict the true class label $\mathsf{y}(\omega)$, for which we need to make some measurements on the object ω—obviously, the object *itself* cannot be used as a predictor here. The measuring procedure is also described as a *measurable* function $\mathsf{x}:\Omega\to\mathcal{X}$. For example, $\mathsf{x}(\omega)$ might be the vector of heart rate, body temperature and blood pressure of the person ω. The joint distribution of x and y is induced from the probability measure P, as follows: for any $A\subseteq\mathcal{X}$ measurable and $y\in\{0,1\}$,

$$\Pr\{\mathsf{x}\in A,\mathsf{y}=y\}:=P(\{\omega\in\Omega\mid\mathsf{x}(\omega)\in A,\,\mathsf{y}(\omega)=y\})\,.$$

Thus, every notion in the traditional framework can also be well defined in the new framework, but not vice versa.

The target of pattern classification is to design a classifier $\hat{y}:\mathcal{X}\to\{0,1\}$ so that some criterion is minimized or maximized. Since $\hat{y}$ is a function on $\mathcal{X}$ rather than Ω, for each $x\in\mathcal{X}$ it does not discriminate between the objects in the set $\Omega_x=\{\omega\in\Omega\mid\mathsf{x}(\omega)=x\}$, which may belong to different classes—that is, $\mathsf{y}(\omega)$ may not assume a constant value on Ω_x. So the best thing we can do is to choose the *best class label* (according to the concerned criterion) for each feature vector $x\in\mathcal{X}$; and assign it to *all* objects in the set Ω_x, regardless of their *true class label*.

To recapitulate, the concept of a true class label can only be defined for objects, not for feature vectors; so it is not well defined under the traditional probabilistic framework, which identify an object with its feature vector. At the feature vector level, the notion of best class labels can be defined; and its definition depends on the performance criterion used. We however had better keep using the term "true label" anyway, for otherwise some commonly used notions such as true positive and misclassification rate would cause even more confusion. Also, dropping this term will make the discussion in Elkan (2001) about "reasonableness" conditions of cost matrices (see also page 1040 of this paper) meaningless. For this reason, we have abused the notion of true labels in the paper even though the traditional framework is adopted. It actually should be understood as *the true class label of the particular object* ω *we are talking about whose feature vector is* x.

B.2 On the Convex Hull of a Given Set in $\mathbb{R}^m$

In this section, we propose a recursive procedure to construct the convex hull of a general subset in the Euclidean space $\mathbb{R}^m$, for which we introduce some basic terminologies first. A set $D\subseteq\mathbb{R}^m$ is said to be *convex* if $\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v}\in D$ for any $\boldsymbol{u},\boldsymbol{v}\in D$ and any $\alpha\in[0,1]$—recall that $\tilde{\alpha}:=1-\alpha$. Let D be a convex set in $\mathbb{R}^m$. A function $f:D\to\mathbb{R}$ is *convex* if for any $\boldsymbol{u},\boldsymbol{v}\in D$ and $\alpha\in[0,1]$, it holds that $f(\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v})\leqslant\alpha\cdot f(\boldsymbol{u})+\tilde{\alpha}\cdot f(\boldsymbol{v})$. If, instead, the reversed inequality holds, then f is a *concave* function. Notice that convex (concave) functions are defined on convex sets. For any $D\subseteq\mathbb{R}^m$, its *convex hull*, denoted as $\mathrm{co}\,D$ in the paper, is defined as the set of all (finite) convex combinations of points in D,

$$\mathrm{co}\,D:=\{\textstyle\sum_{i=1}^n\alpha_i\boldsymbol{u}_i\mid n\in\mathbb{N},\,\boldsymbol{u}_i\in D,\,\alpha_i\geqslant 0,\,\sum_{i=1}^n\alpha_i=1\}\,.$$

Another equivalent definition of $\mathrm{co}\,D$ is that it is the smallest convex set that contains D as a subset. Both definitions will be usee (in proving certain propositions). Furthermore, for any $\boldsymbol{u}\in\mathbb{R}^m$ we shall call the sum $\sum_{i=1}^n\alpha_i\boldsymbol{u}_i$ or the set $\{(\alpha_i,\boldsymbol{u}_i)\}_{i=1}^n$ a *convex decomposition of* $\boldsymbol{u}$ *in* D if it holds that $\alpha_i\geqslant 0$, $\sum_{i=1}^n\alpha_i=1$, $\boldsymbol{u}_i\in D$ and $\boldsymbol{u}=\sum_{i=1}^n\alpha_i\boldsymbol{u}_i$. Notice that a vector $\boldsymbol{u}\in\mathrm{co}\,D$ if and only if it has at least one convex decomposition in D.

Lemma 21 *The convex hull of any subset D of the real line $\mathbb{R}$ is an interval with the endpoints $a=\inf D$ and $b=\sup D$. Moreover, $a\in\mathrm{co}\,D$ iff $a\in D$ and $b\in\mathrm{co}\,D$ iff $b\in D$.*

Proof We only consider the case where $a \in D$ and $b \notin D$; the other three possibilities can be discussed analogously. It is clear that $D \subseteq [a,b)$ and that $[a,b)$ is a convex set. But $\mathrm{co}D$ is the smallest convex set that contains D, thus $\mathrm{co}D \subseteq [a,b)$. It now remains to show that $[a,b) \subseteq \mathrm{co}D$ (hence $\mathrm{co}D = [a,b)$ and we are done). For any $c \in [a,b)$, as $b = \sup D$, there exists a $t \in D$ such that $c < t < b$. Put $\alpha = \frac{t-c}{t-a}$, then $c = \alpha \cdot a + \tilde{\alpha} \cdot t \in \mathrm{co}D$. ■

Although simple, the above lemma characterizes completely the convex hull of subsets in the 1-dimensional space. For the high dimensional case, we need further to introduce some new symbols. For any $m \in \mathbb{N}$ and any $E \subseteq \mathbb{R}^{m+1}$, denote by $E^{\downarrow}$ the projection of E onto $\mathbb{R}^m$ (the subspace of $\mathbb{R}^{m+1}$ spanned by the first m unit vectors):

$$E^{\downarrow} := \{\boldsymbol{u} = [u_1, \ldots, u_m] \in \mathbb{R}^m \mid [\boldsymbol{u}, s] = [u_1, \ldots, u_m, s] \in E \text{ for some } s \in \mathbb{R}\}.$$

For each $\boldsymbol{u} \in \mathbb{R}^m$, we define $E^{\uparrow}_{\boldsymbol{u}} := \{s \in \mathbb{R} \mid [\boldsymbol{u}, s] \in E\}$. Intuitively, $E^{\uparrow}_{\boldsymbol{u}} \subseteq \mathbb{R}$ can be seen as the intersection of the set E and the real line "vertically" placed at the point $\boldsymbol{u}$. Observe that $E^{\uparrow}_{\boldsymbol{u}} \neq \varnothing$ iff $\boldsymbol{u} \in E^{\downarrow}$ and that $[\boldsymbol{u}, s] \in E$ iff $s \in E^{\uparrow}_{\boldsymbol{u}}$. Furthermore, with the notions of $E^{\downarrow}$ and $E^{\uparrow}_{\boldsymbol{u}}$, any $E \subseteq \mathbb{R}^{m+1}$ can be expressed as $E = \{[\boldsymbol{u}, s] \mid \boldsymbol{u} \in E^{\downarrow}, s \in E^{\uparrow}_{\boldsymbol{u}}\}$. In particular, replacing the set E by its convex hull in this identity, we obtain

$$\mathrm{co}E = \{[\boldsymbol{u}, s] \mid \boldsymbol{u} \in (\mathrm{co}E)^{\downarrow}, s \in (\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}\}. \tag{55}$$

Lemma 22 *For any $E \subseteq \mathbb{R}^{m+1}$, it holds that $(\mathrm{co}E)^{\downarrow} = \mathrm{co}E^{\downarrow}$.*

Proof The set $\mathrm{co}E$ is convex, so is its projection $(\mathrm{co}E)^{\downarrow}$—see, for example, Rockafellar (1970, p. 19, Corollary 3.4.1). Moreover, from $E \subseteq \mathrm{co}E$ we know $E^{\downarrow} \subseteq (\mathrm{co}E)^{\downarrow}$. It then follows from the minimality of $\mathrm{co}E^{\downarrow}$ that $\mathrm{co}E^{\downarrow} \subseteq (\mathrm{co}E)^{\downarrow}$. We now show that $(\mathrm{co}E)^{\downarrow} \subseteq \mathrm{co}E^{\downarrow}$. Let $\boldsymbol{u} \in (\mathrm{co}E)^{\downarrow}$, then $[\boldsymbol{u}, s] \in \mathrm{co}E$ for some $s \in \mathbb{R}$. Hence the vector $[\boldsymbol{u}, s]$ has a convex decomposition in E, say $[\boldsymbol{u}, s] = \sum_{i=1}^{n} \alpha_i \cdot [\boldsymbol{u}_i, s_i]$. It follows from $[\boldsymbol{u}_i, s_i] \in E$ that $\boldsymbol{u}_i \in E^{\downarrow}$ and so $\boldsymbol{u} = \sum_{i=1}^{n} \alpha_i \boldsymbol{u}_i \in \mathrm{co}E^{\downarrow}$. This proves $(\mathrm{co}E)^{\downarrow} \subseteq \mathrm{co}E^{\downarrow}$. ■

Lemma 22 links the convex hull of a set in $\mathbb{R}^{m+1}$ to that in $\mathbb{R}^m$ and hence simplifies the first ingredient of Equation (55), $(\mathrm{co}E)^{\downarrow}$. We now analyze its second ingredient, $(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$ with $\boldsymbol{u} \in (\mathrm{co}E)^{\downarrow} = \mathrm{co}E^{\downarrow}$. First of all, since $\mathrm{co}E$ is a convex set, so is $(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$. To see this, let $s_1, s_2 \in (\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$, then $[\boldsymbol{u}, s_1], [\boldsymbol{u}, s_2] \in \mathrm{co}E$. But $\mathrm{co}E$ is a convex set, so for any $\alpha \in [0,1]$ it holds that $\alpha[\boldsymbol{u}, s_1] + \tilde{\alpha}[\boldsymbol{u}, s_2] = [\boldsymbol{u}, \alpha s_1 + \tilde{\alpha} s_2] \in \mathrm{co}E$, that is, $\alpha s_1 + \tilde{\alpha} s_2 \in (\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$. Secondly, $(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$ is a subset of $\mathbb{R}$, it hence must be an interval—one should have no difficulty to see that every convex set in $\mathbb{R}$ is an interval. The problem is thus reduced to determining the two endpoints of the interval, that is, the infimum and supremum of $(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$, for which the following two symbols $\overline{g}(\cdot|\cdot)$ and $\underline{g}(\cdot|\cdot)$ are useful.

By the definition of $E^{\downarrow}$ and $E^{\uparrow}_{\boldsymbol{u}}$, it is obvious that $E^{\uparrow}_{\boldsymbol{u}}$ is nonempty for each $\boldsymbol{u} \in E^{\downarrow}$. Now assume that $E \subseteq \mathbb{R}^{m+1}$ is bounded, that is, $E \subseteq [-b, b]^{m+1}$ for some $b \in \mathbb{R}$, then the set $E^{\uparrow}_{\boldsymbol{u}}$ is also bounded—in fact, $E^{\uparrow}_{\boldsymbol{u}} \subseteq [-b, b]$ for any $\boldsymbol{u} \in E^{\downarrow}$. For such sets E, the functions

$$\overline{g}(\cdot|E) : E^{\downarrow} \to \mathbb{R}, \boldsymbol{u} \mapsto \sup E^{\uparrow}_{\boldsymbol{u}}, \qquad \underline{g}(\cdot|E) : E^{\downarrow} \to \mathbb{R}, \boldsymbol{u} \mapsto \inf E^{\uparrow}_{\boldsymbol{u}} \tag{56}$$

are well defined (and bounded). Note that for general sets $E \subseteq \mathbb{R}^{m+1}$, the above functions could be $\pm\infty$ at some points $\boldsymbol{u}$. So the boundness of the set E is necessary for $\overline{g}$ and $\underline{g}$ to be real-valued. The notation $\overline{g}(\cdot|\cdot)$ and $\underline{g}(\cdot|\cdot)$ allows us to rewrite the supremum of the set $(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$ as $\overline{g}(\boldsymbol{u}|\,\mathrm{co}E)$ and its infimum as $\underline{g}(\boldsymbol{u}|\,\mathrm{co}E)$. Here the functions $\overline{g}(\cdot|\,\mathrm{co}E)$ and $\underline{g}(\cdot|\,\mathrm{co}E)$ are also defined by Equation (56), but with the set E replaced by $\mathrm{co}E$. That is,

$$\begin{aligned}\overline{g}(\cdot|\,\mathrm{co}E) &: (\mathrm{co}E)^{\downarrow} \to \mathbb{R},\, \boldsymbol{u} \mapsto \sup(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}},\\ \underline{g}(\cdot|\,\mathrm{co}E) &: (\mathrm{co}E)^{\downarrow} \to \mathbb{R},\, \boldsymbol{u} \mapsto \inf(\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}.\end{aligned} \tag{57}$$

In the following we aim to relate the above two functions to $\overline{g}(\cdot|E)$ and $\underline{g}(\cdot|E)$.

Lemma 23 *Let $E \subseteq \mathbb{R}^{m+1}$ be a bounded convex set. Then $\underline{g}(\cdot|E)$ is a convex function and $\overline{g}(\cdot|E)$ a concave function on $E^{\downarrow}$.*

Proof Since E is convex, so is its projection $E^{\downarrow}$. Thus, $\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v} \in E^{\downarrow}$ for any $\boldsymbol{u},\boldsymbol{v} \in E^{\downarrow}$ and $\alpha \in [0,1]$. By the definition of $\underline{g}(\cdot|E)$, Equation (56), to prove its convexity we need to show

$$\inf E^{\uparrow}_{\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v}} \leqslant \alpha\cdot\inf E^{\uparrow}_{\boldsymbol{u}} + \tilde{\alpha}\cdot\inf E^{\uparrow}_{\boldsymbol{v}}. \tag{58}$$

For any $\varepsilon > 0$, by the definition of $E^{\uparrow}_{\boldsymbol{u}}$ we know there is an $s \in \mathbb{R}$ such that $[\boldsymbol{u},s] \in E$ and $s < \inf E^{\uparrow}_{\boldsymbol{u}} + \varepsilon$. Similarly, there exists a $t \in \mathbb{R}$ satisfying $[\boldsymbol{v},t] \in E$ and $t < \inf E^{\uparrow}_{\boldsymbol{v}} + \varepsilon$. Then $[\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v}, \alpha s+\tilde{\alpha}t] \in E$ since E is convex. This means that $\alpha s+\tilde{\alpha}t \in E^{\uparrow}_{\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v}}$ and so

$$\inf E^{\uparrow}_{\alpha\boldsymbol{u}+\tilde{\alpha}\boldsymbol{v}} \leqslant \alpha s+\tilde{\alpha}t < \alpha\cdot\inf E^{\uparrow}_{\boldsymbol{u}} + \tilde{\alpha}\cdot\inf E^{\uparrow}_{\boldsymbol{v}} + \varepsilon.$$

Since $\varepsilon > 0$ can be arbitrarily small, we get the desired inequality (58).

The concavity of the function $\overline{g}(\cdot|E)$ can be proven in the similar way. ■

By this lemma, we at once see that the function $\underline{g}(\cdot|\,\mathrm{co}E)$ is convex and $\overline{g}(\cdot|\,\mathrm{co}E)$ concave. Moreover, as $E \subseteq \mathrm{co}E$ and hence $E^{\uparrow}_{\boldsymbol{u}} \subseteq (\mathrm{co}E)^{\uparrow}_{\boldsymbol{u}}$ for any $\boldsymbol{u} \in \mathbb{R}^m$, we know from Equations (56)–(57) that

$$\underline{g}(\boldsymbol{u}|\,\mathrm{co}E) \leqslant \underline{g}(\boldsymbol{u}|E) \leqslant \overline{g}(\boldsymbol{u}|E) \leqslant \overline{g}(\boldsymbol{u}|\,\mathrm{co}E), \qquad \forall\, \boldsymbol{u} \in E^{\downarrow}.$$

Here the domain of $\overline{g}(\cdot|E)$ and $\underline{g}(\cdot|E)$, $E^{\downarrow}$, does not need to be convex; and the domain of $\overline{g}(\cdot|\,\mathrm{co}E)$ and $\underline{g}(\cdot|\,\mathrm{co}E)$, $(\mathrm{co}E)^{\downarrow} = \mathrm{co}E^{\downarrow}$, is the convex hull of the domain of $\overline{g}(\cdot|E)$ and $\underline{g}(\cdot|E)$. These observations motivate us to introduce the concepts of the convex/concave hull of functions defined on a subset of $\mathbb{R}^m$ which is not necessarily convex.

Let $D \subseteq \mathbb{R}^m$ and $f : D \to \mathbb{R}$. The *concave hull* of f is the smallest concave function $f^{\frown} : \mathrm{co}D \to \mathbb{R}$ such that $f^{\frown}(\boldsymbol{u}) \geqslant f(\boldsymbol{u})$ for all $\boldsymbol{u} \in D$; and the *convex hull* of f is the greatest convex function $f_{\smile} : \mathrm{co}D \to \mathbb{R}$ with $f_{\smile}(\boldsymbol{u}) \leqslant f(\boldsymbol{u})$ for $\boldsymbol{u} \in D$. In particular, if the domain D is itself a convex set, then $\mathrm{co}D = D$ and our definition of $f_{\smile}$ and $f^{\frown}$ degenerates into the standard definition. Here both $f_{\smile}$ and $f^{\frown}$ are required to be real-valued. As such, some functions might have no convex or concave hull. For instance, the function $f(t) = t^2$, $t \in \mathbb{R}$ does not have concave hull—it would be $f^{\frown}(t) = \infty$ if the extended real line is considered instead of $\mathbb{R}$.

Lemma 24 *For any bounded subset $E \subseteq \mathbb{R}^{m+1}$, the function $\underline{g}(\cdot|\operatorname{co}E)$ is the convex hull of $\underline{g}(\cdot|E)$; and $\overline{g}(\cdot|\operatorname{co}E)$ is the concave hull of $\overline{g}(\cdot|E)$.*

Proof We have shown that $\underline{g}(\cdot|\operatorname{co}E) : \operatorname{co}E^{\downarrow} \to \mathbb{R}$ is a convex function which for any $\boldsymbol{u} \in E^{\downarrow}$ satisfies $\underline{g}(\boldsymbol{u}|\operatorname{co}E) \leqslant \underline{g}(\boldsymbol{u}|E)$. It thus remains to show that $\underline{g}(\cdot|\operatorname{co}E) \geqslant f(\cdot)$ for any convex function $f : \operatorname{co}E^{\downarrow} \to \mathbb{R}$ satisfying the same condition.

Let $\boldsymbol{u} \in \operatorname{co}E^{\downarrow}$, by definition, $\underline{g}(\boldsymbol{u}|\operatorname{co}E) = \inf(\operatorname{co}E)^{\uparrow}_{\boldsymbol{u}}$. Thus, for any $\varepsilon > 0$, there is an $s \in (\operatorname{co}E)^{\uparrow}_{\boldsymbol{u}}$ such that $s < \underline{g}(\boldsymbol{u}|\operatorname{co}E) + \varepsilon$. By $s \in (\operatorname{co}E)^{\uparrow}_{\boldsymbol{u}}$ we know $[\boldsymbol{u}, s] \in \operatorname{co}E$, so it has a convex decomposition in E, say $[\boldsymbol{u}, s] = \sum_{i=1}^{n} \alpha_i \cdot [\boldsymbol{u}_i, s_i]$. It follows from $[\boldsymbol{u}_i, s_i] \in E$ that $s_i \in E^{\uparrow}_{\boldsymbol{u}_i}$ and hence $\underline{g}(\boldsymbol{u}_i|E) = \inf E^{\uparrow}_{\boldsymbol{u}_i} \leqslant s_i$. Since $f : \operatorname{co}E^{\downarrow} \to \mathbb{R}$ is a convex function and since $f(\cdot) \leqslant \underline{g}(\cdot|E)$ on $E^{\downarrow}$, by Jensen's inequality we have

$$\begin{aligned} f(\boldsymbol{u}) = f(\textstyle\sum_{i=1}^{n} \alpha_i \boldsymbol{u}_i) \leqslant \sum_{i=1}^{n} \alpha_i \cdot f(\boldsymbol{u}_i) \leqslant \sum_{i=1}^{n} \alpha_i \cdot \underline{g}(\boldsymbol{u}_i|E) \quad \dots \\ \dots \quad \leqslant \textstyle\sum_{i=1}^{n} \alpha_i s_i = s < \underline{g}(\boldsymbol{u}|\operatorname{co}E) + \varepsilon. \end{aligned}$$

As $\varepsilon > 0$ can be arbitrarily small, the above inequality implies $f(\boldsymbol{u}) \leqslant \underline{g}(\boldsymbol{u}|\operatorname{co}E)$. We thus have proved that $\underline{g}(\cdot|\operatorname{co}E)$ is the convex hull of $\underline{g}(\cdot|E)$. By the similar argument, we can prove $\overline{g}(\cdot|\operatorname{co}E)$ is the concave hull of $\overline{g}(\cdot|E)$. ■

Lemma 25 *Let $D \subset \mathbb{R}^m$ be an arbitrary set. Then any lower (upper) bounded function $f : D \to \mathbb{R}$ allows for a convex (concave) hull $f_{\smile}(f^{\frown}) : \operatorname{co}D \to \mathbb{R}$.*

Proof On the set $\operatorname{co}D$ define two functions $f^*(\boldsymbol{u})$ and $f_*(\boldsymbol{u})$ by

$$f^*(\boldsymbol{u}) := \sup\{\textstyle\sum_{i=1}^{n} \alpha_i \cdot f(\boldsymbol{u}_i) \mid \{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^{n} \text{ a conv. decomp. of } \boldsymbol{u} \text{ in } D\}, \tag{59}$$

$$f_*(\boldsymbol{u}) := \inf\{\textstyle\sum_{i=1}^{n} \alpha_i \cdot f(\boldsymbol{u}_i) \mid \{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^{n} \text{ a conv. decomp. of } \boldsymbol{u} \text{ in } D\}. \tag{60}$$

As any $\boldsymbol{u} \in \operatorname{co}D$ allows for at least one convex decomposition in D, the above set $\{\sum \dots\}$ is nonempty and hence its supremum and infimum are well defined. We claim that $f_{\smile} = f_*$ when f is lower bounded and that $f^{\frown} = f^*$ when f is upper bounded.

By the definition of $f_{\smile}$, to see that $f_{\smile} = f_*$ it suffices to show

(a) $f_*(\cdot)$ *is a convex function on* $\operatorname{co}D$: Let $\boldsymbol{u}, \boldsymbol{v} \in \operatorname{co}D$ and $t \in [0,1]$, we need to prove $f_*(t\boldsymbol{u} + \tilde{t}\boldsymbol{v}) \leqslant t \cdot f_*(\boldsymbol{u}) + \tilde{t} \cdot f_*(\boldsymbol{v})$. For any $\varepsilon > 0$, by the definition of $f_*(\boldsymbol{u})$, there is a convex decomposition of $\boldsymbol{u}$ in D, $\{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^{n}$, such that $f_*(\boldsymbol{u}) > \sum_{i=1}^{n} \alpha_i \cdot f(\boldsymbol{u}_i) - \varepsilon$. Analogously, $f_*(\boldsymbol{v}) > \sum_{i=1}^{k} \beta_i \cdot f(\boldsymbol{v}_i) - \varepsilon$ for some convex decomposition of $\boldsymbol{v}$, $\{(\beta_i, \boldsymbol{v}_i)\}_{i=1}^{k}$. We thus get

$$t \cdot f_*(\boldsymbol{u}) + \tilde{t} \cdot f_*(\boldsymbol{v}) > \textstyle\sum_{i=1}^{n} t\alpha_i \cdot f(\boldsymbol{u}_i) + \sum_{i=1}^{k} \tilde{t}\beta_i \cdot f(\boldsymbol{v}_i) - \varepsilon.$$

But the set $\{(t\alpha_i, \boldsymbol{u}_i)\}_{i=1}^{n} \cup \{(\tilde{t}\beta_i, \boldsymbol{v}_i)\}_{i=1}^{k}$ forms a convex decomposition of $t\boldsymbol{u} + \tilde{t}\boldsymbol{v}$, so

$$f_*(t\boldsymbol{u} + \tilde{t}\boldsymbol{v}) \leqslant \textstyle\sum_{i=1}^{n} t\alpha_i \cdot f(\boldsymbol{u}_i) + \sum_{i=1}^{k} \tilde{t}\beta_i \cdot f(\boldsymbol{v}_i).$$

It hence follows that $f_*(t\boldsymbol{u} + \tilde{t}\boldsymbol{v}) < t \cdot f_*(\boldsymbol{u}) + \tilde{t} \cdot f_*(\boldsymbol{v}) + \varepsilon$. Since $\varepsilon > 0$ can be arbitrarily small, we conclude that $f_*(t\boldsymbol{u} + \tilde{t}\boldsymbol{v}) \leqslant t \cdot f_*(\boldsymbol{u}) + \tilde{t} \cdot f_*(\boldsymbol{v})$.

(b) $f_*(\boldsymbol{u}) \leqslant f(\boldsymbol{u})$ *for all* $\boldsymbol{u} \in D$: This is obvious as $\{(1,\boldsymbol{u})\}$ is a convex decomposition of $\boldsymbol{u}$ in D.

(c) $g(\boldsymbol{u}) \leqslant f_*(\boldsymbol{u})$ *for any* $g : \mathrm{co}D \to \mathbb{R}$ *satisfying the above conditions (a) and (b), and any* $\boldsymbol{u} \in \mathrm{co}D$: For any $\varepsilon > 0$, by the definition of $f_*(\boldsymbol{u})$, there is a convex decomposition of $\boldsymbol{u}$ in D, $\{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^n$, such that $f_*(\boldsymbol{u}) > \sum_{i=1}^n \alpha_i \cdot f(\boldsymbol{u}_i) - \varepsilon$. As $g(\cdot)$ is a convex function, and as $g \leqslant f$ on D, by Jensen's inequality we have

$$\textstyle\sum_{i=1}^n \alpha_i \cdot f(\boldsymbol{u}_i) \geqslant \sum_{i=1}^n \alpha_i \cdot g(\boldsymbol{u}_i) \geqslant g(\sum_{i=1}^n \alpha_i \boldsymbol{u}_i) = g(\boldsymbol{u}),$$

where the last equality follows from that $\{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^n$ is a convex decomposition of $\boldsymbol{u}$. We thus know $f_*(\boldsymbol{u}) + \varepsilon > g(\boldsymbol{u})$ and so $f_*(\boldsymbol{u}) \geqslant g(\boldsymbol{u})$, since $\varepsilon > 0$ can be arbitrarily small.

By the similar argument, one shows that $f^\frown = f^*$ for upper bounded functions f. ■

The above two lemmas enable us to describe the functions $\overline{g}(\cdot|\mathrm{co}E)$ and $\underline{g}(\cdot|\mathrm{co}E)$ in terms of $\overline{g}(\cdot|E)$ and $\underline{g}(\cdot|E)$, respectively. In fact, by putting $f(\cdot) = \overline{g}(\cdot|E)$ in Equation (59) and $f(\cdot) = \underline{g}(\cdot|E)$ in Equation (60), we get

$$\overline{g}(\boldsymbol{u}|\mathrm{co}E) = \sup\{\textstyle\sum_{i=1}^n \alpha_i \overline{g}(\boldsymbol{u}_i|E) \mid \{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^n \text{ a conv. decomp. of } \boldsymbol{u} \text{ in } E^\downarrow\}, \tag{61}$$

$$\underline{g}(\boldsymbol{u}|\mathrm{co}E) = \inf\{\textstyle\sum_{i=1}^n \alpha_i \underline{g}(\boldsymbol{u}_i|E) \mid \{(\alpha_i, \boldsymbol{u}_i)\}_{i=1}^n \text{ a conv. decomp. of } \boldsymbol{u} \text{ in } E^\downarrow\}. \tag{62}$$

Now let us return to the expression (55), $\mathrm{co}E = \{[\boldsymbol{u}, s] \mid \boldsymbol{u} \in (\mathrm{co}E)^\downarrow, s \in (\mathrm{co}E)^\uparrow_{\boldsymbol{u}}\}$. As has been pointed out earlier, for any $\boldsymbol{u} \in (\mathrm{co}E)^\downarrow = \mathrm{co}E^\downarrow$, the set $(\mathrm{co}E)^\uparrow_{\boldsymbol{u}}$ is an interval in $\mathbb{R}$ with the two endpoints $\overline{g}(\boldsymbol{u}|\mathrm{co}E)$, $\underline{g}(\boldsymbol{u}|\mathrm{co}E)$ determined respectively by Equation (61) and Equation (62). This interval might be open, closed, or half-open-half-closed, depending on whether or not the respective endpoint is in the interval. For simplicity we restrict ourselves to bounded and closed sets E. Then their convex hull $\mathrm{co}E$ are also bounded and closed—see, for example, Aliprantis and Border (2006, p. 185, Corollary 5.33), which in turn implies the set $(\mathrm{co}E)^\uparrow_{\boldsymbol{u}}$ can only be a closed interval, $(\mathrm{co}E)^\uparrow_{\boldsymbol{u}} = [\underline{g}(\boldsymbol{u}|\mathrm{co}E), \overline{g}(\boldsymbol{u}|\mathrm{co}E)]$. Equation (55) can thus be rewritten as

$$\mathrm{co}E = \{[\boldsymbol{u}, s] \mid \boldsymbol{u} \in \mathrm{co}E^\downarrow,\ \underline{g}(\boldsymbol{u}|\mathrm{co}E) \leqslant s \leqslant \overline{g}(\boldsymbol{u}|\mathrm{co}E)\}. \tag{63}$$

The projection $E^\downarrow$ of a bounded closed set E is also bounded and closed, so the above expression of $\mathrm{co}E$ gives naturally rise to a recursive algorithm to construct the convex hull of any bounded and closed set E, as follows. To get $\mathrm{co}E$ we need only to find $\mathrm{co}E^\downarrow$ and the functions $\underline{g}(\cdot|\mathrm{co}E)$ and $\overline{g}(\cdot|\mathrm{co}E)$ as given by Equations (56), (61) and (62); to get $\mathrm{co}E^\downarrow$ we need to find $\mathrm{co}E^{\downarrow\downarrow}$ and the functions $\underline{g}(\cdot|\mathrm{co}E^\downarrow)$ and $\overline{g}(\cdot|\mathrm{co}E^\downarrow)$; and so forth. As $E^\downarrow \subseteq \mathbb{R}^m$ for any $E \subseteq \mathbb{R}^{m+1}$, this procedure terminates with the 1-dimensional case after m steps, which has been fully discussed in Lemma 21.

B.3 The Convex Hull of Three Curves ℓ in the Paper

We now apply the recursive procedure presented in the preceding section to three curves occurred in the paper, to get their convex hull. These curves have been parameterized by the posterior probability $\eta \in [0,1]$, as listed below—to distinguish, a subscript is used to indicate the quantity with

which the curve is associated:

$$\ell_{\underline{\mathrm{CSR}}} = \{[e(\eta), h(\eta)] \mid \eta \in [0,1]\}, \qquad e(\eta) = \min\{c_1\eta, c_0\tilde{\eta}\}; \tag{64}$$

$$\ell_{\underline{\mathrm{BER}}} = \{[\eta, r(\eta), h(\eta)] \mid \eta \in [0,1]\}, \qquad r(\eta) = \min\{\pi^{-1}\eta, \tilde{\pi}^{-1}\tilde{\eta}\}; \tag{65}$$

$$\ell_{\overline{\mathrm{FSC}}} = \{[\eta, u(\eta), h(\eta)] \mid \eta \in [0,1]\}, \qquad u(\eta) = \theta^*\eta - (\eta - \theta^*)^+ \text{ and } \theta^* = \tfrac{1}{2}\overline{\mathrm{FSC}}. \tag{66}$$

In the above, the function $h : [0,1] \to \mathbb{R}$ is concave and satisfies $h(0) = h(1) = 0$.

B.3.1 THE CONVEX HULL OF $\ell_{\underline{\mathrm{CSR}}}$

The curve $\ell_{\underline{\mathrm{CSR}}}$ lies in the *e*-*h* plane; and, by Equation (63), its convex hull can be expressed as

$$\operatorname{co}\ell_{\underline{\mathrm{CSR}}} = \{[e_0, h_0] \mid e_0 \in \operatorname{co}\ell^{\downarrow}_{\underline{\mathrm{CSR}}},\ \underline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}}) \leqslant h_0 \leqslant \overline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}})\}. \tag{67}$$

As $c_0^{-1} + c_1^{-1} = 2$, the range of $e(\eta)$ is $[0, 0.5]$—see the analysis in page 1048. We thus have $\ell^{\downarrow}_{\underline{\mathrm{CSR}}} = \{e(\eta) \mid \eta \in [0,1]\} = [0, 0.5]$ and hence $\operatorname{co}\ell^{\downarrow}_{\underline{\mathrm{CSR}}} = [0, 0.5]$. For each $e_0 \in [0, 0.5]$, the set $(\ell_{\underline{\mathrm{CSR}}})^{\uparrow}_{e_0}$ is computed as follows: $(\ell_{\underline{\mathrm{CSR}}})^{\uparrow}_{e_0} = \{h_0 \in \mathbb{R} \mid [e_0, h_0] \in \ell_{\underline{\mathrm{CSR}}}\} = \{h(\eta) \mid e(\eta) = e_0\}$. But $e(\eta) = e_0$ implies $\eta = c_1^{-1}e_0$ or $\eta = 1 - c_0^{-1}e_0$, so $(\ell_{\underline{\mathrm{CSR}}})^{\uparrow}_{e_0} = \{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}$. It then follows from Equation (56) that

$$\overline{g}(e_0|\ell_{\underline{\mathrm{CSR}}}) = \max\{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}, \qquad \forall e_0 \in [0, 0.5];$$

$$\underline{g}(e_0|\ell_{\underline{\mathrm{CSR}}}) = \min\{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}, \qquad \forall e_0 \in [0, 0.5].$$

By Lemma 24, we know $\overline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}})$ is the concave hull of $\overline{g}(e_0|\ell_{\underline{\mathrm{CSR}}})$ and $\underline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}})$ the convex hull of $\underline{g}(e_0|\ell_{\underline{\mathrm{CSR}}})$, that is,

$$\overline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}}) = [\max\{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}]^{\frown},$$

$$\underline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}}) = [\min\{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}]_{\smile} = 2 \cdot h(\tfrac{1}{2c_1}) \cdot e_0,$$

where the last equality holds because both $h(c_1^{-1}e_0)$ and $h(1 - c_0^{-1}e_0)$ are concave functions of e_0 and they have the same endpoints: $[0,0]$ and $[\frac{1}{2}, h(\frac{1}{2c_1})]$. Substituting the identity $\operatorname{co}\ell^{\downarrow}_{\underline{\mathrm{CSR}}} = [0, 0.5]$ and the above expressions of $\overline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}})$ and $\underline{g}(e_0|\operatorname{co}\ell_{\underline{\mathrm{CSR}}})$ into Equation (67), we obtain

$$\operatorname{co}\ell_{\underline{\mathrm{CSR}}} = \{[e_0, h_0] \mid e_0 \in [0, 0.5],\ 2e_0 \cdot h(\tfrac{1}{2c_1}) \leqslant h_0 \leqslant [\ldots]^{\frown}\}, \tag{68}$$

where the expression in the brackets $[\ldots]$ is $\max\{h(c_1^{-1}e_0), h(1 - c_0^{-1}e_0)\}$.

B.3.2 THE CONVEX HULL OF $\ell_{\underline{\mathrm{BER}}}$ AND $\ell_{\overline{\mathrm{FSC}}}$

The curves $\ell_{\underline{\mathrm{BER}}}$ and $\ell_{\overline{\mathrm{FSC}}}$ are of the same nature: both $r(\eta)$ and $u(\eta)$ are piecewise affine functions whose graph consists of two line segments. They can hence be treated together. By the definition of $r(\eta)$ and $u(\eta)$, we have

$$r(\eta) = \begin{cases} \pi^{-1}\eta & \text{if } \eta \leqslant \pi \\ \tilde{\pi}^{-1}\tilde{\eta} & \text{otherwise} \end{cases}, \qquad u(\eta) = \begin{cases} \theta^*\eta & \text{if } \eta \leqslant \theta^* \\ \theta^* - \tilde{\theta}^*\eta & \text{otherwise} \end{cases}. \tag{69}$$

The graph of the two functions for $\pi = \theta^* = 0.3$ are shown in Figure 11; they also represent the curves $\ell^{\downarrow}_{\underline{\mathrm{BER}}} = \{[\eta, r(\eta)] \mid \eta \in [0,1]\}$ and $\ell^{\downarrow}_{\overline{\mathrm{FSC}}} = \{[\eta, u(\eta)] \mid \eta \in [0,1]\}$, respectively. As both $r(\eta)$

and $u(\eta)$ are concave functions, by Equation (63) it is easy to see that (the detailed derivation is just a routine work and omitted here)

$$\begin{aligned}\operatorname{co}\ell^{\downarrow}_{\underline{\mathrm{BER}}} &= \{[\eta, r_0] \mid \eta \in [0,1],\, 0 \leqslant r_0 \leqslant r(\eta)\}\,,\\ \operatorname{co}\ell^{\downarrow}_{\overline{\mathrm{FSC}}} &= \{[\eta, u_0] \mid \eta \in [0,1],\, (\theta^* - \tilde{\theta}^*)\eta \leqslant u_0 \leqslant u(\eta)\}\,.\end{aligned} \tag{70}$$

This is also clear from Figure 11: they are just the triangles OAB.

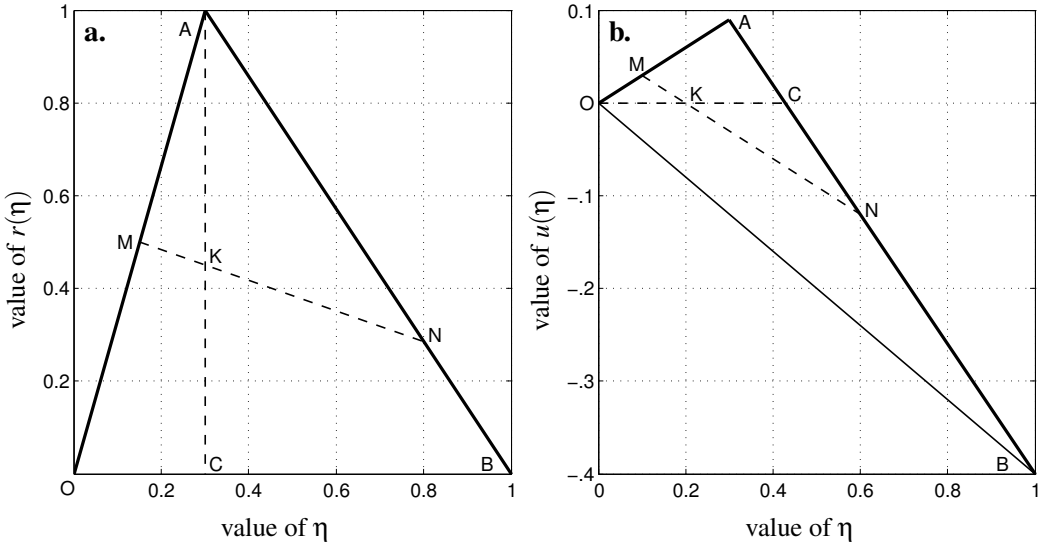


Figure 11: **a.** The graph of the function $r(\eta)$ (the solid line OAB), which can also be expressed as $\ell^{\downarrow}_{\underline{\mathrm{BER}}} = \{[\eta, r(\eta)] \mid \eta \in [0,1]\}$. From the graph one easily sees that the convex hull of $\ell^{\downarrow}_{\underline{\mathrm{BER}}}$ is the area bounded by the triangle OAB.
b. The same graph and curve ($\ell^{\downarrow}_{\underline{\mathrm{CSR}}}$) for the function $u(\eta)$.

As before, to derive the convex hull of the curve $\ell_{\overline{\mathrm{FSC}}}$, we use Equation (63) and obtain

$$\operatorname{co}\ell_{\overline{\mathrm{FSC}}} = \{[\eta, u_0, h_0] \mid [\eta, u_0] \in \operatorname{co}\ell^{\downarrow}_{\overline{\mathrm{FSC}}},\, \underline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant h_0 \leqslant \overline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}})\}. \tag{71}$$

The set $\operatorname{co}\ell^{\downarrow}_{\overline{\mathrm{FSC}}}$ is already known, so it remains to find the expressions of $\overline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}})$ and $\underline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}})$, for which we need first to determine the values of $\overline{g}(\eta, u_0 | \ell_{\overline{\mathrm{FSC}}})$ and $\underline{g}(\eta, u_0 | \ell_{\overline{\mathrm{FSC}}})$ for $[\eta, u_0] \in \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$—see Equation (61) and Equation (62). By the definition of $\ell_{\overline{\mathrm{FSC}}}$, we know $[\eta, u_0] \in \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$ if and only if $u_0 = u(\eta)$; and $(\ell_{\overline{\mathrm{FSC}}})^{\uparrow}_{[\eta, u_0]} = \{h(\eta)\}$ for any $[\eta, u_0] \in \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$. It thus follows from Equation (56) that $\overline{g}(\eta, u_0 | \ell_{\overline{\mathrm{FSC}}}) = \underline{g}(\eta, u_0 | \ell_{\overline{\mathrm{FSC}}}) = h(\eta)$ for any point $[\eta, u_0] \in \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$, that is, for any $\eta \in [0,1]$ and $u_0 = u(\eta)$.

Based upon the above discussion and Equations (61) and (62), we have

$$\overline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}}) = \sup\{\textstyle\sum_{i=1}^{n} \alpha_i \cdot h(\eta_i) \mid \text{condition on } (\alpha_i, \eta_i)\}\,, \tag{72}$$

$$\underline{g}(\eta, u_0 | \operatorname{co}\ell_{\overline{\mathrm{FSC}}}) = \inf\{\textstyle\sum_{i=1}^{n} \alpha_i \cdot h(\eta_i) \mid \text{condition on } (\alpha_i, \eta_i)\}\,, \tag{73}$$

for any $[\eta, u_0] \in \operatorname{co} \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$, where the unspecified condition is that $\{(\alpha_i, \eta_i, u(\eta_i))\}_{i=1}^n$ forms a convex decomposition of the point $[\eta, u_0]$ in $\ell^{\downarrow}_{\overline{\mathrm{FSC}}}$. In other words, here the parameters $\alpha_i, \eta_i \in [0,1]$ should satisfy $\sum_{i=1}^n \alpha_i = 1$, $\sum_{i=1}^n \alpha_i \eta_i = \eta$ and $\sum_{i=1}^n \alpha_i \cdot u(\eta_i) = u_0$. Next we shall prove that $n \leqslant 2$ when the supremum in Equation (72) is obtained; and that the infimum in Equation (73) is attained at $n \leqslant 3$ with $\eta_i \in \{0, \theta^*, 1\}$.

We discuss Equation (73) first. For each η_i in a convex decomposition $\{(\alpha_i, \eta_i, u(\eta_i))\}_{i=1}^n$ of $[\eta, u_0]$, $\eta_i \in [0, \theta^*]$ or $\eta_i \in [\theta^*, 1]$. For the former case, we "split up" the item $(\alpha_i, \eta_i, u(\eta_i))$ into two items at $\eta = 0$ and θ^*, with the α-parameter computed from the condition that the weighted sum of the new items equals to the original one. That is, we construct $(\alpha_i^1, \eta_i^1, u(\eta_i^1))$ and $(\alpha_i^2, \eta_i^2, u(\eta_i^2))$ with $\eta_i^1 = 0$ and $\eta_i^2 = \theta^*$, such that

$$\alpha_i^1 + \alpha_i^2 = \alpha_i\,, \quad \alpha_i^1 \eta_i^1 + \alpha_i^2 \eta_i^2 = \alpha_i \eta_i\,, \quad \alpha_i^1 \cdot u(\eta_i^1) + \alpha_i^2 \cdot u(\eta_i^2) = \alpha_i \cdot u(\eta_i)\,. \tag{74}$$

The third equality of Equation (74) is actually an implication of the first two, because $u(\eta)$ is an affine function on the interval $[0, \theta^*]$—see Equation (69). By the first two equations in Equation (74), we know $\alpha_i^1 = \alpha_i \cdot \frac{\theta^* - \eta_i}{\theta^*}$ and $\alpha_i^2 = \alpha_i \cdot \frac{\eta_i}{\theta^*}$. For the case of $\eta_i \in [\theta^*, 1]$, the split is computed also from Equation (74), but with $\eta_i^1 = \theta^*$ and $\eta_i^2 = 1$. This gives us $\alpha_i^1 = \alpha_i \cdot \frac{1-\eta_i}{1-\theta^*}$ and $\alpha_i^2 = \alpha_i \cdot \frac{\eta_i - \theta^*}{1-\theta^*}$.

In geometry (see Figure 11), the above splitting operation replaces any point M (resp. N) on the line segment OA (resp. AB) by a (unique) convex combination of the two endpoints O and A (resp. A and B). We thus get a new set $\{(\alpha_i^1, \eta_i^1, u(\eta_i^1)), (\alpha_i^2, \eta_i^2, u(\eta_i^2))\}_{i=1}^n$, which, by Equation (74), is obviously a convex decomposition of the point $[\eta, u_0]$ in $\ell^{\downarrow}_{\overline{\mathrm{FSC}}}$. Now, as $h(\eta)$ is a concave function, we know from Lemma 9 that $\alpha_i^1 \cdot h(\eta_i^1) + \alpha_i^2 \cdot h(\eta_i^2) \leqslant \alpha_i \cdot h(\eta_i)$. This implies that the sum $\sum_i \alpha_i \cdot h(\eta_i)$ of the new convex decomposition is no more than that of the original one. Moreover, by its construction, the η-parameter of this new convex decomposition assumes one of the three values: 0, θ^* and 1. We can thus "merge" all items with same η-value into one item (in an obvious way), yielding a convex decomposition of $[\eta, u_0]$ with no more than three items—we wrote "no more than" because any item with $\alpha_i = 0$ can be removed without changing the whole convex decomposition and the value of $\sum_i \alpha_i \cdot h(\eta_i)$.

Thus far, we have shown that for any convex decomposition of $[\eta, u_0]$ another convex decomposition can be constructed which has at most three items whose η-parameter are in the set $\{0, \theta^*, 1\}$, and for which the sum $\sum_i \alpha_i \cdot h(\eta_i)$ is less than or equal to that of the original convex decomposition. Therefore, Equation (73) can be simplified to

$$\underline{g}(\eta, u_0 | \operatorname{co} \ell_{\overline{\mathrm{FSC}}}) = \inf\{\alpha_1 \cdot h(0) + \alpha_2 \cdot h(\theta^*) + \alpha_3 \cdot h(1) \mid \text{condition on } \alpha_{1,2,3}\}\,.$$

In the above expression, $\alpha_i \geqslant 0$ are the coefficients occurred when $[\eta, u_0] \in \operatorname{co} \ell^{\downarrow}_{\overline{\mathrm{FSC}}}$ is written as the (unique) convex combination of the three points $[0, u(0)]$, $[\theta^*, u(\theta^*)]$ and $[1, u(1)]$. In Figure 11, this corresponds with that a point K in the triangle OAB is written as a convex combination of the three extreme points O, A and B. As is well know in geometry, such a convex combination is unique.

By the above discussion, the expression of $\underline{g}(\eta, u_0 | \operatorname{co} \ell_{\overline{\mathrm{FSC}}})$ can be simplified further to

$$\underline{g}(\eta, u_0 | \operatorname{co} \ell_{\overline{\mathrm{FSC}}}) = \alpha_1 \cdot h(0) + \alpha_2 \cdot h(\theta^*) + \alpha_3 \cdot h(1) = \alpha_2 \cdot h(\theta^*)\,,$$

where we have employed the fact $h(0) = h(1) = 0$, and the coefficients α_i are (uniquely) determined by the linear equations with η and u_0 as known constants:

$$\alpha_1 + \alpha_2 + \alpha_3 = 0\,, \qquad \alpha_1 \cdot [0, u(0)] + \alpha_2 \cdot [\theta^*, u(\theta^*)] + \alpha_3 \cdot [1, u(1)] = [\eta, u_0]\,.$$

As $u(0)=0$, $u(\theta^*)=(\theta^*)^2$ and $u(1)=\theta^*-\tilde{\theta}^*$, solving the above equations results in $\alpha_2=(\theta^*\tilde{\theta}^*)^{-1}\cdot[u_0+\eta(\tilde{\theta}^*-\theta^*)]$. Therefore,

$$\underline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})=(\theta^*\tilde{\theta}^*)^{-1}\cdot h(\theta^*)\cdot[u_0+\eta(\tilde{\theta}^*-\theta^*)]\,,\qquad\forall\,[\eta,u_0]\in\operatorname{co}\ell^{\downarrow}_{\overline{\mathrm{FSC}}}. \tag{75}$$

The expression of $\underline{g}(\eta,r_0|\operatorname{co}\ell_{\underline{\mathrm{BER}}})$ can be derived in a similar way, yielding

$$\underline{g}(\eta,r_0|\operatorname{co}\ell_{\underline{\mathrm{BER}}})=r_0\cdot h(\pi)\,,\qquad\forall\,[\eta,r_0]\in\operatorname{co}\ell^{\downarrow}_{\underline{\mathrm{BER}}}. \tag{76}$$

Note that both $\underline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$ and $\underline{g}(\eta,r_0|\operatorname{co}\ell_{\underline{\mathrm{BER}}})$ are affine functions. This fact and Equation (71) reveal that in Figure 6-a (resp. Figure 7-a) the convex hull of the curve $\ell_{\underline{\mathrm{BER}}}$ (resp. $\ell_{\overline{\mathrm{FSC}}}$) is bounded from below by the triangle OAB in the η-r-h (resp. η-u-h) space.

We now study the expression of $\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$, Equation (72). For simplicity, assume that a convex decomposition $\{(\alpha_i,\eta_i,u(\eta_i))\}_{i=1}^n$ of $[\eta,u_0]$ have been ordered such that $\eta_i<\theta^*$ for $i<k$ and $\eta_i\geqslant\theta^*$ for $i\geqslant k$. We can then "merge" the items $\{(\alpha_i,\eta_i,u(\eta_i))\}_{i=1}^{k-1}$ into one, namely, their weighted sum $(\alpha',\eta',u(\eta'))$ with $\alpha'=\sum_{i=1}^{k-1}\alpha_i$ and $\eta'=\frac{1}{\alpha'}\cdot\sum_{i=1}^{k-1}\alpha_i\eta_i$. Similarly, $\{(\alpha_i,\eta_i,u(\eta_i))\}_{i=k}^n$ can be "merged" into $(\alpha'',\eta'',u(\eta''))$ with $\alpha''=\sum_{i=k}^n\alpha_i$ and $\eta''=\frac{1}{\alpha''}\cdot\sum_{i=k}^n\alpha_i\eta_i$. As $u(\eta)$ is an affine function on the intervals $[0,\theta^*]$ and $[\theta^*,1]$—see Equation (69), one easily verifies that $\{(\alpha',\eta',u(\eta')),(\alpha'',\eta'',u(\eta''))\}$ is a convex decomposition of $[\eta,u_0]$ in $\ell^{\downarrow}_{\overline{\mathrm{FSC}}}$. Furthermore, by the concavity of $h(\eta)$ we know $\alpha'\cdot h(\eta')\geqslant\sum_{i=1}^{k-1}\alpha_i\cdot h(\eta_i)$ and $\alpha''\cdot h(\eta'')\geqslant\sum_{i=k}^n\alpha_i\cdot h(\eta_i)$. Hence $\alpha'\cdot h(\eta')+\alpha''\cdot h(\eta'')\geqslant\sum_{i=1}^n\alpha_i\cdot h(\eta_i)$. This enables us to consider only convex decompositions with at most two items when dealing with the function $\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$. That is, Equation (72) can now be simplified to

$$\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})=\sup\{\tilde{t}\cdot h(\eta')+t\cdot h(\eta'')\mid\text{condition on }t,\eta'\text{ and }\eta''\}\,, \tag{77}$$

where t, η' and η'' should be such that $\eta'<\theta^*\leqslant\eta''$ and $\{(\tilde{t},\eta',u(\eta')),(t,\eta'',u(\eta''))\}$ forms a convex decomposition of $[\eta,u_0]$.

In Figure 11, Equation (77) means that for any point $\mathrm{K}=[\eta,u_0]$ in the triangle OAB, we need to find a point $\mathrm{M}=[\eta',u(\eta')]$ on the line segment OA and a point $\mathrm{N}=[\eta'',u(\eta'')]$ on the line segment AB, such that K is on the line segment MN, that is, $\mathrm{K}=\tilde{t}\cdot\mathrm{M}+t\cdot\mathrm{N}$. The value of $\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$ is then the supremum of $t\cdot h(\eta')+\tilde{t}\cdot h(\eta'')$ over all such pairs (M,N). For the curve $\ell_{\underline{\mathrm{BER}}}$, we actually have already carried out this computation in Section 5—see Equations (36)–(38) and (41), whose correctness gets verified by the discussion here. Moreover, the analysis in this section shows that calculating Equations (39) and (40) is in fact unnecessary, which was previously proven in Lemma 10. The similar method can be used to simplify the expression of $\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$ to the maximum of a function of t, like Equation (41).

For the purpose of deriving Theorem 11 and Theorem 15, we will focus only on the case of $u_0=0$ for $\overline{g}(\eta,u_0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$ and the case of $\eta=\pi$ for $\overline{g}(\eta,r_0|\operatorname{co}\ell_{\underline{\mathrm{BER}}})$. The corresponding expressions for these two cases are listed below (the detailed computation is omitted):

$$\overline{g}(\eta,0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})=\sup\{\tilde{t}\cdot h(\tilde{t}^{-1}(\eta\tilde{\theta}^*-t\theta^*))+t\cdot h(\theta^*+t^{-1}\eta\theta^*)\mid t\in[\eta\tfrac{\theta^*}{\tilde{\theta}^*},\eta\tfrac{\tilde{\theta}^*}{\theta^*}]\}\,, \tag{78}$$

$$\overline{g}(\pi,r_0|\operatorname{co}\ell_{\underline{\mathrm{BER}}})=\sup\{\tilde{t}\cdot h(\pi-\tilde{t}^{-1}\pi\tilde{\pi}\tilde{r}_0)+t\cdot h(\pi+t^{-1}\pi\tilde{\pi}\tilde{r}_0)\mid t\in[\pi\tilde{r}_0,r_0+\pi\tilde{r}_0]\}\,. \tag{79}$$

Note that the above Equation (79) is same as Equation (41) (if we replace ρ by r_0).

B.4 The Expectation of a Random Vector and the Convex Hull of Its Range

This section is devoted to proving that *any random vector in $\mathbb{R}^m$ has the expectation lying in the convex hull of its range*. We actually will prove a stronger theorem, which are to be stated in a formal way after we have introduced the necessary definitions and notations.

Modern probability theory defines a random variable as a measurable function on some probability space $(\Omega, \mathscr{F}, P)$. In particular, a random vector is a measurable function from Ω into the Euclidean space $\mathbb{R}^m$ equipped with the σ-algebra of Borel sets. For any $A \in \mathscr{F}$ with $P(A) > 0$ and any random vector $\mathsf{u} : \Omega \to \mathbb{R}^m$, we write

$$\mathsf{u}(A) := \{\mathsf{u}(\omega) \mid \omega \in A\},$$
$$\mathbb{E}_A[\mathsf{u}] := P(A)^{-1} \cdot \textstyle\int_A \mathsf{u}(\omega) \mathrm{d}P.$$

Intuitively, $\mathsf{u}(A) \subseteq \mathbb{R}^m$ is the image of the set $A \subseteq \Omega$ under the mapping u; and $\mathbb{E}_A[\mathsf{u}]$ is the average value (weighted by probability) of u on the set A. Note that, when $A = \Omega$ the above two quantities are the range and the expectation of u, respectively.

We are now ready to formally state the main result of this section.

Theorem 26 *Let $\mathsf{u} : \Omega \to \mathbb{R}^m$ be a random vector and $A \in \mathscr{F}$ satisfy $P(A) > 0$. Then $\mathbb{E}_A[\mathsf{u}] \in \operatorname{co} \mathsf{u}(A)$.*

The following two lemmas discuss the 1-dimensional case (i.e., $m = 1$) and are useful for proving the theorem.

Lemma 27 *Let $A \in \mathscr{F}$ be such that $P(A) > 0$ and the random variable $\mathsf{u} : \Omega \to \mathbb{R}$ satisfy $\mathsf{u}(\omega) > 0$ for any $\omega \in A$. Then $\int_A \mathsf{u}(\omega) \mathrm{d}P > 0$.*

Proof For each $n \in \mathbb{N}$, define $A_n := \{\omega \in A \mid \mathsf{u}(\omega) \geqslant \frac{1}{n}\}$, then $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n \subseteq \cdots$. Furthermore, as $\mathsf{u}(\omega) > 0$ for $\omega \in A$, we have $A = \bigcup_{n=1}^{\infty} A_n$. The continuity of probability measures then implies $\lim_{n\to\infty} P(A_n) = P(A) > 0$. Thus, there exists an $N \in \mathbb{N}$ such that $P(A_N) > 0$. We thus get $\int_A \mathsf{u}(\omega) \mathrm{d}P = \int_{A_N} \mathsf{u}(\omega) \mathrm{d}P + \int_{A \setminus A_N} \mathsf{u}(\omega) \mathrm{d}P \geqslant \frac{1}{N} \cdot P(A_N) > 0$. ■

Lemma 28 *Let $\mathsf{u} : \Omega \to \mathbb{R}$ be a real-valued random variable and let $A \in \mathscr{F}$ be such that $P(A) > 0$. Then $\mathbb{E}_A[\mathsf{u}] \in \operatorname{co} \mathsf{u}(A)$.*

Proof Write $a = \inf \mathsf{u}(A)$, $b = \sup \mathsf{u}(A)$ and assume that $a \in \mathsf{u}(A)$ and $b \notin \mathsf{u}(A)$—there are three other possibilities to which a similar discussion to the one presented here applies. Then by Lemma 21 we have $\operatorname{co} \mathsf{u}(A) = [a, b)$; so it suffices to show $a \leqslant \mathbb{E}_A[\mathsf{u}] < b$.

As $a = \inf \mathsf{u}(A)$, we have $\mathsf{u}(\omega) \geqslant a$ for all $\omega \in A$. Thus, $P(A) \cdot \mathbb{E}_A[\mathsf{u}] = \int_A \mathsf{u}(\omega) \mathrm{d}P \geqslant a \cdot P(A)$ and hence $\mathbb{E}_A[\mathsf{u}] \geqslant a$. To show that $\mathbb{E}_A[\mathsf{u}] < b$, we define $\mathsf{v} = b - \mathsf{u}$. Then $\mathsf{v} > 0$ is a random variable; and it follows from Lemma 27 that $\mathbb{E}_A[\mathsf{v}] = P(A)^{-1} \cdot \int_A \mathsf{v}(\omega) \mathrm{d}P > 0$. But $\mathbb{E}_A[\mathsf{v}] = b - \mathbb{E}_A[\mathsf{u}]$, we thus get $\mathbb{E}_A[\mathsf{u}] < b$. ■

We now prove Theorem 26, by inducting on the dimensionality m.

Proof The case of $m = 1$ has been established in Lemma 28. Assume that the theorem is true in $\mathbb{R}^{m-1}$; and we want to show that it holds also for $\mathbb{R}^m$. If this is not the case, then there exist a random vector $\mathsf{u} : \Omega \to \mathbb{R}^m$ and a set $A \in \mathscr{A}$ such that $P(A) > 0$ and $\mathbb{E}_A[\mathsf{u}] \notin \mathrm{co}\,\mathsf{u}(A)$. Without loss of generality, we can, and do, further assume that $\mathbb{E}_A[\mathsf{u}] = 0$ (otherwise we turn to considering the random vector $\mathsf{u}'(\omega) := \mathsf{u}(\omega) - \mathbb{E}_A[\mathsf{u}]$).

As $\mathbb{E}_A[\mathsf{u}] = 0$ is a point not in the *convex* set $\mathrm{co}\,\mathsf{u}(A)$, there is a hyperplane separating the two—see for example, Boyd and Vandenberghe (2004, Chapter 2.5). That is, there exist $\boldsymbol{w} \in \mathbb{R}^m$ and $c \in \mathbb{R}$ such that $\boldsymbol{w} \cdot \boldsymbol{u} + c \geqslant 0$ for all $\boldsymbol{u} \in \mathrm{co}\,\mathsf{u}(A)$ and that $\boldsymbol{w} \cdot \mathbf{0} + c = c \leqslant 0$, where $\boldsymbol{w} \cdot \boldsymbol{u}$ denotes the standard inner product of $\boldsymbol{w}$ and $\boldsymbol{u}$. Thus, $\boldsymbol{w} \cdot \boldsymbol{u} \geqslant -c \geqslant 0$ for any $\boldsymbol{u} \in \mathrm{co}\,\mathsf{u}(A)$. To simplify the discussion, we assume $\boldsymbol{w}$ is the first standard unit vector, $\boldsymbol{w} = [1, 0, \ldots, 0]$—this can always be obtained by applying a proper rotation operator on the random vector u, so it causes no loss of generality. Under this assumption, the inequality $\boldsymbol{w} \cdot \boldsymbol{u} \geqslant 0$ now reads $u_1 \geqslant 0$, for any $\boldsymbol{u} = [u_1, \ldots, u_m] \in \mathrm{co}\,\mathsf{u}(A)$.

A side remark: intuitively, the above argument says that, since $\mathrm{co}\,\mathsf{u}(A)$ *is convex and* $\mathbb{E}_A[\mathsf{u}] \notin \mathrm{co}\,\mathsf{u}(A)$*, we can first move the origin to the point* $\mathbb{E}_A[\mathsf{u}]$*; then rotate the axes so that* $\mathrm{co}\,\mathsf{u}(A)$ *lies in the half space* $H_{\geqslant 0} := \{\boldsymbol{u} = [u_1, \ldots, u_m] \in \mathbb{R}^m \mid u_1 \geqslant 0\}$ *after the rotation.*

We return and continue the proof. Define

$$\begin{array}{rclcrcl} H_0 & := & \{\boldsymbol{u} \in \mathbb{R}^m \mid u_1 = 0\}, & & A_0 & := & \{\omega \in A \mid \mathsf{u}(\omega) \in H_0\}, \\ H_{>0} & := & \{\boldsymbol{u} \in \mathbb{R}^m \mid u_1 > 0\}, & & A_1 & := & \{\omega \in A \mid \mathsf{u}(\omega) \in H_{>0}\}. \end{array}$$

Then it is clear that $A_0 \cap A_1 = \varnothing$. Furthermore, from $\mathsf{u}(A) \subseteq \mathrm{co}\,\mathsf{u}(A) \subseteq H_{\geqslant 0}$ we know $A_0 \cup A_1 = A$. It hence follows from $\mathbb{E}_A[\mathsf{u}] = 0$ that

$$0 = P(A) \cdot \mathbb{E}_A[\mathsf{u}] = \textstyle\int_A \mathsf{u}(\omega) \mathrm{d}P = \int_{A_0} \mathsf{u}(\omega) \mathrm{d}P + \int_{A_1} \mathsf{u}(\omega) \mathrm{d}P. \tag{80}$$

Extracting the first component of this equality results in $\int_{A_1} \mathsf{u}_1(\omega) \mathrm{d}P = 0$. This is because $\mathsf{u}_1(\omega) = 0$ on A_0 and hence $\int_{A_0} \mathsf{u}_1(\omega) \mathrm{d}P = 0$. But $\mathsf{u}_1(\omega) > 0$ for $\omega \in A_1$, so by Lemma 27 we know $P(A_1) = 0$, which in turn implies $\int_{A_1} \mathsf{u}(\omega) \mathrm{d}P = \mathbf{0}$ and $P(A_0) = P(A) > 0$ (as $A = A_0 \cup A_1$). Equation (80) can then be rewritten as $0 = P(A_0)^{-1} \cdot \mathbb{E}_A[\mathsf{u}] = \int_{A_0} \mathsf{u}(\omega) \mathrm{d}P$, that is, $\mathbb{E}_A[\mathsf{u}] = 0 = P(A_0)^{-1} \cdot \int_{A_0} \mathsf{u}(\omega) \mathrm{d}P = \mathbb{E}_{A_0}[\mathsf{u}]$.

On the other hand, $A_0 \subseteq A$ implies $\mathrm{co}\,\mathsf{u}(A_0) \subseteq \mathrm{co}\,\mathsf{u}(A)$. So it follows from the assumptions $\mathbb{E}_A[\mathsf{u}] \notin \mathrm{co}\,\mathsf{u}(A)$ and $\mathbb{E}_A[\mathsf{u}] = \mathbf{0}$ that $\mathbf{0} \notin \mathrm{co}\,\mathsf{u}(A_0)$. Write $\mathsf{u} = [\mathsf{u}_1, \ldots, \mathsf{u}_m] = [\mathsf{u}_1, \mathsf{v}]$, that is, $\mathsf{v} = [\mathsf{u}_2, \ldots, \mathsf{u}_m]$. As $\mathsf{u}_1(\omega) = 0$ for all $\omega \in A_0$, we have the following "decomposition":

$$\begin{aligned} \mathsf{u}(A_0) &= \{[\mathsf{u}_1(\omega), \mathsf{v}(\omega)] \mid \omega \in A_0\} \\ &= \{(0, \mathsf{v}(\omega)) \mid \omega \in A_0\} \\ &= \{0\} \times \{\mathsf{v}(\omega) \mid \omega \in A_0\} \\ &= \{0\} \times \mathsf{v}(A_0), \end{aligned}$$

and hence $\mathrm{co}\,\mathsf{u}(A_0) = \{0\} \times \mathrm{co}\,\mathsf{v}(A_0)$. This fact together with $\mathbf{0} \notin \mathrm{co}\,\mathsf{u}(A_0)$ implies that $\mathbf{0} \notin \mathrm{co}\,\mathsf{v}(A_0)$. For the $(m-1)$-dimensional random vector v, the induction hypothesis gives $\mathbb{E}_{A_0}[\mathsf{v}] \in \mathrm{co}\,\mathsf{v}(A_0)$. It thus follows that $\mathbb{E}_{A_0}[\mathsf{v}] \neq \mathbf{0}$, which further implies $\mathbb{E}_{A_0}[\mathsf{u}] \neq \mathbf{0}$.

We have proved both $\mathbb{E}_{A_0}[\mathsf{u}] = \mathbf{0}$ and $\mathbb{E}_{A_0}[\mathsf{u}] \neq \mathbf{0}$; this contradiction reveals that the assumption $\mathbb{E}_A[\mathsf{u}] \notin \mathrm{co}\,\mathsf{u}(A)$ must not be true. We thus accomplished the proof. ■

Appendix C. Proofs to the Main Theorems

For those readers who are not satisfied with the presented derivations and who are really enthusiastic about rigorous mathematical proofs, we translate in this section the geometric proofs to the main theorems into the analytical one. We have already done the main job in the preceding section; all we need to do here is to assemble the discussion presented in that section into a proper proof.

C.1 Proof to Theorem 7

By Equations (28), (64) and Theorem 1, we have $[\underline{\mathrm{CSR}}, H(\mathsf{y}|\mathsf{x})] \in \mathrm{co}\,\ell_{\underline{\mathrm{CSR}}}$. It then follows from Equation (68) that

$$2\cdot h(\tfrac{1}{2c_1})\cdot \underline{\mathrm{CSR}} \leqslant H(\mathsf{y}|\mathsf{x}) \leqslant [\max\{h(c_1^{-1}\cdot \underline{\mathrm{CSR}}),\, h(1-c_0^{-1}\cdot \underline{\mathrm{CSR}})\}]^{\frown}.$$

This proves Equation (30). Furthermore, the proofs to Theorem 5 and Corollary 6 can be moved straightforwardly here to show the tightness of the two bounds in Equation (30).

C.2 Proof to Theorem 11

By Equations (34), (65) and Theorem 1, we know $[\pi, 2\underline{\mathrm{BER}}, H(\mathsf{y}|\mathsf{x})]$ is in the set $\mathrm{co}\,\ell_{\underline{\mathrm{BER}}}$, which, by Equation (63), can be written as

$$\mathrm{co}\,\ell_{\underline{\mathrm{BER}}} = \{[\eta, r_0, h_0] \mid [\eta, r_0] \in \mathrm{co}\,\ell^{\downarrow}_{\underline{\mathrm{BER}}},\, \underline{g}(\eta, r_0|\,\mathrm{co}\,\ell_{\underline{\mathrm{BER}}}) \leqslant h_0 \leqslant \overline{g}(\eta, r_0|\,\mathrm{co}\,\ell_{\underline{\mathrm{BER}}})\}.$$

Now fix $\eta = \pi$ in the above expression, and we obtain from Equations (76) and (79) the desired inequality, Equation (44). The tightness of the obtained bounds can be proven similarly to that in Theorem 5 and Corollary 6.

C.3 Theorem 11 is Stronger Than Corollary 8

In this section, we intend to show the upper bound of $H(\mathsf{y}|\mathsf{x})$ as given by Equation (33) is never tighter than that in Equation (44). Mathematically, this amount to proving that

$$\max\{f_3(t) \mid t \in [\pi\tilde{\rho}, \pi\tilde{\rho}+\rho]\} \leqslant \begin{cases} h(\tilde{\pi}\rho) & \text{if } \pi \leqslant 0.5 \\ h(\pi\rho) & \text{if } \pi > 0.5 \end{cases},$$

where $f_3(t) = \tilde{t}\cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t\cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho})$, $\rho = 2\underline{\mathrm{BER}} \in [0,1]$, and $h:[0,1]\to\mathbb{R}$ is a symmetric concave function satisfying $h(0) = h(1) = 1$.

To simplify the proof and notation, we shall consider only the case of $\pi \leqslant 0.5$ under an additional condition that the function $h(\cdot)$ is differentiable[17]. For any function as such and any numbers $\eta, \eta_0 \in [0,1]$, by the concavity of $h(\eta)$ we know $h(\eta) \leqslant h(\eta_0) + h'(\eta_0)\cdot(\eta - \eta_0)$. If $\tilde{\pi}\rho \geqslant \frac{1}{2}$, put $\eta_0 = 1 - \tilde{\pi}\rho = \pi + \tilde{\pi}\tilde{\rho} \leqslant \frac{1}{2}$. Since $h(\eta)$ is symmetric and concave, we know $h(\eta_0) = h(\tilde{\pi}\rho)$ and $h'(\eta_0) \geqslant 0$. It then follows that

$$\begin{aligned} h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) &\leqslant h(\eta_0) + h'(\eta_0)\cdot(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho} - \eta_0) = h(\tilde{\pi}\rho) - h'(\eta_0)\cdot\tilde{\pi}\tilde{\rho}\cdot(1+\tilde{t}^{-1}\pi), \\ h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) &\leqslant h(\eta_0) + h'(\eta_0)\cdot(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho} - \eta_0) = h(\tilde{\pi}\rho) - h'(\eta_0)\cdot\tilde{\pi}\tilde{\rho}\cdot(1-t^{-1}\pi), \end{aligned}$$

17. The case where $\pi > 0.5$ can be discussed similarly. As before, the differentiability assumption is unnecessary: if $h(\cdot)$ is non-differentiable at some point η_0, we can use any number between its right derivative $h'(\eta_0+)$ and left derivative $h'(\eta_0-)$ to replace $h'(\eta_0)$.

and hence $f_3(t) = \tilde{t} \cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t \cdot h(\pi + t^{-1}\pi\tilde{\pi}\tilde{\rho}) \leqslant h(\tilde{\pi}\rho) - h'(\eta_0) \cdot \tilde{\pi}\tilde{\rho} \leqslant h(\tilde{\pi}\rho)$ for any $t \in [0,1]$.

Now suppose that $\tilde{\pi}\rho < \frac{1}{2}$. As $h(\cdot)$ is symmetric, by Jensen's inequality we know

$$f_3(t) = \tilde{t} \cdot h(\pi - \tilde{t}^{-1}\pi\tilde{\pi}\tilde{\rho}) + t \cdot h(\tilde{\pi} - t^{-1}\pi\tilde{\pi}\tilde{\rho}) \leqslant h(\tilde{t}\pi + t\tilde{\pi} - 2\pi\tilde{\pi}\tilde{\rho}).$$

For $t \in [\pi\tilde{\rho}, \pi\tilde{\rho} + \rho]$, by direct computation we have $\tilde{t}\pi + t\tilde{\pi} - 2\pi\tilde{\pi}\tilde{\rho} \in [\pi\rho, \tilde{\pi}\rho]$. By $\pi \leqslant 0.5$ we know $\pi\rho \leqslant \tilde{\pi}\rho < \frac{1}{2}$ and so $h(\pi\rho) \leqslant h(\tilde{\pi}\rho)$. Thus $f_3(t) \leqslant h(\tilde{\pi}\rho)$ for any $t \in [\pi\tilde{\rho}, \pi\tilde{\rho} + \rho]$. So far we have proved that $\max\{f_3(t) \mid t \in [\pi\tilde{\rho}, \rho + \pi\tilde{\rho}]\} \leqslant h(\tilde{\pi}\rho)$.

C.4 Proof to Theorem 15

For any binary classification task (μ, η), let $\theta^* = \frac{1}{2}\overline{\mathrm{FSC}}(\mu, \eta)$ and let the function $u(\eta)$ be as in Equation (69). Then $\mathbb{E}_{x\sim\mu}[u(\eta(x))] = 0$ and hence Equation (49) holds. It follows from Equation (66) and Theorem 1 that $[\pi, 0, H(y|x)]$ is in the set $\operatorname{co}\ell_{\overline{\mathrm{FSC}}}$. So by Equation (71) we know $\underline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant H(y|x) \leqslant \overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}})$, where the range of π is determined by the condition $[\pi, 0] \in \operatorname{co}\ell^{\downarrow}_{\overline{\mathrm{FSC}}}$, which by Equation (70) implies $\pi \in [0, \theta^*/\tilde{\theta}^*]$. It thus follows that

$$\inf_{\pi\in[0,\theta^*/\tilde{\theta}^*]} \underline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant H(y|x) \leqslant \sup_{\pi\in[0,\theta^*/\tilde{\theta}^*]} \overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}). \tag{81}$$

By Equation (75), we have $\underline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) = (\theta^*\tilde{\theta}^*)^{-1} \cdot h(\theta^*) \cdot \pi(\tilde{\theta}^* - \theta^*)$, so the infimum in Equation (81) is 0, which is obtained at $\pi = 0$. Next we will prove briefly that the right hand side of Equation (81) equals to $h(\theta^*/\tilde{\theta}^*) = h\left(\frac{\overline{\mathrm{FSC}}}{2-\overline{\mathrm{FSC}}}\right)$, with the help of Figure 7-b.

For any concave function $h : [0,1] \to \mathbb{R}$ and $\eta_0 \in (0,1)$, it is well known that the left derivative $h'(\eta_0-)$ and the right derivative $h'(\eta_0+)$ exist and satisfy $h'(\eta_0+) \leqslant h'(\eta_0-)$. Moreover, for $\eta_1, \eta_2 \in (0,1)$ with $\eta_1 > \eta_2$, we have $h'(\eta_1-) \leqslant h'(\eta_2+)$. Let $s(\eta_0)$ be a number between $h'(\eta_0+)$ and $h'(\eta_0-)$ and define $f(\eta) := s(\eta_0) \cdot (\eta - \eta_0) + h(\eta_0)$, $\eta \in [0,1]$. As is well known, the affine function $f(\eta)$ satisfies $f(\eta_0) = h(\eta_0)$ and $f(\eta) \geqslant h(\eta)$ for any $\eta \in [0,1]$. Such an affine function is called a *supporting line* of $h(\eta)$ and η_0.

Let $f_1(\eta) = s(\eta_1) \cdot (\eta - \eta_1) + h(\eta_1)$ be a supporting line of $h(\eta)$ at $\eta_1 = \theta^*/\tilde{\theta}^*$. This line intersects with the line $\eta = \theta^*$ at point $\mathrm{K} = [\theta^*, f_1(\theta^*)]$. Through the point K there is a supporting line of $h(\eta)$ at $\eta_2 \leqslant \theta^*$, which we denote as $f_2(\eta) = s(\eta_2) \cdot (\eta - \eta_2) + h(\eta_2)$. As $h(\eta)$ is symmetric and $\eta_2 \leqslant \theta^* \leqslant \frac{1}{2}$, we have $s(\eta_2) \geqslant 0$. Moreover, since $f_1(\theta^*) = f_2(\theta^*)$ and $\eta_2 \leqslant \theta^* \leqslant \eta_1$, by Lemma 18 we know $h(\eta_2) \leqslant h(\eta_1)$.

In Equation (78) let $\eta = \pi$ and relax the resulting expression to

$$\begin{aligned}\overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) &\leqslant \sup\{\tilde{t} \cdot f_2(\tilde{t}^{-1}(\pi\tilde{\theta}^* - t\theta^*)) + t \cdot f_1(\theta^* + t^{-1}\pi\theta^*) \mid t \in [\pi\tfrac{\theta^*}{\tilde{\theta}^*}, \pi\tfrac{\tilde{\theta}^*}{\theta^*}]\} \\ &= f_1(\theta^*) - s(\eta_2) \cdot \theta^* + \pi \cdot [s(\eta_1)\theta^* + s(\eta_2)\tilde{\theta}^*] =: f_0(\pi).\end{aligned}$$

Since $\pi \in [0, \theta^*/\tilde{\theta}^*]$ and $f_0(\pi)$ is an affine function, the above inequality further implies

$$\overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant \max\{f_0(0), f_0(\theta^*/\tilde{\theta}^*)\}.$$

As $s(\eta_2) \geqslant 0$, we know $f_0(0) = f_2(\theta^*) - s(\eta_2) \cdot \theta^* = f_2(0) \leqslant f_2(\eta_2) = h(\eta_2)$. Furthermore,

$$f_0(\theta^*/\tilde{\theta}^*) = f_1(\theta^*) + s(\eta_1) \cdot (\theta^*)^2/\tilde{\theta}^* = f_1\left(\theta^* + (\theta^*)^2/\tilde{\theta}^*\right) = f_1(\eta_1) = h(\eta_1).$$

It thus follows that $f_0(0) \leqslant f_0(\theta^*/\tilde{\theta}^*)$ and so $\overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant h(\eta_1) = h(\theta^*/\tilde{\theta}^*)$ for any $\pi \in [0, \theta^*/\tilde{\theta}^*]$. Thus, $\sup_{\pi\in[0,\theta^*/\tilde{\theta}^*]} \overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \leqslant h(\theta^*/\tilde{\theta}^*)$.

On the other hand, let $\eta = \pi$ and $t = \pi \cdot \tilde{\theta}^*/\theta^*$ in Equation (78), we obtain

$$\overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \geqslant \tilde{t} \cdot h(0) + t \cdot h(\theta^*/\tilde{\theta}^*) = \pi \cdot \tilde{\theta}^*/\theta^* \cdot h(\theta^*/\tilde{\theta}^*).$$

Thus, $\sup_{\pi\in[0,\theta^*/\tilde{\theta}^*]} \overline{g}(\pi, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \geqslant \overline{g}(\theta^*/\tilde{\theta}^*, 0|\operatorname{co}\ell_{\overline{\mathrm{FSC}}}) \geqslant h(\theta^*/\tilde{\theta}^*)$.

So far, we have finished the proof to Equation (50). The tightness of the two inequalities in Equation (50) can be proven by considering the convex decomposition of the extreme points $\mathrm{O} = [0,0,0]$ (or a point arbitrary close to O) and $\mathrm{E} = [\theta^*/\tilde{\theta}^*, 0, h(\theta^*/\tilde{\theta}^*)]$ in Figure 7-a. The detail is similar to that in Theorem 5 and Corollary 6 and omitted here.

References

C.D. Aliprantis and K.C. Border. *Infinite Dimensional Analysis: a Hitchhiker's Guide*. Springer, 2006.

A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

M. Ben-Bassat. f-entropies, probability of error, and feature selection. *Information and Control*, 39 (3):227–242, 1978.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Camgridge University Press, 2004.

R.P. Brent. *Algorithms for Minimization with Derivatives*. Prentice Hall, 1973.

G. Brown, A. Pocock, M.-J. Zhao, and M. Lujan. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13:27–66, 2012.

R. Caruana and A. Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78. ACM, 2004.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 2006.

L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Applications of Mathematics*. Springer, 1996.

W. Duch. *Feature Extraction: Foundation and Applications*, chapter 3, pages 89–117. Springer, 2006. ISBN 3-540-35487-5.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2001.

C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

D. Erdogmus and J.C. Principe. Lower and upper bounds for misclassification probability based on Rényi's information. *Journal of VLSI Signal Processing*, 37:305–317, 2004.

R.M. Fano. *Transmission of Information: a Statistical Theory of Communications*. MIT Press, 1961.

M. Feder and N. Merhav. Relations between entropy and error probability. *IEEE Transactions on Information Theory*, 40:259–266, 1994.

A. Garg and D. Roth. Understanding probabilistic classifiers. In *12th European Conference on Machine Learning (ECML)*, pages 179–191, 2001.

J.D. Golic. On the relationship between the information measures and the bayes probability of error. *IEEE Transactions on Information Theory*, IT-33(5):681–693, 1987.

S. Guiasu. Weighted entropy. *Reports on Mathematical Physics*, 2(3):165–179, 1971.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

M.E. Hellman and J. Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Infomation Theory*, IT-16(4):368–372, 1970.

K.E. Hild II, D. Erdogmus, K. Torkkola, and J.C. Principe. Feature extraction using information-theoretic learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1385–1392, 2006.

M. Jansche. A maximum expected utility framework for binary sequence labeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 736–743, 2007.

T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, February 1967.

R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, special issue on relevance:273–324, 1997.

V.A. Kovalevsky. The problem of character recognition from the point of view of mathematical statistics. In V. A. Kovalevski, editor, *Character Readers and Pattern Recognition*, pages 3–30, New York, 1968.

T.N. Lal, O. Chapelle, J. Weston, and A. Elisseeff. *Feature Extraction: Foundation and Applications*, chapter 5, pages 137–165. Springer, 2006.

D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR*, pages 246–254, 1995.

R. Linsker. Towards an organizing principle for a layered perceptual network. In *Advances in Neural Information Processing Systems (NIPS)*, volume 0, pages 485–494, 1988.

R. Linsker. An application of the priciple of maximum information preservation to linear systems. In *Advances in Neural Information Processing Systems (NIPS)*, volume 1, pages 186–194, 1989.

C.D. Manning, P. Raghavan, and H. Schuetze. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

G.H. Nguyen, A. Bouzerdoum, and S.L. Phung. Learning pattern classification tasks with imbalanced data sets. In P. Yin, editor, *Pattern recognition*, chapter 10, pages 193–208. Vukovar, Croatia: In-Teh., 2009.

J.C. Principe and D. Xu. Information-theoretic learning using Renyi's quadratic entropy. In *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation*, pages 407–412, 1999.

A. Rényi. On measures of entropy and information. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561, 1961.

R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27: 379–423, 623–656, 1948.

I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, 2007.

I.J. Taneja. *Generalized Information Measures and Their Applications*. on-line book, 2001. URL `www.mtm.ufsc.br/ taneja/book/book.html`.

D.L. Tebbe and S.J. Dwyer III. Uncertainty and the probability of error. 14(3):516–518, May 1968.

A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

Journal of Machine Learning Research 14 (2013) 1091-1143 Submitted 11/12; Revised 2/13; Published 4/13

Sparse Activity and Sparse Connectivity in Supervised Learning

Markus Thom MARKUS.THOM@UNI-ULM.DE
driveU / Institute of Measurement, Control and Microtechnology
Ulm University
89081 Ulm, Germany

Günther Palm GUENTHER.PALM@UNI-ULM.DE
Institute of Neural Information Processing
Ulm University
89081 Ulm, Germany

Editor: Aapo Hyvärinen

Abstract

Sparseness is a useful regularizer for learning in a wide range of applications, in particular in neural networks. This paper proposes a model targeted at classification tasks, where sparse activity and sparse connectivity are used to enhance classification capabilities. The tool for achieving this is a sparseness-enforcing projection operator which finds the closest vector with a pre-defined sparseness for any given vector. In the theoretical part of this paper, a comprehensive theory for such a projection is developed. In conclusion, it is shown that the projection is differentiable almost everywhere and can thus be implemented as a smooth neuronal transfer function. The entire model can hence be tuned end-to-end using gradient-based methods. Experiments on the MNIST database of handwritten digits show that classification performance can be boosted by sparse activity or sparse connectivity. With a combination of both, performance can be significantly better compared to classical non-sparse approaches.

Keywords: supervised learning, sparseness projection, sparse activity, sparse connectivity

1. Introduction

Sparseness is a concept of efficiency in neural networks, and exists in two variants in that context (Laughlin and Sejnowski, 2003). The *sparse activity* property means that only a small fraction of neurons is active at any time. The *sparse connectivity* property means that each neuron is connected to only a limited number of other neurons. Both properties have been observed in mammalian brains (Hubel and Wiesel, 1959; Olshausen and Field, 2004; Mason et al., 1991; Markram et al., 1997) and have inspired a variety of machine learning algorithms. A notable result was achieved through the sparse coding model of Olshausen and Field (1996). Given small patches from images of natural scenes, the model is able to produce Gabor-like filters, resembling properties of simple cells found in mammalian primary visual cortex (Hubel and Wiesel, 1959; Vinje and Gallant, 2000). Another example is the optimal brain damage method of LeCun et al. (1990), which can be used to prune synaptic connections in a neural network, making connectivity sparse. Although only a small fraction of possible connections remains after pruning, this is sufficient to achieve equivalent classification results. Since then, numerous approaches on how to measure sparseness have been proposed, see Hurley and Rickard (2009) for an overview, and how to achieve sparse solutions of classical machine learning problems.

The L_0 pseudo-norm is a natural sparseness measure. Its computation consists of counting the number of non-vanishing entries in a vector. Using it rather than other sparseness measures has been shown to induce biologically more plausible properties (Rehn and Sommer, 2007). However, finding of optimal solutions subject to the L_0 pseudo-norm turns out to be NP-hard (Natarajan, 1995; Weston et al., 2003). Analytical properties of this counting measure are very poor, for it is non-continuous, rendering the localization of approximate solutions difficult. The Manhattan norm of a vector is a convex relaxation of the L_0 pseudo-norm (Donoho, 2006), and has been employed in a vast range of applications. This sparseness measure has the significant disadvantage of not being scale-invariant, so that an intuitive notion of sparseness cannot be derived from it.

1.1 Hoyer's Normalized Sparseness Measure

A normalized sparseness measure σ based on the ratio of the L_1 or Manhattan norm and the L_2 or Euclidean norm of a vector has been proposed by Hoyer (2004),

$$\sigma\colon \mathbb{R}^n \setminus \{0\} \to [0,\ 1], \qquad x \mapsto \frac{\sqrt{n} - \|x\|_1 / \|x\|_2}{\sqrt{n} - 1},$$

where higher values indicate more sparse vectors. σ is well-defined because $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\,\|x\|_2$ holds for all $x \in \mathbb{R}^n$ (Laub, 2004). As $\sigma(\alpha x) = \sigma(x)$ for all $\alpha \neq 0$ and all $x \in \mathbb{R}^n \setminus \{0\}$, σ is also scale-invariant. As composition of differentiable functions, σ is differentiable on its entire domain.

This sparseness measure fulfills all criteria of Hurley and Rickard (2009) except for Dalton's fourth law, which states that the sparseness of a vector should be identical to the sparseness of the vector resulting from multiple concatenation of the original vector. This property, however, is not crucial for a proper sparseness measure. For example, sparseness of connectivity in a biological brain increases quickly with its volume, so that connectivity in a human brain is about 170 times more sparse than in a rat brain (Karbowski, 2003). It follows that σ features all desirable properties of a proper sparseness measure.

A sparseness-enforcing projection operator, suitable for projected gradient descent algorithms, was proposed by Hoyer (2004) for optimization with respect to σ. For a pre-defined target degree of sparseness $\sigma^* \in (0,\ 1)$, the operator finds the closest vector of a given scale that has sparseness σ^* given an arbitrary vector. This can be expressed formally as Euclidean projection onto parameterizations of the sets

$$S^{(\lambda_1,\lambda_2)} := \{\, s \in \mathbb{R}^n \mid \|s\|_1 = \lambda_1 \text{ and } \|s\|_2 = \lambda_2 \,\} \text{ and } S_{\geq 0}^{(\lambda_1,\lambda_2)} := S^{(\lambda_1,\lambda_2)} \cap \mathbb{R}^n_{\geq 0}.$$

The first set is for achieving unrestricted projections, whereas the latter set is useful in situations where only non-negative solutions are feasible, for example in non-negative matrix factorization problems. The constants $\lambda_1, \lambda_2 > 0$ are target norms and can be chosen such that all points in these sets achieve a sparseness of σ^*. For example, if λ_2 was set to unity for yielding normalized projections, then λ_1 can be easily derived from the definition of σ.

Hoyer's original algorithm for computation of such a projection is an alternating projection onto a hyperplane representing the L_1 norm constraint, a hypersphere representing the L_2 norm constraint, and the non-negative orthant. A slightly modified version of this algorithm has been proved to be correct by Theis et al. (2005) in the special case when exactly one negative entry emerges that is zeroed out in the orthant projection. However, there is still no mathematically satisfactory proof for the general case.

1.2 Contributions of this Paper

This paper improves upon previous work in the following ways. Section 2 proposes a simple algorithm for carrying out sparseness-enforcing projections with respect to Hoyer's sparseness measure. Further, an improved algorithm is proposed and compared with Hoyer's original algorithm. Because the projection itself is differentiable, it is the ideal tool for achieving sparseness in gradient-based learning. This is exploited in Section 3, where the sparseness projection is used to obtain a classifier that features both sparse activity and sparse connectivity in a natural way. The benefit of these two key properties is demonstrated on a real-world classification problem, proving that sparseness acts as regularizer and improves classification results. The final sections give an overview of related concepts and conclude this paper.

On the theoretical side, a first rigorous and mathematically satisfactory analysis of the properties of the sparseness-enforcing projection is provided. This is lengthy and technical and therefore deferred into several appendixes. Appendix A fixes the notation and gives an introduction to general projections. In Appendix B, certain symmetries of subsets of the Euclidean space and their effect on projections onto such sets is studied. The problem of finding projections onto sets where Hoyer's sparseness measure attains a constant value is addressed in Appendix C. Ultimately, the algorithms proposed in Section 2 are proved to be correct. Appendix D investigates analytical properties of the sparseness projection and concludes with an efficient algorithm that computes its gradient. The gradients for optimization of the parameters of the architecture proposed in Section 3 are collected in the final Appendix E.

2. Algorithms for the Sparseness-Enforcing Projection Operator

The projection onto a set is a fundamental concept, for example see Deutsch (2001):

Definition 1 *Let $x \in \mathbb{R}^n$ and $\varnothing \neq M \subseteq \mathbb{R}^n$. Then every point in*

$$\operatorname{proj}_M(x) := \{\, y \in M \mid \|y-x\|_2 \leq \|z-x\|_2 \text{ for all } z \in M \,\}$$

is called Euclidean projection *of x onto M. When there is exactly one point y in $\operatorname{proj}_M(x)$, then $y = \operatorname{proj}_M(x)$ is used as an abbreviation.*

Because $\mathbb{R}^n$ is finite-dimensional, $\operatorname{proj}_M(x)$ is nonempty for all $x \in \mathbb{R}^n$ if and only if M is closed, and $\operatorname{proj}_M(x)$ is a singleton for all $x \in \mathbb{R}^n$ if and only if M is closed and convex (Deutsch, 2001). In the literature, the elements from $\operatorname{proj}_M(x)$ are also called *best approximations* to x from M.

Projections onto sets that fulfill certain symmetries are of special interest in this paper and are formalized and discussed in Appendix B in greater detail. It is notable that projections onto a *permutation-invariant* set M, that is a set where membership is stable upon coordinate permutation, are *order-preserving*. This is proved in Lemma 9(a). As a consequence, when a vector is sorted in ascending or descending order, then its projection onto M is sorted accordingly. If M is *reflection-invariant*, that is when the signs of arbitrary coordinates can be swapped without violating membership in M, then the projection onto M is *orthant-preserving*, as shown in Lemma 9(b). This means that a point and its projection onto M are located in the same orthant. By exploiting this property, projections onto M can be yielded by recording and discarding the signs of the coordinates of the argument, projecting onto $M \cap \mathbb{R}^n_{\geq 0}$, and finally restoring the signs of the coordinates of the result using the signs of the argument. This is formalized in Lemma 11.

As an example for these concepts, consider the set $Z := \{ x \in \mathbb{R}^n \mid \|x\|_0 = \kappa \}$ of all vectors with exactly $\kappa \in \mathbb{N}$ non-vanishing entries. Z is clearly both permutation-invariant and reflection-invariant. Therefore, the projection with respect to an L_0 pseudo-norm constraint must be both order-preserving and orthant-preserving. In fact, the projection onto Z consists simply of zeroing out all entries but the κ that are greatest in absolute value (Blumensath and Davies, 2009). This trivially fulfills the aforementioned properties of order-preservation and orthant-preservation.

Permutation-invariance and reflection-invariance are closed under intersection and union operations. Therefore, the unrestricted target set $S^{(\lambda_1,\lambda_2)}$ for the σ projection is permutation-invariant and reflection-invariant. It is hence enough to handle projections onto $S^{(\lambda_1,\lambda_2)}_{\geq 0}$ in the first place, as projections onto the unrestricted target set can easily be recovered.

In the remainder of this section, let $n \in \mathbb{N}$ be the problem dimensionality and let $\lambda_1, \lambda_2 > 0$ be the fixed target norms, which must fulfill $\lambda_2 \leq \lambda_1 \leq \sqrt{n}\lambda_2$ to avoid the existence of only trivial solutions. In the applications of the sparseness projection in this paper, λ_2 is always set to unity to achieve normalized projections, and λ_1 is adjusted as explained in Section 1.1 to achieve the target degree of sparseness σ^*. The related problem of finding the best approximation to a point x regardless of the concrete scaling, that is computing projections onto $\{ s \in \mathbb{R}^n \setminus \{0\} \mid \sigma(s) = \sigma^* \}$, can be solved by projecting x onto $S^{(\lambda_1,\lambda_2)}$ and rescaling the result p such as to minimize $\|x - \alpha p\|_2$ under variation of $\alpha \in \mathbb{R}$, which yields $\alpha = \langle x, p \rangle / \|p\|_2^2$. This method is justified theoretically by Remark 5.

2.1 Alternating Projections

First note that the target set can be written as an intersection of simpler sets. Let $e_1, \ldots, e_n \in \mathbb{R}^n$ be the canonical basis of the n-dimensional Euclidean space $\mathbb{R}^n$. Further, let $e := \sum_{i=1}^n e_i \in \mathbb{R}^n$ be the vector where all entries are identical to unity. Then $H := \{ a \in \mathbb{R}^n \mid e^T a = \lambda_1 \}$ denotes the target hyperplane where the coordinates of all points sum up to λ_1. In the non-negative orthant $\mathbb{R}^n_{\geq 0}$, this is equivalent to the L_1 norm constraint. Further, define $K := \{ q \in \mathbb{R}^n \mid \|q\|_2 = \lambda_2 \}$ as the target hypersphere of all points satisfying the L_2 norm constraint. This yields the following factorization:

$$S^{(\lambda_1,\lambda_2)}_{\geq 0} = \mathbb{R}^n_{\geq 0} \cap H \cap K =: D.$$

For computation of projections onto an intersection of a finite number of closed and convex sets, it is enough to perform alternating projections onto the members of the intersection (Deutsch, 2001). As K is clearly non-convex, this general approach has to be altered to work in this specific setup.

First, consider $L := H \cap K$, which denotes the intersection of the L_1 norm target hyperplane and the L_2 norm target hypersphere. L essentially possesses the structure of a hypercircle, that is, all points in L lie also in H and there is a central point $m \in H$ and a real number $\rho \geq 0$ such that all points in L have squared distance ρ from m. It will be shown in Appendix C that $m = \lambda_1/n \cdot e \in \mathbb{R}^n$ and $\rho = \lambda_2^2 - \lambda_1^2/n$. The intersection of the non-negative orthant with the L_1 norm hyperplane, $C := \mathbb{R}^n_{\geq 0} \cap H$, is a scaled canonical simplex. Its barycenter coincides with the barycenter m of L. Finally, for an index set $I \subseteq \{1, \ldots, n\}$ let $L_I := \{ a \in L \mid a_i = 0 \text{ for all } i \notin I \}$ denote the subset of points from L, where all coordinates with index not in I vanish. Its barycenter is given by $m_I = \lambda_1/d \cdot \sum_{i \in I} e_i \in \mathbb{R}^n$. With these preparations, a simple algorithm can be proposed; it computes the sparseness-enforcing projection with respect to a constraint induced by Hoyer's sparseness measure σ.

Theorem 2 *For every $x \in \mathbb{R}^n$, Algorithm 1 computes an element from* $\mathrm{proj}_D(x)$. *If $r \neq m$ after line 1 and $r \neq m_I$ after line 4 in all iterations, then* $\mathrm{proj}_D(x)$ *is a singleton.*

Algorithm 1: Proposed algorithm for computing the sparseness-enforcing projection operator for Hoyer's sparseness measure σ.

Input: $x \in \mathbb{R}^n$ and $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$ with $\lambda_2 \leq \lambda_1 \leq \sqrt{n}\lambda_2$.
Output: $s \in \mathrm{proj}_D(x)$ where $D = S_{\geq 0}^{(\lambda_1,\lambda_2)}$.

// Project onto target hyperplane H and target hypercircle L.
1 $r := \mathrm{proj}_H(x)$;
2 $s \in \mathrm{proj}_L(r)$;
// Perform alternating projections until feasible solution is found.
3 **while** $s \notin \mathbb{R}^n_{\geq 0}$ **do**
  // Project onto scaled canonical simplex C.
4 $r := \mathrm{proj}_C(s)$;
  // Project onto L keeping already vanished coordinates at zero.
5 $s \in \mathrm{proj}_{L_I}(r)$ where $I := \{\, i \in \{1,\ldots,n\} \mid r_i \neq 0 \,\}$;
6 **end**

As already pointed out, the idea of Algorithm 1 is that projections onto D can be computed by alternating projections onto the geometric structures just defined. The rigorous proof of correctness from Appendix C proceeds by showing that the set of solutions is not tampered by projection onto the intermediate structures H, C, L and L_I. Because of the non-convexity of L and L_I, the relation between these sets and the simplex C is non-trivial and needs long arguments to be described further, see especially Lemma 26 and Corollary 27.

The projection onto the hyperplane H is straightforward and discussed in Section C.1.1. As L is essentially a hypersphere embedded in a subspace H of $\mathbb{R}^n$, projections of points from H onto L are achieved by shifting and scaling, see Section C.1.2. The alternating projection onto H and L in the beginning of Algorithm 1 make the result of the projection onto D invariant to positive scaling and arbitrary shifting of the argument, as shown in Corollary 19. This is especially useful in practice, alleviating the need for certain pre-processing methods. The formula for projections onto L can be generalized for projections onto L_I for an index set $I \subseteq \{1,\ldots,n\}$, by keeping already vanished coordinates at zero, see Section C.3.

Projections onto the simplex C are more involved and discussed at length in Section C.2. The most relevant result is that if $x \in \mathbb{R}^n \setminus C$, then there exists a separator $\hat{t} \in \mathbb{R}$ such that $p := \mathrm{proj}_C(x) = \max(x - \hat{t} \cdot e,\, 0)$, where the maximum is taken element-wise (Chen and Ye, 2011). In the cases considered in this paper it is always $\hat{t} \geq 0$ as shown in Lemma 28. This implies that all entries in x that are less than $\hat{t}$ do not survive the projection, and hence the L_0 pseudo-norm of x is strictly greater than that of p. The simplex projection therefore enhances sparseness.

The separator $\hat{t}$ and the number of nonzero entries in the projection onto C can be computed with Algorithm 2, which is an adapted version of the algorithm of Chen and Ye (2011). In line 1, S_n denotes the symmetric group and P_τ denotes the permutation matrix associated with a permutation $\tau \in S_n$. The algorithm works by sorting its argument x and then determining $\hat{t}$ as the mean value of the largest entries of x minus the target L_1 norm λ_1. The number of relevant entries for computation of $\hat{t}$ is equal to the L_0 pseudo-norm of the projection and is found by trying all feasible values, starting with the largest ones. The computational complexity of Algorithm 2 is dominated by sorting the input vector and is thus quasilinear.

Algorithm 2: Computation of information for performing projections onto C, which is a scaled canonical simplex. This is an adapted version of the algorithm of Chen and Ye (2011).

Input: $x \in \mathbb{R}^n \setminus C$ and $\lambda_1 \in \mathbb{R}_{>0}$.
Output: $(\hat{t},\, d) \in \mathbb{R} \times \mathbb{N}$ such that $\mathrm{proj}_C(x) = \max(x - \hat{t} \cdot e,\, 0)$ and $\|\mathrm{proj}_C(x)\|_0 = d$.

// Sort the input vector in descending order.
Let $\tau \in S_n$ such that $x_{\tau(1)} \geq \cdots \geq x_{\tau(n)}$ and $y := P_\tau x \in \mathbb{R}^n$;
// Find the only feasible separator $\hat{t}$.
$s := 0$;
for $i := 1$ **to** $n-1$ **do**
    $s := s + y_i$; $t := \frac{s-\lambda_1}{i}$;
    if $t \geq y_{i+1}$ **then return** $(t,\, i)$;
end
$s := s + y_n$; $t := \frac{s-\lambda_1}{n}$; **return** $(t,\, n)$;

Algorithm 3: Explicit and optimized variant of Algorithm 1.

Input: $x \in \mathbb{R}^n$ and $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$ with $\lambda_2 \leq \lambda_1 \leq \sqrt{n}\lambda_2$.
Output: $s \in \mathrm{proj}_D(x)$ where $D = S_{\geq 0}^{(\lambda_1, \lambda_2)}$.

procedure proj_L($y \in \mathbb{R}^d$)
    $\rho := \lambda_2^2 - \lambda_1^2/d$; // Compute squared radius of L_I (Lemma 15).
    $\varphi := \sum_{i=1}^d y_i^2 - \lambda_1^2/d$; // $\varphi := \|y - m_I\|_2^2$ (Remark 14).
    if $\varphi = 0$ **then**
        $(y_1,\, \ldots,\, y_{d-1})^T := \lambda_1/d + \sqrt{\rho}/\sqrt{d(d-1)}$; // y equals the barycenter of L_I,
        $y_d := \lambda_1/d - \sqrt{\rho(d-1)}/\sqrt{d}$; // pick a sorted projection (Remark 18).
    else $y := \lambda_1/d \cdot e + \sqrt{\rho/\varphi} \cdot (y - \lambda_1/d \cdot e)$; // Pick unique projection (Lemma 17).
end
// Beginning of main body.
Let $\tau \in S_n$ such that $x_{\tau(1)} \geq \cdots \geq x_{\tau(n)}$ and $y := P_\tau x \in \mathbb{R}^n$; // Sort the input vector.
$y := y + 1/n \cdot (\lambda_1 - \sum_{i=1}^n y_i)\, e$; // Project onto H (Lemma 13).
proj_L$(y_1, \ldots, y_n)$; // Project in-place onto L.
// Perform alternating projections until feasible solution is found.
$d := n$; // Store current number of relevant entries of y.
while $(y_1,\, \ldots,\, y_d)^T \notin \mathbb{R}_{\geq 0}^d$ **do**
    $(\hat{t},\, d) :=$ proj_C$(y_1, \ldots, y_d)$; // This is carried out by Algorithm 2.
    $(y_1,\, \ldots,\, y_d)^T := (y_1,\, \ldots,\, y_d)^T - \hat{t}$; // Project onto C (Proposition 24).
    proj_L$(y_1, \ldots, y_d)$; // Project onto L_I where $I = \{1, \ldots, d\}$ (Lemma 30).
end
// Undo sorting permutation and set remaining entries to zero.
$s \in \{0\}^n$; **for** $i := 1$ **to** d **do** $s_{\tau(i)} := y_i$;

2.2 Optimized Variant

Because of the permutation-invariance of the sets involved in the projections, it is enough to sort the vector that is to be projected onto D once. This guarantees that the working vector that emerges from subsequent projections is sorted also. No additional sorting has then to be carried out when using Algorithm 2 for projections onto C. This additionally has the side effect that the non-vanishing entries of the working vector are always concentrated in its first entries. Hence all relevant information can always be stored in a small unit-stride array, to which access is more efficient than to a large sparse array. Further, the index set I of non-vanishing entries in the working vector is always of the form $I = \{1, \ldots, d\}$, where d is the number of nonzero entries.

Algorithm 3 is a variant of Algorithm 1 where these optimizations were applied, and where the explicit formulas for the intermediate projections were used. The following result, which is proved in Appendix C, states that both algorithms always compute the same result:

Theorem 3 *Algorithm 1 is equivalent to Algorithm 3.*

Projections onto C increase the amount of vanishing entries in the working vector, which is of finite dimension n. Hence, at most n alternating projections are carried out, and the algorithm terminates in finite time. Further, the complexity of each iteration is at most linear in the L_0 pseudo-norm of the working vector. The theoretic overall computational complexity is thus at most quadratic in problem dimensionality n.

2.3 Comparison with Hoyer's Original Algorithm

The original algorithm for the sparseness-enforcing projection operator proposed by Hoyer (2004) is hard to understand, and correctness has been proved by Theis et al. (2005) in a special case only. A simple alternative has been proposed with Algorithm 1 in this paper. Based on the symmetries induced by Hoyer's sparseness measure σ and by exploiting the projection onto a simplex, an improved method was given in Algorithm 3.

The improved algorithm proposed in this paper always requires at most the same number of iterations of alternating projections as the original algorithm. The original algorithm uses a projection onto the non-negative orthant $\mathbb{R}^n_{\geq 0}$ to achieve vanishing coordinates in the working vector. This operation can be written as $\mathrm{proj}_{\mathbb{R}^n_{\geq 0}}(x) = \max(x,\ 0)$. In the improved algorithm, a simplex projection is used for this purpose, expressed formally as $\mathrm{proj}_C(x) = \max\left(x - \hat{t} \cdot e,\ 0\right)$ with $\hat{t} \in \mathbb{R}$ chosen accordingly. Due to the theoretical results on simplex geometry from Section C.2 and their application in Lemma 28 in Section C.3, the number $\hat{t}$ is always non-negative. Therefore, at least the same amount of entries is set to zero in the simplex projection compared to the projection onto the non-negative orthant, see also Corollary 29. Hence with induction for the number of non-vanishing entries in the working vector, the number of iterations the proposed algorithm needs to terminate is bounded by the number of iterations the original method needs to terminate given the same input.

The experimental determination of an estimate of the number of iterations required was carried out as follows. Random vectors with sparseness 0.15 were sampled and their sparse projections were computed using the respective algorithms, to gain the best normalized approximations with a target sparseness degree of $\sigma^* := 0.90$. For both algorithms the very same vectors were used as input. During the run-time of the algorithms, the number of iterations that were necessary to compute the result were counted. Additionally, the number of nonzero entries in the working vector was

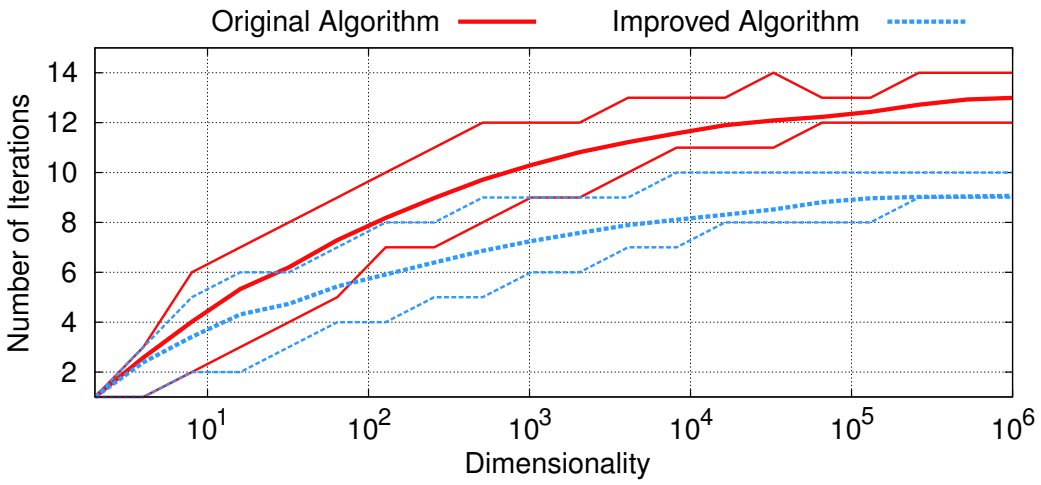


Figure 1: Comparison of the number of iterations of the original algorithm for the projection onto D with the improved version as proposed in this paper. The sparseness-enforcing projection with target sparseness 0.90 was carried out for input vectors of sparseness 0.15. The thick lines indicate the mean number of iterations required for the projection, and the thin lines indicate the minimum and maximum number of iterations, respectively. Even for input vectors with a million entries, less than 14 iterations are required to find the projection. With the improved algorithm, this reduces to at most 10 iterations.

recorded in each iteration. This was done for different dimensionalities, and for each dimensionality 1000 vectors were sampled.

Figure 1 shows statistics on the number of iterations the algorithms needed to terminate. As was already observed by Hoyer (2004), the number of required iterations grows very slowly with problem dimensionality. For $n = 10^6$, only between 12 and 14 iterations were needed with the original algorithm to compute the result. With Algorithm 3, this can be improved to requiring 9 to 10 iterations, which amounts to roughly 30% less iterations. Due to the small slope in the number of required iterations, it can be conjectured that this quantity is at most logarithmic in problem dimensionality n. If this applies, the complexity of Algorithm 3 is at most quasilinear. Because the input vector is sorted in the beginning, it is also not possible to fall below this complexity class.

The progress of working dimensionality reduction for problem dimensionality $n = 1000$ is depicted in Figure 2, averaged over the 1000 input vectors from the experiment. After the first iteration, that is after projecting onto H and L, the working dimensionality still matches the input dimensionality. Starting with the second iteration, dimensions are discarded by projecting onto $\mathbb{R}^n_{\geq 0}$ in the original algorithm and onto C in the improved variant, which yields vanishing entries in the working vectors. With the original algorithm, in the mean 54% of all entries are nonzero after the second iteration, while with the improved algorithm only 27% of the original 1000 dimensions remain in the mean. This trend continues in subsequent iterations such that the final working dimensionality is reached more quickly with the algorithm proposed in this paper. Although using Algorithm 2 to perform the simplex projection is more expensive than just setting negative entries to zero in the orthant projection, the overhead quickly amortizes because of the boost in dimensionality reduction.

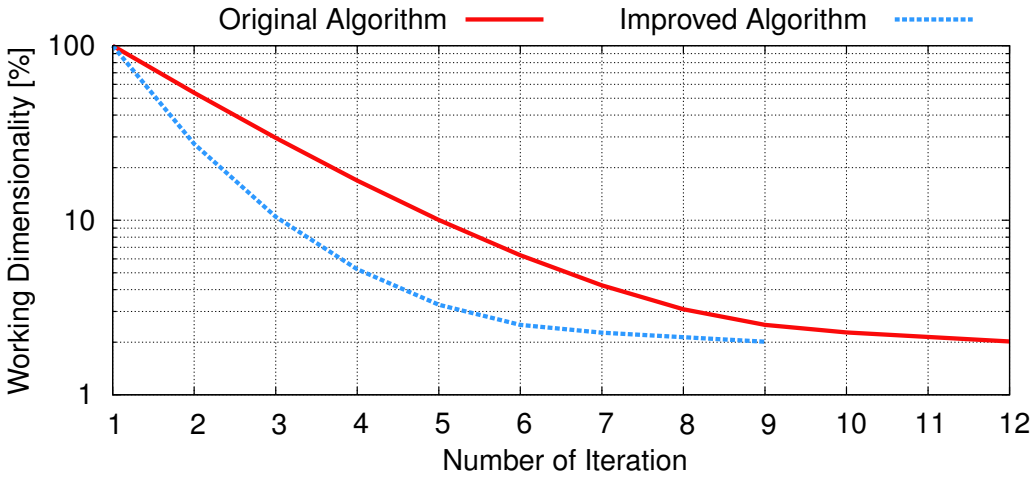


Figure 2: Comparison of the number of non-vanishing entries in the working vectors of the original algorithm and the improved algorithm during run-time. The algorithms were run with input vectors of dimensionality 1000 and initial sparseness 0.15 to compute projections with sparseness 0.90. Standard deviations were always less than 1%; they were omitted in the plot to avoid clutter. The algorithm proposed in this paper reduces dimensionality more quickly and terminates earlier than the original algorithm.

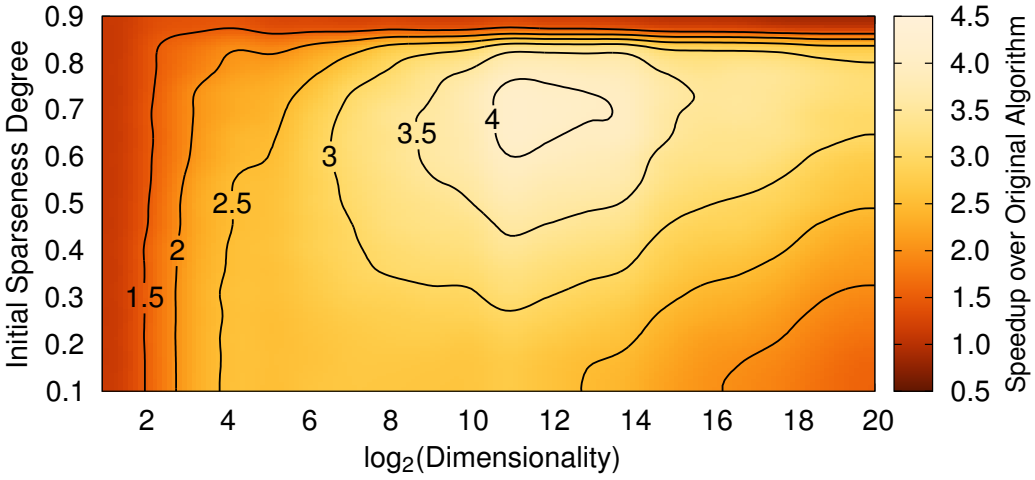


Figure 3: Ratio of the computation time of the original algorithm and the improved algorithm for a variety of input dimensionality and initial vector sparseness. Numbers greater than one indicate parameterizations where the proposed algorithm is more efficient than the original one. There is a large region where the speedup is decent.

For determination of the relative speedup incorporated with both the simplex projection and the access to unit-stride arrays due to the permutation-invariance, both algorithms were implemented as C++ programs using an optimized implementation of the BLAS library for carrying out the vector operations. The employed processor was an Intel Core i7-990X. For a range of different dimensionalities, a set of vectors with varying initial sparseness were sampled. The number of the vectors for every pair of dimensionality and initial sparseness was chosen such that the processing time of the algorithms was several orders of magnitudes greater than the latency time of the operation system. Then the absolute time needed for the algorithms to compute the projections with a target sparseness of 0.90 were measured, and their ratio was taken to compute the relative speedup. The results of this experiment are depicted in Figure 3. It is evident that the maximum speedup is achieved for vectors with a dimensionality between 2^9 and 2^{15}, and an initial sparseness greater than 0.40. For low initial sparseness, as is achieved by randomly sampled vectors, a speedup of about 2.5 can be achieved for a broad spectrum of dimensionality between 2^4 and 2^{13}.

The improvements to the original algorithm are thus not only theoretical, but also noticeable in practice. The speedup is especially useful when the projection is used as a neuronal transfer function in a classifier as proposed in Section 3, because then the computational complexity of the prediction of class membership of unknown samples can be reduced.

2.4 Function Definition and Differentiability

It is clear from Theorem 2 that the projection onto D is unique almost everywhere. Therefore the set $R := \{\, x \in \mathbb{R}^n \mid |\mathrm{proj}_D(x)| \neq 1 \,\}$ is a null set. However, $R \neq \varnothing$ as for example the projection is not unique for vectors where all entries are identical. In other words, for $x := \xi e \in \mathbb{R}^n$ for some $\xi \in \mathbb{R}$ follows $\mathrm{proj}_H(x) = m$ and $\mathrm{proj}_L(m) = L$. If $n = 2$ a possible solution is given by $(\alpha, \beta)^T \in \mathrm{proj}_D(x)$ with α and β given as stated in Remark 18, as in this case α and β are positive. Additionally, another solution is given by $(\beta, \alpha)^T \in \mathrm{proj}_D(x)$ which is unequal to the other solution because of $\alpha \neq \beta$. A similar argument can be used to show non-uniqueness for all $n \geq 2$. As R is merely a small set, non-uniqueness is not an issue in practical applications.

The sparseness-enforcing projection operator that is restricted to non-negative solutions can thus be cast almost everywhere as a function

$$\pi_{\geq 0} \colon \mathbb{R}^n \setminus R \to D, \qquad x \mapsto \mathrm{proj}_D(x).$$

Exploiting reflection-invariance implies that the unrestricted variant of the projection

$$\pi \colon \mathbb{R}^n \setminus R \to S^{(\lambda_1, \lambda_2)}, \qquad x \mapsto s \circ \pi_{\geq 0}\left(|x|\right),$$

is well-defined, where $s \in \{\pm 1\}^n$ is given as described in Lemma 11. Note that computation of $\pi_{\geq 0}$ is a crucial prerequisite to computation of the unrestricted variant π. It will be used exclusively in Section 3 because non-negativity is not necessary in the application proposed there.

If π or $\pi_{\geq 0}$ is employed in an objective function that is to be optimized, the information whether these functions are differentiable is crucial for selecting an optimization strategy. As an example, consider once more projections onto $Z := \{\, x \in \mathbb{R}^n \mid \|x\|_0 = \kappa \,\}$ where $\kappa \in \mathbb{N}$ is a constant. It was already mentioned in Section 2 that the projection onto Z consists simply of zeroing out the elements that are smallest in absolute value. Let $x \in \mathbb{R}^n$ be a point and let $\tau \in S_n$ be a permutation such that $\left|x_{\tau(1)}\right| \geq \cdots \geq \left|x_{\tau(n)}\right|$. Clearly, if $\left|x_{\tau(\kappa)}\right| \neq \left|x_{\tau(\kappa+1)}\right|$ then $\mathrm{proj}_Z(x) = y$ where $y_i = x_i$ for $i \in \{\tau(1), \ldots, \tau(\kappa)\}$ and $y_i = 0$ for $i \in \{\tau(\kappa+1), \ldots, \tau(n)\}$. Moreover, when $\left|x_{\tau(\kappa)}\right| \neq \left|x_{\tau(\kappa+1)}\right|$ then

there exists a neighborhood U of x such that $\mathrm{proj}_Z(s) = \sum_{i=1}^{\kappa} s_{\tau(i)} e_{\tau(i)}$ for all $s \in U$. With this closed-form expression, $s \mapsto \mathrm{proj}_Z(s)$ is differentiable in x with gradient $\partial \mathrm{proj}_Z(x) / \partial x = \mathrm{diag}\left(\sum_{i=1}^{\kappa} e_{\tau(i)}\right)$, that is the identity matrix where the entries on the diagonal belonging to small absolute values of x have been zeroed out. If the requirement on x is not fulfilled, then a small distortion of x is sufficient to find a point in which the projection onto Z is differentiable.

In contrast to the L_0 projection, differentiability of π and $\pi_{\geq 0}$ is non-trivial. A full-length discussion is given in Appendix D, and concludes that both π and $\pi_{\geq 0}$ are differentiable almost everywhere. It is more efficient when only the product of the gradient with an arbitrary vector needs to be computed, see Corollary 36. Such an expression emerges in a natural way by application of the chain rule to an objective function where the sparseness-enforcing projection is used. In practice this weaker form is thus mostly no restriction and preferable for efficiency reasons over the more general complete gradient as given in Theorem 35.

The derivative of $\pi_{\geq 0}$ is obtained by exploiting the structure of Algorithm 1. Because the projection onto D is essentially a composition of projections onto H, C, L and L_I, the overall gradient can be computed using the chain rule. The gradients of the intermediate projections are simple expressions and can be combined to yield one matrix for each iteration of alternating projections. Since these iteration gradients are basically sums of dyadic products, their product with an arbitrary vector can be computed by primitive vector operations. With matrix product associativity, this process can be repeated to efficiently compute the product of the gradient of $\pi_{\geq 0}$ with an arbitrary vector. For this, it is sufficient to record some intermediate quantities during execution of Algorithm 3, which does not add any major overhead to the algorithm itself. The gradient of the unrestricted variant π can be deduced in a straightforward way from the gradient of $\pi_{\geq 0}$ because of their close relationship.

3. Sparse Activity and Sparse Connectivity in Supervised Learning

The sparseness-enforcing projection operator can be cast almost everywhere as vector-valued function π, which is differentiable almost everywhere, see Section 2.4. This section proposes a hybrid of an auto-encoder network and a two-layer neural network, where the sparseness projection is employed as a neuronal transfer function. The proposed model is called supervised online auto-encoder (SOAE) and is intended for classification by means of a neural network that features sparse activity and sparse connectivity. Because of the analytical properties of the sparseness-enforcing projection operator, the model can be optimized end-to-end using gradient-based methods.

3.1 Architecture

Figure 4 depicts the data flow in the proposed model. There is one module for reconstruction capabilities and one module for classification capabilities. The *reconstruction module*, depicted on the left of Figure 4, operates by converting an input sample $x \in \mathbb{R}^d$ into an *internal representation* $h \in \mathbb{R}^n$, and then computing an approximation $\tilde{x} \in \mathbb{R}^d$ to the original input sample. In doing so, the product $u \in \mathbb{R}^n$ of the input sample with a *matrix of bases* $W \in \mathbb{R}^{d \times n}$ is computed, and a transfer function $f \colon \mathbb{R}^n \to \mathbb{R}^n$ is applied. For sparse activity, f can be chosen to be the sparseness-enforcing projection operator π or the projection with respect to the L_0 pseudo-norm. This guarantees that the internal representation is sparsely populated and close to u. The reconstruction is achieved like in a linear generative model, by multiplication of the matrix of bases with the internal representation. Hence the same matrix W is used for both encoding and decoding, rendering the reconstruction module symmetric, or in other words with tied weights. This approach is similar to principal com-

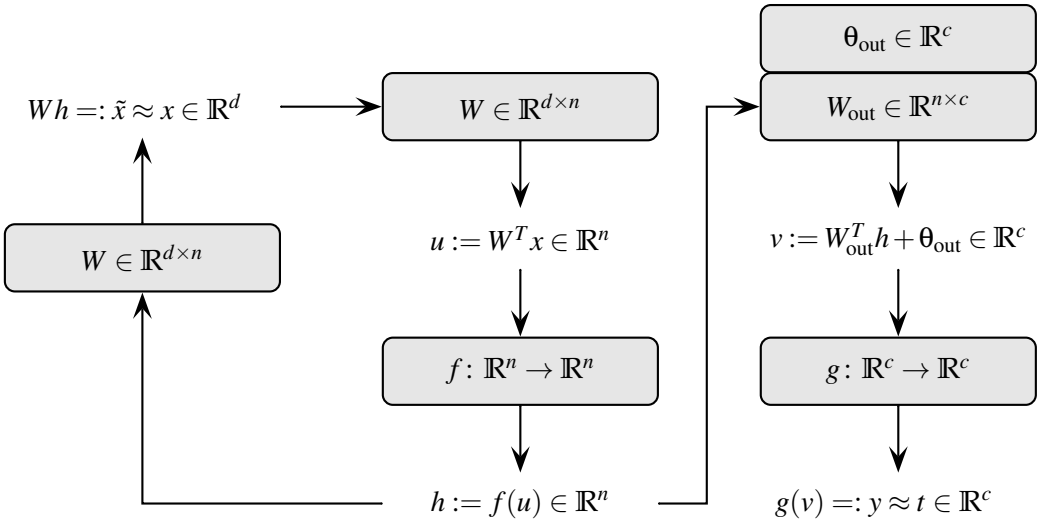


Figure 4: Architecture and data flow of supervised online auto-encoder (SOAE). The circle on the left, mapping from an input sample x to its approximation $\tilde{x}$ comprises the reconstruction module. The classification module consists of the mapping from x to the classification decision y. The matrix of bases W shall be sparsely populated to account for the sparse connectivity property. If the transfer function f is set to the sparseness projection, the internal representation h will be sparsely populated, fulfilling the sparse activity property.

ponent analysis (Hotelling, 1933), restricted Boltzmann machines for deep auto-encoder networks (Hinton et al., 2006) and to sparse encoding symmetric machine (Ranzato et al., 2008).

By enforcing W to be sparsely populated, the sparse connectivity property holds as well. More formally, the aim is that $\sigma(We_i) = \sigma_W$ holds for all $i \in \{1, \ldots, n\}$, where $\sigma_W \in (0,\ 1)$ is the target degree of connectivity sparseness and We_i is the i-th column of W. This condition was adopted from non-negative matrix factorization with sparseness constraints (Hoyer, 2004). In the context of neural networks, the synaptic weights of individual neurons are stored in the columns of the weight matrix W. The interpretation of this formal sparseness constraint is then that each neuron is only allowed to be sparsely connected with the input layer.

The *classification module* is shown on the right-hand side of Figure 4. It computes a *classification decision* $y \in \mathbb{R}^c$ by feeding h through a one-layer neural network. The network output y is yielded through computation of the product with a matrix of weights $W_{\text{out}} \in \mathbb{R}^{n \times c}$, addition of a threshold vector $\theta_{\text{out}} \in \mathbb{R}^c$ and application of a transfer function $g\colon \mathbb{R}^c \to \mathbb{R}^c$. This module shares the inference of the internal representation with the reconstruction module, which can also be considered a one-layer neural network. Therefore the entire processing path from x to y forms a two-layer neural network (Rumelhart et al., 1986), where W stores the synaptic weights of the hidden layer, and W_{out} and θ_{out} are the parameters of the output layer.

The input sample x shall be approximated by $\tilde{x}$, and the target vector for classification $t \in \mathbb{R}^c$ shall be approximated by y. This is achieved by optimization of the parameters of SOAE, that is the quantities W, W_{out} and θ_{out}. The goodness of the approximation $x \approx \tilde{x}$ is estimated using

a differentiable similarity measure $s_R \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and the approximation $y \approx t$ is assessed by another similarity measure $s_C \colon \mathbb{R}^c \times \mathbb{R}^c \to \mathbb{R}$. For minimizing the deviation in both approximations, the objective function

$$E_{\text{SOAE}}(W,\, W_{\text{out}},\, \theta_{\text{out}}) := (1-\alpha) \cdot s_R(\tilde{x},\, x) + \alpha \cdot s_C(y,\, t)$$

shall be optimized, where $\alpha \in [0,\, 1]$ controls the trade-off between reconstruction and classification capabilities. To incorporate sparse connectivity, feasible solutions are restricted to fulfill $\sigma(We_i) = \sigma_W$ for all $i \in \{1, \ldots, n\}$. If $\alpha = 0$, then SOAE is identical to a symmetric auto-encoder network with sparse activity and sparse connectivity. In the case of $\alpha = 1$, SOAE forms a two-layer neural network for classification with a sparsely connected hidden layer and where the activity in the hidden layer is sparse. The parameter α can also be used to blend continuously between these two extremes. Note that $\tilde{x}$ only depends on W but not on W_{out} or θ_{out}, but y depends on W, W_{out} and θ_{out}. Hence W_{out} and θ_{out} are only relevant when $\alpha > 0$, whereas W is essential for all choices of α.

An appropriate choice for s_R is the correlation coefficient (see for example Rodgers and Nicewander, 1988), because it is normed to values in the interval $[-1,\, 1]$, invariant to affine-linear transformations, and differentiable. If f is set to π, then a model that is invariant to the concrete scaling and shifting of the occurring quantities can be yielded. This follows because π is also invariant to such transformations, see Corollary 19. The similarity measure for classification capabilities s_C is chosen to be the cross-entropy error function (Bishop, 1995), which was shown empirically by Simard et al. (2003) to induce better classification capabilities than the mean squared error function. The softmax transfer function (Bishop, 1995) is used as transfer function g of the output layer. It provides a natural pairing together with the cross-entropy error function (Dunne and Campbell, 1997) and supports multi-class classification.

3.2 Learning Algorithm

The proposed optimization algorithm for minimization of the objective function E_{SOAE} is projected gradient descent (Bertsekas, 1999). Here, each update to the degrees of freedom is followed by application of the sparseness projection to the columns of W to enforce sparse connectivity. There are theoretical results on the convergence of projected gradient methods when projections are carried out onto convex sets (Bertsekas, 1999), but here the target set for projection is non-convex. Nevertheless, the experiments described below show that projected gradient descent is an adequate heuristic in the situation of the SOAE framework to tune the network parameters. For completeness, the gradients of E_{SOAE} with respect to the network parameters are given in Appendix E. Update steps are carried out after every presentation of a pair of an input sample and associated target vector. This online learning procedure results in faster learning and improves generalization capabilities over batch learning (Wilson and Martinez, 2003; Bottou and LeCun, 2004).

A learning set with samples from $\mathbb{R}^d$ and associated target vectors from $\{0,1\}^c$ as one-of-c-codes is input to the algorithm. The dimensionality of the internal representation n and the target degree of sparseness with respect to the connectivity $\sigma_W \in (0,\, 1)$ are parameters of the algorithm. Sparseness of connectivity increases for larger σ_W, as Hoyer's sparseness measure is employed in the definition of the set of feasible solutions.

Two possible choices for the hidden layer's transfer function f to achieve sparse activity were discussed in this paper. One possibility is to carry out the projection with respect to the L_0 pseudo-norm. The more sophisticated method is to use the unrestricted sparseness-enforcing projection

operator π with respect to Hoyer's sparseness measure σ, which can be carried out by Algorithm 3. In both cases, a target degree for sparse activity is a parameter of the learning algorithm. In case of the L_0 projection, this sparseness degree is denoted by $\kappa \in \{1, \ldots, n\}$, and sparseness increases with smaller values of it. For the σ projection, $\sigma_H \in (0,\ 1)$ is used, where larger values indicate more sparse activity.

Initialization of the columns of W is achieved by selecting a random subset of the learning set, similar to the initialization of radial basis function networks (Bishop, 1995). This ensures significant activity of the hidden layer from the very start, resulting in strong gradients and therefore reducing training time. The parameters of the output layer, that is W_{out} and θ_{out}, are initialized by sampling from a zero-mean Gaussian distribution with a standard deviation of $1/100$.

In every epoch, a randomly selected subset of samples and associated target vectors from the learning set is used for stochastic gradient descent to update W, W_{out} and θ_{out}. The results from Appendix E can be used to efficiently compute the gradient of the objective function. There, the gradient for the transfer function f only emerges as a product with a vector. The gradient for the L_0 projection is trivial and was given as an example in Section 2.4. If f is Hoyer's sparseness-enforcing projection operator, it is possible to exploit that only the product of the gradient with a vector is needed. In this case, it is more efficient to compute the result of the multiplication implicitly using Corollary 36 and thus avoid the computation of the entire gradient of π.

After every epoch, a sparseness projection is applied to the columns of W. This guarantees that $\sigma(We_i) = \sigma_W$ holds for all $i \in \{1, \ldots, n\}$, and therefore the sparse connectivity property is fulfilled. The trade-off variable α which controls the weight of the reconstruction and the classification term is adjusted according to $\alpha(\nu) := 1 - \exp(-\nu/100)$, where $\nu \in \mathbb{N}$ denotes the number of the current epoch. Thus α starts at zero, increases slowly and asymptotically reaches one. The emphasis at the beginning of the optimization is thus on reconstruction capabilities. Subsequently, classification capabilities are incorporated slowly, and in the final phase of training classification capabilities exclusively are optimized. This continuous variant of unsupervised pre-training (Hinton et al., 2006) leads to parameters in the vicinity of a good minimizer for classification capabilities before classification is preferred over reconstruction through the trade-off parameter α. Compared to the choice $\alpha \equiv 1$ this strategy helps to stabilize the trajectory in parameter space and makes the objective function values settle down more quickly, such that the termination criterion is satisfied earlier.

3.3 Description of Experiments

To assess the classification capabilities and the impact of sparse activity and sparse connectivity, the MNIST database of handwritten digits (LeCun and Cortes, 1998) was employed. It is a popular benchmark data set for classification algorithms, and numerous results with respect to this data set are reported in the literature. The database consists of 70 000 samples, divided into a learning set of 60 000 samples and an evaluation set of 10 000 samples. Each sample represents a digit of size 28×28 pixels and has a class label from $\{0, \ldots, 9\}$ associated with it. Therefore the input and output dimensionalities are $d := 28^2 = 784$ and $c := 10$, respectively. The classification error is given in percent of all 10 000 evaluation samples, hence 0.01% corresponds to a single misclassified digit.

For generation of the original data set, the placement of the digits has been achieved based on their barycenter (LeCun and Cortes, 1998). Because of sampling and rounding errors, the localization uncertainty can hence be assumed to be less than one pixel in both directions. To account for this uncertainty, the learning set was augmented by jittering each sample in each of eight possible

directions by one pixel, yielding 540 000 samples for learning in total. The evaluation set was left unchanged to yield results that can be compared to the literature. As noted by Hinton et al. (2006), the learning problem is no more permutation-invariant due to the jittering, as information on the neighborhood of the pixels is implicitly incorporated in the learning set.

However, classification results improve dramatically when such prior knowledge is used. This was demonstrated by Schölkopf (1997) using the virtual support vector method, which improved a support vector machine with polynomial kernel of degree five from an error of 1.4% to 1.0% by jittering the support vectors by one pixel in four principal directions. This result was extended by DeCoste and Schölkopf (2002), where a support vector machine with a polynomial kernel of degree nine was improved from an error of 1.22% to 0.68% by jittering in all possible eight directions. Further improvements can be achieved by generating artificial training samples using elastic distortions (Simard et al., 2003). This reduced the error of a two-layer neural network with 800 hidden units to 0.7%, compared to the 1.1% error yielded when training on samples created by affine distortions. Very big and very deep neural networks possess a large number of adaptable weights. In conjunction with elastic and affine distortions such neural networks can yield errors as low as 0.35% (Cireşan et al., 2010). The current record error of 0.23% is held by an approach that combines distorted samples with a committee of convolutional neural networks (Cireşan et al., 2012). This is an architecture that has been optimized exclusively for input data that represents images, that is where the neighborhood of the pixels is hard-wired in the classifier. To allow for a plain evaluation that does not depend on additional parameters for creating artificial samples, the jittered learning set with 540 000 samples is used throughout this paper.

The experimental methodology was as follows. The number of hidden units was chosen to be $n := 1000$ in all experiments that are described below. This is an increased number compared to the 800 hidden units employed by Simard et al. (2003), but promises to yield better results when an adequate number of learning samples is used. As all tested learning algorithms are essentially gradient descent methods, an initial step size had to be chosen. For each candidate step size, five runs of a two-fold cross validation were carried out on the learning set. Then, for each step size the median of the ten resulting classification errors was computed. The winning step size was then determined to be the one that achieved a minimum median of classification errors.

In every epoch, 21 600 samples were randomly chosen from the learning set and presented to the network. This number of samples was chosen as it is $1/25$-th of the jittered learning set. The step size was multiplicatively annealed using a factor of 0.999 after every epoch. Optimization was terminated once the relative change in the objective function became very small and no more significant progress on the learning set could be observed. The resulting classifiers were then applied to the evaluation set, and misclassifications were counted.

3.4 Experimental Results

Two variants of the supervised online auto-encoder architecture as proposed in this section were trained on the augmented learning set. In both variants, the target degree of sparse connectivity was set to $\sigma_W := 0.75$. This choice was made because 96% of all samples in the learning set possess a sparseness which is less than 0.75. Therefore, the resulting bases are forced to be truly sparsely connected compared to the sparseness of the digits.

The first variant is denoted by SOAE-σ. Here, the sparseness-enforcing projection operator π was used as transfer function f in the hidden layer. Target degrees of sparse activity σ_H with respect

to Hoyer's sparseness measure σ were chosen from the interval [0.20, 0.95] in steps of size 0.05. This variant was then trained on the jittered learning set using the method described in Section 3.2. For every value of σ_H, the resulting sparseness of activity was measured after training using the L_0 pseudo-norm. For this, each sample of the learning set was presented to the networks, and the number of active units in the hidden layer was counted. Figure 5 shows the resulting mean value and standard deviation of sparse activity. If $\sigma_H = 0.20$ is chosen, then in the mean about 800 of the total 1000 hidden units are active upon presentation of a sample from the learning set. For $\sigma_H = 0.80$ only one hundred units are active at any one time, and for $\sigma_H = 0.95$ there are only eleven active units. The standard deviation of the activity decreases when sparseness increases, hence the mapping from σ_H to the resulting number of active units becomes more accurate.

The second variant, denoted SOAE-L_0, differs from SOAE-σ in that the projection with respect to the L_0 pseudo-norm as transfer function f was used. The target sparseness of activity is given by a parameter $\kappa \in \{1, \ldots, n\}$, which controls the exact number of units that are allowed to be active at any one time. For the experiments, the values for κ were chosen to match the mean activities from the SOAE-σ experiments. This way the results of both variants can be compared based on a unified value of activity sparseness. The results are depicted in Figure 6. Usage of the σ projection consequently outperforms the L_0 projection for all sparseness degrees. Even for high sparseness of activity, that is when only about ten percent of the units are allowed to be active at any one time, good classification capabilities can be obtained with SOAE-σ. For $\kappa \in [242, 558]$, the classification results of SOAE-L_0 reach an optimum. SOAE-σ is more robust, as classification capabilities first begin to collapse when sparseness is below 5%, whereas SOAE-L_0 starts to degenerate when sparseness falls below 20%. For $\sigma_H \in [0.45, 0.85]$, roughly translating to between 5% and 50% activity, about equal classification performance is achieved using SOAE-σ.

It can thus be concluded that using the sparseness-enforcing projection operator as described in this paper yields better results than when the simple L_0 projection is used to achieve sparse activity. To assess the benefit more precisely and to investigate the effect of individual factors, several comparative experiments have been carried out. A summary of these experiments and their outcome is given in Table 1. The variants SOAE-σ and SOAE-L_0 denote the entirety of the respective experiments where sparseness of activity lies in the intervals described above, that is $\sigma_H \in [0.45, 0.85]$ and $\kappa \in [242, 558]$, respectively. Using these intervals, SOAE-σ and SOAE-L_0 achieved a median error of 0.75% and 0.82% on the evaluation set, respectively. Variant SOAE-σ-conn is essentially equal to SOAE-σ, except for sparse connectivity not being incorporated. Sparseness of activity here was also chosen to be $\sigma_H \in [0.45, 0.85]$, which resulted in about equal classification results over the entire range. Dropping of sparse connectivity increases misclassifications, for the median error of SOAE-σ-conn is 0.81% and thereby greater than the median error of SOAE-σ.

The other five approaches included in the comparison are multi-layer perceptrons (MLPs) with the same topology and dynamics as the classification module of supervised online auto-encoder, with two exceptions. First, the transfer function of the hidden layer f was set to a hyperbolic tangent, thus not including explicit sparse activity. Second, in all but one experiment sparse connectivity was either not incorporated, or achieved through other means than by performing a σ projection after each learning epoch. Besides the variation in sparseness of connectivity, the experiments differ in the initialization of the network parameters.

For each variant, 55 runs were carried out and the resulting classifiers were applied to the evaluation set to compute the classification error. Then, the best four and the worst four results were discarded and not included in further analysis. Hence a random sample of size 47 was achieved,

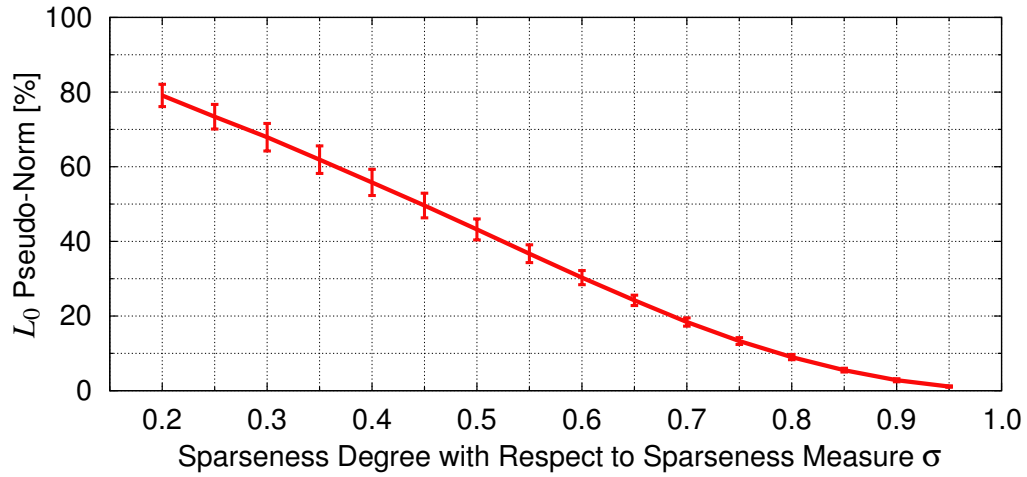


Figure 5: Resulting amount of nonzero entries in an internal representation h with 1000 entries, depending on the target degree of sparseness for activity σ_H with respect to Hoyer's sparseness measure σ. For low values of σ_H, about 80% of the entries are nonzero, whereas for very high sparseness degrees only 1% of the entries do not vanish. The error bars indicate $\pm$ one standard deviation distance from the mean value. Standard deviation shrinks with increasing sparseness degree, making the mapping more accurate.

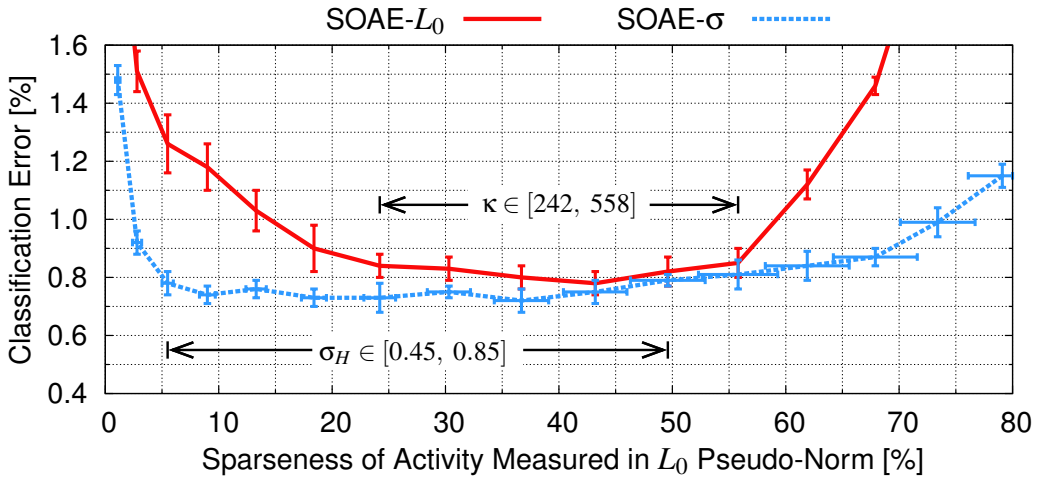


Figure 6: Resulting classification error on the MNIST evaluation set for the supervised online autoencoder network, in dependence of sparseness of activity in the hidden layer. The projection onto an L_0 pseudo-norm constraint for variant SOAE-L_0 and the projection onto a constraint determined by sparseness measure σ for variant SOAE-σ were used as transfer functions. The error bars indicate $\pm$ one standard deviation difference from the mean.

Approach	Sparse Connectivity	Sparse Activity	Result $(W,\ p)$ of Shapiro-Wilk Test	Evaluation Error [%]
SOAE-σ	$\sigma_W = 0.75$	$\sigma_H \in [0.45,\ 0.85]$	$(0.9802,\ 0.60)$	0.75 ± 0.04
SOAE-L_0	$\sigma_W = 0.75$	$\kappa \in [242,\ 558]$	$(0.9786,\ 0.53)$	0.82 ± 0.05
SOAE-σ-conn	none	$\sigma_H \in [0.45,\ 0.85]$	$(0.9747,\ 0.40)$	0.81 ± 0.04
SMLP-SCFC	$\sigma_W = 0.75$	none	$(0.9770,\ 0.47)$	0.81 ± 0.05
MLP-OBD	$\gamma = 12.5\%$	none	$(0.9807,\ 0.62)$	0.89 ± 0.04
MLP-random	none	none	$(0.9798,\ 0.58)$	0.88 ± 0.03
MLP-samples	none	none	$(0.9773,\ 0.49)$	0.91 ± 0.05
MLP-SCFC	none	none	$(0.9794,\ 0.57)$	0.91 ± 0.06

Table 1: Overview of comparative experiments. The second and third columns indicate whether sparse connectivity or sparse activity was incorporated, respectively. The fourth column reports the result of a statistical test for normality, which is interpreted in Section 3.5. The final column gives the median $\pm$ one standard deviation of the achieved classification error on the MNIST evaluation set. The results for each experiment were trimmed to gain a sample of size 47, allowing for statistical robust estimates.

where 15% of the original data were trimmed away. This procedure was also applied to the results of SOAE-σ, SOAE-σ-conn, and SOAE-L_0, to obtain a total of eight random samples of equal size for comparison with another.

The most basic variant, denoted the baseline in this discussion, is MLP-random, where all network parameters were initialized randomly. This achieved a median error of 0.88% on the evaluation set, being considerably worse than SOAE-σ. For variant MLP-samples, the hidden layer was initialized by replication of n randomly chosen samples from the learning set. This did decrease the overall learning time. However, the median classification error was slightly worse with 0.91% compared to MLP-random.

For variant MLP-SCFC, the network parameters were initialized in an unsupervised manner using the sparse coding for fast classification (SCFC) algorithm (Thom et al., 2011a). This method is a precursor to the SOAE proposed in this paper. It also features sparse connectivity and sparse activity but differs in some essential parts. First, sparseness of activity is achieved through a latent variable that stores the optimal sparse code words of all samples simultaneously. Using this matrix of code words, the activity of individual units was enforced to be sparse over time on the entire learning set. SOAE achieves sparseness over space, as for each sample only a pre-defined fraction of units is allowed to be active at any one time. A second difference is that sparse activity is achieved only indirectly by approximation of the latent matrix of code words with a feed-forward representation. With SOAE, sparseness of activity is guaranteed by construction. MLP-SCFC achieved a median classification error of 0.91% on the MNIST evaluation set, rendering it slightly worse than MLP-random and equivalent to MLP-samples.

The first experiment that incorporates only sparse connectivity is SMLP-SCFC. Initialization was done as for MLP-SCFC, but during training sparseness of connectivity was yielded by application of the sparseness-enforcing projection operator to the weights of the hidden layer after

every learning epoch. Hence the sparseness gained from unsupervised initialization was retained. MLP-SCFC features sparse connectivity only after initialization, but loses this property when training proceeds. With this slight modification, the median error of SMLP-SCFC decreases to 0.81%, which is significantly better than the baseline result.

The effect of better generalization due to sparse connectivity has also been observed by LeCun et al. (1990) in the context of convolutional neural networks. It can be explained by the bias-variance decomposition of the generalization error (Geman et al., 1992). When the effective number of the degrees of freedom is constrained, overfitting will be less likely and hence classifiers produce better results on average. The same argument can be applied to SOAE-σ, where additional sparse activity further improves classification results.

The last variant is called MLP-OBD. Here, the optimal brain damage (OBD) algorithm (LeCun et al., 1990) was used to prune synaptic connections in the hidden layer that are irrelevant for the computation of the classification decision of the network. The parameters of the network were first initialized randomly and then optimized on the learning set. Then the impact for each synaptic connection on the objective function was estimated using the Taylor series of the objective function, where a diagonal approximation of the Hessian was employed and terms of cubic or higher order were neglected. Using this information, the number of connections was halved by setting the weight of connections with low impact to zero. The network was then retrained with weights of removed connections kept at zero. This procedure was repeated until a target percentage γ of active synaptic connections in the hidden layer was achieved. For the results reported here, $\gamma = 12.5\%$ was chosen as this reflects the sparse connectivity $\sigma_W = 0.75$ of the other approaches best. MLP-OBD achieved a median classification error of 0.89%, which is comparable to the baseline result.

3.5 Statistical Analysis and Conclusions

A statistical analysis was carried out to assess the significance of the differences in the performance of the eight algorithms. The procedure follows the proposals of Pizarro et al. (2002) and Demšar (2006) for hypothesis testing, and is concluded by effect size estimation as proposed by Grissom (1994) and Acion et al. (2006). For each algorithm, a sample of size 47 was available, allowing for robust analysis results.

First, all results were tested for normality using the test developed by Shapiro and Wilk (1965). The resulting test statistics W and p-values are given in Table 1. As all p-values are large, it cannot be rejected that the samples came from normally distributed populations. Thus normality is assumed in the remainder of this discussion. Next, the test proposed by Levene (1960) was applied to determine whether equality of variances of the groups holds. This resulted in a test statistic $F = 2.7979$ with 7 and 368 degrees of freedom, and therefore a p-value of 0.0075. Hence the hypothesis that all group variances are equal can be rejected with very high significance. Consequently, parametric omnibus and post-hoc tests cannot be applied, as they require the groups to have equal variance.

As an alternative, the nonparametric test by Kruskal and Wallis (1952) which is based on rank information was employed to test whether all algorithms produced classifiers with equal classification errors in the mean. The test statistic was $H = 214.44$ with 7 degrees of freedom, and the p-value was less than 10^{-15}. There is hence a statistically significant difference in the mean classification results. To locate this deviation, a critical difference for comparing the mean ranks of the algorithms was computed. A Tukey-Kramer type modification applied to Dunn's procedure yields this critical difference, which is less conservative than Nemenyi's procedure for the Kruskal-Wallis

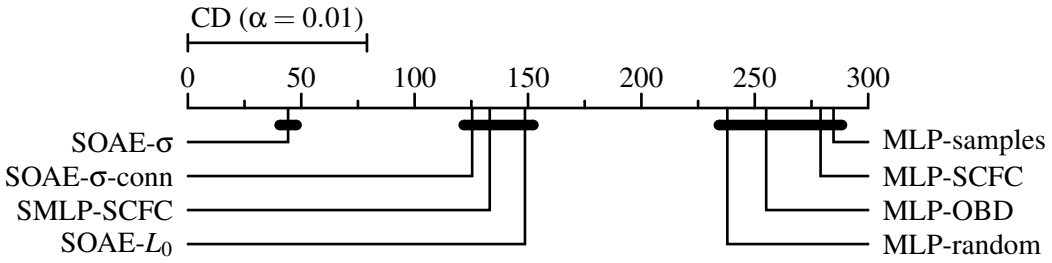


Figure 7: Diagram for multiple comparison of algorithms following Demšar (2006). For each algorithm, the mean rank was computed during the Kruskal-Wallis test. Then, a critical difference (CD) was computed at the $\alpha = 0.01$ significance level. Two algorithms produce classification results that are statistically not equal if the difference between their mean ranks is greater than the critical difference. This induced three groups of algorithms that produced statistically equivalent results, which are marked with black bars.

test (Hochberg and Tamhane, 1987). Note that this approach is nevertheless similar to the post-hoc procedure proposed by Demšar (2006) for paired observations, such that the diagrams proposed there can be adapted to the case for unpaired observations. The result is depicted in Figure 7, where the critical difference for statistical significance at the $\alpha = 0.01$ level is given. This test induces a highly significant partitioning of the eight algorithms, namely three groups A, B and C given by

$$A := \{\text{SOAE-}\sigma\},\ B := \{\text{SOAE-}\sigma\text{-conn},\ \text{SOAE-}L_0,\ \text{SMLP-SCFC}\},$$
$$\text{and } C := \{\text{MLP-OBD},\ \text{MLP-random},\ \text{MLP-samples},\ \text{MLP-SCFC}\}.$$

This partition in turn induces an equivalence relation. Statistical equivalence is hence unambiguous and well-defined at $\alpha = 0.01$. Moreover, the p-value for this partition is 0.007. If the significance level α would have been set lower than this, then groups A and B would blend together.

To assess the benefit when an algorithm from one group is chosen over an algorithm from another group, the probability of superior experiment outcome was estimated (Grissom, 1994; Acion et al., 2006). For this, the classification errors were pooled with respect to membership in the three groups. It was then tested whether these pooled results still come from normal distributions. As group A is a singleton, this is trivially fulfilled with the result from Table 1. For group B, the Shapiro-Wilk test statistic was $W = 0.9845$ and the p-value was 0.11. Group C achieved a test statistic of $W = 0.9882$ and a p-value of 0.12. If a standard significance level of $\alpha = 0.01$ is chosen, then B and C can be assumed to be normally distributed also.

Let E_G be the random variable modeling the classification results of the algorithms from group $G \in \{A,B,C\}$. It is assumed that E_G is normally distributed with unknown mean and unknown variance for all G. Then $E_G - E_{\tilde{G}}$ is clearly normally distributed also for two groups $G, \tilde{G} \in \{A,B,C\}$. Therefore, the probability $P(E_G < E_{\tilde{G}})$ that one algorithm produces a better classifier than another could be computed from the Gaussian error function if the group means and variances were known. However, using Rao-Blackwell theory a minimum variance unbiased estimator $\hat{R}_2$ of this probability can be computed easily (Downton, 1973). Evaluation of the expression for $\hat{R}_2$ shows that $P(E_A < E_B)$ can be estimated by 0.87, $P(E_B < E_C)$ can be estimated by 0.88, and $P(E_A < E_C)$

can be estimated by 0.99. Therefore, the effect of choosing SOAE-σ over any of the seven other algorithms is dramatic (Grissom, 1994).

These results can be interpreted as follows. When neither sparse activity nor sparse connectivity is incorporated, then the worst classification results are obtained regardless of the initialization of the network parameters. The exception is MLP-OBD which incorporates sparse connectivity, although, as its name says, in a destructive way. Once a synaptic connection has been removed, it cannot be recovered, as the measure for relevance of LeCun et al. (1990) vanishes for synaptic connections of zero strength. The statistics for SMLP-SCFC shows that when sparse connectivity is obtained using the sparseness-enforcing projection operator, then superior results can be achieved. Because of the nature of projected gradient descent, it is possible here to restore deleted connections if it helps to decrease the classification error during learning. For SOAE-σ-conn only sparse activity was used, and classification results were statistically equivalent to SMLP-SCFC.

Therefore, using either sparse activity or sparse connectivity improves classification capabilities. When both are used, then results improve even more as variant SOAE-σ shows. This does not hold for SOAE-L_0 however, where the L_0 projection was used as transfer function. As Hoyer's sparseness measure σ and the according projection possess desirable analytical properties, they can be considered smooth approximations to the L_0 pseudo-norm. It is this smoothness which seems to produce this benefit in practice.

4. Related Work

This section reviews work related with the contents of this paper. First, the theoretical foundations of the sparseness-enforcing projection operator are discussed. Next, its application as neuronal transfer function to achieve sparse activity in a classification scenario is put in context with alternative approaches, and possible advantages of sparse connectivity are described.

4.1 Sparseness-Enforcing Projection Operator

The first major part of this paper dealt with improvements to the work of Hoyer (2004) and Theis et al. (2005). Here, an algorithm for the sparseness-enforcing projection with respect to Hoyer's sparseness measure σ was proposed. The technical proof of correctness is given in Appendix C. The set that should be projected onto is an intersection of a simplex C and a hypercircle L, which is a hypersphere lying in a hyperplane. The overall procedure can be described as performing alternating projections onto C and certain subsets of L. This approach is common for handling projections onto intersections of individual sets. For example, von Neumann (1950) proposed essentially the same idea when the investigated sets are closed subspaces, and has shown that this converges to a solution. A similar approach can be carried out for intersections of closed, convex cones (Dykstra, 1983), which can be generalized to translated cones that can be used to approximate any convex set (Dykstra and Boyle, 1987). For these alternating methods, it is only necessary to know how projections onto individual members of the intersection can be achieved.

Although these methods exhibit great generality, they have two severe drawbacks in the scenario of this paper. First, the target set for projection must be an intersection of convex sets. The scaled canonical simplex C is clearly convex, but the hypercircle L is non-convex if it contains more than one point. The condition that generates L cannot easily be weakened to achieve convexity. If the original hypersphere were replaced with a closed ball, then L would be convex. But this changes the meaning of the problem dramatically, as now virtually any sparseness below the original target

degree of sparseness can be obtained. This is because when the target L_1 norm λ_1 is fixed, the sparseness measure σ decreases whenever the target L_2 norm decreases. In geometric terms, the method proposed in this paper performs a projection from within a circle onto its boundary to increase the sparseness of the working vector. This argument is given in more detail in Figure 11 and the proof of Lemma 28(f).

The second drawback of the general methods for projecting onto intersections is that a solution is only achieved asymptotically, even when the convexity requirements are fulfilled. Due to the special structure of C and L, the number of alternating projections that have to be carried out to find a solution using Algorithm 3 is bounded from above by the problem dimensionality. Thus an exact projection is always found in finite time. Furthermore, the solution is guaranteed to be found in time that is at most quadratic in problem dimensionality.

A crucial point is the computation of the projection onto C and certain subsets of L. Due to the nature of the L_2 norm, the latter is straightforward. For the former, efficient algorithms have been proposed recently (Duchi et al., 2008; Chen and Ye, 2011). When only independent solutions are required, the projection of a point x onto a scaled canonical simplex of L_1 norm λ_1 can also be carried out in linear time (Liu and Ye, 2009), without having to sort the vector that is to be projected. This can be achieved by showing that the separator $\hat{t}$ for performing the simplex projection is the unique zero of the monotonically decreasing function $t \mapsto \|\max(|x| - t \cdot e,\ 0)\|_1 - \lambda_1$. The zero of this function can be found efficiently using the bisection method, and exploiting the special structure of the occurring expressions (Liu and Ye, 2009).

In the context of this paper an explicit closed-form expression for $\hat{t}$ is preferable as it permits additional insight into the properties of the projected point. The major part in proving the correctness of Algorithm 1 is the interconnection between C and L, that is that the final solution has zero entries at the according positions in the working vector and thus a chain monotonically decreasing in L_0 pseudo-norm is achieved. This result is established through Lemma 26, which characterizes projections onto certain faces of a simplex, Corollary 27 and their application in Lemma 28.

Analysis of the theoretical properties of the sparseness-enforcing projection is concluded with its differentiability in Appendix D. The idea is to exploit the finiteness of the projection sequence and to apply the chain rule of differential calculus. It is necessary to show that the projection chain is robust in a neighborhood of the argument. This reduces analysis to individual projection steps which have already been studied in the literature. For example, the projection onto a closed, convex set is guaranteed to be differentiable almost everywhere (Hiriart-Urruty, 1982). Here non-convexity of L is not an issue, as the only critical point is its barycenter. For the simplex C, a characterization of critical points is given with Lemma 32 and Lemma 33, and it is shown that the expression for the projection onto C is invariant to local changes. An explicit expression for construction of the gradient of the sparseness-enforcing projection operator is given in Theorem 35. In Corollary 36 it is shown that the computation of the product of the gradient with an arbitrary vector can be achieved efficiently by exploiting sparseness and the special structure of the gradient.

Similar approaches for sparseness projections are discussed in the following. The iterative hard thresholding algorithm is a gradient descent algorithm, where a projection onto an L_0 pseudo-norm constraint is performed (Blumensath and Davies, 2009). Its application lies in compressed sensing, where a linear generative model is used to infer a sparse representation for a given observation. Sparseness here acts as regularizer which is necessary because observations are sampled below the Nyquist rate. In spite of the simplicity of the method, it can be shown that it achieves a good approximation to the optimal solution of this NP-hard problem (Blumensath and Davies, 2009).

Closely related with the work of this paper is the generalization of Hoyer's sparseness measure by Theis and Tanaka (2006). Here, the L_1 norm constraint is replaced with a generalized L_p pseudo-norm constraint, such that the sparseness measure becomes $\sigma_p(x) := {\|x\|_p}/{\|x\|_2}$. For $p = 1$, Hoyer's sparseness measure up to a constant normalization is obtained. When p converges decreasingly to zero, then $\sigma_p(x)^p$ converges point-wise to the L_0 pseudo-norm. Hence for small values of p a more natural sparseness measure is obtained. Theis and Tanaka (2006) also proposed an extension of Hoyer's projection algorithm. It is essentially von Neumann's alternating projection method, where closed subspaces have been replaced by "spheres" that are induced by L_p pseudo-norms. Note that these sets are non-convex when $p < 1$, such that convergence is not guaranteed. Further, no closed-form solution for the projection onto an "L_p-sphere" is known for $p \notin \{1, 2, \infty\}$, such that numerical methods have to be employed.

A problem where similar projections are employed is to minimize a convex function subject to group sparseness (see for example Friedman et al., 2010). In this context, mixed norm balls are of particular interest (Sra, 2012). For a matrix $X \in \mathbb{R}^{n \times g}$, the mixed $L_{p,q}$ norm is defined as the L_p norm of the L_q norms of the columns of X, that is $\|X\|_{p,q} := \left\| \left(\|Xe_1\|_q, \ \ldots, \ \|Xe_g\|_q \right)^T \right\|_p$. Here, X can be interpreted to be a data point with entries partitioned into g groups. When $p = 1$, then the projection onto a simplex can be generalized directly for $q = 2$ (van den Berg et al., 2008) and for $q = \infty$ (Quattoni et al., 2009). The case when $p = 1$ and $q \geq 1$ is more difficult, but can be solved as well (Liu and Ye, 2010; Sra, 2012).

The last problem discussed here is the elastic net criterion (Zou and Hastie, 2005), which is a constraint on the sum of an L_1 norm and an L_2 norm. The feasible set can be written as the convex set $N := \{ s \in \mathbb{R}^n \mid \lambda_1 \|s\|_1 + \lambda_2 \|s\|_2^2 \leq 1 \}$, where $\lambda_1, \lambda_2 \geq 0$ control the shape of N. Note that in N only the sum of two norms is considered, whereas the non-convex set $S^{(\lambda_1, \lambda_2)}$ consists of the intersection of two different constraints. Therefore, the elastic net induces a different notion of sparseness than Hoyer's sparseness measure σ does. As is the case for mixed norm balls, the projection onto a simplex can be generalized to achieve projections onto N (Mairal et al., 2010).

4.2 Supervised Online Auto-Encoder

The sparseness-enforcing projection operator π with respect to Hoyer's sparseness measure σ and the projection onto an L_0 pseudo-norm constraint are differentiable almost everywhere. Thus they are suitable for gradient-based optimization algorithms. In Section 3, they were used as transfer functions in a hybrid of an auto-encoder network and a two-layer neural network to infer a sparse internal representation. This representation was subsequently employed to approximate the input sample and to compute a classification decision. In addition, the matrix of bases which was used to compute the internal representation was enforced to be sparsely populated by application of the sparseness projection after each learning epoch. Hence the supervised online auto-encoder proposed in this paper features both sparse activity and sparse connectivity.

These two key properties have also been investigated and exploited in the context of auto-associative memories for binary inputs. If the entries of the training patterns are sparsely populated, the weight matrix of the memory will be sparsely populated as well after training if Hebbian-like learning rules are used (Kohonen, 1972). The assumption of sparsely coded inputs also results in increased completion capacity and noise resistance of the associative memory (Palm, 1980). If the input data is not sparse inherently, feature detectors can perform a sparsification prior to the actual processing through the memory (Baum et al., 1988).

A purely generative model that also possesses these two key properties is non-negative matrix factorization with sparseness constraints (Hoyer, 2004). This is an extension to plain non-negative matrix factorization (Paatero and Tapper, 1994) which was shown to achieve sparse connectivity on certain data sets (Lee and Seung, 1999). However, there are data sets on which this does not work (Li et al., 2001; Hoyer, 2004). Although Hoyer's model makes sparseness easily controllable by explicit constraints, it is not inherently suited to classification tasks. An extension intended to incorporate class membership information to increase discriminative capabilities was proposed by Heiler and Schnörr (2006). In their approach, an additional constraint was added ensuring that every internal representation is close to the mean of all internal representations that belong to the same class. In other words, the method can be interpreted as supervised clustering, with the number of clusters equal to the number of classes. However, there is no guarantee that a distribution of internal representations exists such that both the reproduction error is minimized and the internal representations can be arranged in such a pattern. Unfortunately, Heiler and Schnörr (2006) used only a subset of a small data set for handwritten digit recognition to evaluate their approach.

A precursor to the supervised online auto-encoder was proposed by Thom et al. (2011a). There, inference of sparse internal representations was achieved by fitting a one-layer neural network to approximate a latent variable of optimal sparse representations. The transfer function used for this approximation was a hyperbolic tangent raised to an odd power greater or equal to three. This resulted in a depression of activities with small magnitude, favoring sparseness of the result. Similar techniques to achieve a shrinkage-like effect for increasing sparseness of activity in a neural network were used by Gregor and LeCun (2010) and Glorot et al. (2011). Information processing is here purely local, that is a scalar function is evaluated entrywise on a vector, and thus no information is interchanged among individual entries.

The use of non-local shrinkage to reduce Gaussian noise in sparse coding has already been described by Hyvärinen et al. (1999). Here, a maximum likelihood estimate with only weak assumptions yields a shrinkage operation, which can be conceived as projection onto a scaled canonical simplex. In the use case of object recognition, a hard shrinkage was also employed to de-noise filter responses (Mutch and Lowe, 2006). Whenever a best approximation from a permutation-invariant set is used, a shrinkage-like operation must be employed. Using a projection operator as neural transfer function is hence a natural extension of these ideas. When the projection is sufficiently smooth, the entire model can be tuned end-to-end using gradient methods to achieve an auto-encoder or a classifier.

The second building block from Thom et al. (2011a) that was incorporated into supervised online auto-encoder is the architectural concept for classification. It is well-known that two layers in a neural network are sufficient to approximate any continuous function on a compactum with arbitrary precision (Cybenko, 1989; Funahashi, 1989; Hornik et al., 1989). Similar architectures have also been proposed for classification in combination with sparse coding of the inputs. However, sparse connectivity was not considered in this context. Bradley and Bagnell (2009) used the Kullback-Leibler divergence as implicit sparseness penalty term and combined this with the back-propagation algorithm to yield a classifier that achieved a 1.30% error rate on the MNIST evaluation set. The Kullback-Leibler divergence was chosen to replace the usual L_1 norm penalty term, as it is smoother than the latter and therefore sparsely coded internal representations are more stable subject to subtle changes of the input. A related technique is supervised dictionary learning by Mairal et al. (2009), where the objective function is an additive combination of a classification error term, a term for the reproduction error, and an L_1 norm constraint. Inference of sparse internal represen-

tations is achieved through solving an optimization problem. Such procedures are time-consuming and greatly increase the computational complexity of classification. With this approach, a classification error of 1.05% on the MNIST evaluation set was achieved. These two approaches used the original MNIST learning set without jittering the digits and can thus be considered permutation-invariant. Augmentation of the learning set with virtual samples would have contributed to improve classification performance, as demonstrated by Schölkopf (1997).

Finally consider once more the sparse connectivity property, which is mostly neglected in the literature in favor of sparse activity. It was shown in this paper that sparse connectivity helps to improve generalization capabilities. In practice, this property can also be used to reduce the computational complexity of classification by one order of magnitude (Thom et al., 2011b). This results from exploiting sparseness and using sparse matrix-vector multiplication algorithms to infer the internal representation, which is the major computational burden in class membership prediction. It was shown in this paper and by Thom et al. (2011b) that a small number of nonzero entries in the weight matrix of the hidden layer is sufficient for achieving good classification results. Furthermore, the additional savings in required storage capacity and bandwidth allow using platforms with modest computational power for practical implementations. Sparseness is therefore an elementary concept of efficiency in artificial processing systems.

5. Conclusions

Without sparseness in their brains, higher mammals probably would not have developed to viable life-forms. This important concept of efficiency was discovered by neuroscientists, and practical benefit was obtained by the engineers of artificial information processing systems. This paper studied Hoyer's sparseness measure σ, and in particular the projection of arbitrary vectors onto sets where σ attains a constant value. A simple yet efficient algorithm for computing this sparseness-enforcing projection operator was proposed in this paper, and its correctness was proved. In addition, it was demonstrated that the proposed algorithm is superior in run-time to Hoyer's original algorithm. The analysis of the theoretical properties of this projection was concluded by showing it is differentiable almost everywhere.

As projections onto σ constraints are well-understood, they constitute the ideal tool for building systems that can benefit from sparseness constraints. An original use case was introduced in this paper. Here, the σ projection was implemented as neuronal transfer function, yielding a differentiable closed-form expression for inference of sparse code words. Besides this sparse activity, the connectivity in this system was also forced to be sparse by performing the σ projection after the presentation of learning examples. Because of its smoothness, the entire system can be optimized end-to-end by gradient-based methods, yielding a classification architecture exhibiting true sparse information processing.

This supervised online auto-encoder was applied on a benchmark data set for pattern recognition. Because sparseness constraints reduce the amount of feasible solutions, it is not clear in the first place whether the same performance can be achieved at all. However, when the target degree of sparseness of the activity is in a reasonable range, classification results are not only equivalent but superior to classical non-sparse approaches. This result is supported by statistical evaluation showing that this performance increase is not merely coincidental, but statistically significant. Therefore, sparseness can be seen as regularizer that offers the potential to improve artificial systems in the same way it seems to improve biological systems.

Acknowledgments

The authors wish to thank Patrik O. Hoyer and Xiaojing Ye for sharing the source code of their algorithms. The authors are also grateful to the anonymous reviewers for their valuable comments and feedback. This work was supported by Daimler AG, Germany.

Appendix A. Notation and Prerequisites

This appendix fixes the notation and provides prerequisites for the following appendices. $\mathbb{N}$ denotes the natural numbers including zero, $\mathbb{R}$ the real numbers and $\mathbb{R}_{\geq 0}$ the non-negative real numbers. $\mathbb{R}^n$ is the n-dimensional Euclidean space with canonical basis $e_1, \ldots, e_n \in \mathbb{R}^n$, and $e := \sum_{i=1}^n e_i \in \mathbb{R}^n$ denotes the vector where all entries are identical to unity. For all other vectors, a subscript denotes the corresponding entry of the vector, that is $x_i = e_i^T x$ for $x \in \mathbb{R}^n$. The amount of nonzero entries in a vector is given by the L_0 pseudo-norm, $\|\cdot\|_0$. $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the Manhattan norm and Euclidean norm, respectively. $\langle \cdot,\ \cdot \rangle$ denotes the canonical dot product in the Euclidean space. Given a vector x, $\mathrm{diag}(x)$ denotes the square matrix with x on its main diagonal and zero entries at all other positions, and $a \circ b = \mathrm{diag}(a)b$ denotes the Hadamard product or entrywise product for vectors a and b. When A and B are square matrices, then $\mathrm{diag}(A,B)$ denotes the block diagonal matrix with the blocks given by A and B. S_n is the symmetric group, and P_τ denotes the permutation matrix for $\tau \in S_n$. For a set $M \subseteq U$, M^C denotes its complement in the universal set U, where $U \in \{\mathbb{R}^n, \{1, \ldots, n\}\}$ is clear from the context. The power set of M is denoted by $\wp(M)$. If $M \subseteq \mathbb{R}^n$, then ∂M denotes its boundary in the topological sense. The sign function is denoted by $\mathrm{sgn}(\cdot)$. A list of symbols that are frequently used throughout the paper is given in Table 2.

The important concept of the projection onto a set was given in Definition 1. The following basic statement will be used extensively in this paper and follows from $\langle x,\ x \rangle = \|x\|_2^2$ for all $x \in \mathbb{R}^n$ and the fact that the scalar product is a symmetric bilinear form (Laub, 2004):

Proposition 4 *Let $a, b \in \mathbb{R}^n$. Then $\|a \pm b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 \pm 2\langle a,\ b \rangle$. Further it is $\|a-b\|_2^2 = \|a-p\|_2^2 + \|p-b\|_2^2 + 2\langle a-p,\ p-b \rangle$ for all $p \in \mathbb{R}^n$.*

As an example, note that the outcome of the sparseness-enforcing projection operator depends only on the target sparseness degree up to scaling:

Remark 5 *Let $\lambda_1, \lambda_2 > 0$ and $\tilde{\lambda}_1, \tilde{\lambda}_2 > 0$ be pairs of target norms such that $\lambda_1/\lambda_2 = \tilde{\lambda}_1/\tilde{\lambda}_2$. Then*

$$\mathrm{proj}_{S^{(\lambda_1,\lambda_2)}}(x) = \tilde{\lambda}_2/\lambda_2 \cdot \mathrm{proj}_{S^{(\tilde{\lambda}_1,\tilde{\lambda}_2)}}(x) \text{ for all } x \in \mathbb{R}^n.$$

Proof It is sufficient to show only one inclusion. Let $x \in \mathbb{R}^n$ be arbitrary, $p \in \mathrm{proj}_{S^{(\lambda_1,\lambda_2)}}(x)$ and $\tilde{r} \in S^{(\tilde{\lambda}_1,\tilde{\lambda}_2)}$. Define $\tilde{p} := \tilde{\lambda}_1/\lambda_1 \cdot p = \tilde{\lambda}_2/\lambda_2 \cdot p \in \mathbb{R}^n$, then $\|\tilde{p}\|_1 = \left|\tilde{\lambda}_1/\lambda_1\right| \cdot \|p\|_1 = \tilde{\lambda}_1$ and analogously $\|\tilde{p}\|_2 = \tilde{\lambda}_2$, hence $\tilde{p} \in S^{(\tilde{\lambda}_1,\tilde{\lambda}_2)}$. For the claim to hold it has now to be shown that $\|\tilde{p}-x\|_2 \leq \|\tilde{r}-x\|_2$. Write $r := \lambda_2/\tilde{\lambda}_2 \cdot \tilde{r} \in \mathbb{R}^n$, which in fact lies in $S^{(\lambda_1,\lambda_2)}$. So $\|p-x\|_2 \leq \|r-x\|_2$ by definition of p, and with Proposition 4 follows $\|\tilde{r}-x\|_2^2 - \|\tilde{p}-x\|_2^2 = \|\tilde{r}\|_2^2 + \|x\|_2^2 - 2\langle \tilde{r},\ x \rangle - \|\tilde{p}\|_2^2 - \|x\|_2^2 + 2\langle \tilde{p},\ x \rangle = 2\langle \tilde{p}-\tilde{r},\ x \rangle = \tilde{\lambda}_2/\lambda_2 \cdot 2\langle p-r,\ x \rangle = \tilde{\lambda}_2/\lambda_2 \cdot \left(\|r-x\|_2^2 - \|p-x\|_2^2\right) \geq 0$. ■

Hence only the ratio of the target L_1 norm to the target L_2 norm is important and not their actual scale. This argument can be generalized to projections onto any scale-invariant set and therefore naturally holds also for $S_{\geq 0}^{(\lambda_1,\lambda_2)}$.

Symbol and Definition	Meaning
σ (see Section 1.1)	Sparseness measure by Hoyer (2004)
π and $\pi_{\geq 0}$ (see Section 2.4)	Sparseness projection cast as function
$n \in \mathbb{N}$	Problem dimensionality
$e_1, \ldots, e_n \in \mathbb{R}^n$	Canonical basis of $\mathbb{R}^n$
$e := \sum_{i=1}^n e_i \in \mathbb{R}^n$	Vector where all entries are one
$\lambda_1 \in \mathbb{R}_{>0}$	Target L_1 or Manhattan norm
$\lambda_2 \in \mathbb{R}_{>0}$	Target L_2 or Euclidean norm
$S^{(\lambda_1,\lambda_2)} \subseteq \mathbb{R}^n$ (see Section 1.1)	Target set for sparseness projection
$S_{\geq 0}^{(\lambda_1,\lambda_2)} := S^{(\lambda_1,\lambda_2)} \cap \mathbb{R}^n_{\geq 0}$	Target set for non-negative sparseness projection
$D := S_{\geq 0}^{(\lambda_1,\lambda_2)}$	Short for the non-negative target set
$H := \{ a \in \mathbb{R}^n \mid e^T a = \lambda_1 \}$	Target hyperplane
$K := \{ q \in \mathbb{R}^n \mid \|q\|_2 = \lambda_2 \}$	Target hypersphere
$L := H \cap K$	Target hypercircle
$C := \mathbb{R}^n_{\geq 0} \cap H$	Scaled canonical simplex
$m := \lambda_1/n \cdot e \in \mathbb{R}^n$	Barycenter of L and C
$\rho := \lambda_2^2 - \lambda_1^2/n \in \mathbb{R}$	Squared radius of L
$I \subseteq \{1, \ldots, n\}$	Index set of nonzero entries
$d := \lvert I \rvert \in \mathbb{N}$	Working dimensionality
$L_I := \{ a \in L \mid a_i = 0 \text{ for all } i \notin I \}$	Points in L where certain coordinates vanish
$C_I := \{ c \in C \mid c_i = 0 \text{ for all } i \notin I \}$	Face of simplex C
$m_I := \lambda_1/d \cdot \sum_{i \in I} e_i \in \mathbb{R}^n$	Barycenter of L_I and C_I
$\rho_I := \lambda_2^2 - \lambda_1^2/d \in \mathbb{R}$	Squared radius of L_I

Table 2: A list of symbols used frequently in this paper and their meaning.

Appendix B. Projections onto Symmetric Sets

This appendix investigates certain symmetries of sets and their effect on projections onto such sets. A great variety of sparseness measures fulfills certain symmetries as vector entries are equally weighted, see Hurley and Rickard (2009). This means that no entry is preferred over another, and for negative entries usually the absolute value or the squared value is taken, such that the signs of the entries are ignored. Consider the following definition of symmetries that are to be analyzed:

Definition 6 *Let $\emptyset \neq M \subseteq \mathbb{R}^n$. Then M is called* permutation-invariant *if and only if $P_\tau x \in M$ for all $x \in M$ and all permutations $\tau \in S_n$. Further, M is called* reflection-invariant *if and only if $b \circ x \in M$ for all $x \in M$ and all $b \in \{ \pm 1 \}^n$.*

In other words, a subset M of the Euclidean space is permutation-invariant if set membership is invariant to permutation of individual coordinates. M is reflection-invariant if single entries can be

negated without violating set membership. This is equivalent to $x - 2\sum_{i \in I} x_i e_i \in M$ for all $x \in M$ and all index sets $I \subseteq \{1, \ldots, n\}$, which is a condition that is technically easier to handle. The following observation states that these symmetries are closed under common set operations:

Remark 7 *Let $\varnothing \neq A, B \subseteq \mathbb{R}^n$. When A and B are permutation-invariant or reflection-invariant, then so are $A \cup B$, $A \cap B$ and A^C.*

The proof is obvious by elementary set algebra. Now consider the following general properties of functions mapping to power sets:

Definition 8 *Let $\varnothing \neq M \subseteq \mathbb{R}^n$, let $\wp(M)$ be its power set and let $f \colon \mathbb{R}^n \to \wp(M)$ be a function. f is called* order-preserving *if and only if $x_i > x_j$ implies $p_i \geq p_j$ for all $x \in \mathbb{R}^n$, for all $p \in f(x)$ and for all $i, j \in \{1, \ldots, n\}$. f is called* absolutely order-preserving *if and only if from $|x_i| > |x_j|$ follows $|p_i| \geq |p_j|$ for all $x \in \mathbb{R}^n$, for all $p \in f(x)$ and for all $i, j \in \{1, \ldots, n\}$. f is called* orthant-preserving *if and only if $\operatorname{sgn}(x_i) = \operatorname{sgn}(p_i)$ or $x_i = 0$ or $p_i = 0$ for all $x \in M$ and all $p \in f(x)$.*

Hence, a function f is order-preserving if the relative order of entries of its arguments does not change upon function evaluation. Thus if the entries of x are sorted in ascending or descending order, then so are the entries of every vector in $f(x)$. Orthant-preservation denotes the fact that x and every vector from $f(x)$ are located in the same orthant. The link between set symmetries and projection properties is established by the following result. A weaker form of its statements has been described by Duchi et al. (2008) in the special case of a projection onto a simplex.

Lemma 9 *Let $\varnothing \neq M \subseteq \mathbb{R}^n$ and $p \colon \mathbb{R}^n \to \wp(M)$, $x \mapsto \operatorname{proj}_M(x)$. Then the following holds:*

(a) When M is permutation-invariant, then p is order-preserving.

(b) When M is reflection-invariant, then p is orthant-preserving.

Proof (a) Let $x \in \mathbb{R}^n$ and $p \in \operatorname{proj}_M(x)$. Let $i, j \in \{1, \ldots, n\}$ with $x_i > x_j$. Assume that $p_i < p_j$. Let $\tau := (i, j) \in S_n$ and $q := P_\tau p$, then $q \in M$ because of M being permutation-invariant. Consider $d := \|p - x\|_2^2 - \|q - x\|_2^2$. Because τ is a single transposition, application of Proposition 4 yields $d = 2(p_j - p_i)(x_i - x_j)$. By requirement $d > 0$, which contradicts the minimality of p as being a projection of x onto M. Hence $p_i \geq p_j$ must hold.

(b) Let $x \in \mathbb{R}^n$ and $p \in \operatorname{proj}_M(x)$. Define $I := \{\, i \in \{1, \ldots, n\} \mid \operatorname{sgn}(x_i) \neq \operatorname{sgn}(p_i) \,\}$. The claim holds trivially if $I = \varnothing$. Assume $I \neq \varnothing$ and define $q := p - 2\sum_{i \in I} p_i e_i$. It follows $q \in M$ because M is reflection-invariant. Proposition 4 implies $\|q\|_2^2 = \|p\|_2^2$, and clearly $\langle q,\, x \rangle = \langle p,\, x \rangle - 2\sum_{i \in I} p_i x_i$. Therefore application of Proposition 4 yields $d := \|p - x\|_2^2 - \|q - x\|_2^2 = -4\sum_{i \in I} p_i x_i$. By the definition of I one obtains $p_i x_i \in \{-1, 0\}$. Hence would there be an index $i \in I$ with $p_i \neq 0$ and $x_i \neq 0$, then $d > 0$, but $\|p - x\|_2^2 > \|q - x\|_2^2$ would contradict the minimality of p. Therefore $I = \{\, i \in \{1, \ldots, n\} \mid p_i = 0 \text{ or } x_i = 0 \,\}$, and the claim follows. ■

When the projection onto a permutation-invariant set is unique, then equal entries of the argument cause equal entries in the projection:

Remark 10 *Let $\varnothing \neq M \subseteq \mathbb{R}^n$ be permutation-invariant and $x \in \mathbb{R}^n$. When $p = \operatorname{proj}_M(x)$ is unique, then $p_i = p_j$ follows for all $i, j \in \{1, \ldots, n\}$ with $x_i = x_j$.*

Proof Let $x \in \mathbb{R}^n$, $p = \text{proj}_M(x)$ and $i, j \in \{1,\ldots,n\}$ with $x_i = x_j$. Assume $p_i \neq p_j$ would hold and let $\tau := (i,j) \in S_n$ and $q := P_\tau p \neq p$. With the permutation-invariance of M follows $q \in M$, and $\|q-x\|_2 = \|p-x\|_2$ with $x_i = x_j$. Hence $q \in \text{proj}_M(x)$, so $q = p$ with the uniqueness of the projection, which contradicts $q \neq p$. Therefore, $p_i = p_j$. ■

The next result shows how solutions to a projection onto reflection-invariant sets can be turned into non-negative solutions and vice-versa. Its second part was already observed by Hoyer (2004), in the special case of the sparseness-enforcing projection operator, and by Duchi et al. (2008), when the connection between projections onto a simplex and onto an L_1 ball was studied. Both did not provide a proof, but in the latter work a hint to a possible proof was given. With Lemma 11 it suffices to consider non-negative solutions for projections onto reflection-invariant sets.

Lemma 11 *Let $\varnothing \neq A \subseteq \mathbb{R}^n$ be reflection-invariant, $B := A \cap \mathbb{R}^n_{\geq 0}$ and $p, x \in \mathbb{R}^n$. Then:*

(a) If $p \in \text{proj}_A(x)$, then $|p| \in \text{proj}_B(|x|)$.

(b) If $p \in \text{proj}_B(|x|)$, then $s \circ p \in \text{proj}_A(x)$ where $s \in \{\pm 1\}^n$ is given by $s_i := 1$ if $x_i \geq 0$ and $s_i := -1$ otherwise for all $i \in \{1,\ldots,n\}$.

Proof First note that if $q \in \text{proj}_A(x)$, then $\text{sgn}(x_i) = \text{sgn}(q_i)$ or $x_i = 0$ or $q_i = 0$ with Lemma 9(b). Hence for all $i \in \{1,\ldots,n\}$ follows $(|q_i| - |x_i|)^2 = (q_i \cdot \text{sgn}(q_i) - x_i \cdot \text{sgn}(x_i))^2 = (q_i - x_i)^2$, and therefore $\|q-x\|_2^2 = \||q| - |x|\|_2^2$. Furthermore, $|q| \in A$ because of A reflection-invariant and $|q| \in \mathbb{R}^n_{\geq 0}$, so $|q| \in B$.

(a) Let $p \in \text{proj}_A(x)$ and $q \in B$, then $|p| \in B$ and it has to be shown that $\||p| - |x|\|_2 \leq \|q - |x|\|_2$. Define $I := \{\, i \in \{1,\ldots,n\} \mid x_i < 0 \,\}$ and $\tilde{q} := q - 2\sum_{i \in I} q_i e_i$, that is the signs of entries in I are flipped. Clearly $\tilde{q} \in A$, so in conjunction with the remark at the beginning of the proof follows $\||p| - |x|\|_2^2 = \|p-x\|_2^2 \leq \|\tilde{q} - x\|_2^2$. For $i \notin I$ one obtains $x_i \geq 0$ and $\tilde{q}_i = q_i$, hence $\tilde{q}_i - x_i = q_i - |x_i|$. For $i \in I$ follows $x_i < 0$ and $\tilde{q}_i = -q_i$, hence $\tilde{q}_i - x_i = -(q_i - |x_i|)$. This yields $(\tilde{q}_i - x_i)^2 = (q_i - |x_i|)^2$ for all $i \in \{1,\ldots,n\}$, thus $\|\tilde{q} - x\|_2^2 = \|q - |x|\|_2^2$, and the claim follows.

(b) Let $p \in \text{proj}_B(|x|)$. If $i \in \{1,\ldots,n\}$ with $x_i \geq 0$, then clearly $s_i p_i - x_i = p_i - |x_i|$. For $i \in \{1,\ldots,n\}$ with $x_i < 0$ follows $s_i p_i - x_i = -(p_i - |x_i|)$. Therefore, $\|s \circ p - x\|_2^2 = \|p - |x|\|_2^2$. Let $q \in \text{proj}_A(x)$, then the remark at the beginning of the proof yields $\|q-x\|_2^2 = \||q| - |x|\|_2^2$ and $|q| \in B$. $p \in \text{proj}_B(|x|)$ yields $\|p - |x|\|_2^2 \leq \||q| - |x|\|_2^2$, and the claim follows. ■

Using this result immediately yields a condition for projections to be absolutely order-preserving:

Lemma 12 *Let $\varnothing \neq M \subseteq \mathbb{R}^n$ be both permutation-invariant and reflection-invariant. Then the function $p\colon \mathbb{R}^n \to \wp(M)$, $x \mapsto \text{proj}_M(x)$, is absolutely order-preserving.*

Proof Let $x \in \mathbb{R}^n$, $p \in \text{proj}_M(x)$, and $i, j \in \{1,\ldots,n\}$ with $|x_i| > |x_j|$. Define $L := M \cap \mathbb{R}^n_{\geq 0}$, which is permutation-invariant with Remark 7. Lemma 11 implies that $|p| \in \text{proj}_L(|x|)$, and with Lemma 9 follows $|p_i| \geq |p_j|$. ■

The application of these elementary results to projections onto sets on which functions achieve constant values is straightforward. Examples were given in Section 2 with the sets Z and $S^{(\lambda_1,\lambda_2)}$.

Appendix C. Proof of Correctness of Algorithm 1 and Algorithm 3

The purpose of this appendix is to rigorously prove correctness of Algorithm 1 and Algorithm 3, that is that they compute projections onto $S^{(\lambda_1,\lambda_2)}_{\geq 0}$. Projections onto $S^{(\lambda_1,\lambda_2)}$ can then be inferred easily as explained in Appendix B.

C.1 Geometric Structures and First Considerations

The aim is to compute projections onto D, which is the intersection of the non-negative orthant $\mathbb{R}^n_{\geq 0}$, the target hyperplane H and the target hypersphere K, see Section 2.1. Further, the intersection of H and K yields a hypercircle L, and the intersection of $\mathbb{R}^n_{\geq 0}$ and H yields a scaled canonical simplex C. The structure of H and L will be analyzed in Section C.1.1 and Section C.1.2, respectively. The properties of C are discussed in Section C.2 and Section C.3. These results will then be used in Section C.4 to prove Theorem 2 and Theorem 3.

For the analysis of subsets where certain coordinates vanish, it is useful to define the following quantities for an index set $I \subseteq \{1,\ldots,n\}$ with cardinality $d := |I|$. The corresponding face of C is denoted by $C_I := \{\, c \in C \mid c_i = 0 \text{ for all } i \notin I \,\}$ and has barycenter $m_I := {}^{\lambda_1}/_{d} \cdot \sum_{i \in I} e_i \in C_I$. Further, $L_I := \{\, a \in L \mid a_i = 0 \text{ for all } i \notin I \,\}$ denotes the hypercircle with according vanishing entries, and $\rho_I := \lambda_2^2 - {}^{\lambda_1^2}/_{d}$ is the squared radius of L_I. Note that m_I is also the barycenter of L_I.

With these definitions the intermediate goal is now to prove that projections onto D can be computed by alternating projections onto the geometric structures defined earlier. The idea is to show that the set of solutions is not tampered by alternating projections onto H, C, L and L_I.

C.1.1 L_1 Norm Constraint—Target Hyperplane

First, the projection onto the target hyperplane H is considered. Lemma 13 is an elaborated version of a result from Theis et al. (2005), which is included here for completeness. Using its statements, it can be assumed that the considered point lies on H without modification of the solution set of the projection onto the target set D.

Lemma 13 *Let $x \in \mathbb{R}^n$. Then the following holds:*

(a) $\operatorname{proj}_H(x) = x + {}^{1}/_{n} \cdot \left(\lambda_1 - e^T x\right) e$.

(b) Let $r := \operatorname{proj}_H(x)$. *Then* $\operatorname{proj}_D(x) = \operatorname{proj}_D(r)$.

Proof (a) This is essentially a projection onto a hyperplane, yielding a unique result.

(b) With (a) follows $r - x = {}^{1}/_{n} \cdot \left(\lambda_1 - e^T x\right) e$. Hence $\langle h,\ r - x\rangle = {}^{\lambda_1}/_{n} \cdot \left(\lambda_1 - e^T x\right)$ holds for arbitrary $h \in H$. This expression is independent of the entries of h, which yields $\langle a - b,\ r - x\rangle = 0$ for every $a, b \in H$.

Now let $p \in \operatorname{proj}_D(x)$, that is $\|p - x\|_2 \leq \|q - x\|_2$ for all $q \in D$. Let $q \in D$ be arbitrary. With $D \subseteq H$ follows $\langle q - p,\ r - x\rangle = 0$, and thus Proposition 4 yields $\|q - r\|_2^2 - \|p - r\|_2^2 = \|q - x\|_2^2 - \|p - x\|_2^2 + 2\langle q - p,\ x - r\rangle = \|q - x\|_2^2 - \|p - x\|_2^2 \geq 0$, hence $\|p - r\|_2^2 \leq \|q - r\|_2^2$, so $p \in \operatorname{proj}_D(r)$.

For the converse let $p \in \operatorname{proj}_D(r)$. Analogously $\|q - x\|_2^2 - \|p - x\|_2^2 = \|q - r\|_2^2 - \|p - r\|_2^2 \geq 0$, hence $p \in \operatorname{proj}_D(x)$. ■

Therefore, the barycenter m is the projection of the origin onto H. The next remark gathers additional information on the norm of m and dot products with this point.

Remark 14 *It is $\|m\|_2^2 = {}^{\lambda_1^2}/_{n}$. Further, $\langle m,\ h\rangle = {}^{\lambda_1^2}/_{n}$ for all $h \in H$, and thus $\|h - m\|_2^2 = \|h\|_2^2 - {}^{\lambda_1^2}/_{n}$ with Proposition 4.*

C.1.2 L_1 and L_2 Norm Constraint—Target Hypersphere

After the projection onto the target hyperplane H has been carried out, consider now the joint constraint of H and the target hypersphere K. First note that $L = H \cap K$ is a hypercircle, that is a hypersphere in the subspace H, with intrinsic dimensionality reduced by one:

Lemma 15 *Consider $L = H \cap K$ and $\rho = \lambda_2^2 - \lambda_1^2/n$. Then the following holds:*

(a) $L = \tilde{L} := \{\, q \in H \mid \|q - m\|_2^2 = \rho \,\}$.

(b) $L \neq \emptyset$ *if and only if* $\lambda_2 \geq \lambda_1/\sqrt{n}$.

Proof (a) Follows immediately from Remark 14.

(b) L is nonempty if and only if $\rho \geq 0$ using (a), and $\rho = (\lambda_2 + \lambda_1/\sqrt{n})\,(\lambda_2 - \lambda_1/\sqrt{n})$. Hence, with $\lambda_1, \lambda_2 > 0$ one obtains $\rho \geq 0$ if and only if $\lambda_2 - \lambda_1/\sqrt{n} \geq 0$. ■

Hence $L \neq \emptyset$ by the requirement that $\lambda_2 \leq \lambda_1 \leq \sqrt{n}\lambda_2$. Further, the following observation follows immediately from Proposition 4 and $\|a\|_2 = \|b\|_2 = \lambda_2$ for all $a, b \in L$:

Remark 16 *For all $a, b \in L$ it is $\|a - b\|_2^2 = 2\left(\lambda_2^2 - \langle a,\, b\rangle\right)$, hence $\langle a,\, b\rangle = \lambda_2^2 - 1/2 \cdot \|a - b\|_2^2$.*

Therefore, on L the dot product is equal to the Euclidean norm up to an additive constant. Next consider projections onto L and note that the solution set with respect to D is not changed by this operation. The major arguments for this result have been taken over from Theis et al. (2005). Here, the statements from Lemma 15 have been incorporated and the resulting quadratic equation was solved explicitly, simplifying the original version of Theis et al. (2005).

Lemma 17 *Let $r \in H$ with $r \neq m$. Let $s := m + \delta(r - m)$ where $\delta := \sqrt{\rho}/\|r - m\|_2$. Then:*

(a) $\delta > 0$, *$s \in L$, and $\|q - r\|_2^2 - \|s - r\|_2^2 = 1/\delta \cdot \|q - s\|_2^2$ for all $q \in L$.*

(b) $s = \mathrm{proj}_L(r)$.

(c) $\mathrm{proj}_D(r) = \mathrm{proj}_D(s)$.

Proof (a) First note that $\delta > 0$ because of $r \neq m$. Clearly $s = (1 - \delta)m + \delta r$. Further, $s \in H$ because of $e^T s = \lambda_1$, and $s \in L$ because of $\|s - m\|_2^2 = \rho$ and Lemma 15.

Let $q \in L$ be arbitrary. One obtains $\|q\|_2 = \|s\|_2$ with $q, s \in K$ and therefore application of Proposition 4 yields $\|q - r\|_2^2 - \|s - r\|_2^2 = 2\,\langle s - q,\, r\rangle$. With Remark 14 follows $\langle m,\, r\rangle = \lambda_1^2/n$ and $\|r\|_2^2 = \|r - m\|_2^2 + \lambda_1^2/n$. Hence, $\langle s,\, r\rangle = (1 - \delta)\,\langle m,\, r\rangle + \delta\,\langle r,\, r\rangle = \lambda_1^2/n + \delta\,\|r - m\|_2^2$. On the other hand, from $s - m = \delta(r - m)$ and hence $r = (1 - 1/\delta)\,m + 1/\delta \cdot s$, and using Remark 16 it follows that $\langle q,\, r\rangle = (1 - 1/\delta)\,\lambda_1^2/n + 1/\delta \cdot \left(\lambda_2^2 - 1/2 \cdot \|q - s\|_2^2\right)$. Therefore with $\delta\,\|r - m\|_2^2 = \rho/\delta$ one obtains

$$\langle s - q,\, r\rangle = \delta\,\|r - m\|_2^2 + 1/\delta \cdot \left(\lambda_1^2/n - \lambda_2^2 + 1/2 \cdot \|q - s\|_2^2\right) = \tfrac{1}{2\delta}\,\|q - s\|_2^2,$$

and the claim follows directly by substitution.

(b) Let $q \in L$, then (a) implies $\|q - r\|_2^2 - \|s - r\|_2^2 = 1/\delta \cdot \|q - s\|_2^2 \geq 0$ with equality if and only if $q = s$ because of $\|\cdot\|_2$ being positive definite. Thus s is the unique projection of r onto L.

(c) With (a) follows $\|q - s\|_2^2 - \|p - s\|_2^2 = \delta\left(\|q - r\|_2^2 - \|p - r\|_2^2\right)$ for all $p, q \in D$ because of $D \subseteq L$. For $\mathrm{proj}_D(r) \subseteq \mathrm{proj}_D(s)$, let $p \in \mathrm{proj}_D(r)$ and $q \in D$. By definition $\|p - r\|_2^2 \leq \|q - r\|_2^2$, and thus $\|q - s\|_2^2 - \|p - s\|_2^2 \geq 0$ with $\delta > 0$, hence $p \in \mathrm{proj}_D(s)$. For the converse, let $p \in \mathrm{proj}_D(s)$ and $q \in D$. Similarly, $\|q - r\|_2^2 - \|p - r\|_2^2 = 1/\delta \cdot \left(\|q - s\|_2^2 - \|p - s\|_2^2\right) \geq 0$, thus $p \in \mathrm{proj}_D(r)$. ■

Lemma 17 does not hold when $r = m$, which forms a null set. In practice, however, this can occur when the input vector x for Algorithm 1 is poorly chosen, for example if all entries are equal. In this case, $\mathrm{proj}_L(r) = L$, hence any point from L can be chosen for further processing.

Remark 18 *One possibility in the case $r = m$ would be to choose the point $s := \alpha \sum_{i=1}^{n-1} e_i + \beta e_n$ where $\alpha, \beta \in \mathbb{R}$ for $s \in \mathrm{proj}_L(r)$, that is forcing the last entry to be unequal to the other ones. For satisfying $s \in L$, set $\alpha := \lambda_1/n + \sqrt{\rho}/\sqrt{n(n-1)}$ and $\beta := \lambda_1 - \alpha(n-1) = \lambda_1/n - \sqrt{\rho(n-1)}/\sqrt{n}$. This yields $\alpha - \beta = \sqrt{\rho}\left(1/\sqrt{n(n-1)} + \sqrt{n-1}/\sqrt{n}\right) > 0$, hence $\alpha \neq \beta$. This choice has the convenient side effect of s being sorted in descending order.*

Combining these properties of H and L, it can now be shown that projections onto D are invariant to affine-linear transformations with positive scaling:

Corollary 19 *Let $\alpha > 0$, $\beta \in \mathbb{R}$ and $x \in \mathbb{R}^n$. Then $\mathrm{proj}_D(\alpha x + \beta e) = \mathrm{proj}_D(x)$.*

Proof Let $\alpha > 0$, $\beta \in \mathbb{R}$ and $x \in \mathbb{R}^n$. With Lemma 13 and Lemma 17 it is enough to show that $\mathrm{proj}_L(\mathrm{proj}_H(\alpha x + \beta e)) = \mathrm{proj}_L(\mathrm{proj}_H(x))$. Let $\tilde{x} := \alpha x + \beta e$, $\tilde{r} := \mathrm{proj}_H(\tilde{x})$ and $\tilde{s} := \mathrm{proj}_L(\tilde{r})$. Lemma 13 and $e^T e = n$ yield $\tilde{r} = (\alpha x + \beta e) + 1/n \cdot \left(\lambda_1 - \alpha e^T x - \beta e^T e\right) e = \alpha x + 1/n \cdot \left(\lambda_1 - \alpha e^T x\right) e$. Hence $\tilde{r}$ is independent of β. Lemma 17 yields $\tilde{s} = m + \tilde{\delta}(\tilde{r} - m)$, where $\tilde{\delta} := \sqrt{\rho}/\|\tilde{r} - m\|_2$. Application of Proposition 4 yields

$$\begin{aligned}\|\tilde{r}\|_2^2 &= \|\alpha x\|_2^2 + \left\|1/n \cdot \left(\lambda_1 - \alpha e^T x\right) e\right\|_2^2 + 2\left\langle \alpha x,\ 1/n \cdot \left(\lambda_1 - \alpha e^T x\right) e\right\rangle \\ &= \alpha^2 \|x\|_2^2 + 1/n \cdot \left(\lambda_1 - \alpha e^T x\right)\left(\lambda_1 + \alpha e^T x\right) = \alpha^2 \|x\|_2^2 + 1/n \cdot \left(\lambda_1^2 - \alpha^2 (e^T x)^2\right),\end{aligned}$$

and with Remark 14 follows $\|\tilde{r} - m\|_2^2 = \|\tilde{r}\|_2^2 - \lambda_1^2/n = \alpha^2\left(\|x\|_2^2 - 1/n \cdot (e^T x)^2\right)$. Let $r := \mathrm{proj}_H(x)$ and $s := \mathrm{proj}_L(r)$, then Lemma 13 and Lemma 17 imply $r = x + 1/n \cdot \left(\lambda_1 - e^T x\right) e$ and $s = m + \delta(r - m)$, where $\delta := \sqrt{\rho}/\|r - m\|_2$. Likewise $\|r - m\|_2^2 = \|x\|_2^2 - 1/n \cdot (e^T x)^2$, and hence $\delta/\tilde{\delta} = \|\tilde{r} - m\|_2/\|r - m\|_2 = \alpha$, where $\alpha > 0$ must hold. This yields

$$\tilde{s} = m + \tilde{\delta}(\tilde{r} - m) = m + \delta/\alpha \cdot \left(\alpha x + \lambda_1/n \cdot e - \alpha/n \cdot e^T x e - \lambda_1/n \cdot e\right) = m + \delta\left(x - 1/n \cdot e^T x e\right) = s,$$

which shows that the projection is invariant. ■

Therefore, shifting and positive scaling of the argument of Algorithm 1 do not change the outcome. An overview of the steps carried out this far is given in Figure 8. Consider a point $x \in \mathbb{R}^n$ and $s := \mathrm{proj}_L(\mathrm{proj}_H(x))$. When $s \in \mathbb{R}^n_{\geq 0}$, then already $s \in D$ and hence $s \in \mathrm{proj}_D(x)$. Therefore only situations in which $s \notin \mathbb{R}^n_{\geq 0}$ holds are relevant in the remainder of this discussion.

C.2 Simplex Geometry

The joint constraint of the target hyperplane H with non-negativity yields simplex C. The following definition is likewise to definitions from Chen and Ye (2011) and Michelot (1986):

Definition 20 *For $n \in \mathbb{N}$, $n \geq 1$, the set $\triangle^n := \{\, \alpha \in \mathbb{R}^n_{\geq 0} \mid e^T \alpha = 1 \,\}$ is called* canonical n-simplex.

It is clear that $C = \mathbb{R}^n_{\geq 0} \cap H = \{\, \lambda_1 \alpha \mid \alpha \in \triangle^n \,\}$ is a scaled canonical simplex. Further, for an index set $I \subseteq \{1, \ldots, n\}$ the set $C_I = \{\, c \in C \mid c_i = 0 \text{ for all } i \notin I \,\}$ is a face of the simplex, which intrinsically possesses the structure of a simplex itself—although of reduced intrinsic dimensionality. Consider the following observation on the topology of C embedded in the subspace H:

Proposition 21 *Let $c = \lambda_1 \alpha \in C$ with $\alpha \in \triangle^n$. Then $c \in \partial C$ in the metric space $(H,\ \|\cdot\|_2)$ if and only if there is a $j \in \{1, \ldots, n\}$ with $\alpha_j = 0$.*

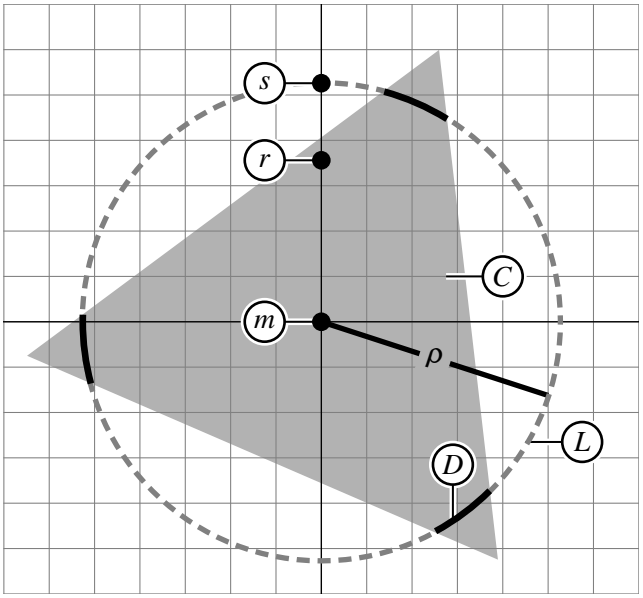


Figure 8: Sketch of the situation in Section C.1, projected onto target hyperplane H. r is the projection of the input point x onto H. s is the projection of r onto the hypercircle L, which has squared radius ρ. The intersection of H with the non-negative orthant is a simplex and denoted by C. The feasible set D is the intersection of C and L, and is marked with solid black lines. With Lemma 13 and Lemma 17 follows that $\mathrm{proj}_D(x) = \mathrm{proj}_D(r) = \mathrm{proj}_D(s)$, hence the next steps consist of projecting s onto C for finding the projection of x onto D.

The proof is simple and omitted as it does not contribute to deeper insight. Hence the faces C_I are subsets of ∂C, which is the topological border of C in $(H,\ \|\cdot\|_2)$. Using Proposition 21 a statement on the inradius of C can be made, which in turn can be used to show that for $n = 2$ no simplex projection has to be carried out at all:

Proposition 22 *The squared inradius of C is $\rho_{\text{in}} := \frac{\lambda_1^2}{n(n-1)}$. It is $L \subseteq C$ for $n = 2$.*

Proof Because C is closed and convex, it is enough to consider the distance between interior points and boundary points. Hence the insphere radius of a point $p \in C$ can be computed as being the minimum distance to any of the boundary points. With Proposition 21 these points can be characterized as points where at least one entry vanishes. Using Lagrange multipliers it can be shown that $\min_{c \in \partial C} \|m - c\|_2^2 = \rho_{\text{in}}$. Further, it can be shown that no point other than m is center of a larger insphere. This is achieved by constructing projections on certain faces of C, as is discussed in detail in the forthcoming Lemma 26. When $n = 2$ and $\lambda_2 \leq \lambda_1$, which is fulfilled by requirement on λ_1 and λ_2, then $\rho = \lambda_2^2 - \lambda_1^2/n \leq \lambda_1^2/2 = \rho_{\text{in}}$, and $L \subseteq C$ follows with Lemma 15. ■

The projection within H from outside a simplex is unique and must be located on its boundary:

Remark 23 *Let $s \in H \setminus C$. Then $\mathrm{proj}_C(s) \in \partial C$ in $(H,\ \|\cdot\|_2)$.*

The proof is obvious because C is closed and convex. By combination of Proposition 21 and Remark 23, it is now evident that the projection within H onto C yields vanishing entries. After the

first projection onto H, this subspace is never left throughout the arguments presented here, such that Remark 23 always applies. It has yet to be shown that the projection onto D possesses zero entries in the same coordinates. This way, a reduction of problem dimensionality can be achieved, and an iterative algorithm can be constructed to compute the projection onto D. The algorithm is guaranteed to terminate at the latest when the problem dimensionality equals two with Proposition 22.

C.2.1 PROJECTION ONTO A SIMPLEX

Quite a few methods have been proposed for carrying out projections onto canonical simplexes. An iterative algorithm was developed by Michelot (1986) which is very similar to Hoyer's original method for computation of the projection onto D. A simpler and more effective algorithm has been developed by Duchi et al. (2008). Building upon this work, Chen and Ye (2011) have proposed and rigorously proved correctness of a very similar algorithm, which is more explicit than that of Duchi et al. (2008). Their algorithm can be adapted to better suit the needs for the sparseness-enforcing projection. This adapted version was given by Algorithm 2 in Section 2. The following note makes the adaptations explicit.

Proposition 24 *Let $x \in \mathbb{R}^n \setminus C$ and $p := \mathrm{proj}_C(x)$. Then the following holds:*

(a) There exists $\hat{t} \in \mathbb{R}$ such that $p = \max(x - \hat{t} \cdot e,\ 0)$, where the maximum is taken element-wise.

(b) Algorithm 2 computes $\hat{t}$ such that (a) holds and the number of nonzero entries in p.

Proof The arguments from Chen and Ye (2011) hold for projections onto $\triangle^n$. The case of the scaled canonical simplex can be recovered using $p = \lambda_1 \cdot \mathrm{proj}_{\triangle^n}(x/\lambda_1)$. Therefore lines 4 and 7 of Algorithm 2 can be adapted from $t := \frac{s-1}{i}$ and $t := \frac{s-1}{n}$ to $t := \frac{s-\lambda_1}{i}$ and $t := \frac{s-\lambda_1}{n}$, respectively. The correct number of nonzero entries in p follows immediately from its expression from (a), the fact that y is sorted in descending order and the termination criterion of Algorithm 2. ■

As already described in Section 2.2, symmetries can be exploited for projections onto C:

Remark 25 *When x is already sorted in descending order, then no sorting is needed at the beginning of Algorithm 2. The projection p is then sorted also, because C is permutation-invariant. In this case, the nonzero entries of p are located in the first $d := \|p\|_0$ entries, while $p_{d+1} = \cdots = p_n = 0$.*

This fact is useful for optimizing access to the relevant entries of the working vector, which can then be stored contiguously in memory.

C.2.2 PROJECTION ONTO A FACE OF A SIMPLEX

The projection within H onto C yields zero entries in the working vector. It still remains to be shown that the projection onto D possesses zero entries at the same coordinates as the projection onto C. If this holds true, then the dimensionality of the original problem can be reduced, and iterative arguments can be applied. The main building block in the proof is the explicit construction of projections from within the simplex onto a certain face. The next Lemma is fundamental for proving correctness of Algorithm 1. It describes the construction of the result of the projection onto a simplex face and poses a statement on its norm, which in turn is used to prove that the position of vanishing entries does not change upon projection.

Lemma 26 *Let $q \in C$ and let $\emptyset \neq I \subseteq \{1,\ldots,n\}$ be an arbitrary index set. Then there exists an $s \in C_I$ with $\|q-v\|_2^2 = \|q-s\|_2^2 + \|s-v\|_2^2$ for all $v \in C_I$. If additionally $\max_{j\in J} q_j \leq \min_{i\in I} q_i$ holds for $J := I^C$, then $\|s\|_2 \geq \|q\|_2$ with equality if and only if $q_j = 0$ for all $j \in J$.*

More precisely, let $h := |J|$ and let $J = \{j_1,\ldots,j_h\}$ such that $q_{j_1} \leq \cdots \leq q_{j_h}$. Consider the sequence $s^{(0)},\ldots,s^{(h)} \in \mathbb{R}^n$ defined iteratively by $s^{(0)} := q$ and

$$s^{(k)} := s^{(k-1)} - s_{j_k}^{(k-1)} e_{j_k} + \tfrac{1}{n-k} s_{j_k}^{(k-1)} \left(e - \textstyle\sum_{i=1}^{k} e_{j_i}\right)$$

for $k \in \{1,\ldots,h\}$. Write $s := s^{(h)}$. Then the following holds:

(a) $s^{(k)} \in C_{\{j_1,\ldots,j_k\}^C}$ *for all* $k \in \{1,\ldots,h\}$.

(b) $\left\langle s^{(0)} - s^{(k)},\ s^{(k)} - s^{(k+1)} \right\rangle = 0$ *for all* $k \in \{0,\ldots,h-1\}$.

(c) $\left\|s^{(k)} - q\right\|_2^2 = \sum_{i=1}^{k} \left\|s^{(i)} - s^{(i-1)}\right\|_2^2$ *for all* $k \in \{0,\ldots,h\}$.

(d) $s_{j_k}^{(k-1)} = q_{j_k} + \frac{1}{n-k+1} \sum_{i=1}^{k-1} q_{j_i}$ *for all* $k \in \{1,\ldots,h\}$.

(e) $\left\langle s^{(0)} - s^{(k)},\ s^{(k)} - v \right\rangle = 0$ *for all* $k \in \{0,\ldots,h\}$ *and for all* $v \in C_I$.

(f) $\|q-v\|_2^2 = \left\|q - s^{(k)}\right\|_2^2 + \left\|s^{(k)} - v\right\|_2^2$ *for all* $k \in \{0,\ldots,h\}$ *and for all* $v \in C_I$.

(g) $s = \mathrm{proj}_{C_I}(q)$.

If $\max_{j\in J} q_j \leq \min_{i\in I} q_i$, then the following holds as well:

(h) $s_{j_1}^{(k)} \leq \cdots \leq s_{j_h}^{(k)} \leq \min_{i\in I} s_i^{(k)}$ *for all* $k \in \{0,\ldots,h\}$.

(i) $s_{j_k}^{(k-1)} \leq \frac{\lambda_1}{n-k+1}$ *for all* $k \in \{1,\ldots,h\}$.

(j) $\left\|s^{(k-1)}\right\|_2 \leq \left\|s^{(k)}\right\|_2$ *for all* $k \in \{1,\ldots,h\}$, *and hence* $\|s\|_2 \geq \|q\|_2$.

(k) $\|s\|_2 = \|q\|_2$ *if and only if* $q_j = 0$ *for all* $j \in J$.

Proof In other words, $s^{(k)}$ is constructed from $s^{(k-1)}$ by setting entry j_k to zero, and adjusting all remaining entries, but the ones previously set to zero, such that the L_1 norm is preserved. This generates a finite series of points progressively approaching C_I, see (a), where the final point is from C_I. As all relevant dot products vanish, see (b) and (e), this is a process of orthogonal projections. Hence the distance between points can be computed using the Pythagorean theorem, see (c) and (f). In (g) it is then shown that s is the unique projection of q onto C_I.

If the entry j_k in $s^{(k-1)}$ does not vanish, then the L_2 norm of the newly constructed point is greater than that of the original point, see (j) and (k). The entries with indices from J must be sufficiently small for this non-decreasing norm property to hold, see (h) and (i). The magnitude of these entries, however, is strongly connected with the magnitudes of respective entries from the original point q, that is, the rank is preserved from one point to its successor. Figure 9 gives an example for $n = 3$ in which cases the non-decreasing norm property holds.

Let $a_k := \frac{1}{n-k}\left(e - \sum_{i=1}^{k} e_{j_i}\right) - e_{j_k} \in \mathbb{R}^n$ for $k \in \{1,\ldots,h\}$. Then $s^{(k)} = s^{(k-1)} + s_{j_k}^{(k-1)} a_k$, and with induction follows $s^{(k)} = s^{(0)} + \sum_{i=1}^{k} s_{j_i}^{(i-1)} a_i$ for $k \in \{1,\ldots,h\}$.

(a) First note that $e^T a_k = \frac{1}{n-k}(n-k) - 1 = 0$ and that $a_k \in \mathbb{R}^n_{\geq 0}$ for all $k \in \{1,\dots,h\}$. It is now shown by induction that $s^{(k)}$ lies on the claimed face of C. For $k=1$, one obtains $s^{(1)}_{j_1} = 0$ and $s^{(1)}_i = q_i + \frac{1}{n-1} q_{j_1}$ for $i \neq j_1$. Thus $s^{(1)}_i \geq 0$ for all $i \in \{1,\dots,n\}$ because of $q_i \geq 0$ for all $i \in \{1,\dots,n\}$. Further $e^T s^{(1)} = e^T q + q_{j_1} e^T a_1 = e^T q = \lambda_1$, hence $s^{(1)} \in C_{\{1,\dots,n\}\setminus\{j_1\}}$.

For $k-1 \to k$, assume $s^{(k-1)}_i = 0$ for all $i \in \{j_1,\dots,j_{k-1}\}$, $s^{(k-1)}_i \geq 0$ for all $i \in \{1,\dots,n\}$ and $e^T s^{(k-1)} = \lambda_1$. Clearly, $s^{(k)}_i = s^{(k-1)}_i = 0$ holds for $i \in \{j_1,\dots,j_{k-1}\}$. Furthermore, one obtains $s^{(k)}_{j_k} = s^{(k-1)}_{j_k} - s^{(k-1)}_{j_k} = 0$. With $s^{(k-1)} \in \mathbb{R}^n_{\geq 0}$ and $a_k \in \mathbb{R}^n_{\geq 0}$ follows that $s^{(k)} \in \mathbb{R}^n_{\geq 0}$. Finally it is $e^T s^{(k)} = e^T s^{(k-1)} + s^{(k-1)}_{j_k} e^T a_k = e^T s^{(k-1)} = \lambda_1$. Hence $s^{(k)} \in C_{\{1,\dots,n\}\setminus\{j_1,\dots,j_k\}}$.

(b) For $i \in \{1,\dots,k\}$ follows

$$\begin{aligned}\left\langle e - \textstyle\sum_{\mu=1}^{i} e_{j_\mu},\, e - \sum_{\nu=1}^{k+1} e_{j_\nu} \right\rangle &= \langle e,\, e\rangle - \left\langle e,\, \textstyle\sum_{\nu=1}^{k+1} e_{j_\nu}\right\rangle - \left\langle \sum_{\mu=1}^{i} e_{j_\mu},\, e\right\rangle + \left\langle \sum_{\mu=1}^{i} e_{j_\mu},\, \sum_{\nu=1}^{k+1} e_{j_\nu}\right\rangle \\ &= n - (k+1) - i + i = n-k-1,\end{aligned}$$

and therefore

$$\begin{aligned}\langle a_i,\, a_{k+1}\rangle &= \tfrac{1}{(n-i)(n-k-1)} \left\langle e - \textstyle\sum_{\mu=1}^{i} e_{j_\mu},\, e - \sum_{\nu=1}^{k+1} e_{j_\nu}\right\rangle + \left\langle e_{j_i},\, e_{j_{k+1}}\right\rangle \\ &\quad - \tfrac{1}{n-i}\left\langle e - \textstyle\sum_{\mu=1}^{i} e_{j_\mu},\, e_{j_{k+1}}\right\rangle - \tfrac{1}{n-k-1}\left\langle e_{j_i},\, e - \sum_{\nu=1}^{k+1} e_{j_\nu}\right\rangle \\ &= \tfrac{n-k-1}{(n-i)(n-k-1)} + 0 - \tfrac{1}{n-i} - \tfrac{0}{n-k-1} = 0.\end{aligned}$$

Thus

$$\left\langle s^{(0)} - s^{(k)},\, s^{(k)} - s^{(k+1)}\right\rangle = \left\langle \textstyle\sum_{i=1}^{k} s^{(i-1)}_{j_i} a_i,\, s^{(k)}_{j_{k+1}} a_{k+1}\right\rangle = \sum_{i=1}^{k} s^{(i-1)}_{j_i} s^{(k)}_{j_{k+1}} \langle a_i,\, a_{k+1}\rangle = 0.$$

(c) Follows by induction using Proposition 4 and (b).

(d) Clearly, $\langle e_{j_k},\, a_i\rangle = \frac{1}{n-i}$ for $i \in \{1,\dots,k-1\}$, hence with induction follows

$$\begin{aligned}s^{(k-1)}_{j_k} &= \left\langle e_{j_k},\, s^{(k-1)}\right\rangle = \left\langle e_{j_k},\, s^{(0)}\right\rangle + \textstyle\sum_{i=1}^{k-1} s^{(i-1)}_{j_i} \langle e_{j_k},\, a_i\rangle = q_{j_k} + \sum_{i=1}^{k-1} \frac{1}{n-i} s^{(i-1)}_{j_i} \\ &\overset{\text{IH}}{=} q_{j_k} + \textstyle\sum_{i=1}^{k-1} \frac{1}{n-i} q_{j_i} + \sum_{i=1}^{k-1}\sum_{\mu=1}^{i-1} \frac{1}{n-i}\frac{1}{n-i+1} q_{j_\mu} = q_{j_k} + \sum_{i=1}^{k-1} q_{j_i}\left[\frac{1}{n-i} + \sum_{\mu=i+1}^{k-1} \frac{1}{n-\mu}\frac{1}{n-\mu+1}\right].\end{aligned}$$

Using $\sum_{i=1}^{k-1}\sum_{\mu=1}^{i-1} = \sum_{1\leq\mu<i\leq k-1} = \sum_{\mu=1}^{k-1}\sum_{i=\mu+1}^{k-1}$ the order of summation was changed after the induction step, and then the variables i and μ were swapped. For the claim to hold it is enough to show that $\frac{1}{n-i} + \sum_{\mu=i+1}^{k-1} \frac{1}{n-\mu}\frac{1}{n-\mu+1} = \frac{1}{n-k+1}$ for all $i \in \{1,\dots,k-1\}$, which follows by reverse induction.

(e) Proof by induction. Let $v \in C_I$, that is $e^T v = \lambda_1$ and $v_j = 0$ for all $j \in J$. For $k=0$, $s^{(0)} - s^{(k)} = 0$ and the claim follows. For $k-1 \to k$, first note that $\langle a_k,\, q\rangle = \frac{1}{n-k}\left(\lambda_1 - \sum_{i=1}^{k} q_{j_i}\right) - q_{j_k}$, $\left\langle a_k,\, s^{(k-1)}\right\rangle = \left\langle a_k,\, s^{(0)}\right\rangle + \sum_{i=1}^{k-1} s^{(i-1)}_{j_i} \langle a_k,\, a_i\rangle = \langle a_k,\, q\rangle$ because $\langle a_k,\, a_i\rangle = 0$ for all $i \in \{1,\dots,k-1\}$ as shown in (b), thus $\left\langle a_k,\, s^{(0)} - 2s^{(k-1)}\right\rangle = -\langle a_k,\, q\rangle$. Furthermore $\langle a_k,\, v\rangle = \frac{\lambda_1}{n-k}$ because $\langle e_{j_i},\, v\rangle = v_{j_i} = 0$ for all $i \in \{1,\dots,h\}$, and $\langle a_k,\, a_k\rangle = \|a_k\|_2^2 = \frac{1}{(n-k)^2}\left\|e - \sum_{i=1}^{k} e_{j_i}\right\|_2^2 + \left\|e_{j_k}\right\|_2^2 = 1 + \frac{1}{n-k}$.

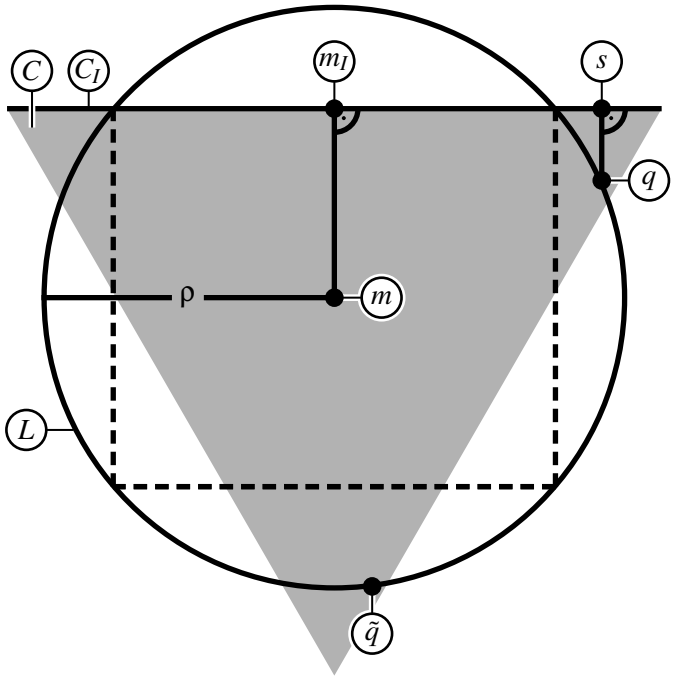


Figure 9: Situation of Lemma 26: The projection of a point q from within the simplex C onto one of its faces C_I yields point s. When q is sufficiently close to C_I, then $\|s\|_2 \geq \|q\|_2$. This does not hold when the point projected onto C_I is too far away. For example, the point $\tilde{q}$ which is located outside the dashed square with edge length $2\|m_I - m\|_2$ would yield a projection with a smaller Euclidean norm.

Therefore,

$$\begin{aligned}
\left\langle s^{(0)} - s^{(k)},\, s^{(k)} - v \right\rangle &= \left\langle s^{(0)} - s^{(k-1)} - s_{j_k}^{(k-1)} a_k,\, s^{(k-1)} + s_{j_k}^{(k-1)} a_k - v \right\rangle \\
&= \left\langle s^{(0)} - s^{(k-1)},\, s^{(k-1)} - v \right\rangle + s_{j_k}^{(k-1)} \left\langle a_k,\, s^{(0)} - 2s^{(k-1)} - s_{j_k}^{(k-1)} a_k + v \right\rangle \\
&\overset{\text{IH}}{=} s_{j_k}^{(k-1)} \left(- \langle a_k,\, q \rangle - s_{j_k}^{(k-1)} \langle a_k,\, a_k \rangle + \langle a_k,\, v \rangle \right) \\
&= s_{j_k}^{(k-1)} \left(-\tfrac{\lambda_1}{n-k} + \tfrac{1}{n-k} \textstyle\sum_{i=1}^{k} q_{j_i} + q_{j_k} - s_{j_k}^{(k-1)} \left(1 + \tfrac{1}{n-k}\right) + \tfrac{\lambda_1}{n-k} \right) \\
&= s_{j_k}^{(k-1)} \left(q_{j_k} \left(1 + \tfrac{1}{n-k}\right) + \tfrac{1}{n-k} \textstyle\sum_{i=1}^{k-1} q_{j_i} - s_{j_k}^{(k-1)} \left(1 + \tfrac{1}{n-k}\right) \right) = 0,
\end{aligned}$$

where the final equality was yielded using the statement from (d) and $\left(1 + \frac{1}{n-k}\right) \cdot \frac{1}{n-k+1} = \frac{1}{n-k}$.

(f) Follows using Proposition 4 and (e).

(g) A unique Euclidean projection exists because C_I is closed and convex. $s \in C_I$ with (a), and $\|q - s\|_2^2 \leq \|q - v\|_2^2$ for all $v \in C_I$ with (f). Therefore $s = \text{proj}_{C_I}(q)$.

(h) In the remainder of the proof assume that $\max_{j \in J} q_j \leq \min_{i \in I} q_i$ holds. It is first shown by induction that $s_{j_1}^{(k)} \leq \cdots \leq s_{j_h}^{(k)}$ for all $k \in \{0, \ldots, h\}$. For $k = 0$ this is fulfilled as requirement on q and by definition of J. For $k - 1 \to k$, let $\mu, \nu \in \{j_1, \ldots, j_h\}$ with $\mu < \nu$. Then with χ denoting the

indicator function and with $A := \chi_{\{j_k\}}(\mu) - \chi_{\{j_k\}}(\nu) + \frac{1}{n-k}\left(\chi_{\{j_1,\ldots,j_k\}^C}(\nu) - \chi_{\{j_1,\ldots,j_k\}^C}(\mu)\right)$ follows $s_\nu^{(k)} - s_\mu^{(k)} = s_\nu^{(k-1)} - s_\mu^{(k-1)} + s_{j_k}^{(k-1)} A$. Clearly, when $A \geq 0$ then the claim follows with the induction hypothesis and with $s_{j_k}^{(k-1)} \geq 0$ due to (a).

First consider the case of $\mu \in \{j_1,\ldots,j_{k-1}\}$. If $\nu \in \{j_1,\ldots,j_{k-1}\}$ also, then $A = 0$. If $\nu = j_k$, then $A = -1$, and hence $s_\nu^{(k)} - s_\mu^{(k)} = s_\nu^{(k-1)} - s_\mu^{(k-1)} - s_\nu^{(k-1)} = -s_\mu^{(k-1)}$ which however vanishes with (a). If $\nu \in \{j_{k+1},\ldots,j_h\}$, then $A = \frac{1}{n-k} \geq 0$. If $\mu = j_k$, then $\nu \in \{j_{k+1},\ldots,j_h\}$, and then $A = 1 + \frac{1}{n-k} \geq 0$. If $\mu \in \{j_{k+1},\ldots,j_h\}$, then $\nu \in \{j_{\mu+1},\ldots,j_h\}$, thus $A = 0$. Hence the first claim is always fulfilled.

Next, it is shown that $\max_{j\in J} s_j^{(k)} \leq \min_{i\in I} s_i^{(k)}$ for all $k \in \{0,\ldots,h\}$. For $k = 0$ this is the requirement on q. For $k-1 \to k$, let $i \in I$ and $j \in J$. It is then $\chi_{\{j_k\}}(i) = 0$, $\chi_{\{j_k\}}(j) \in \{0,1\}$, $\chi_{\{j_1,\ldots,j_k\}^C}(i) = 1$ and $\chi_{\{j_1,\ldots,j_k\}^C}(j) = 0$, therefore

$$\begin{aligned} s_i^{(k)} - s_j^{(k)} = \;& s_i^{(k-1)} - s_{j_k}^{(k-1)}\chi_{\{j_k\}}(i) + \tfrac{1}{n-k} s_{j_k}^{(k-1)} \chi_{\{j_1,\ldots,j_k\}^C}(i) \\ & - s_j^{(k-1)} + s_{j_k}^{(k-1)}\chi_{\{j_k\}}(j) - \tfrac{1}{n-k} s_{j_k}^{(k-1)} \chi_{\{j_1,\ldots,j_k\}^C}(j) \\ = \;& s_i^{(k-1)} - s_j^{(k-1)} + s_{j_k}^{(k-1)}\left(\tfrac{1}{n-k} + \chi_{\{j_k\}}(j)\right) \geq 0, \end{aligned}$$

where $s_i^{(k-1)} - s_j^{(k-1)} \geq 0$ holds by induction hypothesis.

(i) With (a) follows $\lambda_1 = e^T s^{(k-1)}$ and $s_{j_i}^{(k-1)} = 0$ for all $i \in \{1,\ldots,k-1\}$. $s_i^{(k-1)} \geq s_{j_k}^{(k-1)}$ holds for all $i \in I$ with (h), and $s_{j_i}^{(k-1)} \geq s_{j_k}^{(k-1)}$ for all $i \in \{k+1,\ldots,h\}$. Therefore,

$$\begin{aligned} \lambda_1 &= \textstyle\sum_{i=1}^n s_i^{(k-1)} = \sum_{i\in I} s_i^{(k-1)} + \sum_{i=1}^{k-1} s_{j_i}^{(k-1)} + s_{j_k}^{(k-1)} + \sum_{i=k+1}^{h} s_{j_i}^{(k-1)} \\ &\geq \big((n-h) + 1 + (h-k)\big) s_{j_k}^{(k-1)} = (n-k+1)\, s_{j_k}^{(k-1)}, \end{aligned}$$

and the claim follows because $n-k+1 > 0$.

(j) In (e) it was shown that $\|a_k\|_2^2 = 1 + \frac{1}{n-k}$. Furthermore,

$$\left\langle s^{(k-1)},\, a_k \right\rangle = \tfrac{1}{n-k}\left(\lambda_1 - \textstyle\sum_{i=1}^{k-1} s_{j_i}^{(k-1)} - s_{j_k}^{(k-1)}\right) - s_{j_k}^{(k-1)} = \tfrac{1}{n-k}\left(\lambda_1 - s_{j_k}^{(k-1)}\right) - s_{j_k}^{(k-1)},$$

because all $s_{j_i}^{(k-1)}$ vanish for $i \in \{1,\ldots,k-1\}$ using (a). Application of Proposition 4 yields

$$\begin{aligned} \left\|s^{(k)}\right\|_2^2 - \left\|s^{(k-1)}\right\|_2^2 &= \left\|s_{j_k}^{(k-1)} a_k\right\|_2^2 + 2 s_{j_k}^{(k-1)} \left\langle s^{(k-1)},\, a_k \right\rangle \\ &= s_{j_k}^{(k-1)}\left[s_{j_k}^{(k-1)}\left(1 + \tfrac{1}{n-k}\right) + \tfrac{2}{n-k}\left(\lambda_1 - s_{j_k}^{(k-1)}\right) - 2 s_{j_k}^{(k-1)}\right] \\ &= s_{j_k}^{(k-1)}\left[\tfrac{2\lambda_1}{n-k} - s_{j_k}^{(k-1)}\left(1 + \tfrac{1}{n-k}\right)\right], \end{aligned}$$

which is non-negative when $s_{j_k}^{(k-1)} \leq \left(1 + \frac{1}{n-k}\right)^{-1} \cdot \frac{2\lambda_1}{n-k} = \frac{2\lambda_1}{n-k+1}$. With (i) this is always fulfilled, hence $\left\|s^{(k-1)}\right\|_2 \leq \left\|s^{(k)}\right\|_2$, and $\|s\|_2 \geq \|q\|_2$ follows immediately using a telescoping sum argument.

(k) When $q_j = 0$ for all $j \in J$, then $s = q$ and the claim follows. When there is a $j_k \in \{j_1,\ldots,j_h\}$ with $q_{j_k} \neq 0$, let k be minimal such that either $k = 1$ or $q_{j_{k-1}} = 0$, hence $s_{j_k}^{(k-1)} = q_{j_k}$. With (i) follows $0 < s_{j_k}^{(k-1)} \leq \frac{\lambda_1}{n-k+1} < \frac{2\lambda_1}{n-k+1}$, and hence $\left\|s^{(k)}\right\|_2^2 - \left\|s^{(k-1)}\right\|_2^2 > 0$ with (j), thus $\left\|s\right\|_2 > \left\|q\right\|_2$. ■

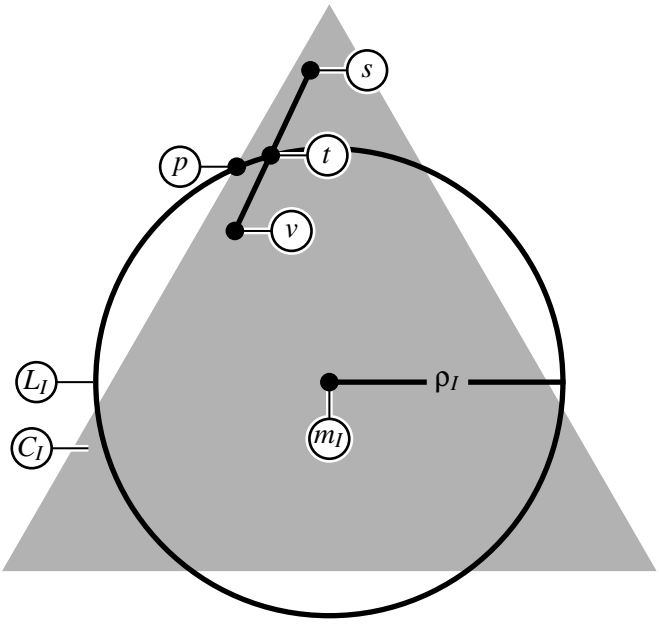


Figure 10: Sketch of the proof of Corollary 27: The projection of v onto D must be located on simplex face C_I. Assume there is a projection $q \notin C_I$, then it must be sufficiently close to C_I for application of Lemma 26. The projection s of q onto C_I is located outside $L_I = \{\, a \in L \mid a_i = 0 \text{ for all } i \notin I \,\}$. Hence the intersection t of the line between v and s and L_I is in C_I due to its convexity, and it is farther than the projection p of v onto $\tilde{D} = \{\, a \in D \mid a_i = 0 \text{ for all } i \notin I \,\}$. Therefore q cannot be the projection of v onto D.

The application of Lemma 26 then shows that the projection of a point from a face C_I of C onto D must reside on the same face C_I, given the original point is located within a sphere with squared radius ρ_I around m_I. As will be shown in Lemma 28, this is automatically fulfilled for projections from L onto C.

Corollary 27 *Let $I \subseteq \{1,\dots,n\}$, let $v \in C_I$ with $\|v\|_2 < \lambda_2$, and let $q \in \mathrm{proj}_D(v)$. Then $q \in C_I$.*

Proof Let $J := \{1,\dots,n\} \setminus I$, and let $q \in \mathrm{proj}_D(v)$. Assume there is at least one $j \in J$ with $q_j \neq 0$. For showing $\max_{j\in J} q_j \leq \min_{i\in I} q_i$, assume there are $i \in I$ and $j \in J$ with $q_j > q_i$. Then $v_i > 0$ and $v_j = 0$ because of $v \in C_I$. D is permutation-invariant using Remark 7 as intersection of permutation-invariant sets. Hence let $\tau := (i,j) \in S_n$ be the transposition swapping i and j, and consider

$$d := \|q - v\|_2^2 - \|P_\tau q - v\|_2^2 = 2\,(q_j - q_i)\,(v_i - v_j)\,.$$

It is $d > 0$ because of $q_j - q_i > 0$ and $v_i - v_j = v_i > 0$. Hence $\|P_\tau q - v\|_2 < \|q - v\|_2$ and $P_\tau q \in D$, which violates the minimality of q. Therefore, $\max_{j\in J} q_j \leq \min_{i\in I} q_i$ must hold.

A drawing for the next arguments is given in Figure 10. With Lemma 26 there is an $s \in C_I$ such that $\|q - v\|_2^2 = \|q - s\|_2^2 + \|s - v\|_2^2$ and $\|s\|_2 > \lambda_2$. Consider $f\colon [0,\,1] \to \mathbb{R},\ \beta \mapsto \|v + \beta\,(s - v)\|_2$. Clearly $f(0) = \|v\|_2 < \lambda_2$ and $f(1) = \|s\|_2 > \lambda_2$, hence with the intermediate value theorem there

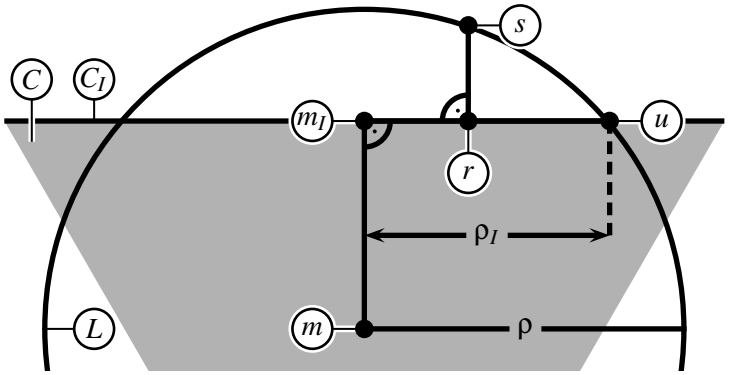


Figure 11: Situation of Lemma 28 and Lemma 30: The point s is projected onto simplex C yielding r, which resides on one of its faces C_I. From there point u can be constructed by projecting within C_I onto the target hypersphere.

exists a $\beta^* \in (0,\ 1)$ with $f(\beta^*) = \lambda_2$. Let $t := v + \beta^*(s-v) \in \mathbb{R}^n$, which lies in C_I because of $v, s \in C_I$ and C_I is convex. By construction $\|t\|_2 = \lambda_2$, hence $t \in \tilde{D} := \{\, a \in D \mid a_i = 0 \text{ for all } i \notin I \,\}$. Clearly, $\|v-t\|_2 + \|t-s\|_2 = |\beta^*| \cdot \|s-v\|_2 + |1-\beta^*| \cdot \|v-s\|_2 = \|s-v\|_2$. Let $p \in \text{proj}_{\tilde{D}}(v)$, then $\|v-p\|_2 \leq \|v-t\|_2$. Therefore,

$$\|q-v\|_2^2 = \|q-s\|_2^2 + \|s-v\|_2^2 = \|q-s\|_2^2 + (\|v-t\|_2 + \|t-s\|_2)^2 > \|v-t\|_2^2 \geq \|v-p\|_2^2.$$

Because of $\tilde{D} \subseteq D$, $p \in D$ also, hence $\|q-v\|_2 \leq \|p-v\|_2$, which contradicts $\|q-v\|_2 > \|v-p\|_2$. Hence, $q_j = 0$ for all $j \in J$ must hold, and thus $q \in C_I$. ■

Because C_I is isomorphic to a simplex, but with lower dimensionality than C, an algorithm can be constructed to compute the projection onto D, as discussed in the following.

C.3 Self-Similarity of the Feasible Set

The next Lemma summarizes previous results and analyzes projections from L onto C in greater detail. It shows that the solution set with respect to the projection onto D is not tampered, and that all solutions have zeros at the same positions as the projection onto C. Figure 11 provides orientation on the quantities discussed in Lemma 28.

Lemma 28 *Let $s \in L \setminus C$ and $r \in \text{proj}_C(s)$. Let $I := \{\, i \in \{1,\ldots,n\} \mid r_i \neq 0 \,\}$ and $d := |I|$. Then:*

(a) There exists $\hat{t} \in \mathbb{R}_{\geq 0}$ such that $r = \max(s - \hat{t} \cdot e,\ 0)$, with the maximum taken element-wise.

(b) $I \neq \emptyset$ and $I \neq \{1,\ldots,n\}$.

(c) $s_i > \hat{t}$ and $s_i - r_i = \hat{t}$ for all $i \in I$. $s_i \leq \hat{t}$ and $s_i - r_i = s_i$ for all $i \notin I$.

(d) $\langle r,\ s-r \rangle = \lambda_1 \hat{t}$.

(e) $\langle m_I,\ s-r \rangle = \lambda_1 \hat{t}$, thus $\langle m_I - r,\ s-r \rangle = 0$.

(f) $\text{proj}_D(r) \subseteq C_I$, hence from $q \in \text{proj}_D(r)$ follows $q_i = 0$ for all $i \notin I$.

(g) Let $q \in \mathrm{proj}_D(r)$. *Then* $\langle q-r,\ s-r\rangle = 0$, *and thus* $\|q-s\|_2^2 = \|q-r\|_2^2 + \|r-s\|_2^2$.

(h) $\mathrm{proj}_D(r) = \mathrm{proj}_D(s) \subseteq C_I$.

Proof (a) The existence of $\hat{t} \in \mathbb{R}$ such that $r = \max(s - \hat{t}\cdot e,\ 0)$ is guaranteed by Proposition 24. It remains to be shown that $\hat{t} \geq 0$. Consider the index set $J := \{\, j \in \{1,\ldots,n\} \mid s_j \geq 0 \,\}$ of non-negative entries of s. Because of $s \in L \setminus C$ one obtains $e^T s = \lambda_1$ and there is an index i with $s_i < 0$, hence $J \neq \{1,\ldots,n\}$. Therefore, $\lambda_1 = \sum_{j\in J} s_j + \sum_{j\notin J} s_j < \sum_{j\in J} s_j$. Now assume $\hat{t} < 0$, then $s_j - \hat{t} > 0$ for all $j \in J$, and hence $r_j = s_j - \hat{t}$ for all $j \in J$. Using $r \in C$ yields

$$\lambda_1 = \sum_{j=1}^n r_j \geq \sum_{j\in J} r_j = \sum_{j\in J}(s_j - \hat{t}) = \sum_{j\in J} s_j - |J|\cdot\hat{t} > \sum_{j\in J} s_j,$$

which contradicts $\sum_{j\in J} s_j > \lambda_1$. Hence $\hat{t} \geq 0$ must hold.

(b) Would $I = \varnothing$ hold, then $r = 0$, which is impossible because of $r \in C$. $I = \{1,\ldots,n\}$ would violate the existence of vanishing entries in r, as is guaranteed by Remark 23.

(c) Using (a): When $i \in I$, then $0 < r_i = s_i - \hat{t}$ and the claim follows. When $i \notin I$, then $r_i = 0$, hence $s_i - \hat{t} \leq 0$ and the claim follows.

(d) $e^T r = e^T s = \lambda_1$ because $r, s \in H$, hence using (c) yields

$$0 = \langle e,\ s-r\rangle = \sum_{i\in I}(s_i - r_i) + \sum_{i\notin I}(s_i - r_i) = \sum_{i\in I}\hat{t} + \sum_{i\notin I} s_i = d\hat{t} + \lambda_1 - \sum_{i\in I} s_i,$$

thus

$$\langle r,\ s-r\rangle = \sum_{i\in I} r_i(s_i - r_i) = \sum_{i\in I}(s_i - \hat{t})\,\hat{t} = \hat{t}\left(\sum_{i\in I} s_i - d\hat{t}\right) = \lambda_1\hat{t}.$$

(e) $\langle m_I,\ s-r\rangle = {}^{\lambda_1}\!/_d \cdot \sum_{i\in I}(s_i - r_i) = {}^{\lambda_1}\!/_d \cdot \sum_{i\in I}\hat{t} = \lambda_1\hat{t}$ with (c), and the claim follows with (d).

(f) $r \in C_I$ by definition of I. Using this, (c) and $\hat{t} \geq 0$ from (a) yields

$$\begin{aligned}\lambda_2^2 = \|s\|_2^2 &= \sum_{i\in I} s_i^2 + \sum_{i\notin I} s_i^2 = \sum_{i\in I}(r_i + \hat{t})^2 + \sum_{i\notin I} s_i^2 \\ &> \sum_{i\in I} r_i^2 + d\hat{t}^2 + 2\hat{t}\sum_{i\in I} r_i = \|r\|_2^2 + d\hat{t}^2 + 2\lambda_1\hat{t} \geq \|r\|_2^2,\end{aligned}$$

thus the claim holds using Corollary 27.

(g) With (f) follows $q \in C_I$. Hence $\langle q,\ s-r\rangle = \sum_{i\in I} q_i(s_i - r_i) = \hat{t}\sum_{i\in I} q_i = \lambda_1\hat{t}$ with (c), and the claims follow with (d) and Proposition 4.

(h) Let $p \in \mathrm{proj}_D(s)$ and $q \in \mathrm{proj}_D(r)$. Then using (g) and Proposition 4 one obtains that $\|q-s\|_2^2 - \|p-s\|_2^2 = \|q-r\|_2^2 - \|p-r\|_2^2 + 2\langle p-r,\ s-r\rangle$. With (c) and (d) follows

$$\begin{aligned}\langle p-r,\ s-r\rangle &= \sum_{i\in I} p_i(s_i - r_i) + \sum_{i\notin I} p_i(s_i - r_i) - \langle r,\ s-r\rangle = \sum_{i\in I} p_i\hat{t} + \sum_{i\notin I} p_i s_i - \lambda_1\hat{t} \\ &= \hat{t}\left(\lambda_1 - \sum_{i\notin I} p_i\right) + \sum_{i\notin I} p_i s_i - \lambda_1\hat{t} = \sum_{i\notin I} p_i(s_i - \hat{t}) \leq 0.\end{aligned}$$

Now $q \in \mathrm{proj}_D(r)$ yields $\|q-r\|_2^2 \leq \|p-r\|_2^2$, and hence $\|q-s\|_2^2 - \|p-s\|_2^2 \leq 0$, thus $q \in \mathrm{proj}_D(s)$. Similarly, $\|q-s\|_2^2 - \|p-s\|_2 \geq 0$ with $p \in \mathrm{proj}_D(s)$, so $\|q-r\|_2^2 - \|p-r\|_2^2 \geq -2\langle p-r,\ s-r\rangle \geq 0$, therefore $p \in \mathrm{proj}_D(r)$. ■

The following corollary states a similar result as in Theis et al. (2005). However, the proof here uses the notion of simplex projections instead of relying on pure analytical statements. The result presented here is stronger, as multiple entries of the vector can be set to zero simultaneously, while in Theis et al. (2005) at most one entry can be zeroed out in a single iteration.

Corollary 29 *Let* $s \in L \setminus C$ *and* $p \in \mathrm{proj}_D(s)$. *Then* $p_i = 0$ *for all* $i \in \{1,\ldots,n\}$ *with* $s_i \leq 0$.

Proof Let $i \in \{1,\ldots,n\}$ with $s_i \leq 0$. Let $r \in \text{proj}_C(s)$. With Lemma 28(a) follows $r_i = 0$ because $\hat{t} \geq 0$, and the claim follows with Lemma 28(h). ■

The final step is to meet the hypersphere constraint again. For this, the simplex projection r is projected onto the target hypersphere, simultaneously keeping already vanished entries at zero, yielding a point u. Lemma 30 gives an explicit formulation of this projection and shows that the solution set with respect to the projection onto D stays the same. Refer to Figure 11 for a sketch of the construction of u.

Lemma 30 *Let* $s \in L \setminus C$, $r := \text{proj}_C(s) = \max(s - \hat{t} \cdot e,\, 0)$ *with* $\hat{t} \in \mathbb{R}_{\geq 0}$ *using Lemma 28. Let* $I := \{\, i \in \{1,\ldots,n\} \mid r_i \neq 0 \,\}$ *and* $d := |I|$. *Let* $u := m_I + \delta(r - m_I)$ *where* $\delta := \sqrt{\rho_I}/\|r - m_I\|_2$. *Then:*

(a) $u \in L$ *and* $u_i = 0$ *for all* $i \notin I$, *hence* $u \in L_I$.

(b) $\text{proj}_D(u) \subseteq C_I$.

(c) $u = \text{proj}_{L_I}(r)$.

(d) $\text{proj}_D(u) = \text{proj}_D(r) = \text{proj}_D(s) \subseteq C_I$.

Proof (a) Clearly $u \in H$. With $m_I, r \in H$ and Remark 14 follows that $\langle m,\, m_I \rangle = \langle m,\, r \rangle = \lambda_1^2/n$. Moreover, $\langle m_I,\, m_I \rangle = \lambda_1^2/d$ and $\langle r,\, m_I \rangle = \sum_{i \in I} r_i \cdot \lambda_1/d = \lambda_1^2/d$, therefore $\langle r,\, m_I - m \rangle = \langle m_I,\, m_I - m \rangle$. With $u = (1-\delta)\, m_I + \delta r$ it is $\langle u,\, m_I - m \rangle = \langle m_I,\, m_I - m \rangle = \lambda_1^2\,(1/d - 1/n)$. Hence $\langle u - m_I,\, m_I - m \rangle = 0$. Further, $\|m - m_I\|_2^2 = \|m\|_2^2 + \|m_I\|_2^2 - 2\,\langle m,\, m_I \rangle = \lambda_1^2\,(1/d - 1/n)$.

Thus with Proposition 4, $\|u - m\|_2^2 = \|u - m_I\|_2^2 + \|m_I - m\|_2^2 = \rho_I + \lambda_1^2\,(1/d - 1/n) = \rho$, and with Lemma 15 follows $u \in L$. For $i \notin I$, one obtains $u_i = (1-\delta)\, e_i^T m_I + \delta r_i = 0$, hence $u \in L_I$.

(b) If $u \in C$, then $u \in D$ because of $u \in L$ with (a), and hence $u = \text{proj}_D(u)$. The claim then follows with $u_i = 0$ for all $i \notin I$.

If $u \notin C$, then let $q \in \text{proj}_D(u)$. With Corollary 29 applied to u follows that $q_i = 0$ for all i with $u_i \leq 0$, especially for all $i \notin I$. Hence the claim follows.

(c) Write $I = \{i_1,\ldots,i_d\}$ and consider $\varphi\colon \mathbb{R}^n \to \mathbb{R}^d$, $(x_1,\, \ldots,\, x_n)^T \mapsto (x_{i_1},\, \ldots,\, x_{i_d})^T$. Further, let $\tilde{H} := \{\, a \in \mathbb{R}^d \mid e^T a = \lambda_1 \,\}$, $\tilde{K} := \{\, q \in \mathbb{R}^d \mid \|q\|_2 = \lambda_2 \,\}$, $\tilde{L} := \tilde{H} \cap \tilde{K}$ and $\tilde{D} := \mathbb{R}^d_{\geq 0} \cap \tilde{H} \cap \tilde{K}$. Clearly, when $x_i = 0$ for all $i \notin I$, then $e^T x = e^T \varphi(x)$ and $\|x\|_2 = \|\varphi(x)\|_2$. Thus in this case membership of x in one of H, K, L or D implies membership of $\varphi(x)$ in $\tilde{H}$, $\tilde{K}$, $\tilde{L}$ or $\tilde{D}$, respectively.

Application of Lemma 17 to $\varphi(r)$ and $\varphi(u)$ implies that $\varphi(u) = \text{proj}_{\tilde{L}}(\varphi(r))$. Let $q \in L_I$, then $\varphi(q) \in \tilde{L}$, hence $\|\varphi(u) - \varphi(r)\|_2 \leq \|\varphi(q) - \varphi(r)\|_2$. From $i \notin I$ follows $r_i = u_i = q_i = 0$, hence $\|u - r\|_2 = \|\varphi(u) - \varphi(r)\|_2$ and $\|q - r\|_2 = \|\varphi(q) - \varphi(r)\|_2$, and the claim follows.

(d) For the converse of φ, let $\psi\colon \mathbb{R}^d \to \mathbb{R}^n$, $\tilde{x} \mapsto x$ where $x_i = 0$ for all $i \notin I$ and $x_i = \tilde{x}_j$ when there is a $j \in \{1,\ldots,d\}$ with $i = i_j$. Analogous to the above, membership of $\tilde{y}$ in one of $\tilde{H}$, $\tilde{K}$, $\tilde{L}$ or $\tilde{D}$ implies membership of $\psi(\tilde{y})$ in H, K, L or D, respectively.

With Lemma 28(f) and Lemma 28(h) it is enough to show $\text{proj}_D(u) = \text{proj}_D(r)$. Like in (c), from Lemma 17 follows as well that $\text{proj}_{\tilde{D}}(\varphi(r)) = \text{proj}_{\tilde{D}}(\varphi(u))$. Let $p \in \text{proj}_D(u)$ and $q \in \text{proj}_D(r)$, then $p \in C_I$ with (b) and $q \in C_I$ with Lemma 28(f), and thus $\varphi(p), \varphi(q) \in \tilde{D}$. Assume $\varphi(p) \notin \text{proj}_{\tilde{D}}(\varphi(u))$, then there exists an $a \in \tilde{D}$ with $\|\psi(a) - u\|_2 = \|a - \varphi(u)\|_2 < \|\varphi(p) - \varphi(u)\|_2 = \|p - u\|_2$, violating the minimality of p. Hence $\varphi(p) \in \text{proj}_{\tilde{D}}(\varphi(u))$, and analogously follows $\varphi(q) \in \text{proj}_{\tilde{D}}(\varphi(r))$. Now $\text{proj}_{\tilde{D}}(\varphi(r)) = \text{proj}_{\tilde{D}}(\varphi(u))$ implies that $\varphi(p) \in \text{proj}_{\tilde{D}}(\varphi(r))$ and $\varphi(q) \in \text{proj}_{\tilde{D}}(\varphi(u))$. Thus, $\|p - r\|_2 = \|\varphi(p) - \varphi(r)\|_2 = \|\varphi(q) - \varphi(r)\|_2 = \|q - r\|_2$, so $p \in \text{proj}_D(r)$, and one obtains analogously that $q \in \text{proj}_D(u)$. Therefore $\text{proj}_D(u) = \text{proj}_D(r)$. ■

With Lemma 30 a point u is constructed. If $u \in C$, then it is already the solution for the projection onto D. Otherwise, Lemma 28 and Lemma 30 can be applied once more, gaining a new point u. Lemma 28(b) states that the amount of nonzero entries of u must decrease, hence this process can be repeated for at most n iterations. If a point with only two non-vanishing entries results, it is guaranteed to be a solution by Proposition 22.

C.4 Proof of Theorem 2 and Theorem 3

Using the previous results it can now be shown that the proposed Algorithm 1 actually computes a correct solution, and that the algorithm always terminates in finite time.

Proof of Theorem 2 For proving partial correctness, let $x \in \mathbb{R}^n$ be arbitrary. Lemma 13 yields $\mathrm{proj}_D(x) = \mathrm{proj}_D(r)$ after line 1, and with Lemma 17 follows $\mathrm{proj}_D(x) = \mathrm{proj}_D(s)$ after line 2. There is a pre-test loop in line 3, and it has to be shown that the loop-invariant is $\mathrm{proj}_D(x) = \mathrm{proj}_D(s)$. At the beginning of the loop, $s \notin \mathbb{R}^n_{\geq 0}$ must hold, thus $s \in L \setminus C$. After line 4, $\mathrm{proj}_D(x) = \mathrm{proj}_D(r)$ holds with Lemma 28. Then with Lemma 30, $\mathrm{proj}_D(x) = \mathrm{proj}_D(s)$ is ensured after line 5, hence the loop-invariant holds. Thus, after the loop it is $\mathrm{proj}_D(x) = \mathrm{proj}_D(s)$ and $s \in D$, so $\mathrm{proj}_D(x) = s$. If $r = m$ in line 2 or $r = m_I$ in line 5, s can be chosen to be any point from L or L_I, respectively, for example the point given in Remark 18. In this case, the projection is not unique, but a valid representative is found.

To prove total correctness, it has to be shown that the loop in line 3 terminates. Remark 23 applied to C_I guarantees that the number of nonzero entries in s is strictly less at the end of the loop than the number of nonzero entries upon entering the loop. Hence, at most n iterations of the loop can be carried out, and when $|I| = 2$ the solution is already in D with Proposition 22. Thus the algorithm terminates in finite time. ■

It remains to be shown that the optimized variant is also correct.

Proof of Theorem 3 First note that Algorithm 3 consists of a procedure `proj_L` carrying out projections onto L and L_I in-place, and a main body. A function `proj_C` is called to obtain the information on how to perform projections onto C. This is carried out by Algorithm 2. Upon entry of the main body, the input vector x is sorted in descending order, yielding a vector y. The algorithm then operates on the sorted vector y, and undoes the sorting permutation at the end. Because H, L and C are permutation-invariant, the projections onto the respective sets are guaranteed to remain sorted with Lemma 9.

Therefore, y has not to be sorted again for the simplex projection, as Algorithm 2 would require. Also note from Lemma 28 that in the simplex projection the smallest elements are set to zero, and the original Algorithm 1 continues working on the d non-vanishing entries. Because of the order-preservation, entries $d+1, \ldots, n$ of y are zero, and all relevant information is concentrated in $y_1, \ldots, y_d$. Therefore, Algorithm 3 can continue working on these first d entries only, and the index set of non-vanishing entries is always $I = \{1, \ldots, d\}$. As the nonzero elements are stored contiguously in memory, access to y can be realized as a small unit-stride array. This is more efficient than working on a large and sparsely populated vector. Therefore, the loop starting at line 13 corresponds to the loop starting at line 3 in Algorithm 1. At the end of the main body, the sorting permutation τ is inverted and the entries from the sorted result vector y are stored in a new vector s. Because $y_{d+1}, \ldots, y_n = 0$, these entries can be ignored by setting the entire vector s to zero before-hand. ■

The proposed optimizations hence lead to the same solution which the original algorithm computes.

Appendix D. Analytical Properties of the Sparseness-Enforcing Projection Operator

In this appendix, it is studied in which situations $\pi_{\geq 0}$ and π as defined in Section 2.4 are differentiable, and hence continuous. Further, an explicit expression for their gradient is sought. It is clear by Theorem 2 that the projection of any point onto D can be written as finite composition of projections onto H, L, C and L_I, respectively. In other words, for all points $x \in \mathbb{R}^n \setminus R$ there exists a finite sequence of index sets $I_1, \ldots, I_h \subseteq \{1, \ldots, n\}$ with $I_j \supsetneq I_{j+1}$ for $j \in \{1, \ldots, h-1\}$ such that

$$\pi_{\geq 0}(x) = \left[\bigcirc_1^{j=h} \left(\mathrm{proj}_{L_{I_j}} \circ \mathrm{proj}_C \right) \circ \mathrm{proj}_L \circ \mathrm{proj}_H \right](x),$$

where $\bigcirc_1^{j=h}$ denotes iterated composition of functions, starting with $j = h$ and decreasing until $j = 1$, that is $\pi_{\geq 0}(x) = \mathrm{proj}_{L_{I_h}}(\mathrm{proj}_C(\cdots \mathrm{proj}_{L_{I_1}}(\mathrm{proj}_C(\mathrm{proj}_L(\mathrm{proj}_H(x)))) \cdots))$. The sequence $I_1, \ldots, I_h$ here depends on x. The intermediate goal is to show that this sequence remains fixed in a neighborhood of x, and that each projection in the chain is differentiable almost everywhere. This then implies differentiability of $\pi_{\geq 0}$ except for a null set. Because of the close relationship of π with $\pi_{\geq 0}$, π is then also differentiable almost everywhere as shown in the end of this appendix.

The projection onto H is differentiable everywhere, as is clear from its explicit formula given in Lemma 13. Considering L and L_I for $I \subseteq \{1, \ldots, n\}$, the projection is unique and can be cast as function $\mathbb{R}^n \to \mathbb{R}^n$ unless the point to be projected is equal to the barycenters m and m_I, respectively. By considering the explicit formulas given in Lemma 17 and Lemma 30, it is clear that these functions are differentiable as composition of differentiable functions. Thus only the projection onto the simplex C demands attention. Note that the number $\hat{t}$ from Proposition 24 is equal to the mean value of the entries of the argument, that survive the projection, modulo an additive constant:

Proposition 31 *Let $x \in \mathbb{R}^n \setminus C$ and $p := \mathrm{proj}_C(x)$. Then there is a set $I \subseteq \{1, \ldots, n\}$ such that $p = \max(x - \hat{t} \cdot e,\ 0)$ where $\hat{t} = {}^1\!/_{|I|} \cdot (\sum_{i \in I} x_i - \lambda_1)$.*

Proof Follows directly from Proposition 24 and Algorithm 2 by undoing the permutation τ. ■

Note that when $I = \{1, \ldots, n\}$, this is very similar to the projection onto H, see Lemma 13. The next result states a condition under which I is locally constant, and hence identifies points where the projection onto C is differentiable with a closed form expression:

Lemma 32 *Let $x \in \mathbb{R}^n \setminus C$, and let $p := \mathrm{proj}_C(x)$, $I \subseteq \{1, \ldots, n\}$ and $\hat{t} \in \mathbb{R}$ be given as in Proposition 31. When $x_i \neq \hat{t}$ for all $i \in \{1, \ldots, n\}$, then the following holds where $u := \sum_{i \in I} e_i \in \mathbb{R}^n$ and $v := e - u \in \mathbb{R}^n$ are the indicator vectors of I and I^C, respectively:*

(a) $p_i > 0$ *if and only if* $i \in I$.

(b) $p = x + {}^1\!/_d \cdot \left(\lambda_1 - u^T x\right) u - v \circ x$, *where $\circ$ denotes the Hadamard product and $d := |I|$.*

(c) There exists a constant $\varepsilon > 0$ and a neighborhood $U := \{ s \in \mathbb{R}^n \mid \|x - s\|_2 < \varepsilon \}$ of x, such that $\mathrm{sgn}(\mathrm{proj}_C(s)) = \mathrm{sgn}(p)$ for all $s \in U$.

(d) $\mathrm{proj}_C(s) = s + {}^1\!/_d \cdot \left(\lambda_1 - u^T s\right) u - v \circ s$ *for all* $s \in U$.

(e) $s \mapsto \mathrm{proj}_C(s)$ *is differentiable in x.*

Proof Only a sketch of a proof is presented here.

(a) Follows from the characterization of $\hat{t}$ given in Proposition 31.

(b) The identity can be validated directly using (a) and Proposition 31.

(c) Follows by choosing $\varepsilon := 1/2 \cdot \min_{i \in \{1,\dots,n\}} |x_i - \hat{t}|$, which is positive by requirement on x.

(d) Validation follows like in (b) using (c).

(e) The projection onto C can be written locally in closed form using (d). In the same neighborhood, the index set of vanishing entries of the projected points does not change. Hence, the projection is differentiable as a composition of differentiable functions. ■

It is clear that there are points in which $s \mapsto \mathrm{proj}_C(s)$ is continuous but not differentiable, for example points that are projected onto one of the vertices of C. The structure in this situation is locally equivalent to that of the absolute value function. However, for every point where the projection onto C is not differentiable, a subtle change is sufficient to find a point where the projection is differentiable:

Lemma 33 *Consider the function $p\colon \mathbb{R}^n \setminus C \to C$, $s \mapsto \mathrm{proj}_C(s)$ and let $x \in \mathbb{R}^n \setminus C$ be a point such that p is not differentiable in x. Then for all $\varepsilon > 0$ there exists a point $y \in \mathbb{R}^n$ with $\|x-y\|_2 < \varepsilon$ such that p is differentiable in y.*

Proof Let $\hat{t} \in \mathbb{R}$ be the separator from Proposition 31 for the projection onto C. Let the index set of all collisions with $\hat{t}$ be denoted by $J := \{\, j \in \{1,\dots,n\} \mid x_j = \hat{t} \,\}$, which is nonempty with Lemma 32 because p is not differentiable in x. Define $\delta := \varepsilon / \sqrt{4|J|} > 0$ and consider $y := x - \delta \sum_{j \in J} e_j \in \mathbb{R}^n$. Clearly, $\|x-y\|_2 = \varepsilon/2$. From Proposition 31 follows that the separating $\hat{t}$ for the projection onto C is independent of the entries of x with indices in J, as long as they are less than or equal to $\hat{t}$. Because $\delta > 0$, these entries in y are strictly smaller than $\hat{t}$, hence p is differentiable in y with Lemma 32. ■

Therefore the set on which $s \mapsto \mathrm{proj}_C(s)$ is not differentiable forms a null set. The next result gathers the gradients of the individual projections involved in the computation of the sparseness-enforcing projection operator with respect to σ. Using the chain rule, the gradient of $\pi_{\geq 0}$ can be derived afterwards as multiplication of the individual gradients.

Lemma 34 *The individual projections for $\pi_{\geq 0}$ are differentiable almost everywhere. Their gradients are given as follows:*

(a) $\frac{\partial \mathrm{proj}_H(x)}{\partial x} = E_n - 1/n \cdot ee^T$, *where $E_n \in \mathbb{R}^{n \times n}$ is the identity matrix.*

(b) $\frac{\partial \mathrm{proj}_L(x)}{\partial x} = \frac{\sqrt{\rho}}{\|x-m\|_2} \left(E_n - 1/\|x-m\|_2^2 \cdot (x-m)(x-m)^T \right)$.

(c) $\frac{\partial \mathrm{proj}_C(x)}{\partial x} = E_n - 1/d \cdot uu^T - \mathrm{diag}(v)$. *Here, $I := \{\, i \in \{1,\dots,n\} \mid e_i^T \mathrm{proj}_C(x) \neq 0 \,\}$ is the index set of nonzero entries of the projection onto C, $d := |I|$, $u := \sum_{i \in I} e_i \in \mathbb{R}^n$ and $v := e - u \in \mathbb{R}^n$.*

(d) $\frac{\partial \mathrm{proj}_{L_I}(x)}{\partial x} = \frac{\sqrt{\rho_I}}{\|x-m_I\|_2} \left(E_n - 1/\|x-m_I\|_2^2 \cdot (x-m_I)(x-m_I)^T \right)$.

Proof (a) Follows from the closed form expression in Lemma 13.

(b) Lemma 17 yields $\mathrm{proj}_L(x) = m + \delta(x) \cdot (x-m)$ with $\delta(x) = \sqrt{\rho}/\|x-m\|_2$. With the quotient rule follows $\partial \delta(x)/\partial x = -\sqrt{\rho}/\|x-m\|_2^3 \cdot (x-m)^T$, as ρ does not depend on x. The claim then follows by application of the product rule.

(c) Follows from Lemma 32, similar to (a), using $v \circ x = \mathrm{diag}(v)x$.

(d) Follows exactly as in (b). ■

Clearly, the gradients for proj_H and for proj_L are special cases of the gradients of proj_C and proj_{L_I}, respectively. Therefore, they need no separate handling in the computation of the overall gradient. Exploiting the special structure of the matrices involved and the readily sorted input as in Algorithm 3, the gradient computation can be further optimized. For the remainder of this appendix, let $O_{a\times b} \in \{0\}^{a\times b}$ and $J_{a\times b} \in \{1\}^{a\times b}$ denote the matrices with a rows and b columns where all entries equal zero and unity, respectively.

Theorem 35 *Let $x \in \mathbb{R}^n$ be sorted in descending order and $\pi_{\geq 0}$ be differentiable in x. Let $h \in \mathbb{N}$ denote the number of iterations Algorithm 3 needs to terminate. In every iteration of the algorithm, store the following values for $i \in \{1,\ldots,h\}$, where line numbers reference Algorithm 3:*

- *$d_i \in \mathbb{N}$ denoting the current dimensionality as determined by lines 12 and 14.*
- *$\delta_i := \sqrt{\rho/\varphi} \in \mathbb{R}$ where ρ and φ are determined in lines 2 and 3, respectively.*
- *$r(i) := y - m_I \in \mathbb{R}^{d_i}$ as computed in line 7.*

Let $N := d_h = \|\pi_{\geq 0}(x)\|_0$ denote the number of nonzero entries in the projection onto D, and define $s(i) := \left(e_1^T r(i),\ \ldots,\ e_N^T r(i)\right)^T \in \mathbb{R}^N$ as the first N entries of each $r(i)$. For $i \in \{1,\ldots,h\}$ let

$$A_i := \delta_i E_N - {}^{\delta_i}/_{d_i} \cdot J_{N\times N} - \alpha_i s(i)s(i)^T + {}^{\alpha_i}/_{d_i} \cdot s(i)s(i)^T J_{N\times N} \in \mathbb{R}^{N\times N} \text{ where } \alpha_i := {}^{\delta_i}/_{\|r(i)\|_2^2},$$

and let $A := \prod_1^{i=h} A_i = A_h \cdots A_1 \in \mathbb{R}^{N\times N}$. Then the gradient of $\pi_{\geq 0}$ in x is $\text{diag}\left(A,\ O_{(n-N)\times(n-N)}\right)$, that is a block diagonal matrix where the quadratic submatrix with row and column indices from 1 to N is given by A, and where all other entries vanish.

Proof The gradient of projections onto H is merely a special case of projections onto C, which also applies to the respective projections onto L and L_I, see Lemma 34. Hence, the very first iteration is a special case of iterations with $i > 1$. Consider one single iteration $i \in \{1,\ldots,h\}$ of Algorithm 3, that is the computation of $\text{proj}_{L_I} \circ \text{proj}_C$ for some $I \subseteq \{1,\ldots,n\}$. Write $d := d_i$, $\delta := \delta_i$, $\alpha := \alpha_i$ and $r := r(i)$ for short. Because the input vector x is sorted by requirement, all intermediate vectors that are projected are sorted as well using Lemma 9. Thus $I = \{1,\ldots,d\}$ holds.

With Lemma 34, the gradient $G_C \in \mathbb{R}^{n\times n}$ of the projection onto both H and C is of the form $G_C := E_n - {}^1/_d \cdot uu^T - \text{diag}(v)$, where $u := \sum_{i=1}^d e_i$ and $v := e - u$. Let $q := (r_1,\ldots,r_d,0,\ldots,0)^T \in \mathbb{R}^n$ be a copy of r padded with zeros to achieve full dimensionality n. The gradient of the projection onto L, and in general L_I, is given by $G_L := \delta E_n - \alpha qq \in \mathbb{R}^{n\times n}$ using Lemma 34. The gradient of the whole iteration is then given by the chain rule, yielding

$$G := G_L G_C = \delta E_n - {}^{\delta}/_d \cdot uu^T - \delta\,\text{diag}(v) - \alpha qq^T + {}^{\alpha}/_d \cdot qq^T uu^T + \alpha qq^T \text{diag}(v) \in \mathbb{R}^{n\times n}.$$

Write $O := O_{(n-d)\times(n-d)}$, then G is a block diagonal matrix of a matrix from $\mathbb{R}^{d\times d}$ and O: Note that $E_n - \text{diag}(v) = \text{diag}(E_d, O)$, $uu^T = \text{diag}(J_{d\times d}, O)$, and $qq^T = \text{diag}(rr^T, O)$. Therefore, $qq^T \text{diag}(v) = \text{diag}(rr^T, O) \cdot \left(\begin{smallmatrix} 0 & 0 \\ 0 & E_{n-d} \end{smallmatrix}\right) = 0$, and $qq^T uu^T = \text{diag}(rr^T J_{d\times d}, O)$. Thus

$$\begin{aligned} G &= \delta\,\text{diag}(E_d, O) - {}^{\delta}/_d \cdot \text{diag}(J_{d\times d}, O) - \alpha\,\text{diag}(rr^T, O) + {}^{\alpha}/_d \cdot \text{diag}(rr^T J_{d\times d}, O) \\ &= \text{diag}\left(\delta E_d - {}^{\delta}/_d \cdot J_{d\times d} - \alpha rr^T + {}^{\alpha}/_d \cdot rr^T J_{d\times d},\ O\right). \end{aligned}$$

By denoting the gradient of iteration i by matrix $G_i \in \mathbb{R}^{n\times n}$ for $i \in \{1,\ldots,h\}$ and by application of the chain rule follows that the gradient of all iterations is given by $\prod_1^{i=h} G_i$. In this matrix, all

entries but the top left submatrix of dimensionality $N \times N$ are vanishing, where $N = d_h$. This is because the according statement applies to G_i for the top left submatrix of dimensionality $d_i \times d_i$, and $d_1 > \cdots > d_h$ holds, and only the according entries survive the matrix multiplication. Therefore it is sufficient to compute only the top left $N \times N$ entries of the gradients of the individual iterations, as the remaining entries are not relevant for the final gradient. This is reflected by the definition of the matrices A_i for $i \in \{1, \ldots, h\}$ from the claim. ■

The gradient can thus be computed using matrix-matrix multiplications, where the matrices are square and the edge length is the number of nonzero entries in the result of the projection. This computation is more efficient than using the $n \times n$ matrices of the individual projections. However, when the target degree of sparseness is low, and thus the amount of nonzero entries N in the result of the projection is large, gradient computation can become very inefficient. In practice, often only the product of the gradient with an arbitrary vector is required. In this case, the procedure can be sped up by exploiting the special structure of the gradient of $\pi_{\geq 0}$:

Corollary 36 *Let $x \in \mathbb{R}^n$ be sorted in descending order and $\pi_{\geq 0}$ be differentiable in x. The product of the gradient of $\pi_{\geq 0}$ in x with an arbitrary vector can be computed using vector operations only.*

Proof Note that because of the associativity of the matrix product it is enough to consider the product of the gradient $G \in \mathbb{R}^{n \times n}$ of one iteration of Algorithm 3 with one vector $y \in \mathbb{R}^n$. Because of the statements of Theorem 35, it suffices to consider the top left $N \times N$ entries of G and the first N entries of y, as all other entries vanish. Therefore let $A := \delta E_N - \delta/d \cdot J_{N \times N} - \alpha s s^T + \alpha/d \cdot s s^T J_{N \times N} \in \mathbb{R}^{N \times N}$ be the non-vanishing block of G as given by Theorem 35, let $u := J_{N \times 1} \in \mathbb{R}^N$ be the vector of ones such that $uu^T = J_{N \times N}$, and let $z := (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ denote the vector with the first entries of y. Using matrix product associativity and distributivity over multiplication with a scalar yields

$$Az = \delta\,(z - 1/d \cdot \langle z,\, u\rangle \cdot u) + \alpha\,(1/d \cdot \langle s,\, u\rangle\, \langle z,\, u\rangle - \langle s,\, z\rangle)\, s,$$

where $\langle z,\, u\rangle = \sum_{i=1}^N z_i$ and $\langle s,\, u\rangle = \sum_{i=1}^N s_i$. Hence Az can be computed in-place from z by subtraction of a scalar value from all entries, rescaling by δ, and adding a scaled version of vector s. ■

Although in Theorem 35 and Corollary 36 it was necessary that the input vector is sorted, the general case can easily be recovered:

Proposition 37 *Let $x \in \mathbb{R}^n$ be a point, $\tau \in S_n$ such that $y := P_\tau x \in \mathbb{R}^n$ is sorted in descending order and $\pi_{\geq 0}$ be differentiable in y with gradient $G \in \mathbb{R}^{n \times n}$. Then $\pi_{\geq 0}$ is also differentiable in x, and the gradient is $P_\tau^T G P_\tau$.*

Proof Follows with $P_\tau^T = P_{\tau^{-1}} = P_\tau^{-1}$, $\pi_{\geq 0}(x) = P_\tau^T \pi_{\geq 0}(P_\tau x) = P_\tau \pi_{\geq 0}(y)$ and the chain rule. ■

Likewise, the gradient for the unrestricted projection π can be computed from the gradient for $\pi_{\geq 0}$:

Proposition 38 *Let $x \in \mathbb{R}^n$ be a point such that $\pi_{\geq 0}$ is differentiable in $|x|$ with gradient $G \in \mathbb{R}^{n \times n}$. Let $s \in \{\pm 1\}^n$ be given such that $\pi(x) = s \circ \pi_{\geq 0}(|x|)$. Then π is differentiable in x, and the gradient is* $\operatorname{diag}(s) G \operatorname{diag}(s)$.

Proof Follows analogously to Proposition 37, using $|x| = s \circ x = \operatorname{diag}(s) x$. ■

Summing up, the gradient of the projection onto $S_{\geq 0}^{(\lambda_1, \lambda_2)}$ and $S^{(\lambda_1, \lambda_2)}$ can be computed efficiently by bookkeeping a few values as discussed in Theorem 35, and applying simple operations to recover the general case. When only the product of the gradient with a vector is required, the computation can be made more efficient as stated in Corollary 36. Direct application of Theorem 35 should be avoided in this situation because of the high computational complexity.

Appendix E. Gradients for SOAE Learning

The objective function E_{SOAE} is a convex combination of two similarity measures s_R and s_C. The degrees of freedom W, W_{out} and θ_{out} of the SOAE architecture should be tuned by gradient-based methods to minimize these functions. This appendix reports the gradient information needed for reproduction of the experiments. The first statement addresses the reconstruction module.

Proposition 39 *It is* $\left(\partial s_R(\tilde{x},\, x)/\partial W\right)^T = xgWf'(u) + g^T h^T \in \mathbb{R}^{d\times n}$ *where* $g := \partial s_R(\tilde{x},x)/\partial \tilde{x} \in \mathbb{R}^{1\times d}$ *is the gradient of the similarity measure with respect to its first argument. Additionally,* $\left(\partial s_R(\tilde{x},\, x)/\partial W_{\text{out}}\right)^T = 0 \in \mathbb{R}^{n\times c}$ *and* $\partial s_R(\tilde{x},\, x)/\partial \theta_{\text{out}} = 0 \in \mathbb{R}^{1\times c}$.

Proof As s_R does not depend on W_{out} or θ_{out}, the respective gradients vanish. The symmetry between encoding and decoding yields $\tilde{x} := W \cdot f\left(W^T x\right)$. The gradient for W follows using the chain rule and the product rule for matrix calculus, see Neudecker (1969) and Vetter (1970). ■

The correlation coefficient is the recommended choice for the similarity measure of the reconstruction module because it is normed and invariant to affine-linear transformations. It is also differentiable almost everywhere:

Proposition 40 *If* s_R *is the correlation coefficient and* $x, \tilde{x} \in \mathbb{R}^d \setminus \{0\}$*, then*

$$\left(\frac{\partial s_R(\tilde{x},x)}{\partial \tilde{x}}\right)^T = \frac{1}{\sqrt{\lambda\mu}}\left(x - \frac{\langle e,\, x\rangle}{d}e\right) - \frac{s_R(\tilde{x},x)}{\lambda}\left(\tilde{x} - \frac{\langle e,\, \tilde{x}\rangle}{d}e\right) \in \mathbb{R}^d,$$

where all entries of $e \in \mathbb{R}^d$ *are unity,* $\lambda := \|\tilde{x}\|_2^2 - 1/d \cdot \langle e,\, \tilde{x}\rangle^2 \in \mathbb{R}$ *and* $\mu := \|x\|_2^2 - 1/d \cdot \langle e,\, x\rangle^2 \in \mathbb{R}$.

Proof One obtains $\sqrt{\lambda\mu} \cdot s_R(\tilde{x},x) = \langle x,\, \tilde{x}\rangle - 1/d \cdot \langle e,\, \tilde{x}\rangle \langle e,\, x\rangle$ because s_R is the correlation coefficient. The claim then follows with the quotient rule. ■

The gradients of the similarity measure for classification capabilities are essentially equal to those of an ordinary two-layer neural network, and can be computed using the back-propagation algorithm (Rumelhart et al., 1986). However, the pairing of the softmax transfer function with the cross-entropy error function provides a particularly simple structure of the gradient (Dunne and Campbell, 1997). For completeness, the gradients of the classification module of SOAE are summarized:

Proposition 41 *If* s_C *is the cross-entropy error function,* g *is the softmax transfer function and the target vector for classification* t *is a one-of-c code, then* $\left(\partial s_C(y,\, t)/\partial W_{\text{out}}\right)^T = h \cdot (y-t)^T \in \mathbb{R}^{n\times c}$, $\partial s_C(y,\, t)/\partial \theta_{\text{out}} = (y-t)^T \in \mathbb{R}^{1\times c}$ *and* $\left(\partial s_C(y,\, t)/\partial W\right)^T = x \cdot \left((y-t)^T W_{\text{out}}^T f'(u)\right) \in \mathbb{R}^{d\times n}$.

Proof Basic matrix calculus (Neudecker, 1969; Vetter, 1970) yields $\partial s_C(y,\, t)/\partial \theta_{\text{out}} = \left(\partial s_C(y,\, t)/\partial y\right) \cdot g'(y)$, $\left(\partial s_C(y,\, t)/\partial W_{\text{out}}\right)^T = h \cdot \left(\partial s_C(y,\, t)/\partial \theta_{\text{out}}\right)$ and $\left(\partial s_C(y,\, t)/\partial W\right)^T = x \cdot \left(\left(\partial s_C(y,\, t)/\partial \theta_{\text{out}}\right) \cdot W_{\text{out}}^T f'(u)\right)$. By requirement $\partial s_C(y,\, t)/\partial y = -(t \circledcirc y)^T$, where $\circledcirc$ denotes the element-wise quotient, $g'(y) = \text{diag}(y) - yy^T$ and $\sum_{i=1}^c t_i = 1$. Therefore $\left(\partial s_C(y,\, t)/\partial \theta_{\text{out}}\right)^T = \left(yy^T - \text{diag}(y)\right) \cdot (t \circledcirc y) = y \cdot \langle y,\, t \circledcirc y\rangle - y \circ t \circledcirc y = y - t$ using $\langle y,\, t \circledcirc y\rangle = \sum_{i=1}^c y_i \cdot t_i/y_i = 1$, and the claim follows. ■

As E_{SOAE} is a convex combination of the reconstruction error and the classification error, its overall gradient follows immediately from Proposition 39 and Proposition 41. Proposition 40, the results from Appendix D, and the gradient of the L_0 projection as described in Section 2.4 can then be used to compute the explicit gradients for the procedure proposed in this paper.

References

L. Acion, J. J. Peterson, S. Temple, and S. Arndt. Probabilistic index: An intuitive non-parametric approach to measuring the size of treatment effects. *Statistics in Medicine*, 25(4):591–602, 2006.

E. B. Baum, J. Moody, and F. Wilczek. Internal representations for associative memory. *Biological Cybernetics*, 59(4–5):217–228, 1988.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

T. Blumensath and M. E. Davies. A simple, efficient and near optimal algorithm for compressed sensing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3357–3360, 2009.

L. Bottou and Y. LeCun. Large scale online learning. In *Advances in Neural Information Processing Systems 16*, pages 217–224, 2004.

D. M. Bradley and J. A. Bagnell. Differentiable sparse coding. In *Advances in Neural Information Processing Systems 21*, pages 113–120, 2009.

Y. Chen and X. Ye. Projection onto a simplex. Technical Report arXiv:1101.6081v2, University of Florida, 2011.

D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010.

D. C. Cireşan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

D. DeCoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46 (1–3):161–190, 2002.

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

F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer, 2001.

D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6): 797–829, 2006.

F. Downton. The estimation of Pr(Y < X) in the normal case. *Technometrics*, 15(3):551–558, 1973.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, pages 272–279, 2008.

R. A. Dunne and N. A. Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proceedings of the Australasian Conference on Neural Networks*, pages 181–185, 1997.

R. L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

R. L. Dykstra and J. P. Boyle. An algorithm for least squares projections onto the intersection of translated, convex cones. *Journal of Statistical Planning and Inference*, 15(3):391–399, 1987.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. Technical Report arXiv:1001.0736v1, Stanford University, 2010.

K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the International Conference on Machine Learning*, pages 399–406, 2010.

R. J. Grissom. Probability of the superior outcome of one treatment over another. *Journal of Applied Psychology*, 79(2):314–316, 1994.

M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *Journal of Machine Learning Research*, 7:1385–1407, 2006.

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

J.-B. Hiriart-Urruty. At what points is the projection mapping differentiable? *The American Mathematical Monthly*, 89(7):456–458, 1982.

Y. Hochberg and A. C. Tamhane. *Multiple Comparison Procedures*. John Wiley & Sons, Inc., 1987.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology*, 148(3):574–591, 1959.

N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

A. Hyvärinen, P. O. Hoyer, and E. Oja. Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation. In *Advances in Neural Information Processing Systems 11*, pages 473–478, 1999.

J. Karbowski. How does connectivity between cortical areas depend on brain size? Implications for efficient computation. *Journal of Computational Neuroscience*, 15(3):347–356, 2003.

T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, C-21(4):353–359, 1972.

W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

A. J. Laub. *Matrix Analysis for Scientists and Engineers*. Society for Industrial and Applied Mathematics, 2004.

S. B. Laughlin and T. J. Sejnowski. Communication in neuronal networks. *Science*, 301(5641): 1870–1874, 2003.

Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998. Available electronically at `http://yann.lecun.com/exdb/mnist`.

Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems 2*, pages 598–605, 1990.

D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755):788–791, 1999.

H. Levene. Robust tests for equality of variances. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press, 1960.

S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 207–212, 2001.

J. Liu and J. Ye. Efficient euclidean projections in linear time. In *Proceedings of the International Conference on Machine Learning*, pages 657–664, 2009.

J. Liu and J. Ye. Efficient ℓ_1/ℓ_q norm regularization. Technical Report arXiv:1009.4766v1, Arizona State University, 2010.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems 21*, pages 1033–1040, 2009.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

H. Markram, J. Lübke, M. Frotscher, A. Roth, and B. Sakmann. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *Journal of Physiology*, 500(2):409–440, 1997.

A. Mason, A. Nicoll, and K. Stratford. Synaptic transmission between individual pyramidal neurons of the rat visual cortex in vitro. *Journal of Neuroscience*, 11(1):72–84, 1991.

C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–18, 2006.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24 (2):227–234, 1995.

H. Neudecker. Some theorems on matrix differentiation with special reference to Kronecker matrix products. *Journal of the American Statistical Association*, 64(327):953–963, 1969.

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.

B. A. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, 2004.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

G. Palm. On associative memory. *Biological Cybernetics*, 36(1):19–31, 1980.

J. Pizarro, E. Guerrero, and P. L. Galindo. Multiple comparison procedures applied to model selection. *Neurocomputing*, 48(1–4):155–173, 2002.

A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for $l_{1,\infty}$ regularization. In *Proceedings of the International Conference on Machine Learning*, pages 857–864, 2009.

M. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems 20*, pages 1185–1192, 2008.

M. Rehn and F. T. Sommer. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience*, 22(2): 135–146, 2007.

J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

B. Schölkopf. *Support Vector Learning*. PhD thesis, Technische Universität Berlin, 1997.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4):591–611, 1965.

P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 958–962, 2003.

S. Sra. Fast projections onto mixed-norm balls with applications. *Data Mining and Knowledge Discovery*, 25(2):358–377, 2012.

F. J. Theis and T. Tanaka. Sparseness by iterative projections onto spheres. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 709–712, 2006.

F. J. Theis, K. Stadlthanner, and T. Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *Proceedings of the European Signal Processing Conference*, pages 1672–1675, 2005.

M. Thom, R. Schweiger, and G. Palm. Supervised matrix factorization with sparseness constraints and fast inference. In *Proceedings of the International Joint Conference on Neural Networks*, pages 973–979, 2011a.

M. Thom, R. Schweiger, and G. Palm. Training of sparsely connected MLPs. In *Lecture Notes in Computer Science*, volume 6835, pages 356–365, 2011b.

E. van den Berg, M. Schmidt, M. P. Friedlander, and K. Murphy. Group sparsity via linear-time projection. Technical Report TR-2008-09, Department of Computer Science, University of British Columbia, 2008.

W. J. Vetter. Derivative operations on matrices. *IEEE Transactions on Automatic Control*, 15(2): 241–244, 1970.

W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276, 2000.

J. von Neumann. *Functional Operators, Volume II: The Geometry of Orthogonal Spaces*. Princeton University Press, 1950.

J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.

D. R. Wilson and T. R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(11):1429–1451, 2003.

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

Journal of Machine Learning Research 14 (2013) 1145-1173 Submitted 8/09; Revised 11/12; Published 4/13

Stress Functions for Nonlinear Dimension Reduction, Proximity Analysis, and Graph Drawing

Lisha Chen LISHA.CHEN@YALE.EDU
Department of Statistics
Yale University
New Haven, CT 06511, USA

Andreas Buja
Department of Statistics
University of Pennsylvania
Philadelphia, PA 19104, USA

Editor: Mikhail Belkin

Abstract

Multidimensional scaling (MDS) is the art of reconstructing pointsets (embeddings) from pairwise distance data, and as such it is at the basis of several approaches to nonlinear dimension reduction and manifold learning. At present, MDS lacks a unifying methodology as it consists of a discrete collection of proposals that differ in their optimization criteria, called "stress functions". To correct this situation we propose (1) to embed many of the extant stress functions in a parametric family of stress functions, and (2) to replace the ad hoc choice among discrete proposals with a principled parameter selection method. This methodology yields the following benefits and problem solutions: (a) It provides guidance in tailoring stress functions to a given data situation, responding to the fact that no single stress function dominates all others across all data situations; (b) the methodology enriches the supply of available stress functions; (c) it helps our understanding of stress functions by replacing the comparison of discrete proposals with a characterization of the effect of parameters on embeddings; (d) it builds a bridge to graph drawing, which is the related but not identical art of constructing embeddings from graphs.

Keywords: multidimensional scaling, force-directed layout, cluster analysis, clustering strength, unsupervised learning, Box-Cox transformations

1. Introduction

In the last decade and a half an important line of work in machine learning has been nonlinear dimension reduction and manifold learning. Many approaches used in this area are based on inter-object distances and the faithful reproduction of such distances by so-called "embeddings," that is, mappings of the objects of interest (e.g., images, signals, documents, genes, network vertices) to points in a low dimensional space such that the low-dimensional distances mimic the "true" inter-object distances as best as possible. Examples of distance-based methods include, among many others: kernel PCA (KPCA; Schölkopf et al., 1998) "Isomap" (Tenenbaum, De Silva, and Langford, 2000) , kernel-based semidefinite programming (SDP; Lu, Keleş, Wright, and Wahba, 2005; Weinberger, Sha, Zhu, and Saul, 2007), and two very different methods that both go under the name "local multidimensional scaling" by Venna and Kaski (2006) and by the present authors (Chen

and Buja, 2009). These can all be understood as outgrowths of various forms of multidimensional scaling (MDS).

MDS approaches are divided into two distinct classes: (1) classical scaling of the Torgerson-Gower type (the older approach) is characterized by the indirect approximation of target distances through inner products; (2) distance scaling of the Kruskal-Shepard type is characterized by the direct approximation of target distances. The relative merits are as follows: classical scaling approaches often reduce to eigendecompositions that provide hierarchical solutions (increasing the embedding dimension means adding more coordinates to an existing embedding); distance scaling approaches are non-hierarchical and require high-dimensional optimizations, but they tend to force more information into any given embedding dimension. It is this class of distance scaling approaches for which the present article provides a unified methodology.

Distance scaling approaches differ in their choices of a "stress function", that is, a criterion that measures the mismatch between target distances (the data) and embedding distances. Distance scaling and the first stress function were first introduced by Kruskal (1964a,b), followed with proposals by Sammon (1969), Takane, Young, and De Leeuw (ALSCAL, 1977), Kamada and Kawai (1989), among others. The problem with a proliferation of proposals is that proposers invariably manage to find situations in which their methods shine, yet no single method is universally superior to all others across all data situations in any meaningful sense, nor does one single stress function necessarily exhaust all possible insights to be gained even from a single data set. For example, embeddings from two stress functions on the same data may both be insightful in that one better reflects local structure, the other global structure.

This situation calls for a rethinking that goes beyond the addition of further proposals. Needed is a methodology that organizes stress functions and provides guidance to their specific performance on any given data set. To satisfy this need we will execute the following program: (1) We embed extant stress functions in a multi-parameter family of stress functions that ultimately extends to incomplete distance data or distance graphs, thereby encompassing "energy functions" for graph drawing; (2) we interpret the effects of some of these parameters on embeddings in terms of a theory that describes how different stress functions entail different compromises in the face of conflicting distance information; (3) we use meta-criteria to measure the quality of embeddings independently of the stress functions, and we use these meta-criteria to select stress functions that are in well-specified senses (near) optimal for a given data set. We have used meta-criteria earlier (Chen and Buja, 2009) in a single-parameter selection problem, and a variation of the approach proves critical in a multi-parameter setting.

For part (1) of the program we took a page from graph drawing which had been in a situation similar to MDS: a collection of discrete proposals for so-called "energy functions", the analogs of stress functions for graph data. This state of affairs changed with the work by Noack (2003) who embedded extant energy functions in single-parameter families of energy functions. Inspired by this work, the first author (Chen, 2006) proposed in her thesis the four-parameter family of distance-based stress functions presented here for the first time. These stress functions are based on Box-Cox transforms and named the "B-C family"; it includes power laws and logarithmic laws for attracting and repulsing energies, a power law for up- or down-weighting of small or large distances, as well as a regularization parameter for incomplete distance data. This family provides an umbrella for several stress functions from the MDS literature as well as energy functions from the graph drawing literature. A related two-parameter family of energy functions for weighted graph data was proposed by Noack (2009), and we study its connection to stress functions for distance data in Section 2.5.

For part (2) of the program, the analysis and interpretation of the stress function parameters, we develop the nucleus of a theory that explains the effects of the some of the parameters on embeddings. Here, too, we looked to Noack (2003, 2007, 2009) for a template of a theory, but it turns out that distance data, considered by us, and weighted graph data, considered by Noack (2009), require different theories. For one thing, distance data, unlike weighted graph data, have a natural concept of "perfect embedding", which is achieved when the target distance data are perfectly matched by the embedding distances. We show that all members in the B-C family of stress functions for complete distance data have the property that they are minimized by perfect embeddings if such exist (Section 2.3) because they satisfy what we call "edgewise unbiasedness". In the general case, when there exists no perfect embedding, a natural question is how the minimization of stress functions creates compromises between conflicting distance information. To answer this question we introduce the notion of "scale sensitivity", which is the degree to which the compromise is dominated by small or large distances through the interaction of two stress function parameters (Section 2.4).

Before we outline step (3) of our program, we make a point that is of interest to machine learning: The B-C family of stress functions encompasses energy functions for graph drawing through an extension from complete to incomplete distance data. First we note that MDS based on complete distance data has been successfully applied to graph drawing through the device of shortest-path length computation for all pairs of nodes in a graph; see, for example, Gansner et al. (2005). Underlying this device is the interpretation of (unweighted) graphs as incomplete distance data whereby edges carry a distance of +1 and non-edges have missing distances. Similarly, the ISOMAP method of nonlinear dimension reduction relies on a complete distance matrix consisting of shortest path lengths computed from a local distance graph. There exists, however, another device for extending MDS to graphs: It is possible to canonically extend all B-C stress functions from complete to incomplete distance data by constructing a limit whereby intuitively non-edges are imputed with an infinite distance that has infinitesimally small weight, creating a pervasive repulsing energy that spreads out embeddings and prevents them from crumpling up. This limiting process offers up a parameter to control the relative strength of the pervasive repulsion vis-à-vis the partial stress for the known distances, thereby acting as a regularization parameter that stabilizes embeddings by reducing variance at the cost of some bias. This device, first applied by the authors (Chen and Buja, 2009) to Kruskal's stress function, brings numerous energy functions for unweighted graphs under the umbrella of the B-C family of stress functions.

Finally, in step (3) of our program, we turn to the problem of selecting "good embeddings" from the multitude that can be obtained from the B-C family of stress functions. This problem can be approached in a principled way with a method that was first used by the authors again in the case of Kruskal's stress function (Chen and Buja, 2009; Chen, 2006; Akkucuk and Carroll, 2006): We employ "meta-criteria" that judge how well embeddings preserve the input topology in a manner that is independent of the stress function used to create the embedding. These meta-criteria measure the degree to which K-nearest neighborhoods are preserved in the mapping of objects to their images in an embedding. K-NN structure is insensitive to nonlinear monotone transformations of the distances in both domains, implying that the meta-criteria allow even quite biased (distorted) configurations to be recognized as performing well in the minimalist sense of preserving K-NNs. Thus the parameters of the B-C family of stress functions can be chosen to optimize a meta-criterion. In this way we turn the ad hoc trial-and-error search for good embeddings into a parameter selection problem.

This article proceeds as follows: Section 2 introduces the B-C family of stress functions in steps. Section 3 introduces the meta-criteria, and Section 4 illustrates the methodology with simulated examples and two sets of real data. Section 5 concludes with a discussion.

2. MDS Stress Functions Based on Power Laws

This section first interprets Kruskal's stress (Kruskal, 1964a) in the framework of attracting and repulsing energies (Section 2.1); it then generalizes these energies with general power laws (Section 2.2), discusses the notions of edgewise unbiasedness (Section 2.3) and scale sensitivity (Section 2.4), as well as the relation between distance- and weight-based approaches (Section 2.5), and generalizes the family to the case of incomplete distance data (Section 2.6). The section concludes with technical aspects concerning the irrelevance of the relative strengths of attracting and repulsing energies (Section 2.7) and the unit invariance of the repulsion parameter for incomplete distance data (Section 2.8).

2.1 Kruskal's Stress as Sum of Attracting and Repulsing Energies

To start we assume a generic MDS situation in which a full set of target distance data $D = (D_{i,j})_{i,j=1,\dots,N}$ is given for all pairs of objects of interest. We assume $D_{i,i} = 0$ and $D_{i,j} > 0$ for $i \neq j$. MDS solves what we may call the "Rand McNalley Road Atlas problem": Given a table showing the distances between all pairs of cities, draw a map of the cities that reproduces the given distances.

Kruskal's original MDS proposal (Kruskal, 1964a) solves the problem by proposing a stress function that is essentially a residual sum of squares (RSS) between the target distances given as data and the distances in the embedding. An embedding (configuration, graph drawing) is a set of points $\mathbf{X} = (\mathbf{x}_i)_{i=1,\dots,N}$, $\mathbf{x}_i \in \mathbb{R}^p$, so that

$$d_{i,j} \;=\; \|\mathbf{x}_i - \mathbf{x}_j\|$$

are the embedding distances (we limit ourselves to Euclidean distances). The goal is to find an embedding $\mathbf{X}$ whose distances $d_{i,j}$ fit the target distances $D_{i,j}$ as best as possible. Kruskal's stress function is therefore

$$S(d|D) \;=\; \sum_{i,j}(d_{i,j} - D_{i,j})^2,$$

where we let $d = (d_{i,j})_{i,j} = (\|\mathbf{x}_i - \mathbf{x}_j\|)_{i,j}$ and $D = (D_{i,j})_{i,j}$. Optimization is carried out over all $N \times p$ coordinates of the configuration $\mathbf{X}$.

Taking a page from the graph drawing literature, we interpret Kruskal's stress function as composed of an "attracting energy" and a "repulsing energy" as follows:

$$S(d|D) \;=\; \sum_{i,j}({d_{i,j}}^2 - 2D_{i,j}\,d_{i,j}) \;+\; \text{const.}$$

The term ${d_{i,j}}^2$ represents an "attracting energy" because in isolation it is minimized by $d_{i,j} = 0$. The term $-2D_{i,j}d_{i,j}$ represents a "repulsing energy" because again in isolation it is minimized by $d_{i,j} = \infty$. (The term ${D_{i,j}}^2$ is a constant that does not affect the minimization; it calibrates the minimum energy level at zero.) A stress term $(d_{i,j} - D_{i,j})^2$ is therefore seen to be equivalent to a sum of an attracting and a repulsing energy term that balance each other in such a way that the minimum energy is achieved at $d_{i,j} = D_{i,j}$.

2.2 The B-C Family of Stress Functions

We next introduce a family of stress functions whose attracting and repulsing energies follow power laws, in analogy to Noack's generalized energy functions for graph drawing (Noack, 2003, 2007, 2009). However, we would like this family to also include logarithmic laws, as in Noack's "LinLog" energy (Noack, 2003, 2007). To accommodate logarithms in the family of power transformations, statisticians have long used the so-called Box-Cox family of transformations, defined for $d > 0$ by

$$BC_\alpha(d) \;=\; \begin{cases} \frac{d^\alpha - 1}{\alpha} & (\alpha \neq 0), \\ \log(d) & (\alpha = 0). \end{cases}$$

This modification of the raw power transformations d^α not only affords analytical fill-in with the natural logarithm for $\alpha = 0$, it also extends the family to $\alpha < 0$ while preserving increasing monotonicity of the transformations: for $\alpha < 0$ raw powers d^α are decreasing while $BC_\alpha(d)$ is increasing. The derivative is

$$BC_\alpha{}'(d) \;=\; d^{\alpha-1} > 0 \quad \forall d > 0,\ \forall \alpha \in \mathbb{R}\,.$$

By subtracting the (otherwise irrelevant) constant 1 in the numerator and dividing by α, Box-Cox transformations are affinely matched to the natural logarithm at $d = 1$ for all powers α:

$$BC_\alpha(1) = 0, \quad BC_\alpha{}'(1) = 1.$$

See Figure 1 for an illustration of Box-Cox transformations.

Using Box-Cox transformations we construct a generalization of Kruskal's stress function by allowing arbitrary power laws for the attracting and the repulsing energies, subject to the constraint that the attracting power is greater than the repulsing power to guarantee that the minimum combined energy is finite ($> -\infty$). We denote the attracting power by $\mu + \lambda$ and the repulsing power by μ with the understanding that $\lambda > 0$ and $-\infty < \mu < +\infty$.

Definition 1 *The **B-C family of stress functions** for complete distance data $D = (Dij)_{i,j}$ is given by*

$$S(d|D) \;=\; \sum_{i,j=1,\ldots,N} {D_{i,j}}^\nu \left(BC_{\mu+\lambda}(d_{i,j}) \;-\; {D_{i,j}}^\lambda\, BC_\mu(d_{i,j}) \right). \tag{1}$$

As we assume $D_{i,j} > 0$ for $i \neq j$ the weight term ${D_{i,j}}^\nu$ is meaningful for all powers $-\infty < \nu < +\infty$. Thus ${D_{i,j}}^\nu$ upweights the summands for large $D_{i,j}$ when $\nu > 0$ and downweights them when $\nu < 0$; for $\nu = 0$ the stress function is an unweighted sum. The parameter ν allows us to capture a couple of extant stress functions; see Table 1. Kruskal's stress function does not require ν as it arises from $\mu = 1$, $\lambda = 1$ and $\nu = 0$. The idea of using general power laws in an attraction-repulsion paradigm arose independently in the first author's PhD thesis (Chen, 2006) and in Noack (2009). For a discussion of the relationship between the two proposals see Section 2.5.

2.3 Edgewise Unbiasedness of Stress Functions

The reason for introducing the multiplier ${D_{i,j}}^\lambda$ in the repulsing energy is to grant what we call **edgewise unbiasedness**: If there exist only two objects, $N = 2$, with target distance D, then the stress function $S(d) = D^\nu \left(BC_{\mu+\lambda}(d) - D^\lambda BC_\mu(d) \right)$ should be minimized by $d = D$:

$$D \;=\; \operatorname{argmin}_d\, D^\nu \left(BC_{\mu+\lambda}(d) - D^\lambda BC_\mu(d) \right).$$

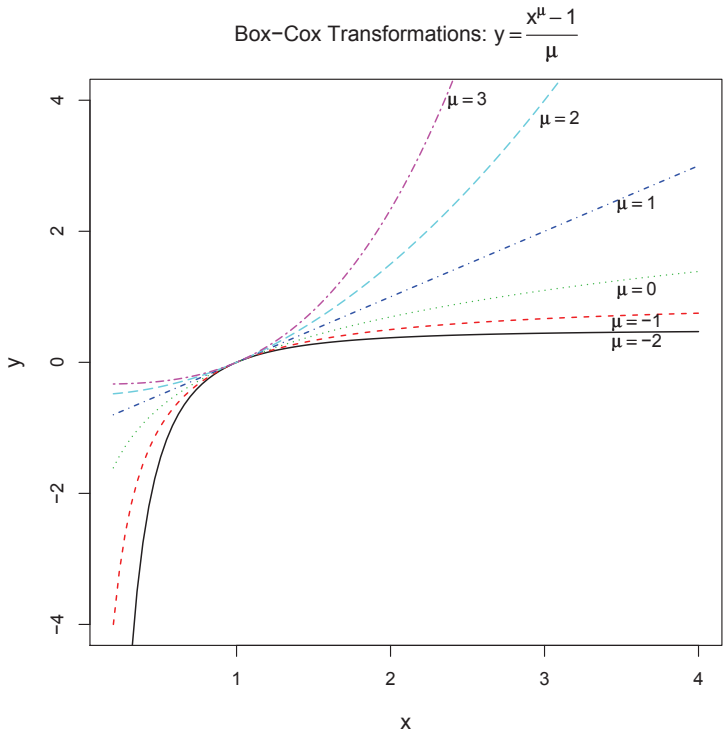


Figure 1: Box-Cox Transformations: $y = \frac{x^\mu - 1}{\mu}$

This property is easily verified using $\lambda > 0$: $S'(d) = D^{\nu+\mu-1}\left(d^\lambda - D^\lambda\right)$, hence $S'(d) < 0$ for $d \in (0, D)$ and $S'(d) > 0$ for $d \in (D, \infty)$, so that $S(d)$ is strictly descending on $(0, D)$ and strictly ascending on (D, ∞). This property holds only for this particular choice of the power D^λ in the repulsing energy term.

Edgewise unbiasedness is essential to grant the following exact reconstruction property:

Proposition 2 *If the target data $D_{i,j}$ form a set of Euclidean distances in the embedding dimension, $D_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ $(i, j = 1, \ldots, N)$, then all B-C stress functions are minimized by the embeddings that reproduce the target distances exactly: $d_{i,j} = D_{i,j}$.*

Note that embeddings are unique only up to rotations, translations and reflections. They may have additional non-uniqueness properties that may be peculiar to the data.

2.4 Scale Sensitivity

Next we analyze the role of the parameters ν and λ. As we will see, they determine the degree to which conflicting metric information is decided in favor of small or large target distances. It is a major goal of MDS procedures to reach good compromises to obtain informative embeddings in the general situation when distance data are not perfectly embeddable in a Euclidean space of a given dimension, be it due to error in the target distances, or due to the distance interpretation

of what is really just dissimilarity data, or due to intrinsic higher dimensionality of the underlying objects. To gain insight into the nature of the compromises, it is beneficial to construct a simple paradigmatic situation in which contention between conflicting distance data can be analyzed. One such situation is as follows: Assume again that there are only two objects ($N = 2$), but that target distances were obtained twice for this same pair of objects, resulting in different values D_1 and D_2 (due to observation error, say). In practice, one often reduces multiple distances by averaging them, but a more principled approach is to form a stress function with multiple stress terms per object pair (i, j). In general, if target distances $D_{i,j,k}$ for the object pair (i, j) are observed $K_{i,j}$ times, the B-C stress function will be

$$S \;=\; \sum_{i,j=1,\ldots,N} \sum_{k=1,\ldots,K_{i,j}} {D_{i,j,k}}^{\nu} \left(BC_{\mu+\lambda}(d_{i,j}) - {D_{i,j,k}}^{\lambda} BC_{\mu}(d_{i,j}) \right).$$

With this background, the paradigmatic situation of two target distances D_1 and D_2 observed on one object pair is the simplest case that exhibits contention between conflicting distance information. The stress function for the single embedding distance d is

$$S \;=\; {D_1}^{\nu} \left(BC_{\mu+\lambda}(d) - {D_1}^{\lambda} BC_{\mu}(d) \right) \;+\; {D_2}^{\nu} \left(BC_{\mu+\lambda}(d) - {D_2}^{\lambda} BC_{\mu}(d) \right).$$

It is minimized by

$$d_{\min} \;=\; \left(\alpha_1 {D_1}^{\lambda} + \alpha_2 {D_2}^{\lambda} \right)^{1/\lambda}, \quad \text{where } \alpha_1 = \frac{{D_1}^{\nu}}{{D_1}^{\nu} + {D_2}^{\nu}},\ \alpha_2 = \frac{{D_2}^{\nu}}{{D_1}^{\nu} + {D_2}^{\nu}}, \tag{2}$$

so that $\alpha_1 + \alpha_2 = 1$. Thus $d_{\min}$ is the Lebesgue L_λ norm of the 2-vector (D_1, D_2) with regard to the Bernoulli distribution with probabilities α_1 and α_2 (an improper norm for $0 < \lambda < 1$). However, α_1 and α_2 are also functions of (D_1, D_2), hence the minimizing distance $d_{\min} = d(D_1, D_2)$ is a function of the target distances in a complex way. Yet, the Lebesgue norm interpretation is useful because it allows us to analyze the dependence of d on the parameters λ and ν separately:

- For fixed $D_1 \neq D_2$, the minimizing distance d is a monotone increasing function of ν for $-\infty < \nu < \infty$, and we have

$$d_{\min} \;=\; \left(\alpha_1 {D_1}^{\lambda} + \alpha_2 {D_2}^{\lambda} \right)^{1/\lambda} \quad \begin{cases} \uparrow\ \max(D_1, D_2) & \text{as} \quad \nu \uparrow \infty, \\ \downarrow\ \min(D_1, D_2) & \text{as} \quad \nu \downarrow -\infty. \end{cases}$$

  The reason is that if $D_1 > D_2$ we have $\alpha_1 \uparrow 1$ as $\nu \uparrow \infty$, and $\alpha_2 \uparrow 1$ as $\nu \downarrow -\infty$.

- For fixed $D_1 \neq D_2$, the minimizing distance d is a monotone increasing function of λ for $0 < \lambda < \infty$, and we have

$$d_{\min} \;=\; \left(\alpha_1 {D_1}^{\lambda} + \alpha_2 {D_2}^{\lambda} \right)^{1/\lambda} \quad \begin{cases} \uparrow\ \max(D_1, D_2) & \text{as} \quad \lambda \uparrow \infty, \\ \downarrow\ {D_1}^{\alpha_1} {D_2}^{\alpha_2} & \text{as} \quad \lambda \downarrow 0. \end{cases}$$

(These facts generalize in the obvious manner to K distances $D_1, D_2, \ldots, D_K$ observed on the pair of objects.) While large distances win out in the limit for $\lambda \uparrow +\infty$, fixed small distances > 0 will never win out entirely for $\lambda \downarrow 0$, although for ever smaller λ the compromise will be shifted ever more toward the smaller distance.

Conclusion: *Embeddings that minimize B-C stress compromise ever more in favor of ...*
... larger distances as $\lambda \uparrow \infty$ *or* $\nu \uparrow \infty$*, with full max-dominance in either limit;*
... smaller distances as $\lambda \downarrow 0$ *or* $\nu \downarrow -\infty$*, with full min-dominance only in the* ν*-limit.*

We use the term "**small scale sensitivity**" for the behavior of stress functions as $\lambda \downarrow 0$ and/or $\nu \downarrow -\infty$. It has the effect of reinforcing local structure because object pairs with small target distances will preferentially be placed close together in the embedding. A related observation was made by Noack (2003) for $\lambda \downarrow 0$ in graph drawing and called "clustering strength"; this concept is not identical to small distance sensitivity, however; see Section 2.5.

2.5 Distances versus Weights

Noack (2009) presents a family of "energy functions" for weighted graphs/networks that should be discussed here because it might be thought to be identical to the B-C family of stress functions—which it is not, though there exists a connection. The following discussion is meant to clarify the difference between specifying the relation among object pairs in terms of weights and in terms of distances.

Underlying the idea of mapping weighted graph data to graph drawings is a density paradigm. The intuition is that objects connected by edges with large weights should be represented by embedding points that are near each other so as to form high density areas. Hence large weights play a similar role as small distances in their intended effects on embeddings. Weights and distances are therefore in an inverse relation to each other, a fact that will be made precise below.

Next we follow Noack (2009) and consider data given as edge weights $w_{i,j} \geq 0$ for all pairs (i, j) with the interpretation that an edge in a graph "exists" between objects i and j if $w_{i,j} > 0$. (He also allows node weights w_i, but we set these to 1 as they add no essential freedom of functional form.) The family of energy functions he considers uses a general form of power laws for attracting and repulsing energies:

$$U(d|W) \;=\; \sum_{i,j=1,\ldots,N} \left(w_{i,j} \frac{{d_{i,j}}^{a+1}}{a+1} \;-\; \frac{{d_{i,j}}^{r+1}}{r+1} \right), \tag{3}$$

where we write $W = (w_{i,j})_{i,j=1,\ldots,N}$. It is assumed that $a > r$ in order to grant finitely sized minimizing embeddings for connected graphs. In the spirit, though not the letter, of Box-Cox transforms, Noack imputes natural logarithms for $a+1 = 0$ or $r+1 = 0$. Unweighted graphs are characterized by $w_{i,j} \in \{0, 1\}$, in which case the total energy (3) amounts to (1) the sum of attracting energies limited to the edges in the graph, and (2) the sum of repulsing energies for *all* pairs of nodes. This functional form is suggested by traditional energy functions in graph drawing where an attracting force holds the embedding points $\mathbf{x}_i$ and $\mathbf{x}_j$ together if there exists an edge between them and where the repulsing force is pervasive and exists for all pairs so as to disentangle the embedding points by spreading them out.

We now ask how the energy functions (3) and the B-C stress functions (1) relate to each other. A simple answer can be given by drawing on the notion of edgewise unbiasedness: in a two-node situation with single weight w, find the embedding distance $d_{\min}$ that minimizes the energy function (3); this distance $d_{\min} = d(w)$ can be interpreted as the target distance D for which the energy function is edgewise unbiased. Thus the canonical relation between weights and target distances is $D = d(w)$. For an energy function (3) the specialization to two nodes is $U = w d^{a+1}/(a+1) - d^{r+1}/(r+1)$, whose stationarity condition is $U' = w d^a - d^r = 0$, hence $w = 1/d^{a-r}$ and $d(w) = 1/w^{1/(a-r)}$, as noted by Noack (2009, Equation (3)). Thus the correspondence between w and its

edgewise unbiased target distance D is

$$D \;=\; \frac{1}{w^{1/(a-r)}}. \tag{4}$$

Using the translation $w_{i,j} = {D_{i,j}}^{-(a-r)}$ and the convention $w_{i,j} = 0 \Rightarrow D_{i,j} = +\infty \Rightarrow {D_{i,j}}^{-(a-r)} = 0$, we can rewrite the energy function (3) modulo irrelevant constants as

$$U(d|D) \;\sim\; \sum_{i,j=1,\ldots,N} \Big({D_{i,j}}^{-(a-r)}\, BC_{a+1}(d_{i,j}) \;-\; BC_{r+1}(d_{i,j}) \Big). \tag{5}$$

A comparison with (1) shows that the 2-parameter family of energy functions (5) forms a subfamily of the 3-parameter family of distance-based B-C stress functions (1) as follows:

$$\nu = -(a-r), \quad \mu = r+1, \quad \lambda = a-r.$$

Thus the essential constraint is that $\lambda = -\nu$, entailing $\nu < 0$. In light of the results of Section 2.4 this constraint implies a counterbalancing of distance sensitivities implied by these parameters: as $\lambda \uparrow \infty$ large distance sensitivity increases, but simultaneously $\nu = -\lambda \downarrow -\infty$ and hence small scale sensitivity increases as well. Full clarity of the interplay is gained by repeating the exercise of Section 2.4 in the case $\nu = -\lambda$: Given two target distances D_1 and D_2 for $N = 2$ objects, the minimizing distance is obtained by specializing (2) to $\nu = -\lambda$:

$$d_{\min} \;=\; \frac{1}{\left(\frac{1}{2}{D_1}^{-\lambda} + \frac{1}{2}{D_2}^{-\lambda}\right)^{1/\lambda}} \quad \begin{cases} \downarrow \; \min(D_1, D_2) & \text{as} \quad \lambda \uparrow \infty, \\ \uparrow \; \sqrt{D_1 D_2} & \text{as} \quad \lambda \downarrow 0. \end{cases}$$

Thus the minimizing distance $d_{\min}$ is the reciprocal of the Lebesgue L_λ norm of the vector $({D_1}^{-1}, {D_2}^{-1})$ with regard to a uniform distribution $\alpha_1 = \alpha_2 = 1/2$. The identification $\nu = -\lambda$ has therefore a considerable degree of small scale sensitivity for all values of $\lambda > 0$, and counterintuitively it increases with increasing λ: apparently the increasing small scale sensitivity incurred from the parameter $\nu \downarrow -\infty$ outweighs the diminished small scale sensitivity due to $\lambda \uparrow +\infty$.

It follows that Noack's notion of "clustering strength" (Noack, 2003) is not identical to our notion of small scale sensitivity because clustering strength increases for $\lambda = -\nu \downarrow 0$. Rather, clustering strength has to do with the implied translation of a fixed weight w to a target distance $D = 1/w^{1/\lambda}$ according to (4): relatively large weights w will result in relatively ever smaller target distances D as $\lambda \downarrow 0$, thus reinforcing the clustering effect by the simple translation $w \mapsto D$. Diminishing small scale sensitivity for $\lambda = -\nu \downarrow 0$ is a lesser effect by comparison.

2.6 B-C Stress Functions for Incomplete Distance Data or Distance Graphs

In order to arrive at stress functions for non-full graphs, we extend a device we used previously to transform Kruskal-Shepard MDS into a localized or graph version called "local MDS" or "LMDS" (Chen and Buja, 2009). We now assume target distances $D_{i,j}$ are given only for edges $(i,j) \in E$ in a graph. Starting with stress functions (1) for full graphs, we replace the dissimilarities $D_{i,j}$ for non-edges $(i,j) \notin E$ with a single large dissimilarity D_∞ which we let go to infinity. We down-weight these terms with a weight w in such a way that $w {D_\infty}^{\lambda+\nu} = t^{\lambda+\nu}$ is constant:

$$\begin{aligned} S \;&=\; \sum_{(i,j)\in E} {D_{i,j}}^{\nu} \Big(BC_{\mu+\lambda}(d_{i,j}) \;-\; {D_{i,j}}^{\lambda}\, BC_{\mu}(d_{i,j}) \Big) \\ &+\; w \sum_{(i,j)\notin E} {D_\infty}^{\nu} \Big(BC_{\mu+\lambda}(d_{i,j}) \;-\; {D_\infty}^{\lambda}\, BC_{\mu}(d_{i,j}) \Big). \end{aligned}$$

As $D_\infty \to \infty$, we have $w = (t/D_\infty)^{\nu+\lambda} \to 0$ and $wD_\infty{}^{\nu} \to 0$, hence in the limit we obtain:

$$S \;=\; \sum_{(i,j)\in E} D_{i,j}{}^{\nu} \Big(BC_{\mu+\lambda}(d_{i,j}) \;-\; D_{i,j}{}^{\lambda}\, BC_{\mu}(d_{i,j}) \Big) \;-\; t^{\nu+\lambda} \sum_{(i,j)\notin E} BC_{\mu}(d_{i,j})\,. \tag{6}$$

This procedure justifies wiping out the attracting energy outside the graph. We call (6) the *B-C family of stress functions* for distance graphs. The parameter t balances the relative strength of the combined attraction and repulsion inside the graph with the repulsion outside the graph. For completeness, we list the assumed ranges of the parameters:

$$t \geq 0\,, \quad \lambda > 0\,, \quad -\infty < \mu < \infty\,, \quad -\infty < \nu < \infty.$$

An interesting variation of the idea of pervasive repulsion is proposed by Koren and Çivril (2009) who use finite rather than limiting energies.

Choice of Parameters	Special Cases
$E = V^2$, $\lambda = 1$, $\mu = 1$, $\nu = 0$	MDS (Kruskal, 1964a; Kruskal and Seery, 1980)
$E = V^2$, $\lambda = 2$, $\mu = 2$, $\nu = 0$	ALSCAL (Takane, Young, and De Leeuw, 1977)
$E = V^2$, $\lambda = 1$, $\mu = 1$, $\nu = -2$	Kamada and Kawai (1989)
$E = V^2$, $\lambda = 1$, $\mu = 1$, $\nu = -1$	Sammon (1969)
$E \subset V^2$, $\lambda = 1$, $\mu = 1$, $\nu = 0$, $t > 0$	LMDS (Chen and Buja, 2009)
$E \subset V^2$, $\lambda = 3$, $\mu = 0$, $D_{i,j} = 1$, $t = 1$	Fruchterman and Reingold (1991)
$E \subset V^2$, $\lambda = 4$, $\mu = -2$, $D_{i,j} = 1$, $t = 1$	Davidson and Harel (1996)
$E \subset V^2$, $\lambda = 1$, $\mu = 0$, $D_{i,j} = 1$, $t = 1$	LinLog (Noack, 2003)
$E \subset V^2$, $\lambda = 1$, $\mu = 1$, $D_{i,j} = 1$, $t = 1$	QuadLin (Noack, 2003)
$E \subset V^2$, $\lambda > 0$, $\mu = 0$, $D_{i,j} = 1$, $t = 1$	PolyLog family (Noack, 2003, his $r = \lambda$)

Table 1: Some special cases of stress functions and their parameters in the B-C family. The first four entries refer to stress functions for complete distance data; the last five entries refer to energy functions for plain graphs (in which case $D_{i,j} = 1$ for all edges and hence ν is vacuous). LMDS applies to incomplete distance data or distance graphs, as do all members of the B-C family. (Not included is the family of power laws for weighted graphs by Noack (2009) because they become stress functions for distance graphs only after a mapping of weights to distances.)

2.7 An Irrelevant Constant: Weighting the Attraction

Noack (2003, Section 5.5) observed that for his LinLog energy function the relative weighting of the attracting energy relative to the repulsing energy is irrelevant in the sense that such weighting would only change the scale of the minimizing layout but not the shape. A similar statement can be made for all members of the B-C family of stress functions. To demonstrate this effect, we introduce B-C stress functions whose attraction is weighted by a factor c^λ ($c > 0$):

$$S_c(d) \;=\; \sum_{(i,j)\in E} D_{i,j}{}^{\nu} \Big(c^{\lambda}\, BC_{\mu+\lambda}(d_{i,j}) \;-\; D_{i,j}{}^{\lambda}\, BC_{\mu}(d_{i,j}) \Big) \;-\; t^{\nu+\lambda} \sum_{(i,j)\notin E} BC_{\mu}(d_{i,j})\,,$$

where $d = (d_{i,j})$ is the set of all configuration distances for all pairs (i, j), including those not in the graph E. The repulsion terms are still differentially weighted depending on whether (i, j) is an edge of the graph E or not, which is in contrast to most energy functions proposed in the graph layout literature where invariably $t = 1$.

In analogy to Noack's argument, we observe the following form of scale equivariance:

$$S_1(c\,d) \;=\; c^{\mu}\, S_c(d) \;+\; \text{const.}$$

As a consequence, if d is a minimizing set of configuration distances for $S_c(\cdot)$, then the distances $c\,d$ of the scaled embedding $c\,\mathbf{X}$ minimize the original unweighted B-C stress function $S_1(\cdot)$.

It is in this sense that Noack's PolyLog family of stress functions can be considered as a special case of the B-C family: PolyLog energies agree with B-C stress functions for unweighted graphs ($D_{i,j} = 1$) for $\mu = 0$ and $t = 1$ up to a multiplicative factor in the attracting energy.

2.8 Unit-Invariant Forms of the Repulsion Weight

In the B-C family of stress functions (6), the relative strength of attracting and repulsing forces is balanced by the parameter t. This parameter, however, has two deficiencies: (1) It suffers from a lack of invariance under a change of units in the target distances $D_{i,j}$; (2) it has stronger effects in sparse graphs than dense graphs because the number of terms in the summations over E and $V \setminus E$ vary with the size of the graph E. Both deficiencies can be corrected by reparametrizing t in terms of a new parameter τ as follows:

$$t^{\lambda+\nu} \;=\; \frac{|E|}{|V^2| - |E|} \cdot \big(\text{median}_{(i,j)\in E}\, D_{i,j}\big)^{\lambda+\nu} \cdot \tau^{\lambda+\nu}\,.$$

This new parameter τ is unit free and adjusted for graph size. (Obviously the median can be replaced with any other statistic $S(D)$ that is positively homogeneous of first order: $S(cD) = cS(D)$ for $c > 0$.) These features enable us to formulate past experience in a problem-independent fashion as follows: in the examples we have tried, $\tau = 1$ has yielded satisfactory results. In light of this experience, there may arise few occasions in practice where there is a need to tune τ. As users work with different units in $D_{i,j}$ or different neighborhood sizes when defining NN-graphs, the recommendation $\tau = 1$ stands. Just the same, we will illustrate the effect of varying τ in an artificial example (Section 4.1).

3. Meta-Criteria for Parameter Selection

Following Chen and Buja (2009) and Akkucuk and Carroll (2006), we describe "meta-criteria" to measure the quality of configurations independently of the primary stress functions. The main purpose of these meta-criteria is to guide the selection of parameters such as those in the B-C family, λ, μ and τ. The idea is to compare "input neighborhoods" defined in terms of $D_{i,j}$ with "output neighborhoods" defined in terms of $d_{i,j}$ by measuring the size of their overlaps. Such neighborhoods are typically constructed as K-NN sets or, less frequently, in metric terms as ε-neighborhoods. In a dimension reduction setting one may define for the i'th point the input neighborhood $\mathcal{N}_D(i)$ as the set of K-NNs with regard to $D_{i,j}$ and similarly the output neighborhood $\mathcal{N}_d(i)$ as the set of K-NNs with regard to $d_{i,j}$. In an unweighted graph setting, one may define $\mathcal{N}_D(i)$ as the metric $\varepsilon = 1$ neighborhood, that is, the set of points connected with the i'th point in the graph E, and hence the neighborhood size $K(i) = |\mathcal{N}_D(i)|$ is the degree of the i'th point in the graph E and will vary from

point to point. The corresponding output neighborhood $\mathcal{N}_d(i)$ can then be defined as the $K(i)$-NN set with regard to $d_{i,j}$. The pointwise meta-criterion at the i'th point is defined as size of the overlap between $\mathcal{N}_d(i)$ and $\mathcal{N}_D(i)$, hence it is in frequency form

$$N_d(i) \;=\; |\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|\,,$$

and in proportion form, using $|\mathcal{N}_D(i)|$ as the baseline,

$$M_d(i) \;=\; \frac{|\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|}{|\mathcal{N}_D(i)|}\,.$$

The global meta-criteria are simply the averages over all points:

$$N_d \;=\; \frac{1}{|V|}\sum_i N_d(i) \qquad \text{and} \qquad M_d = \frac{1}{|V|}\sum_i M_d(i)\,.$$

Only when all input neighborhood sizes are equal, $|\mathcal{N}_D(i)| = K$, is there a simple relationship between N_d and M_d: $M_d = \frac{1}{K} N_d$. We subscript these quantities with d because they serve to compare different outputs $(\mathbf{x}_i)_{i=1\ldots N}$ (configurations, embeddings, graph drawings), but all that is used are the interpoint distances $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$. The proportion form M_d is obviously advantageous because it allows comparisons across different K (or ε).

Whether the meta-criterion values are small or large should be judged not against their possible ranges ($[0,1]$ for M_d) but against the possibility that $d_{i,j}$ (hence the embedding) and $D_{i,j}$ are entirely unrelated and generate only random overlap in their respective neighborhoods $\mathcal{N}_d(i)$ and $\mathcal{N}_D(i)$. The expected value of random overlap is not zero, however; rather, it is $\mathrm{E}[|\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|] = |\mathcal{N}_d(i)| \cdot |\mathcal{N}_D(i)|/(|V|-1)$ because random overlap should be modeled by a hypergeometric distribution with $|\mathcal{N}_D(i)|$ "defectives" and $|\mathcal{N}_d(i)|$ "draws" from a total of $|V|-1$ "items." The final adjusted forms of the meta-criteria are therefore:

$$\begin{aligned}
N_d^{adj}(i) &= |\mathcal{N}_d(i) \cap \mathcal{N}_D(i)| \;-\; \frac{1}{|V|-1}\,|\mathcal{N}_d(i)| \cdot |\mathcal{N}_D(i)|\,, \\
M_d^{adj}(i) &= \frac{|\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|}{|\mathcal{N}_D(i)|} \;-\; \frac{1}{|V|-1}\,|\mathcal{N}_d(i)|\,, \\
N_d^{adj} &= \frac{1}{|V|}\sum_i N_d^{adj}(d)\,, \qquad M_d^{adj} \;=\; \frac{1}{|V|}\sum_i M_d^{adj}(d)\,.
\end{aligned}$$

When the neighborhoods are all K-NN sets, $|\mathcal{N}_d(i)| = |\mathcal{N}_D(i)| = K$, these expressions simplify:

$$\begin{aligned}
N_d^{adj}(i) &= |\mathcal{N}_d(i) \cap \mathcal{N}_D(i)| \;-\; \frac{K^2}{|V|-1}\,, \\
M_d^{adj}(i) &= \frac{|\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|}{K} \;-\; \frac{K}{|V|-1} \;=\; \frac{N_d^{adj}(i)}{K}\,, \\
N_d^{adj} &= N_d \;-\; \frac{K^2}{|V|-1}, \qquad M_d^{adj} \;=\; M_d \;-\; \frac{K}{|V|-1} \;=\; \frac{N_d^{adj}}{K}\,.
\end{aligned}$$

An important general observation is that if the neighborhoods are defined as K-NN sets, the meta-criteria are invariant under monotone transformations of both inputs $D_{i,j}$ and outputs $d_{i,j}$.

Methods that have this invariance are called "non-metric" in proximity analysis/multidimensional scaling because they depend only on the ranks and not the actual values of the distances.

In what follows, we will report M_d^{adj} for each configuration shown in the figures, and we will also use the pointwise values $M_d(i)$ as a diagnostic by highlighting points with $M_d(i) < 1/2$ as problematic in some of the figures.

Remark 3 *Venna and Kaski (2006, and references therein) introduce an interesting distinction between "trustworthiness" and "continuity" measurement. In our notation the points in $\mathcal{N}_d(i) \setminus \mathcal{N}_D(i)$ violate trustworthiness because they are shown near but are not near in truth (near = being in the $K(i)$-NN), whereas the points in $\mathcal{N}_D(i) \setminus \mathcal{N}_d(i)$ violate continuity because they are near in truth but not shown as near. Venna and Kaski (2006) measure both violations separately based on distance-ranks. We implicitly also measure both, but more crudely by unweighted counting of violations. It turns out, however, that the two violation counts are the same: $|\mathcal{N}_d(i) \setminus \mathcal{N}_D(i)| = |\mathcal{N}_D(i) \setminus \mathcal{N}_d(i)| = K(i) - |\mathcal{N}_d(i) \cap \mathcal{N}_D(i)|$. Thus our meta-criterion is simultaneously a measure of trustworthiness and of continuity. Lee and Verleysen (2008) introduce a larger class of potentially interesting meta-criteria that include ours and Venna and Kaski (2006) as special cases.*

4. B-C Stress Functions Applied

In this section, we illustrate the methodology with simulated data (Section 4.1), the Olivetti face data (Section 4.2) and the Frey face data (Section 4.3).

4.1 Simulated Data

We introduced three parameters in the B-C stress functions for complete distance data, namely, λ, μ and ν, and a fourth parameter, τ, in the B-C stress functions for incomplete distance graph data. In this subsection, we will examine how three of the four parameters affect configurations in terms of their local and global structure by experimenting with an artificial example. We will simplify the task and eliminate the parameter ν by setting it to zero, so that the weight $D_{i,j}{}^{\nu} = 1$ disappears from the stress functions. The reason for doing so is that both parameters ν and λ play a role in determining scale sensitivity, and, while they are not redundant, the weighting power ν is the more dangerous of the two because it can single-handedly destabilize stress functions as $\nu \downarrow -\infty$ through unlimited outweighting of large distances. By comparison, the small scale sensitivity caused by small values of the parameter $\lambda > 0$ is limited as the analysis of Section 2.4 shows.

To illustrate the effects of the remaining parameters λ, μ and τ on embeddings, we constructed an artificial data example consisting of 83 points that form a geometric shape represented in Figure 2 (top). The design was inspired by a simulation example used by Trosset (2006). The distance between any pair of adjacent points is set to 1. To define an initial local graph, as input to the stress functions, we connected each point with its adjacent neighbors with distance 1 (Figure 2, bottom). That is, we used metric nearest neighborhoods with radius 1. Thus, interior points have node degree 4, corner points and connecting points have node degree 2, and the remaining peripheral points have node degree 3. The geometry of this input graph is intended to represent three connected clusters with an internal structure that is relatively tight compared to the connecting structure.

We produced 2-D configurations using B-C stress functions, and to explore the effect of the parameters on the configurations, we varied each of the three parameters one at a time. For each combination of parameters, we used two different starting configurations: the inputs from Figure

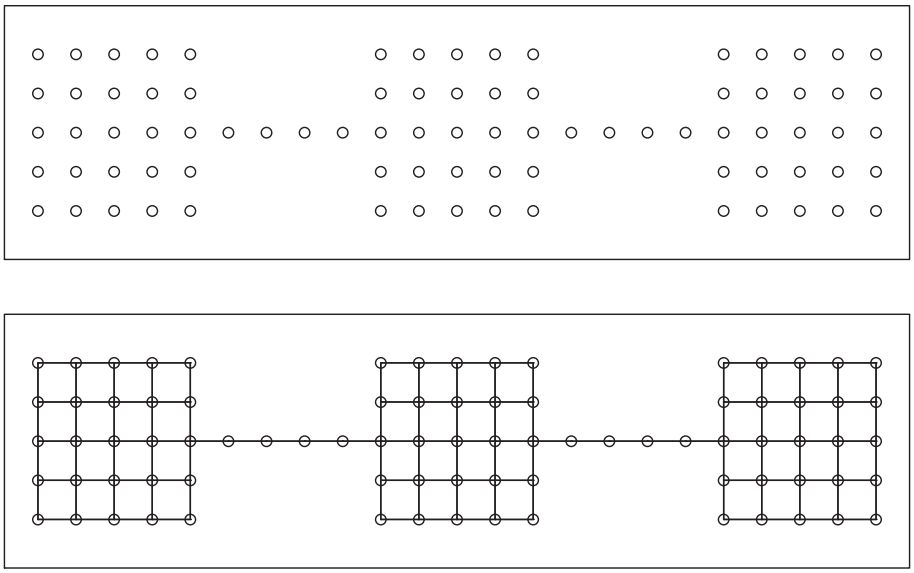


Figure 2: The original configuration (top) and initial local graph (bottom)

2, and a random start, respectively. Starting from the true input configurations is of course not actionable in practice, but it serves a purpose: it demonstrates the biases and distortions implied by minimization of the stress function under the best of circumstances. Starting from a random configuration, on the other hand, gives indications about the stability of the solutions in terms of local minima, as well as the effort required to get from an uninformed starting configuration to a meaningful local minimum. (In practice, one never knows how truly optimal any configuration is that has been obtained by a numerical algorithm, and a better sense of the issue is often obtained only by analyzing the solutions obtained from multiple restarts.)

For starts from the input configurations, the results are shown in Figures 3, 5, and 7, and for starts from random configurations they are shown in Figures 4, 6, and 8, along with their M^{adj} values that measure the local faithfulness of the configuration. We also colored red (symbolled triangles) the points whose neighborhood structure is not well preserved in terms of a proportion $< 1/2$ of shared neighbors between input and output configurations ($M(i) < 1/2$).

Parameter λ*:* We set $\mu = 0$ and $\tau = 1$ and let λ vary as follows: $\lambda = 5,\ 2,\ 1,\ 0.5$. The resulting configurations are shown in Figures 3 and 4. The overall observation from both figures is that for smaller λ the greater small-scale sensitivity causes the configurations to cluster more strongly. We also notice in both figures that M^{adj} increases with decreasing λ, which indicates local structure within clusters is better recovered when the clusters are well separated. To confirm this, we show a zoom on the nearly collapsed points in the bottom right configurations and observe the square structure in the input configuration is almost perfectly recovered. This indicates that by tuning λ properly the resulting configurations can reveal both macro structure in terms of relative cluster placement as well as micro structure within each cluster.

Parameter μ*:* In Figures 5 and 6, we examine the effect of μ. We fix $\lambda = 5$ and $\tau = 1$ and let μ vary as follows: $\mu = -1,\ 0,\ 1,\ 2$. An overall observation is that the larger μ, the more spread out

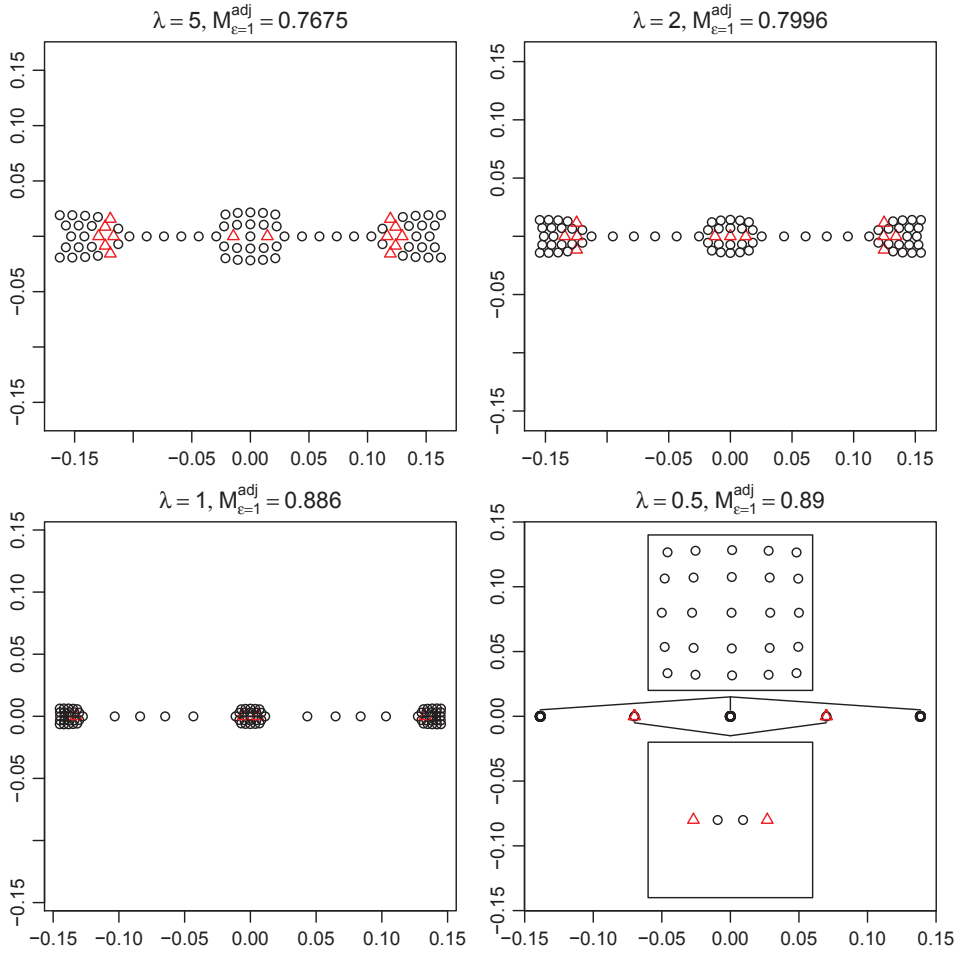


Figure 3: The configurations with varying λ with other parameters fixed at $\mu = 0,\ \tau = 1$, starting from the input configuration (Figure 2).

are the points. This is consistent with the interpretation of μ as the power law of repulsion. With smaller μ (such as $\mu = -1$) the stress function flattens out as the distance increases (Figure 1) and the points are subject to very weak repulsion. The top left plots ($\mu = -1$) in both figures show that weak repulsion is suitable for generating locally faithful structure. However, the repulsion can be too weak to generate globally meaningful structure, as illustrated by the configurations obtained from random starts (Figure 6). In the top left plot of Figure 6, the three clusters are not aligned properly due to the weak repulsion. With stronger repulsion the points are placed in a globally more correct position, as shown in bottom two plots in Figure 6, with a sacrifice of local faithfulness (as reflected by the lower value of M^{adj}). The distortion of local structure is not surprising considering the fact that the repulsion is stronger in the direction in which points line up, which in this case is the horizontal direction. By comparison, points are squeezed flat in the vertical direction because repulsion has no traction vertically.

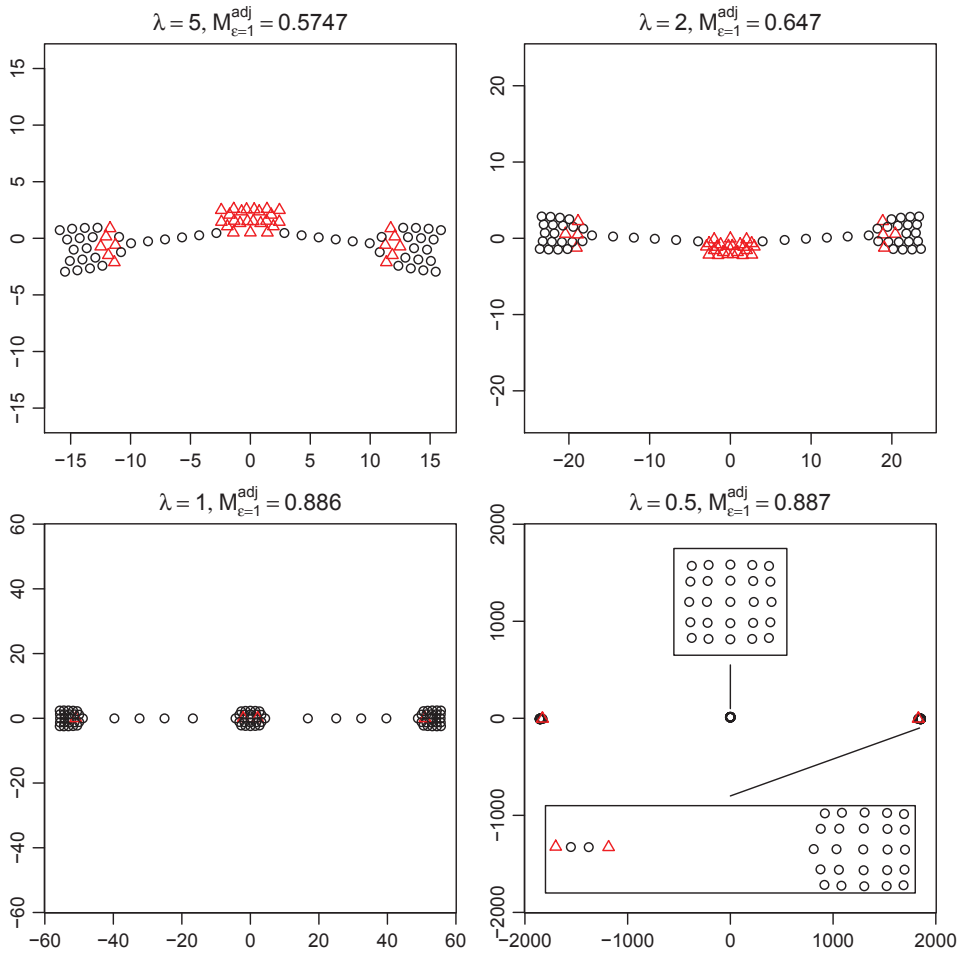


Figure 4: The configurations with varying λ with other parameters fixed at $\mu = 0$, $\tau = 1$, starting from a random configuration.

Parameter τ*:* We fix $\lambda = 5$ and $\mu = 0$ and vary τ as follows: $\tau = 0.01,\ 1,\ 10^3,\ 10^5$. The configurations starting from the original design and from a random start are shown in Figures 7 and 8. Figure 7 shows that the configuration closest to the input configuration is achieved with relatively small τ ($\tau = 0.01$). This indicates that the B-C stress functions for small τ are quite successful in recreating local distances as they are supposed to. From a random start, however, the configuration can be easily trapped in a local minimum with small τ (Figure 8, top two plots), which indicates that the relative weight of repulsion to attraction controlled by τ plays an essential role in achieving a stable configuration. With relatively larger τ, the points are more spread-out and configurations reveal the underlying structure more faithfully, both locally and globally (bottom two plots, Figure 8).

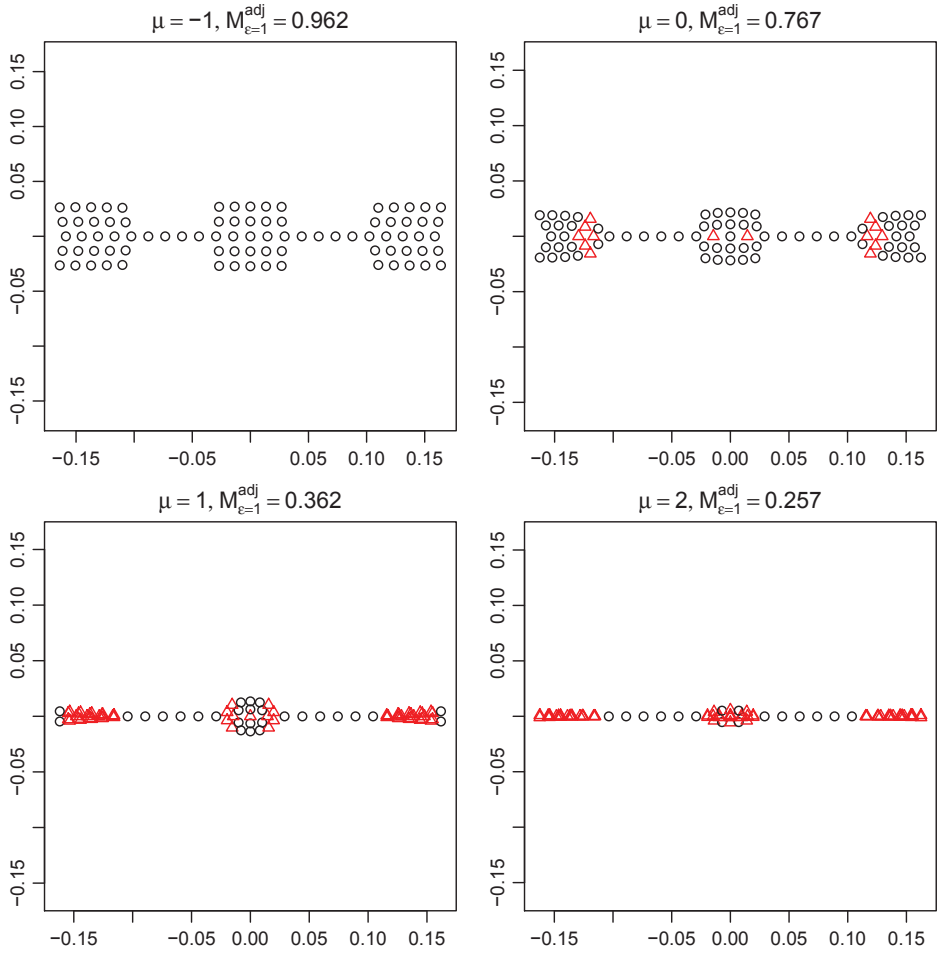


Figure 5: The configurations with varying μ with other parameters fixed at $\lambda = 5$, $\tau = 1$, starting optimization from the input configuration (Figure 2).

4.2 Olivetti Faces Data

This data set, published on Sam Roweis' website `http://www.cs.toronto.edu/~roweis/data.html`, contains 400 facial images of 40 people, 10 images for each person. All images are of size 64 by 64. Mathematically, each image can be represented by a long vector, each element of which records the light intensity of one pixel in the image. Given this representation, we treat each image as a data point lying in a 4096-dimensional space ($64 \times 64 = 4096$). For visualization purposes we reduce the dimension from 4096 to 2. As the 40 faces form natural groups, we would expect effective dimension reduction methods to show clusters in their configurations. If this expectation is correct, then this data set provides an excellent test bed for the effect of the clustering power λ.

We first centered the data, a 400×4096 matrix, at their row means to adjust the brightness of the images to the same level. We constructed a pairwise distance matrix using Euclidean distances in the original high dimensional space R^{4096}. We then defined a local graph using 4-NN, that is,

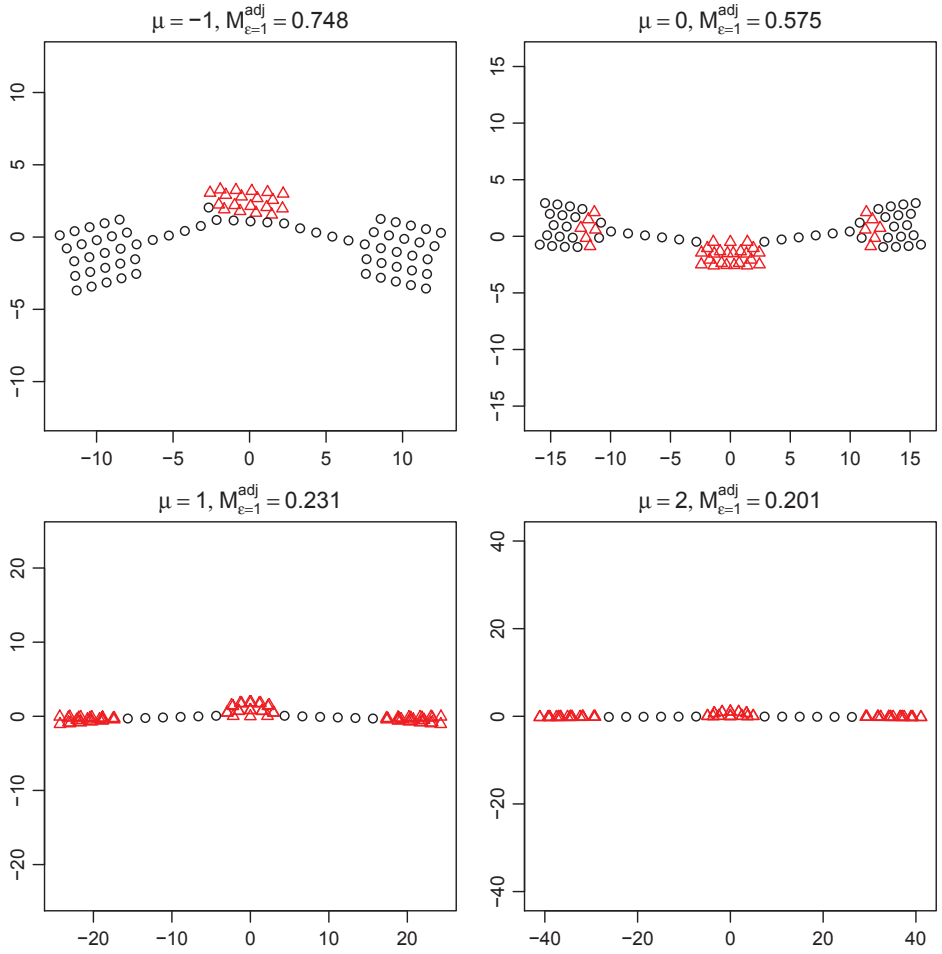


Figure 6: The configurations with varying μ with other parameters fixed at $\lambda = 5,\ \tau = 1$, starting from a random configuration.

we connected each point to its four nearest neighbors. In the resulting graph five small components were disconnected from the main component of the graph. Each of them contained images from a single person, with 5 images for one person and 5 images for another four persons. Since the disconnected components are trivially pushed away from the main component in any embedding due to the complete absence of attraction, we discarded them and kept the 355 images representing 36 people for further analysis. We created for each person a unique combination of color and symbol to code the points representing it.

Figure 9 shows 2D configurations generated by different stress functions (6) with different clustering powers, $\lambda = 2,\ 1,\ 2/3,\ 1/2$, while the other parameters are fixed at $\mu = 0$ and $\tau = 1$. For the largest value, $\lambda = 2$, we do not see any clusters; for $\lambda = 1$, we see some fuzzy clusters forming; as λ decreases the clusters become clearer. The colors and symbols show that these clusters are not artificial but real, mostly representing the images of the same person. The configurations do not

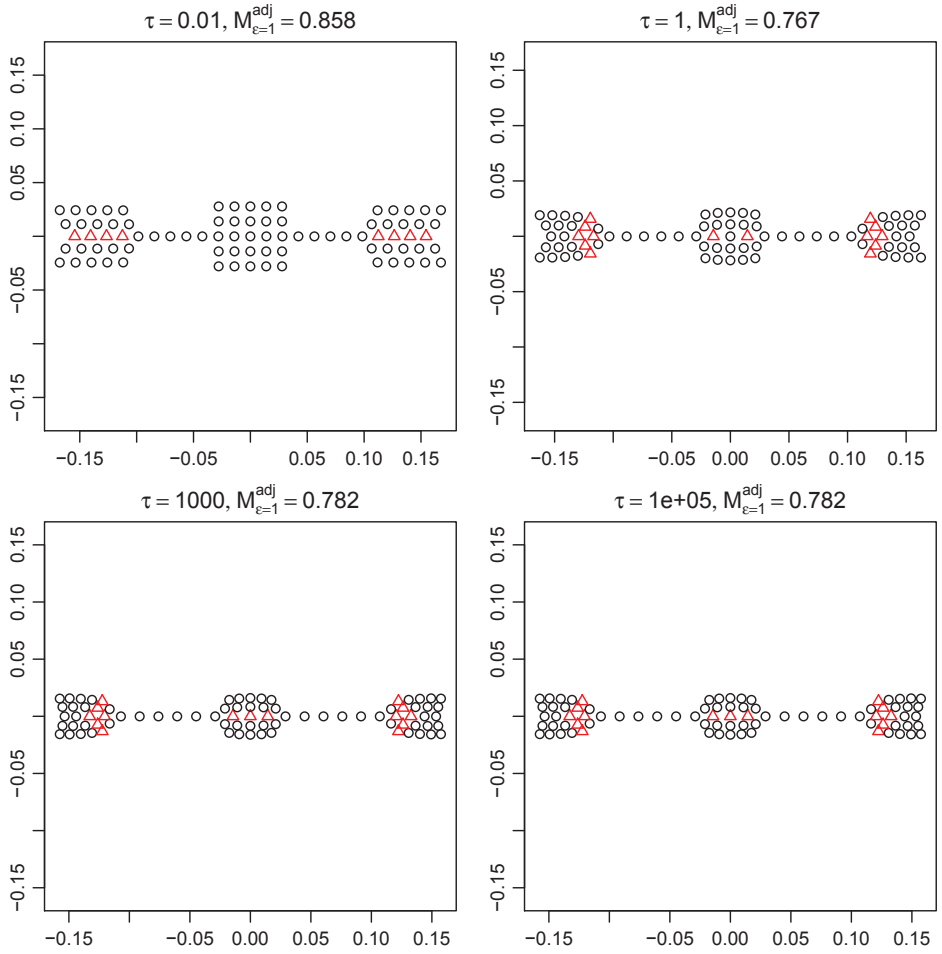


Figure 7: The configurations with varying τ with other parameters fixed at $\lambda = 5$, $\mu = 0$, starting from the input configuration (Figure 2).

produce exactly 36 clusters: some images of different people are not quite distinguishable in the configurations and some images of the same person are torn apart. The former could really present similar images; the latter could be due to the placement of images in the random start. However, the overall impression is to confirm the clustering effect due to small scale sensitivity for small values of λ. An interesting observation is that the meta-criterion M_k^{adj} increases as the small scale sensitivity strengthens, which assures us of the faithfulness of local topology.

For comparison, Figure 10 shows 2D configurations generated from four popular dimension reduction methods: PCA, MDS, Isomap and LLE. PCA and MDS did not find any clusters. Isomap and LLE did reveal a few clusters, but not as many nor as clearly as some of those obtained from the BC-family, even though we tuned the neighborhood size for Isomap and LLE to achieve the best visualization. For example, we chose a different neighborhood size $K = 8$ for LLE and Isomap

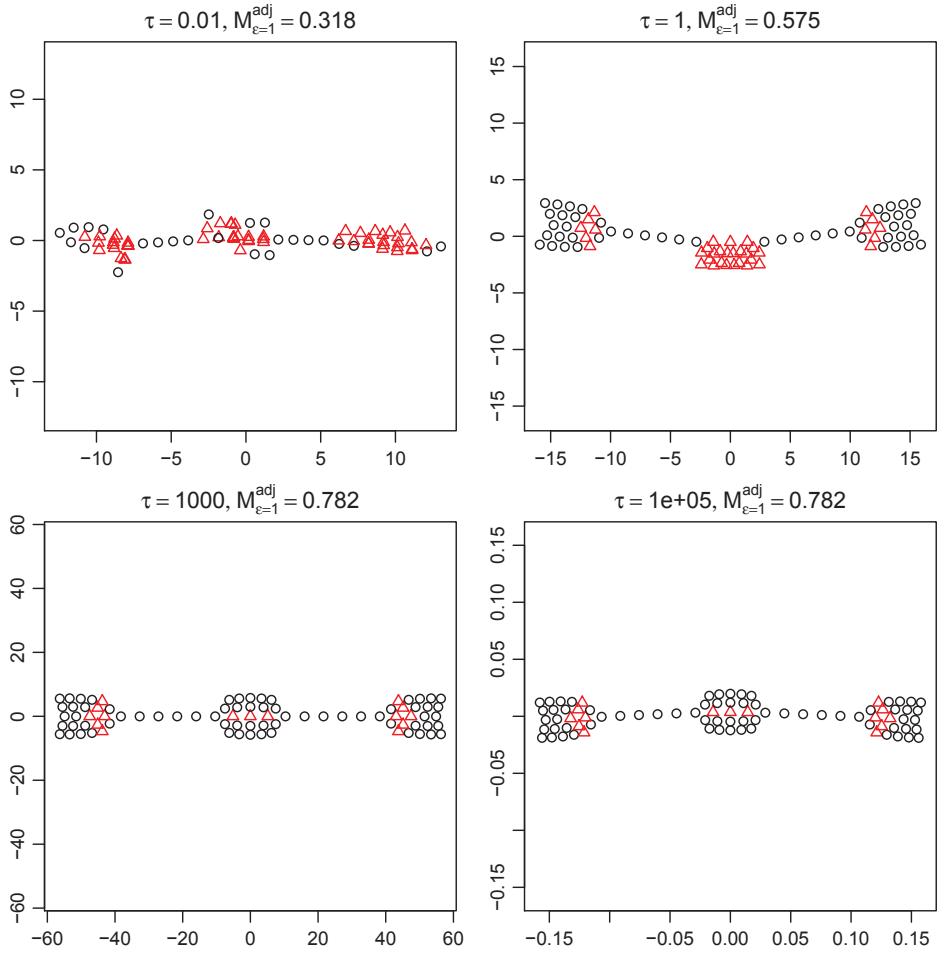


Figure 8: The configurations with varying τ with other parameters fixed at $\lambda = 5$, $\mu = 0$, starting from a random configuration.

$K = 4$; LLE configurations degenerated to lines or lines and a big cluster when $K = 4$ and 6, respectively.

4.3 Frey Face Data

In the Olivetti data we were able to show successful use of the small scale sensitivity for small values of λ. In the present example, the Frey face data, we study the effect of μ and its interaction with λ. The data were originally published with the LLE article (Roweis and Saul, 2000). We studied its low dimensional configurations from various dimension reduction methods in Chen and Buja (2009). The data contains 1965 facial images of "Branden Frey," which are stills of a short video clip recorded when Frey was making different facial expressions. Each image is of size 20×28 which can be thought of as a data point in 560-dimensional space. In our experiments we use a subset of 500 images in order to save on computations and in order to obtain less cluttered low

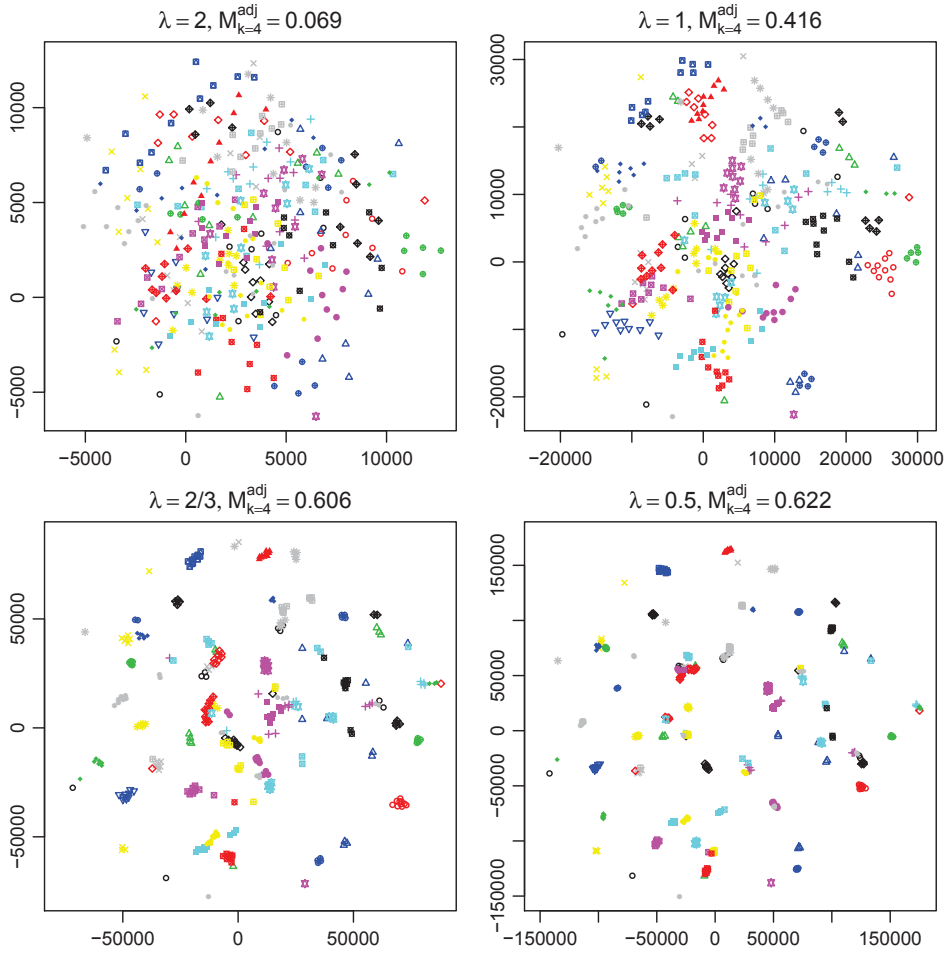


Figure 9: Olivetti Faces. Configurations with varying λ with other parameters fixed at $\mu = 0,\ \tau = 1$.

dimensional embeddings. The fact is that the intrinsic structure of the full data set is well preserved in this subset, partly due to the inherent redundancies in video sequences: the images close in order are very similar because the stills are taken more frequently than Frey's facial expression changes.

In Figure 11 we show the first two principal components of the 3D configurations for varying μ as λ and τ are fixed at one. The neighborhood size is $K = 6$. The coloring/symbolling scheme is adapted from the LMDS configurations of Chen and Buja (2009) where the points were colored/symbolled to highlight the clustering structure found in those configurations. The bottom left configuration with parameters $\lambda = \mu = 1$ is an LMDS configuration. We observe that the small scale sensitivity of a fixed value $\lambda = 1$ varies as μ varies. With smaller μ such as -1 and 0, the small scale sensitivity is most pronounced: we see bigger clusters in the configurations with larger μ are split into smaller ones. On the other hand, larger values such as $\mu = 1,\ 2$ clearly provide connectivity between clusters and therefore better capture the global structure in the data. The local neighborhood

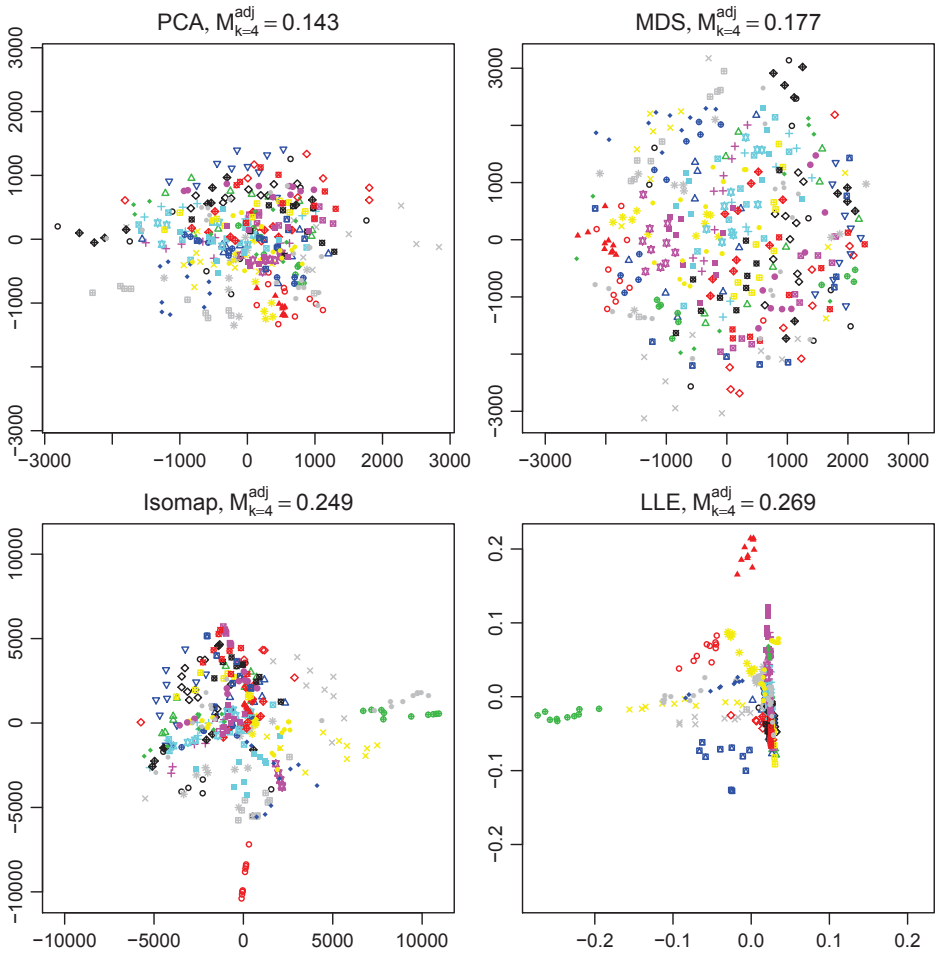


Figure 10: Olivetti Faces. Configurations from PCA, MDS, Isomap and LLE.

structure, though, is better reflected in the configurations with smaller values of μ, as suggested by the values of meta-criteria.

5. Summary and Discussion

Our work contributes to the literature on proximity analysis, nonlinear dimension reduction and graph drawing by systematizing the class of distance-based approaches whose commonality is that they generate embeddings (maps, configurations, graph drawings) of objects in such a way that given input distances between the objects are well-approximated by output distances in the embedding. The systematization consists of devising a multi-parameter family of stress functions that comprises many published proposals from the literature on proximity analysis (MDS) and graph drawing. A benefit is that the seemingly arbitrary selection of a loss function is turned into a parameter selection problem based on external "meta-criteria" that measure the quality of embeddings independently of the stress functions.

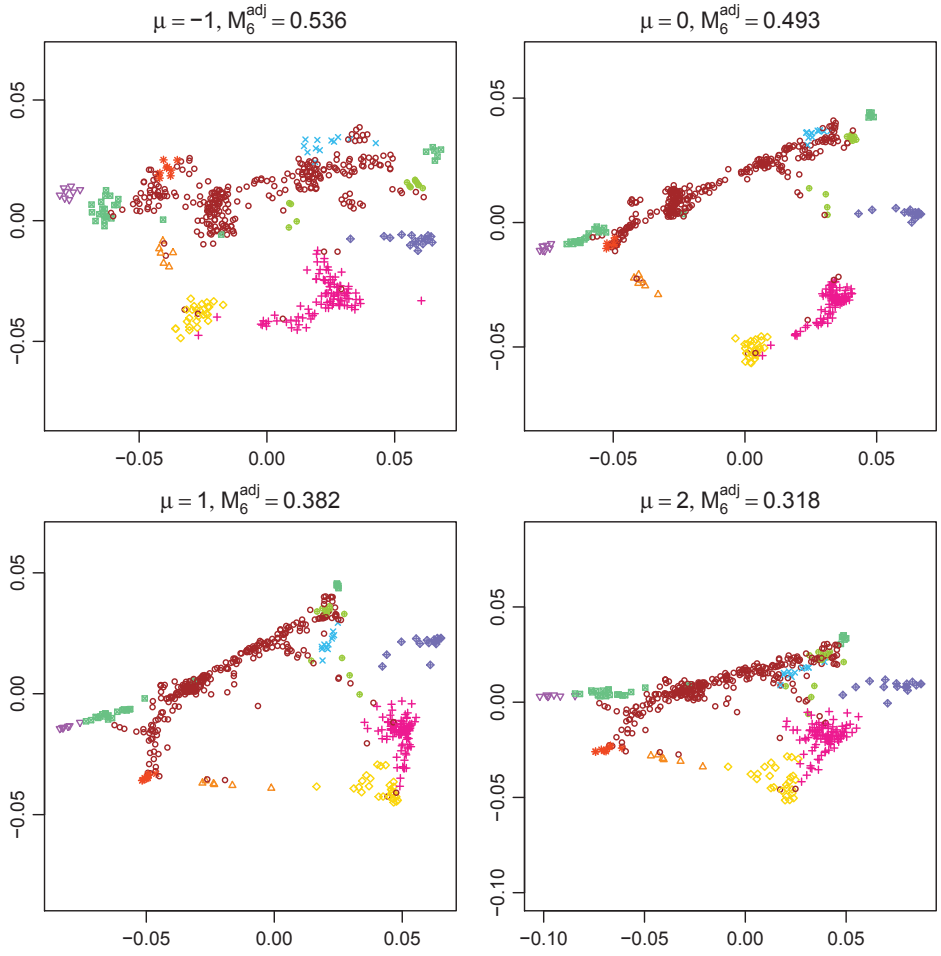


Figure 11: Frey Face Data. Configurations with varying μ when $\lambda = 1$.

The parameters of the proposed family have the following interpretations:

- λ: This parameter determines the relative strengths of the attracting and repulsing forces to each other, while maintaining "edgewise unbiasedness" of the stress function. In practical terms, this parameter strongly influences "small scale sensitivity": For decreasing λ, it increases the small-scale sensitivity, that is, the tendency to group together nearby points in the embedding. Range: $\lambda > 0$.

- μ: This parameter is the power law of the repulsing energy. The greater μ, the greater is the tendency to suppress large discrepancies between inputs $D_{i,j}$ and outputs $d_{i,j}$. Range: $-\infty < \mu < +\infty$.

- ν: This is a weighting parameter that allows up- and down-weighting of pairs of objects as a function of the input distance $D_{i,j}$. For example, as ν decreases below zero, stress terms for large distances $D_{i,j}$ will be progressively down-weighted. Range: $-\infty < \nu < +\infty$.

- τ: A regularization parameter that stabilizes configurations for incomplete distance data, that is, distance graphs, at the cost of some bias (stretching of configurations), achieved by imputing infinite input distances with infinitesimal repulsion. Range: $\tau > 0$.

The power laws for attracting and repulsing energies are interpreted as Box-Cox transformations, which has two benefits: (1) Box-Cox transformations encompass a logarithmic attracting law for $\mu + \lambda = 0$ and a logarithmic repulsing law for $\mu = 0$; (2) they permit negative powers for both laws because the Box-Cox transformations are monotone increasing for powers in the whole range of real numbers. The regularization parameter τ plays a role only when the input distance matrix is incomplete, as in the case of a distance graph or in the case of localization by restricting the loss function to small scale (as in LMDS; Chen and Buja, 2009).

The problem of incomplete distance information is often solved by completing it with additive imputations provided by the shortest-path algorithm, so that MDS-style stress functions can be used—the route taken by Isomap (Tenenbaum et al., 2000). The argument against such completion is that stress functions tend to be driven by the largest distances, which are imputed and hence noisy. Conversely the argument against not completing is that the use of pervasive repulsion to stabilize configurations amounts to imputation also, albeit of an uninformative kind. A full understanding of the trade-offs between completion and repulsion is currently lacking, but practitioners can meanwhile experiment with both approaches and compare them on their data. In both cases the family of loss functions proposed here offers control over the scale sensitivity parameter λ, the repulsion power μ, and the weighting power ν.

Another issue with distance-based approaches is that there is often much freedom in choosing the distances, in particular when applied to dimension reduction. There is therefore a need to systematize the choices and provide guidance for "distance selection."

Appendix A. Stress Minimization

Minimizing stress functions can be a very high-dimensional optimization problem involving all coordinates of all points in an embedding, amounting to Np parameters. For this reason, minimization algorithms tend to be based on simple gradient descent (Kruskal, 1964b) or on majorization (Borg and Groenen, 2005). We limit ourselves in this appendix to providing gradients, though with one innovation to solve the following problem: optimization of stress functions tends to spend much effort on getting the size of the embedding right, which is not only unnecessary but also may cause delay of convergence when in fact the shape of the embedding is already optimized, or misjudgement of convergence when the size has been gotten right but the shape has not. This appendix proceeds therefore in three steps: Section A.1 provides gradients for plain stress functions as presented in the body of this article; Section A.2 derives size-invariant versions of stress functions; Section A.3 provides gradients for the latter. (In order to make the formulas more readable, we set the parameter ν to zero and hence ignore it; it would be a simple matter to put it back in the formulas.)

A.1 Gradients for Stress Functions

Let the $N \times p$ matrix $\mathbf{X} = (x_1, \cdots, x_N)^T$ represent the embedding consisting of n points in d dimensions. As always let $D_{i,j}$ be the input distances and $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ the output distances. The B-C

stress function for $\mu \neq 0$ and $\mu+\lambda \neq 0$ (but $\nu = 0$) is

$$S(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \sum_{(i,j)\in E} \left(\frac{{d_{i,j}}^{\mu+\lambda} - 1}{(\mu+\lambda)} - {D_{i,j}}^{\lambda} \frac{{d_{i,j}}^{\mu} - 1}{\mu} \right) - t^{\lambda} \sum_{(i,j)\in E^C} \frac{{d_{i,j}}^{\mu} - 1}{\mu}.$$

Let $\nabla S = (\nabla_1, \cdots, \nabla_N)^T$ be the gradient of the stress function with respect to $\mathbf{X}$:

$$\nabla_i = \frac{\partial S}{\partial \mathbf{x}_i} = \sum_{j\in\mathcal{N}_D(i)} \left({d_{i,j}}^{\mu+\lambda-2} - {D_{i,j}}^{\lambda} {d_{i,j}}^{\mu-2} \right) (\mathbf{x}_i - \mathbf{x}_j) - t^{\lambda} \sum_{j\in\mathcal{N}_D^S(i)} {d_{i,j}}^{\mu-2} (\mathbf{x}_i - \mathbf{x}_j).$$

Define a $N \times N$ matrix M as follows:

$$M_{ij} = \begin{cases} {d_{i,j}}^{\mu+\lambda-2} - {D_{i,j}}^{\lambda} {d_{i,j}}^{\mu-2} & \text{if } j \in E(i), \\ -t^{\lambda} {d_{i,j}}^{\mu-2} & \text{if } j \notin E(i). \end{cases}$$

Note that M is symmetric. The gradient can be simplified to

$$\nabla_i = \frac{\partial S}{\partial \mathbf{x}_i} = \sum_j M_{ji} (\mathbf{x}_i - \mathbf{x}_j) = (\sum_j M_{ji}) \mathbf{x}_i - \sum_j M_{ji} x_j \ ,$$

and

$$\nabla S = \mathbf{X} * (M \cdot E) - M \cdot \mathbf{X} \ ,$$

where E is a $N \times d$ matrix with all elements being 1. The symbol '$*$' represents elementwise multiplication of the two matrices of the same size, and the symbol '$\cdot$' stands for regular matrix multiplication.

A.2 Size-Invariant Forms of B-C Stress Functions

As mentioned, it is a common experience that algorithms for minimizing stress functions spend much effort on getting the size of the embedding right. Size, however, is not of interest—shape is. We have therefore a desire to re-express stress in a manner that is independent of size. Fortunately, there exists a general method that achieves this goal: For any configuration, minimize stress with regard to size and replace the original stress with its size-minimized value. This works because the minimization with regard to size can be carried out explicitly with elementary calculus. The result is a new form of stress that is minimized by the same shapes as the original stress, but it is independent of size and hence purely driven by shape. The computational advantage of size-invariant stress is that gradient-based optimization descends along directions that change shape, not size. We sketch the derivation (again for $\nu = 0$ for less unwieldy formulas).

It is convenient to collect the repulsion terms inside and outside the graph because they share the power law:

$$S \ = \ \sum_{(i,j)\in E} BC_{\mu+\lambda}(d_{i,j}) \ - \ \sum_{(i,j)\in V^2} \tilde{D}_{i,j}^{\lambda} \, BC_{\mu}(d_{i,j}) \, ,$$

where

$$\tilde{D}_{i,j} = \begin{cases} D_{i,j}, & (i,j) \in E, \\ t & (i,j) \notin E. \end{cases}$$

Next, consider a configuration $\mathbf{X} = (x_i)_{i=1\ldots N}$ and resized versions $s\mathbf{X} = (s\,x_i)_{i=1\ldots N}$ thereof ($s > 0$). The configuration distances scale along with size: $d_{i,j}(s\mathbf{X}) = s\,d_{i,j}(\mathbf{X})$. To find the stationary size factor s of the stress as a function of s, $S = S(s)$, we observe that

$$\frac{\partial}{\partial s} BC_\mu(s\,d_{i,j}) \;=\; s^{\mu-1} {d_{i,j}}^\mu \qquad (\forall \mu \in \mathbb{R}).$$

In particular, this holds even for $\mu = 0$. Next we solve the stationary equation and check second derivatives:

$$\begin{aligned} S(sd) &= \sum_E BC_{\mu+\lambda}(s\,d_{i,j}) \;-\; \sum_{V^2} \tilde{D}^\lambda_{i,j}\, BC_\mu(s\,d_{i,j}), \\ S'(sd) &= s^{\mu+\lambda-1}\, T_{den} \;-\; s^{\mu-1} T_{num}, \qquad (7) \\ S''(sd) &= (\mu+\lambda-1)\, s^{\mu+\lambda-2}\, T_{den} \;-\; (\mu-1)\, s^{\mu-1} T_{num}, \qquad (8) \end{aligned}$$

where $T_{den} = T_{den}(d)$ and $T_{num} = T_{num}(d)$ are defined for $d = (d_{i,j})$ by

$$T_{den} = \sum_E {d_{i,j}}^{\mu+\lambda}, \qquad T_{num} = \sum_{V^2} \tilde{D}^\lambda_{i,j}\, {d_{i,j}}^\mu.$$

Again (7) and (8) hold even for $\mu = 0$ and $\mu + \lambda = 0$. The stationary size factor s_* that satisfies $S'(s_*) = 0$ is

$$s_* \;=\; \left(\frac{T_{num}}{T_{den}}\right)^\lambda \qquad (\forall \mu \in \mathbb{R},\;\; \lambda > 0). \qquad (9)$$

The factor s_* is a strict minimum:

$$S''(s_* d) \;=\; \lambda \frac{T_{num}^{\lambda\mu+1-2\lambda}}{T_{den}^{\lambda\mu-2\lambda}} \;>\; 0 \qquad (\forall \mu \in \mathbb{R},\; \lambda > 0).$$

Evaluating $S(s_*)$ we arrive at a **size-invariant** yet **shape-equivalent** form of the stress function. For the evaluation we need to separate power laws from the two logarithmic cases:

$$S(sd) \;\approx\; \begin{cases} \frac{1}{\mu+\lambda}\, s^{\mu+\lambda}\, T_{den} \;-\; \frac{1}{\mu}\, s^\mu\, T_{num} & (\mu+\lambda \neq 0,\; \mu \neq 0), \\ |E| \log(s) \;+\; \sum_E \log(d_{i,j}) \;-\; \frac{1}{\mu}\, s^\mu\, T_{num} & (\mu+\lambda = 0), \\ \lambda s^\lambda\, T_{den} \;-\; \sum_{V^2} \tilde{D}^\lambda_{i,j}\, (\log(s) + \log(d_{i,j})) & (\mu = 0), \end{cases}$$

where "$\approx$" means "equal up to additive constants that are irrelevant for optimization." We calculate $\tilde{S} = S(s_*)$ separately in the three cases with s_* from (9):

- $\mu+\lambda \neq 0$, $\mu \neq 0$: Several algebraic simplifications produce the following.

$$\tilde{S} \;=\; \left(\frac{1}{\mu+\lambda} - \frac{1}{\mu}\right) \frac{T_{num}^{\lambda\mu+1}}{T_{den}^{\lambda\mu}} \;=\; \left(\frac{1}{\mu+\lambda} - \frac{1}{\mu}\right) \frac{\left(\sum_{V^2} \tilde{D}^\lambda_{i,j}\, {d_{i,j}}^\mu\right)^{\lambda\mu+1}}{\left(\sum_E {d_{i,j}}^{\mu+\lambda}\right)^{\lambda\mu}}. \qquad (10)$$

  [Note that $\tilde{S}$ gets minimized, hence the ratio on the right gets minimized when the left factor is positive (i.e., $\mu < 0 < \mu + 1/\lambda$), and it gets maximized when the left factor is negative (i.e., $\mu > 0$ or $\mu + 1/\lambda < 0$).]

- $\mu+\lambda=0$: We take advantage of the fact that $T_{den}=|E|$ and $\mu=-\lambda$. We have

$$\tilde{S} \;\approx\; |E|\,\lambda \log \sum_{V^2} \left(\frac{\tilde{D}_{i,j}}{d_{i,j}}\right)^{\lambda} \;+\; \sum_{E} \log(d_{i,j}).$$

- $\mu=0$: We take advantage of the fact that $T_{den}=\sum_E {d_{i,j}}^{\lambda}$ and also that $T_{num}=\sum_{V^2}\tilde{D}_{i,j}^{\lambda}$ is a constant for optimization with regard to $d=(d_{i,j})$, and any additive term that is just a function of $\tilde{D}_{i,j}$ but not $d_{i,j}$ can be neglected. We have

$$\tilde{S} \;\approx\; \lambda \left(\sum_{V^2} \tilde{D}_{i,j}^{\lambda}\right) \log(\sum_{E} {d_{i,j}}^{\lambda}) \;-\; \sum_{V^2} \tilde{D}_{i,j}^{\lambda} \log(d_{i,j}).$$

Even though size-invariance holds by construction, one checks it easily in all three cases: $\tilde{S}(sd)=\tilde{S}(d)$.

A.3 Gradients for Size-Invariant Stress Functions

To describe the gradient of the size-invariant stress function $\tilde{S}(d)$. We only consider the case $\mu+\lambda\neq 0$ and $\mu\neq 0$, which is shown in Equation (10).

Let $\nabla S=((\nabla S)_1,...,(\nabla S)_n)^T$ be the gradient the $\tilde{S}(d)$ with respect to configuration $\mathbf{X}=(\mathbf{x}_1,...,\mathbf{x}_n)^T$. We have

$$\begin{aligned}
(\nabla S)_i &= \left(\frac{1}{\mu+\lambda}-\frac{1}{\mu}\right)\left[(\lambda\mu+1)\left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu}(\nabla T_{num})_i-(\lambda\mu)\left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu+1}(\nabla T_{den})_i\right],\\
(\nabla T_{num})_i &= \frac{\partial T_{num}(d)}{\partial \mathbf{x}_i}=\mu\sum_j \tilde{D}_{i,j}d_{i,j}^{\mu-2}(\mathbf{x}_i-\mathbf{x}_j),\\
(\nabla T_{den})_i &= \frac{\partial T_{den}(d)}{\partial \mathbf{x}_i}=(\mu+\lambda)\sum_{j\in E(i)} d_{i,j}^{\mu+\lambda-2}(\mathbf{x}_i-\mathbf{x}_j).
\end{aligned}$$

Plug $(\nabla T_{num})_i$ and $(\nabla T_{den})_i$ into the $(\nabla S)_i$, and we have

$$\begin{aligned}
(\nabla S)_i &= \left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu}\left(\frac{T_{num}}{T_{den}}\sum_{j\in E(i)} d_{i,j}^{\mu+\lambda-2}(\mathbf{x}_i-\mathbf{x}_j)-\sum_j \tilde{D}_{i,j}d_{i,j}^{\mu-2}(\mathbf{x}_i-\mathbf{x}_j)\right)\\
&= \left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu}\left(\sum_{j\in E(i)}\left(\frac{T_{num}}{T_{den}}d_{i,j}^{\mu+\lambda-2}-D_{ji}^{\lambda}d_{i,j}^{\mu-2}\right)(\mathbf{x}_i-\mathbf{x}_j)-\sum_{j\in E^c(i)} t\; d_{i,j}^{\mu-2}(\mathbf{x}_i-\mathbf{x}_j)\right).
\end{aligned}$$

Define a $N\times N$ matrix M by

$$M_{ij} \;=\; \begin{cases} \left(\frac{T_{num}}{T_{den}}d_{i,j}^{\mu+\frac{1}{\lambda}-2}-{D_{i,j}}^{\lambda}d_{i,j}^{\mu-2}\right) & \text{for } j\in E(i),\\ t\; d_{i,j}^{\mu-2} & \text{for } j\notin E(i). \end{cases}$$

The gradient can be simplified to

$$\begin{aligned}(\nabla S)_i &= \left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu} \sum_j M_{ij}\,(\mathbf{x}_i - \mathbf{x}_j) \\ &= \left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu} \left(\sum_j M_{ij}\mathbf{x}_i - \sum_j M_{ij}\mathbf{x}_j\right),\end{aligned}$$

and

$$\nabla S = \left(\frac{T_{num}}{T_{den}}\right)^{\lambda\mu} (\mathbf{X} * (M \cdot E) - M \cdot \mathbf{X}).$$

We did the calculation separately for $\mu = 0$ and $\lambda + \mu = 0$ which resulted in the following:

$$\mu = 0: \qquad \tilde{S}(d) \sim \lambda \left(\sum_{(i,j)\in V^2} \tilde{D}_{ij}\right) \log\left(\sum_{(i,j)\in E} {D_{i,j}}^{\lambda}\right) - \sum_{(i,j)\in V^2} \tilde{D}_{ij} \log D_{i,j},$$

$$\mu + \lambda = 0: \qquad \tilde{S}(d) \sim \lambda \left(\sum_{(i,j)\in E}\right) \log\left(\sum_{(i,j)\in V^2} \tilde{D}_{ij} {D_{i,j}}^{\mu}\right) + \sum_{(i,j)\in E} \log D_{i,j}.$$

References

Ulas Akkucuk and J. Douglas Carroll. PARAMAP vs. Isomap: A comparison of two nonlinear mapping algorithms. *Journal of Classification*, 23(2):221–254, 2006.

Ingwer Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, 2005.

Lisha Chen. *Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Layout and Proximity Analysis*. PhD thesis, Ph.d. Thesis, University of Pennsylvania, Philadelphia, Pennsylvania, 2006.

Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104 (485):209–219, 2009.

Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics (TOG)*, 15(4):301–331, 1996.

Thomas M.J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

Emden Gansner, Yehuda Koren, and Stephen North. Graph drawing by stress majorization. In *Graph Drawing*, pages 239–250. Springer, 2005.

Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

Yehuda Koren and Ali Çivril. The binary stress model for graph drawing. In *Graph Drawing*, pages 193–205. Springer, 2009.

Joseph B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.

Joseph B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29 (2):115–129, 1964b.

Joseph B. Kruskal and Judith B. Seery. Designing network diagrams. In *Proc. First General Conference on Social Graphics*, pages 22–50, 1980.

John A. Lee and Michel Verleysen. Rank-based quality assessment of nonlinear dimensionality reduction. In *16th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium*, pages 49–54, 2008.

Fan Lu, Sündüz Keleş, Stephen J Wright, and Grace Wahba. Framework for kernel regularization with application to protein clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12332–12337, 2005.

Andreas Noack. Energy models for drawing clustered small world graphs. Technical report, Inst. of Computer Science, Brandenburg Technical University, Cottbus, Germany, 2003.

Andreas Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007.

Andreas Noack. Modularity clustering is force-directed layout. *Physical Review E*, 79(2):026102, 2009.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

John W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 100(5):401–409, 1969.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Yoshio Takane, Forrest W. Young, and Jan De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977.

Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Michael W. Trosset. Classical multidimensional scaling and laplacian eigenmaps. *Presentation given at the 2006 Joint Statistical Meeting (Session 411)*, 2006.

Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6):889–899, 2006.

Kilian Q. Weinberger, Fei Sha, Qihui Zhu, and Lawrence K. Saul. Graph Laplacian regularization for large-scale semidefinite programming. *Advances in Neural Information Processing Systems*, 19:1489, 2007.

Journal of Machine Learning Research 14 (2013) 1175-1179 Submitted 6/12; Revised 10/12; Published 4/13

GPstuff: Bayesian Modeling with Gaussian Processes

Jarno Vanhatalo* JARNO.VANHATALO@HELSINKI.FI
Department of Environmental Sciences
University of Helsinki
P.O. Box 65
FI-00014 Helsinki, Finland

Jaakko Riihimäki JAAKKO.RIIHIMAKI@AALTO.FI
Jouni Hartikainen JOUNI.HARTIKAINEN@AALTO.FI
Pasi Jylänki PASI.JYLANKI@AALTO.FI
Ville Tolvanen VILLE.TOLVANEN@AALTO.FI
Aki Vehtari AKI.VEHTARI@AALTO.FI
Department of Biomedical Engineering and Computational Science
Aalto University School of Science
P.O. Box 12200
FI-00076 Aalto, Finland

Editor: Balazs Kegl

Abstract

The GPstuff toolbox is a versatile collection of Gaussian process models and computational tools required for Bayesian inference. The tools include, among others, various inference methods, sparse approximations and model assessment methods.

Keywords: Gaussian process, Bayesian hierarchical model, nonparametric Bayes

1. Introduction

Gaussian process (GP) prior provides a flexible building block for many hierarchical Bayesian models (Rasmussen and Williams, 2006). GPstuff (v4.1) is a versatile collection of computational tools for GP models and it has already been used in several published projects, for example, in epidemiology, species distribution modeling and building energy usage modeling (see Vanhatalo et al., 2013, and project web pages for references). GPstuff combines models and inference tools in a modular format. It also provides various sparse GP models and methods for model assessment. The toolbox is compatible with Unix and Windows Matlab (at least r2009b or later). Most features work also with Octave (tested with 3.6.4). The toolbox is available from `http://becs.aalto.fi/en/research/bayes/gpstuff/` and also `http://mloss.org/software/view/451/`.

2. Implementation

In many practical GP models, the observations $\mathbf{y} = [y_1, ..., y_n]^{\mathrm{T}}$ related to inputs (covariates) $\mathbf{X} = \{\mathbf{x}_i = [x_{i,1}, ...x_{i,d}]^{\mathrm{T}}\}_{i=1}^n$ are assumed to be conditionally independent given a latent function (or predictor) $f(\mathbf{x})$ so that the likelihood $p(\mathbf{y}|\mathbf{f},\boldsymbol{\gamma}) = \prod_{i=1}^n p(y_i|f_i,\boldsymbol{\gamma})$, where $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_n)]^{\mathrm{T}}$, fac-

∗. Work done mainly while at BECS, Aalto University.

torizes over cases. The latent function is given a GP prior, $f \sim GP(m(\mathbf{x}|\phi), k(\mathbf{x},\mathbf{x}'|\theta))$ which is defined by the mean and covariance function, $m(\mathbf{x}|\phi)$ and $k(\mathbf{x},\mathbf{x}'|\theta)$ respectively. The parameters, $\vartheta = \{\gamma,\phi,\theta\}$, are given a hyperprior after which the posterior $p(\mathbf{f}|\mathbf{y},\mathbf{X})$ is approximated and used for prediction. Most of the models in GPstuff follow the above single latent dependency, but there are also models where each factor depends on multiple latent values.

We illustrate the construction and inference of a GP model with a regression example. First, we assume $y_i = f(\mathbf{x}_i) + \varepsilon_i$, $\varepsilon_i \sim N(0,\sigma^2)$, and give $f(\mathbf{x})$ a GP prior with a squared exponential covariance function, $k(\mathbf{x},\mathbf{x}') = \sigma^2_{\text{se}} \exp(||\mathbf{x}-\mathbf{x}'||^2/2l^2)$.

```
lik = lik_gaussian('sigma2', 0.2^2);                    % init. the likelihood
gpcf = gpcf_sexp('lengthScale', 1, 'magnSigma2', 0.2^2) % init. the cov. function
gp = gp_set('lik', lik, 'cf', gpcf);                    % init. the model struct
% Find MAP estimate of the parameters and predict to new inputs
opt=optimset('TolFun',1e-3,'TolX',1e-3,'Display','iter');  % optimization settings
gp=gp_optim(gp,x,y,'optimf',@fminscg,'opt',opt);        % x,y = training data
[Ef, Varf] = gp_pred(gp, x, y, xt);                     % xt = test inputs
```

The model is constructed modularly so that each mathematical function or distribution is represented by an "object" style structure. The structures `lik` and `gpcf` contain all the essential information about the likelihood and covariance function such as parameter values and function handles to construct a covariance matrix and its gradient with respect to the parameters. All the model blocks are collected into a GP structure constructed by `gp_set`.

There are two lines of approach for the inference. The first assumes a Gaussian observation model which enables an analytic solution for the marginal likelihood $p(\mathbf{y}|\mathbf{X},\vartheta)$ and the conditional posterior $p(\mathbf{f}|\mathbf{X},\mathbf{y},\vartheta)$. Using the relation $p(\vartheta|\mathbf{y},\mathbf{X}) \propto p(\mathbf{y}|\mathbf{X},\vartheta)p(\vartheta)$ the parameters, ϑ, can be optimized to the maximum a posterior (MAP) estimate or marginalized over with grid, central composite design (CCD), importance sampling (IS) or Markov chain Monte Carlo (MCMC) integration (Vanhatalo et al., 2010). With other observation models the marginal likelihood and the conditional posterior have to be approximated either with Laplace's method (LA) or expectation propagation (EP) (Rasmussen and Williams, 2006). An alternative approach is to sample from the joint posterior $p(\mathbf{f},\vartheta|\mathbf{X},\mathbf{y})$ with MCMC by alternating sampling from $p(\mathbf{f}|\mathbf{X},\mathbf{y},\vartheta)$ and $p(\vartheta|\mathbf{X},\mathbf{y},\mathbf{f})$.

Above, `gp_optim` returns a redefined model structure with parameter values optimized to their MAP estimate. Any optimizer with similar arguments to Matlab's optimizers can be used. `gp_pred` returns the conditional posterior predictive mean, $\mathrm{E}[f|\mathbf{y},\mathbf{X},\vartheta]$ and variance $\mathrm{Var}[f|\mathbf{y},\mathbf{X},\vartheta]$ at the test inputs.

Many sparse GPs have been proposed to speed up the computations with large data sets. GPstuff includes FI(T)C, PIC, SOR, DTC (Quiñonero-Candela and Rasmussen, 2005), VAR (Titsias, 2009), CS+FIC (Vanhatalo and Vehtari, 2008) sparse approximations, and several compactly supported (CS) covariance functions. For example, CS+FIC can be used with the following modification to the model initialization.

```
gpcf2 = gpcf_ppcs2('nin', nin, 'lengthScale', 5, 'magnSigma2', 1);
gp = gp_set('type','CS+FIC','lik',lik,'cf',{gpcf,gpcf2},'X_u',Xu)
```

In the first line, a CS covariance function, piecewise polynomial of second order, is created. It is then given to the GP structure together with inducing inputs (`Xu`) and sparse GP type definition.

We can tailor the above model, for example, by replacing the Gaussian observation model with a more robust Student-t observation model (Jylänki et al., 2011).

```
lik = lik_t('nu', 4, 'sigma2', 10, 'nu_prior', prior_logunif);
gp = gp_set('lik', lik, 'cf', gpcf, 'jitterSigma2', 1e-6, 'latent_method', 'EP');
```

Here we set explicitly the prior for the degrees of freedom parameter, ν in the Student-t distribution, add jitter on the diagonal of the covariance matrix and define EP as the means to approximate the marginal likelihood.

GPstuff has wide variety of observation models (see Table 1) of which we want to highlight implementations of recently proposed multinomial probit with EP (Riihimäki et al., 2013) and logistic GP density estimation and regression with Laplace approximation (Riihimäki and Vehtari, 2012).

The constructed models could be compared, for example, with deviance information criterion (DIC), widely applicable information criterion (WAIC), leave-one-out or k-fold cross-validation (LOO/kf-CV) (Vehtari and Ojanen, 2012) with functions `gp_dic`, `gp_waic`, `gp_loopred` and `gp_kfcv`.

New models can be implemented by modifying the existing model blocks, such as covariance functions. Adding new inference methods is more laborious since they require summaries from model blocks which may not be provided by the current version of GPstuff. A thorough introduction to GPstuff is provided by demo programs and Vanhatalo et al. (2013).

3. Related Software

Perhaps the best known GP software packages are the Gaussian processes for Machine Learning (GPML) (Rasmussen and Nickisch, 2010) and the flexible Bayesian modelling (FBM) (Neal, 1998). Overviews of alternatives are provided by the Gaussian processes website (`http://www.gaussianprocess.org/`) and the R Archive Network (`http://cran.r-project.org/`). The main advantage of GPstuff over the other GP software is its versatile collection of models and computational tools. Its most important features and comparison to GPML and FBM are presented in Table 1. GPstuff project was started in 2006 based on the MCMCstuff-toolbox (`http://becs.aalto.fi/en/research/bayes/mcmcstuff/`), which was based on Netlab (Nabney, 2001) and influenced by FBM. The INLA software (Rue et al., 2009) and the book by Rasmussen and Williams (2006) have motivated some of the technical details in GPstuff. In addition, the implementation of sparse matrix routines, used with the CS covariance functions, rely on the SuiteSparse toolbox (Davis, 2005).

Acknowledgments

The work for GPstuff has been partially funded by the Academy of Finland (grant 218248). Pieces of code have been written by other people than us. At BECS, Aalto University these persons are: T. Auranen, T. Nikoskinen, T. Peltola, E. Pennala, H. Peura, V. Pietiläinen, M. Siivola, S. Särkkä and E. Ulloa. People outside Aalto University are: C. M. Bishop, T. A. Davis, M. D. Hoffman, K. Hornik, D.-J. Kroon, I. Murray, I. T. Nabney, R. M. Neal and C. E. Rasmussen. We thank them all for sharing their code under a free software license.

	GPstuff	GPML	FBM
Covariance functions			
number of elementary functions	13	10	4
sums of elements, masking of inputs	x	x	x
delta distance	x		x
products, positive scaling of elements	x	x	
Mean functions			
number of elementary functions	4	4	0
sums of elements, masking of inputs	x	x	
products, power, scaling of elements		x	
marginalized parameters	x		
Single latent likelihood/observation models			
Gaussian	x	x	x
logistic/logit, erf/probit	x	x	MCMC
Poisson	x	LA/EP/MCMC	MCMC
Gaussian scale mixture	MCMC		MCMC
Student-*t*	x	LA/VB/MCMC	
Laplacian		EP/VB/MCMC	
mixture of likelihoods		LA/EP/MCMC	
sech-squared, uniform for classification		x	
derivative observations	for sexp covf only		
binomial, negative binomial, zero-trunc. negative binomial, log-Gaussian Cox process; Weibull, log-Gaussian and log-logistic with censoring	x		
quantile regression	MCMC/EP		
Multilatent likelihood/observation models			
multinomial, Cox proportional hazard model, density estimation, density regression, input dependent noise, input dependent overdispersion in Weibull, zero-inflated negative binomial	MCMC/LA		
multinomial logit (softmax)	MCMC/LA		MCMC
multinomial probit	EP		MCMC
Priors for parameters (ϑ)			
several priors, hierarchical priors	x		x
Sparse models			
FITC	x	exact/EP/LA	
CS, FIC, CS+FIC, PIC, VAR, DTC, SOR	x		
PASS-GP	LA/EP		
Latent inference			
exact (Gaussian only)	x	x	x
scaled Metropolis, HMC	x		x
LA, EP, elliptical slice sampling	x	x	
variational Bayes (VB)		x	
scaled HMC (with inverse of prior cov.)		x	
scaled HMC (whitening with approximate posterior covariance)	x		
parallel EP, robust EP	x		
marginal corrections (cm2 and fact)	x		
Hyperparameter inference			
type II ML	x	x	x
type II MAP, Metropolis, HMC	x		x
LOO-CV for Gaussian	x	x	
least squares LOO-CV for non-Gaussian		some likelihoods	
LA/EP LOO-CV for non-Gaussian, k-fold CV	x		
NUTS, slice sampling (SLS), surrogate SLS, shrinking-rank SLS, covariance-matching SLS, grid, CCD, importance sampling	x		
Model assessment			
marginal likelihood	MAP,ML	ML	
LOO-CV for fixed hyperparameters	x	x	
LOO-CV for integrated hyperparameters, k-fold CV, WAIC, DIC	x		
average predictive comparison	x		

Table 1: The comparison of features in GPstuff (v4.1), GPML (v3.2) and FBM (2004-11-10) toolboxes. In case of model blocks the notation x means that it can be inferred with any inference method (EP, LA (Laplace), MCMC and in case of GPML also with VB). In case of sparse approximations, inference methods and model assessment methods x means that the method is available for all model blocks.

References

Timothy A. Davis. Algorithm 849: A concise sparse Cholesky factorization package. *ACM Trans. Math. Softw.*, 31:587–591, 2005. ISSN 0098-3500.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust Gaussian process regression with a Student-*t* likelihood. *Journal of Machine Learning Research*, 12:3227–3257, 2011.

Ian T. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer, 2001.

Radford Neal. Regression and classification using Gaussian process priors. In J. M. Bernardo, J. O. Berger, A. P. David, and A. P. M. Smith, editors, *Bayesian Statistics 6*, pages 475–501. Oxford University Press, 1998.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(3):1939–1959, 2005.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Jaakko Riihimäki and Aki Vehtari. Laplace approximation for logistic Gaussian process density estimation. *ArXiv e-prints*, (1211.0174), 2012. URL `http://arxiv.org/abs/1211.0174`.

Jaakko Riihimäki, Pasi Jylänki, and Aki Vehtari. Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *Journal of Machine Learning Research*, 14:75–109, 2013.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society B*, 71(2):1–35, 2009.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. *JMLR Workshop and Conference Proceedings*, 5:567–574, 2009.

Jarno Vanhatalo and Aki Vehtari. Modelling local and global phenomena with sparse Gaussian processes. In David A. McAllester and Petri Myllymäki, editors, *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 571–578, 2008.

Jarno Vanhatalo, Ville Pietiläinen, and Aki Vehtari. Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine*, 29(15):1580–1607, 2010.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Bayesian modeling with Gaussian processes using the GPstuff toolbox. *ArXiv e-prints*, (1206.5754), 2013. URL `http://arxiv.org/abs/1206.5754`.

Aki Vehtari and Janne Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

Journal of Machine Learning Research 14 (2013) 1181-1227 Submitted 10/11; Revised 10/12; Published 4/13

Performance Bounds for λ Policy Iteration and Application to the Game of Tetris

Bruno Scherrer BRUNO.SCHERRER@INRIA.FR
MAIA Project-Team, INRIA Lorraine
615 rue du Jardin Botanique
54600 Villers-lès-Nancy
FRANCE

Editor: Shie Mannor

Abstract

We consider the discrete-time infinite-horizon optimal control problem formalized by Markov decision processes (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). We revisit the work of Bertsekas and Ioffe (1996), that introduced λ policy iteration—a family of algorithms parametrized by a parameter λ—that generalizes the standard algorithms value and policy iteration, and has some deep connections with the temporal-difference algorithms described by Sutton and Barto (1998). We deepen the original theory developed by the authors by providing convergence rate bounds which generalize standard bounds for value iteration described for instance by Puterman (1994). Then, the main contribution of this paper is to develop the theory of this algorithm when it is used in an approximate form. We extend and unify the separate analyzes developed by Munos for approximate value iteration (Munos, 2007) and approximate policy iteration (Munos, 2003), and provide performance bounds in the discounted and the undiscounted situations. Finally, we revisit the use of this algorithm in the training of a Tetris playing controller as originally done by Bertsekas and Ioffe (1996). Our empirical results are different from those of Bertsekas and Ioffe (which were originally qualified as "paradoxical" and "intriguing"). We track down the reason to be a minor implementation error of the algorithm, which suggests that, in practice, λ policy iteration may be more stable than previously thought.

Keywords: stochastic optimal control, reinforcement learning, Markov decision processes, analysis of algorithms

1. Introduction

We consider the discrete-time infinite-horizon optimal control problem formalized by Markov decision processes (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). We revisit the λ policy iteration algorithm introduced by Bertsekas and Ioffe (1996), also published in the reference textbook of Bertsekas and Tsitsiklis (1996),[1] that (as stated by the authors) *"is primarily motivated by the case of large and complex problems where the use of approximation is essential"*. It is a family of algorithms parametrized by a parameter λ that generalizes the standard dynamic-programming algorithms value iteration (which corresponds to the case $\lambda = 0$) and policy iteration (case $\lambda = 1$), and has some deep connections with the temporal-difference algorithms that are well known to the reinforcement-learning community (Sutton and Barto, 1998; Bertsekas and Tsitsiklis, 1996).

1. The work of Bertsekas and Ioffe (1996) being historically anterior to the textbook of Bertsekas and Tsitsiklis (1996), we only refer to the former in the rest of the paper.

In their original paper, Bertsekas and Ioffe (1996) show the convergence of λ policy iteration for its exact version and provide its *asymptotic* convergence rate. The authors also describe a case study involving an instance of approximate λ policy iteration, but neither their paper nor (to the best of our knowledge) any subsequent work show that this makes sense: two important issues are whether approximations can be controlled throughout the iterations and checking that the approach does not break when considering an undiscounted problem like Tetris. In this paper, we extend the theory on this algorithm in several ways. We derive its *non-asymptotic* convergence rate for its exact version. More importantly, we develop the theory of λ policy iteration for its main purpose, that is—recall the above quote—when it is run in an approximate form. We show that the performance loss due to using the greedy policy with respect to the current value estimate instead of the optimal policy can be made arbitrarily small by controlling the error along the iterations. Last but not least, we show that our analysis can be extended to the undiscounted case.

The rest of the paper is organized as follows. In Section 2, we introduce the framework of Markov decision processes, describe the two standard algorithms, value and policy iteration. Section 3 describes λ policy iteration in an original way that makes its connection with these standard algorithms obvious. We discuss there the close connection with TD(λ) (Sutton and Barto, 1998) and recall the main results obtained by Bertsekas and Ioffe (1996): convergence and asymptotic rate of convergence of the exact algorithm. Our main results are stated in Section 4. We first argue that the analysis of λ policy iteration is more involved than that of value and policy iteration since neither contraction nor monotonicity arguments, that analysis of these two algorithms rely on, hold for λ policy iteration. We provide a non-asymptotic analysis of λ policy iteration and several asymptotic performance bounds for its approximate version. We close this section by presenting performance bounds of approximate λ policy iteration that also apply to the undiscounted case. We discuss in Section 5 the relations between our results and those previously obtained for approximate value and policy iteration by Munos (2003, 2007). Last but not least, Section 6 revisits the empirical part of the work of Bertsekas and Ioffe (1996), where an approximate version of λ policy iteration is used for training a Tetris controller.

2. Framework And Standard Algorithms

We begin by describing the framework of Markov decision processes we consider throughout the paper. We go on by describing the two main algorithms of the literature, value and policy iteration, for solving the related problem.

We consider a discrete-time dynamic system whose state transition depends on a control. We assume that there is a **state space** X of finite[2] size N. When at state $i \in \{1,..,N\}$, an action is chosen from a finite **action space** A. The action $a \in A$ specifies the **transition probability** $p_{ij}(a)$ to the next state j. At each transition, the system is given a reward $r(i,a,j)$ where r is the instantaneous **reward function**. In this context, we look for a stationary deterministic policy (a function $\pi : X \to A$ that maps states into actions[3]) that maximizes the expected discounted sum of rewards from any state i,

2. We restrict our attention to finite state space problems for simplicity. The extension of our study to infinite/continuous state spaces is straightforward.
3. Restricting our attention to stationary deterministic policies is not a limitation. Indeed, for the optimality criterion to be defined soon, it can be shown that there exists at least one stationary deterministic policy that is optimal (Puterman, 1994).

called the **value of policy** π at state i:

$$v^\pi(i) := E_\pi\left[\sum_{k=0}^{\infty}\gamma^k r(i_k,\pi(i_k),i_{k+1})\middle| i_0 = i\right]$$

where E_π denotes the expectation conditional on the fact that the actions are selected with the policy π, and $0<\gamma<1$ is a discount factor.[4] The tuple $\langle X,A,p,r,\gamma\rangle$ is called a **Markov decision process (MDP)** (Puterman, 1994; Bertsekas and Tsitsiklis, 1996).

The **optimal value** starting from state i is defined as

$$v_*(i) := \max_\pi v^\pi(i).$$

We write P^π for the $N\times N$ stochastic matrix whose elements are $p_{ij}(\pi(i))$ and r^π the vector whose components are $\sum_j p_{ij}(\pi(i))r(i,\pi(i),j)$. The value functions v^π and v_* can be seen as vectors on X. It is well known that v^π is a solution of the following Bellman equation:

$$v^\pi = r^\pi + \gamma P^\pi v^\pi.$$

The value function v^π is thus a fixed point of the linear operator $T^\pi v := r^\pi + \gamma P^\pi v$. As P^π is a stochastic matrix, its eigenvalues cannot be greater than 1, and consequently $I-\gamma P^\pi$ is invertible. This implies that

$$v^\pi = (I-\gamma P^\pi)^{-1} r^\pi = \sum_{i=0}^{\infty}(\gamma P^\pi)^i r^\pi. \tag{1}$$

It is also well known that the optimal value v_* satisfies the following Bellman equation:

$$v_* = \max_\pi(r^\pi + \gamma P^\pi v_*) = \max_\pi T^\pi v_*$$

where the max operator is component-wise. In other words, v_* is a fixed point of the nonlinear operator $Tv := \max_\pi T^\pi v$. For any value vector v, we call a **greedy policy with respect to the value** v a policy π that satisfies:

$$\pi \in \arg\max_{\pi'} T^{\pi'} v$$

or equivalently $T^\pi v = Tv$. We write, with some abuse of notation[5] greedy(v) any policy that is greedy with respect to v. The notions of optimal value function and greedy policies are fundamental to optimal control because of the following property: any policy π_* that is greedy with respect to the optimal value is an **optimal policy** and its value v^{π_*} is equal to v_*.

The operators T^π and T are γ-contraction mappings with respect to the **max norm** $\|.\|_\infty$ (Puterman, 1994) defined as follows for all vector u:

$$\|u\|_\infty := \max_x |u(x)|.$$

In what follows, we only describe what this means for T but the same holds for T^π. Being a γ-contraction mapping for the max norm means that for all pairs of vectors (v,w),

$$\|Tv - Tw\|_\infty \le \gamma\|v-w\|_\infty.$$

4. We will consider the undiscounted situation ($\gamma = 1$) in Section 4.4, and introduce appropriate related assumptions there.

5. There might be several policies that are greedy with respect to some value v.

Algorithm 1 `Value iteration`

Input: An MDP, an initial value v_0
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $v_{k+1} \leftarrow Tv_k$ // Update the value
    $k \leftarrow k+1$
  until some stopping criterion
  Return greedy(v_k)

This ensures that the fixed point v_* of T exists and is unique. Furthermore, for any initial vector v_0,

$$\lim_{k\to\infty} T^k v_0 = v_*. \tag{2}$$

Given an MDP, standard algorithmic solutions for computing an optimal value-policy pair are value and policy iteration (Puterman, 1994). The rest of this section describes both algorithms with some of the relevant properties for the subject of this paper.

The **value iteration** algorithm for computing the value of a policy π and the value of the optimal policy π_* rely on Equation 2. Algorithm 1 provides a description of value iteration for computing an optimal policy (replace T by T^π in it and one gets value iteration for computing the value of some policy π). The contraction property induces some interesting properties for value iteration. Not only does it ensure convergence, but it also implies a linear rate of convergence of the value v_k to v_*: for all $k \geq 0$,

$$\|v_* - v_k\|_\infty \leq \gamma^k \|v_* - v_0\|_\infty .$$

It is possible to derive a performance bound, that is a bound on the difference between the real value of a policy produced by the algorithm and the value of the optimal policy π_* by using the following well-known property (Puterman, 1994): For all v, if $\pi =$greedy(v) then

$$\|v_* - v^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} \|v_* - v\|_\infty .$$

Let π_k denote the policy that is greedy with respect to v_{k-1}. Then,

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma} \|v_* - v_0\|_\infty . \tag{3}$$

Policy iteration is an alternative method for computing an optimal policy for an infinite-horizon discounted Markov decision process. This algorithm is based on the following property: if π is some policy, then any policy π' that is greedy with respect to the value of π, that is any π' satisfying $\pi' =$ greedy(v^π), is better than π in the sense that $v^{\pi'} \geq v^\pi$. Policy iteration exploits this property in order to generate a sequence of policies with increasing values. It is described in Algorithm 2. Note that we use the analytical form of the value of a policy given by Equation 1. When the state space and the action space are finite, policy iteration converges to an optimal policy π_* in a finite number of iterations (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). In infinite state spaces, if the function $v \mapsto P^{\text{greedy}(v)}$ is Lipschitz, then it can be shown that policy iteration has a quadratic convergence rate (Puterman, 1994).

Algorithm 2 `Policy iteration`

Input: An MDP, an initial policy π_0
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $v_k \leftarrow (I - \gamma P^{\pi_k})^{-1} r^{\pi_k}$ // Estimate the value of π_k
    $\pi_{k+1} \leftarrow \text{greedy}(v_k)$ // Update the policy
    $k \leftarrow k+1$
  until some stopping criterion
  Return π_k

3. The λ Policy Iteration Algorithm

In this section, we describe the family of algorithms that is the main topic of this paper, "λ policy iteration,"[6] originally introduced by Bertsekas and Ioffe (1996). λ policy iteration is parametrized by a coefficient $\lambda \in (0,1)$ and generalizes value and policy iteration. When $\lambda = 0$, λ policy iteration reduces to value iteration while it reduces to policy iteration when $\lambda = 1$. We also recall the fact discussed by Bertsekas and Ioffe (1996) that λ policy iteration draws some connections with temporal-difference algorithms (Sutton and Barto, 1998).

We begin by giving some intuition about how one can make a connection between value and policy iteration. At first sight, value iteration builds a sequence of value functions and policy iteration a sequence of policies. In fact, both algorithms can be seen as updating a sequence of value-policy pairs. With some little rewriting—by decomposing the (nonlinear) Bellman operator T into *(i)* the maximization step and *(ii)* the application of the (linear) Bellman operator—it can be seen that each iterate of value iteration is equivalent to the two following updates:

$$\left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & T^{\pi_{k+1}} v_k \end{array} \right. \Leftrightarrow \left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & r^{\pi_{k+1}} + \gamma P^{\pi_{k+1}} v_k. \end{array} \right.$$

The left hand side of the above equation uses the operator $T^{\pi_{k+1}}$ while the right hand side uses its definition. Similarly—by inverting in Algorithm 2 the order of *(i)* the estimation of the value of the current policy and *(ii)* the update of the policy, and by using the fact that the value of the policy π_{k+1} is the fixed point of $T^{\pi_{k+1}}$ (Equation 2)—it can be argued that every iteration of policy iteration does the following:

$$\left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (T^{\pi_{k+1}})^{\infty} v_k \end{array} \right. \Leftrightarrow \left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (I - \gamma P^{\pi_{k+1}})^{-1} r^{\pi_{k+1}}. \end{array} \right.$$

This rewriting makes both algorithms look close to each other. Both can be seen as having an estimate v_k of the value of policy π_k, from which they deduce a potentially better policy π_{k+1}. The corresponding value $v^{\pi_{k+1}}$ of this better policy may be regarded as a target which is tracked by the next estimate v_{k+1}. The difference is in the update that enables to go from v_k to v_{k+1}: while policy iteration directly *jumps to* the value of π_{k+1} (by applying the Bellman operator $T^{\pi_{k+1}}$ an infinite number of times), value iteration only *makes one step* towards it (by applying $T^{\pi_{k+1}}$ only once).

6. It was also called "temporal-difference based policy iteration" in the original paper, but we take the name λ policy iteration, as it was the name picked by most subsequent works.

From this common view of value iteration, it is natural to introduce the well-known modified policy iteration algorithm (Puterman and Shin, 1978) which *makes n steps* at each update:

$$\left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (T^{\pi_{k+1}})^n v_k\end{array}\right. \Leftrightarrow \left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & \left[I + ... + (\gamma P^{\pi_{k+1}})^{n-1}\right] r^{\pi_{k+1}} + (\gamma P^{\pi_{k+1}})^n v_k.\end{array}\right.$$

The above common view is actually here interesting because it also leads to a natural introduction

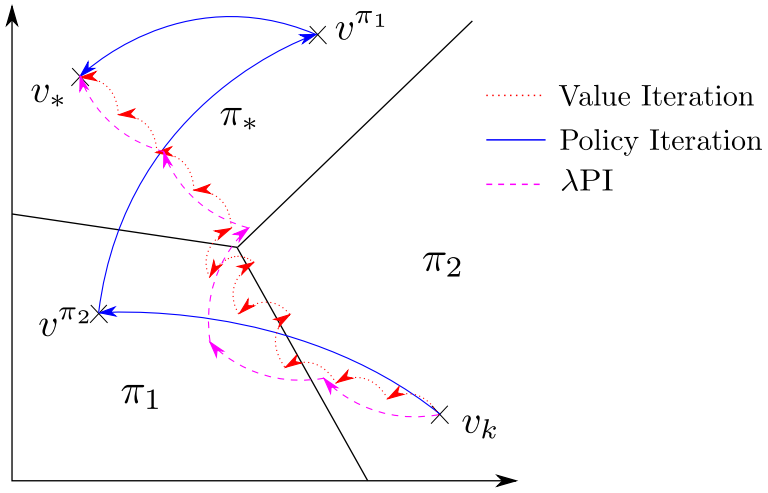


Figure 1: *Visualizing λ policy iteration in the greedy partition.* Following Bertsekas and Tsitsiklis (1996, p. 226), one can decompose the value space as a collection of polyhedra, such that each polyhedron corresponds to a region where one policy is greedy. This is called the *greedy partition*. In the above example, there are only 3 policies, π_1, π_2 and π_*. v_k is the initial value. $\text{greedy}(v_k) = \pi_2$, $\text{greedy}(v^{\pi_2}) = \pi_1$, and $\text{greedy}(v^{\pi_1}) = \pi_*$. Therefore policy iteration (or "1 policy iteration") generates the sequence $((\pi_2, v^{\pi_2}), (\pi_1, v^{\pi_1}), (\pi_*, v^{\pi_*}))$. Value iteration (or "0 policy iteration") starts by slowly updating v_k towards v^{π_2} until it crosses the boundary π_1/π_2, after which it tracks alternatively v^{π_1} and v^{π_2}, until it reaches the π_* part. In other words, value iteration makes small steps. λ policy iteration is doing something intermediate: it makes steps of which the length is controlled by λ.

of λ policy iteration. λ policy iteration is doing a λ-*adjustable step* towards the value of π_{k+1}:

$$\left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (1-\lambda)\sum_{j=0}^{\infty} \lambda^j (T^{\pi_{k+1}})^{j+1} v_k\end{array}\right.$$

$$\Leftrightarrow \left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (I - \lambda\gamma P^{\pi_{k+1}})^{-1}(r^{\pi_{k+1}} + (1-\lambda)\gamma P^{\pi_{k+1}} v_k).\end{array}\right.$$

The equivalence between the left and the right representation of λ policy iteration needs here to be proved. For all $k \geq 0$ and all function v, Bertsekas and Ioffe (1996) introduce the following operator[7]

$$M_k v := (1-\lambda) T^{\pi_{k+1}} v_k + \lambda T^{\pi_{k+1}} v \quad (4)$$

$$= r^{\pi_{k+1}} + (1-\lambda)\gamma P^{\pi_{k+1}} v_k + \lambda\gamma P^{\pi_{k+1}} v \quad (5)$$

7. The equivalence between Equations 4 and 5 follows trivially from the definition of $T^{\pi_{k+1}}$.

Algorithm 3 λ `policy iteration`

Input: An MDP, $\lambda \in (0,1)$, an initial value v_0
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $\pi_{k+1} \leftarrow \text{greedy}(v_k)$ // Update the policy
    $v_{k+1} \leftarrow T_\lambda^{\pi_{k+1}} v_k + \varepsilon_{k+1}$ // Update the estimate of the value of policy π_{k+1}
    $k \leftarrow k+1$
  until some convergence criterion
  Return greedy(v_k)

and prove that

- M_k is a contraction mapping of modulus $\lambda\gamma$ for the max norm ;
- The next iterate v_{k+1} of λ policy iteration is the (unique) fixed point of M_k.

The left representation of λ policy iteration is obtained by "unrolling" Equation 4 an infinite number of times, while the right one is obtained by using Equation 5 and solving the linear system $v_{k+1} = M_k v_{k+1}$.

As illustrated in figure 1, the parameter λ (or n in the case of modified policy iteration) can informally be seen as adjusting the size of the step for tracking the target $v^{\pi_{k+1}}$: the bigger the value, the longer the step. Formally, λ policy iteration (consider the above left hand side) consists in doing a geometric average of parameter λ of the terms $(T^{\pi_{k+1}})^j v_k$ for all values of j. The right hand side is here interesting because it clearly shows that λ policy iteration generalizes value iteration (when $\lambda = 0$) and policy iteration (when $\lambda = 1$). The operator M_k gives some insight on how one may concretely implement one iteration of λ policy iteration: it can for instance be done through a value-iteration like algorithm which applies M_k iteratively. Then, the fact that its contraction factor is $\lambda\gamma$ is interesting: when $\lambda < 1$, finding the corresponding fixed point can generally be done in fewer iterations than that of $T^{\pi_{k+1}}$, which is only γ-contracting.

In order to fully describe the λ policy iteration algorithm, we introduce an operator that corresponds to the computation of the fixed point of M_k. For any value v and any policy π, define:

$$\begin{aligned} T_\lambda^\pi v &:= v + (I - \lambda\gamma P^\pi)^{-1}(T^\pi v - v) && (6)\\ &= (I - \lambda\gamma P^\pi)^{-1}(v - \lambda\gamma P^\pi v + T^\pi v - v) \\ &= (I - \lambda\gamma P^\pi)^{-1}(r^\pi + (1-\lambda)\gamma P^\pi v) && (7)\\ &= (I - \lambda\gamma P^\pi)^{-1}(\lambda r^\pi + (1-\lambda) T^\pi v), \end{aligned}$$

where the different equalities are due to basic algebra and the fact that $T^\pi v = r^\pi + \gamma P^\pi v$.

λ policy iteration is formally described in Algorithm 3. Our description includes a potential error term ε_k when updating the value, which stands for several possible sources of error at each iteration: this error might be the computer round off, the fact that we use an approximate architecture for representing v, a stochastic approximation of P^{π_k}, etc... or a combination of these. It is

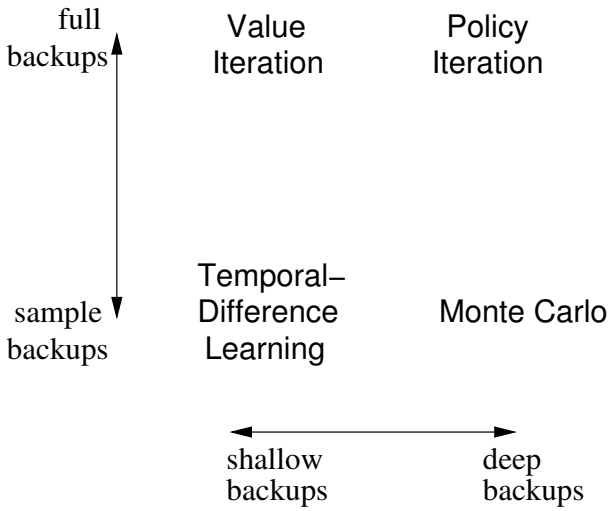


Figure 2: λ *policy iteration, a fundamental algorithm for reinforcement learning.* We represent a picture of the family of algorithms corresponding to λ policy iteration. The vertical axis corresponds to whether one does full backup (exact computation of the expectations) or stochastic approximation (estimation through samples). The horizontal axis corresponds to the depth of the backups, and is controlled by the parameter λ. This drawing is reminiscent of the picture that appears in chapter 10.1 of the textbook by Sutton and Barto (1998) that represents *"two of the most important dimensions"* of reinforcement-learning methods along the same dimensions. In that drawing, from top to bottom and left to right, the authors labeled the corners "Dynamic Programming", "Exhaustive search", "Temporal-Difference learning" and "Monte-Carlo". It is interesting to notice that Sutton and Barto (1998) comment their drawing as follows: *"At three of the four corners of the space are the three primary methods for estimating values: DP, TD, and Monte Carlo"*. They do not recognize the fourth corner as one of the reinforcement-learning *primary methods*. Our representation of λ policy iteration actually suggests that in place of "Exhaustive search", policy iteration, which consists in computing the value of the current policy, is the *deepest backup method*, and can be considered as the batch version of Monte Carlo.

straightforward to see that the λ policy iteration reduces to value iteration (Algorithm 1) when $\lambda = 0$ and to policy iteration[8] (Algorithm 2) when $\lambda = 1$.

The definition of the operator T_λ^π given by Equation 7 is the form we have used for the introduction of λ policy iteration as an intermediate algorithm between value and policy iteration. The equivalent form given by Equation 6 can be used to make a connection with the TD(λ) algorithm[9]

8. Policy iteration starts with an initial policy while λ policy iteration starts with some initial value. To be precise, "1 policy iteration" starting with v_0 is equivalent to policy iteration starting with the greedy policy with respect to v_0.
9. TD stands for temporal difference. As we have mentioned in Footnote 6, λ policy iteration was originally also called "temporal-difference based policy iteration" and the presentation of Bertsekas and Ioffe (1996) starts from the formulation of Equation 6 (which is close to TD(λ)), and afterwards makes the connection with value and policy iteration.

(Sutton and Barto, 1998). Indeed, through Equation 6, the evaluation phase of λ policy iteration can be seen as an incremental additive procedure:

$$v_{k+1} \leftarrow v_k + \Delta_k$$

where

$$\Delta_k := (I - \lambda\gamma P^{\pi_{k+1}})^{-1}(T^{\pi_{k+1}} v_k - v_k)$$

is zero if and only if the value v_k is equal to the optimal value v_*. It can be shown (Bertsekas and Ioffe, 1996) that the vector Δ_k has components given by:

$$\Delta_k(i) = E_{\pi_{k+1}}\left[\sum_{t=0}^{\infty}(\lambda\gamma)^t \delta_k(i_t, i_{t+1}) \,\middle|\, i_0 = i\right] \tag{8}$$

with

$$\delta_k(i,j) := r(i, \pi_{k+1}(i), j) + \gamma v(j) - v(i)$$

being the temporal difference associated to transition $i \to j$, as defined by Sutton and Barto (1998). When one uses a stochastic approximation of λ policy iteration, that is when the expectation $E_{\pi_{t+1}}$ is approximated by sampling, λ policy iteration reduces to the algorithm TD(λ) which is described in chapter 7 of Sutton and Barto (1998). In particular, when $\lambda = 1$, the terms in the above sum collapse and become the exact discounted return:

$$\begin{aligned}\sum_{j=0}^{\infty}\gamma^j \delta_k(i_j, i_{j+1}) &= \sum_{j=0}^{\infty}\gamma^j \left[r(i_j, \pi_{k+1}(i_j), i_{j+1}) + \gamma v(i_{j+1}) - v(i_j)\right] \\ &= \sum_{j=0}^{\infty}\gamma^j r(i_j, \pi_{k+1}(i_j), i_{j+1})\end{aligned}$$

and the stochastic approximation matches the Monte-Carlo method. Also, Bertsekas and Ioffe (1996) show that approximate TD(λ) with a linear feature architecture, as described in chapter 8.2 of Sutton and Barto (1998), corresponds to a natural approximate version of λ policy iteration where the value is updated by least squares fitting using a gradient-type iteration after each sample. Last but not least, as illustrated in figure 2, the reader might notice that the "unified view" of reinforcement-learning algorithms which is depicted in chapter 10.1 of Sutton and Barto (1998) is in fact a picture of λ policy iteration.

To our knowledge, little has been done concerning the analysis of λ policy iteration: the only results available concern the exact case (when $\varepsilon_k = 0$). Define the following factor

$$\beta = \frac{(1-\lambda)\gamma}{1-\lambda\gamma}. \tag{9}$$

We have $0 \leq \beta \leq \gamma < 1$. If $\lambda = 0$ (value iteration) then $\beta = \gamma$, and if $\lambda = 1$ (policy iteration) then $\beta = 0$. In the original article introducing λ policy iteration, Bertsekas and Ioffe (1996) show the convergence and provide the following asymptotic rate of convergence.

Proposition 1 (Convergence of λPI, Bertsekas and Ioffe, 1996)
The sequence v_k converges to v_*. Furthermore, after some index k_*, the rate of convergence is linear in β as defined in Equation 9, that is

$$\forall k \geq k_*, \;\; \|v_{k+1} - v_*\| \leq \beta \|v_k - v_*\|.$$

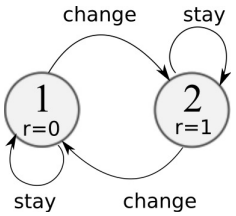


Figure 3: This simple deterministic MDP is used to show that λ policy iteration cannot be analyzed in terms of contraction (see text for details).

By making λ close to 1, β can be arbitrarily close to 0 so the above rate of convergence might look overly impressive. This needs to be put into perspective: the index k_* is the index after which the policy π_k does not change anymore (and is equal to the optimal policy π_*). As we said when we introduced the algorithm, λ controls the speed at which one wants v_k to "track the target" $v^{\pi_{k+1}}$; when $\lambda = 1$, this is done in one step (and if $\pi_{k+1} = \pi_*$ then $v_{k+1} = v_*$).

4. Analysis Of λ Policy Iteration

λ policy iteration is conceptually nice since it generalizes the two most well-known algorithms for solving Markov decision processes. In the literature, lines of analysis are different for value and policy iteration. Analyzes of value iteration are based on the fact that it computes the fixed point of the Bellman operator which is a γ-contraction mapping in max norm (Bertsekas and Tsitsiklis, 1996). Unfortunately, it can be shown that the operator by which policy iteration updates the value from one iteration to the next is in general not a contraction in max norm. In fact, this observation can be drawn for λ policy iteration as soon as it does not reduce to value iteration:

Proposition 2 *If $\lambda > 0$, there exists no norm for which the operator $v \mapsto T_\lambda^{greedy(v)} v$ by which λ policy iteration updates the value from one iteration to the next is a contraction.*

Proof To see this, consider the deterministic MDP (shown in figure 3) with two states $\{1,2\}$ and two actions $\{change, stay\}$. The instantaneous rewards of being in state 1 and 2 are respectively $r_1 = 0$ and $r_2 = 1$ (they do not depend on the action nor the resulting state), and the transitions are characterized as follows: $P_{change}(2|1) = P_{change}(1|2) = P_{stay}(1|1) = P_{stay}(2|2) = 1$. Consider the following two value functions $v = (\varepsilon, 0)$ and $v' = (0, \varepsilon)$ with $\varepsilon > 0$. Their corresponding greedy policies are $\pi = (stay, change)$ and $\pi' = (change, stay)$. Then, we can compute the next iterates of v

and v' (using Equation 7):

$$\begin{aligned} r^{\pi} + (1-\lambda\gamma)P^{\pi}v &= \begin{pmatrix} (1-\lambda)\gamma\varepsilon \\ 1+(1-\lambda)\gamma\varepsilon \end{pmatrix}, \\ T_{\lambda}^{\pi}v &= \begin{pmatrix} \frac{(1-\lambda)\gamma\varepsilon}{1-\lambda\gamma} \\ 1+\frac{(1-\lambda)\gamma\varepsilon}{1-\lambda\gamma} \end{pmatrix}, \\ r^{\pi'} + (1-\lambda\gamma)P^{\pi'}v' &= \begin{pmatrix} (1-\lambda)\gamma\varepsilon \\ 1+(1-\lambda)\gamma\varepsilon \end{pmatrix}, \\ \text{and } T_{\lambda}^{\pi'}v' &= \begin{pmatrix} \frac{1+(1-\lambda)\gamma\varepsilon}{1-\lambda\gamma} - 1 \\ \frac{1+(1-\lambda)\gamma\varepsilon}{1-\lambda\gamma} \end{pmatrix}. \end{aligned}$$

Then

$$T_{\lambda}^{\pi'}v' - T_{\lambda}^{\pi}v = \begin{pmatrix} \frac{1}{1-\lambda\gamma} - 1 \\ \frac{1}{1-\lambda\gamma} - 1 \end{pmatrix}$$

while

$$v' - v = \begin{pmatrix} -\varepsilon \\ \varepsilon \end{pmatrix}.$$

As ε can be arbitrarily small, the norm of $T_{\lambda}^{\pi}v - T_{\lambda}^{\pi'}v'$ can be arbitrarily larger than that of $v - v'$ when $\lambda > 0$. ■

Analyzes of policy iteration usually rely on the fact that the sequence of values generated is non-decreasing (Bertsekas and Tsitsiklis, 1996; Munos, 2003). Unfortunately, it can easily be seen that as soon as λ is smaller than 1, the value functions may decrease (it suffices to take a very high initial value). For non trivial values of λ, λ policy iteration is neither contracting nor non-decreasing, so we need a new proof technique.

4.1 Main Proof Ideas

The rest of this section provides an overview of our analysis. We show how to compute an upper bound of the loss for λ policy iteration in the general (possibly approximate) case. It is the basis for the derivation of component-wise bounds for exact λ policy iteration (Section 4.2) and approximate λ policy iteration (Section 4.3). Consider λ policy iteration as described in Algorithm 3, and the sequence of value-policy-error triplets $(v_k, \pi_k, \varepsilon_k)$ it generates.

Our goal is to provide a bound of the **loss** of using policy π_k instead of the optimal policy:

$$l_k := v_* - v^{\pi_k}.$$

Our analysis amounts to decompose the loss as follows:

$$v_* - v^{\pi_k} = \underbrace{v_* - w_k}_{d_k} + \underbrace{w_k - v^{\pi_k}}_{s_k},$$

where w_k is the **value** of the k^{th} before the approximation ε_k is incurred:

$$w_k := v_k - \varepsilon_k = T_{\lambda}^{\pi_k} v_{k-1}.$$

We shall call the term $d_k = v_* - w_k$ the **distance** as it is a measure of distance between the optimal value and the k^{th} value w_k. Similarly, we shall call the term $s_k = w_k - v^{\pi_k}$ the **shift** as it shows the shift between the k^{th} value w_k and the value of the k^{th} policy (as mentioned before, the former can indeed be understood as tracking the latter). As it will appear shortly, we will be able to bound both quantities, and thus deduce a bound on the loss. Actual bounds on d_k and s_k will be based on a bound on the **Bellman residual** of the k^{th} value:

$$b_k := T_{k+1}v_k - v_k = Tv_k - v_k.$$

To lighten the notations, from now on we write: $P_k := P^{\pi_k}$, $T_k := T^{\pi_k}$, $P_* := P^{\pi_*}$. We refer to the factor β as introduced by Bertsekas and Ioffe (Equation 9 page 1189). Also, the following stochastic matrix plays a recurrent role in our analysis:[10]

$$A_k := (1-\lambda\gamma)(I-\lambda\gamma P_k)^{-1}P_k.$$

For a vector u, we use the notation $\overline{u}$ for an upper bound of u and $\underline{u}$ for a lower bound.

Our analysis relies on a series of lemmas that we now state (for clarity, all the proofs are deferred to appendix B).

Lemma 3 *The shift is related to the Bellman residual as follows:*

$$s_k = \beta(I-\gamma P_k)^{-1}A_k(-b_{k-1}).$$

Lemma 4 *The Bellman residual at iteration $k+1$ cannot be much lower than that at iteration k:*

$$b_{k+1} \geq \beta A_{k+1}b_k + x_{k+1}$$

where $x_k := (\gamma P_k - I)\varepsilon_k$ only depends on the approximation error.

As a consequence, a lower bound of the Bellman residual is:[11]

$$b_k \geq \sum_{j=1}^{k} \beta^{k-j}\left(A_kA_{k-1}...A_{j+1}\right)x_j + \beta^k\left(A_kA_{k-1}...A_1\right)b_0 := \underline{b_k}.$$

Using Lemma 3, the bound on the Bellman residual also provides an upper bound on the shift:[12]

$$s_k \leq \beta(I-\gamma P_k)^{-1}A_k(-\underline{b_{k-1}}) := \overline{s_k}.$$

Lemma 5 *The distance at iteration $k+1$ cannot be much greater than that at iteration k:*

$$d_{k+1} \leq \gamma P_* d_k + y_k$$

where $y_k := \frac{\lambda\gamma}{1-\lambda\gamma}A_{k+1}(-\underline{b_k}) - \gamma P_*\varepsilon_k$ depends on the lower bound of the Bellman residual and the approximation error.

10. The fact that this is indeed a stochastic matrix is explained at the beginning of the appendices.
11. We use the property here that if some vectors satisfy the component-wise inequality $x \leq y$, and if P is a stochastic matrix, then the component-wise inequality $Px \leq Py$ holds.
12. We use the fact that $(1-\gamma)(I-\gamma P_k)^{-1}$ is a stochastic matrix (see Footnote 10) and Footnote 11.

Then, an upper bound of the distance is:[13]

$$d_k \leq \sum_{j=0}^{k-1} \gamma^{k-1-j}(P_*)^{k-1-j} y_j + \gamma^k (P_*)^k d_0 = \overline{d_k}.$$

Eventually, as

$$l_k = d_k + s_k \leq \overline{d_k} + \overline{s_k},$$

the upper bounds on the distance and the shift enable us to derive the upper bound on the loss.

The above derivation is a generalization of that of Munos (2003) for approximate policy iteration. Note however that it is not a trivial generalization: when $\lambda = 1$, that is when both proofs coincide, $\beta = 0$ and Lemmas 3 and 4 have the following particularly simple form: $s_k = 0$ and $b_{k+1} \geq x_{k+1}$.

The next two subsections contain our main results, which take the form of performance bounds when using λ policy iteration. Section 4.2 gathers the results concerning exact λ policy iteration, while Section 4.3 presents those concerning approximate λ policy iteration.

4.2 Performance Bounds For Exact λ Policy Iteration

Consider exact λ policy iteration for which we have $\varepsilon_k = 0$ for all k. By exploiting the recursive relations we have described in the previous section (this process is detailed in appendix C), we can derive the following component-wise bounds for the loss.

Lemma 6 (Component-Wise rate of convergence of exact λPI)
For all $k > 0$, the following matrices

$$\begin{aligned} E_k &:= (1-\gamma)(P_*)^k (I - \gamma P_*)^{-1}, \\ E'_k &:= \left(\frac{1-\gamma}{\gamma^k}\right) \left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j (P_*)^{k-1-j} A_{j+1} A_j \ldots A_1 \right. \\ &\qquad \left. + \beta^k (I - \gamma P_k)^{-1} A_k A_{k-1} \ldots A_1 \right), \\ \text{and } F_k &:= (1-\gamma) P_*^k + \gamma E'_k P_* \end{aligned}$$

are stochastic and the performance of the policies generated by λ policy iteration satisfies

$$v_* - v^{\pi_k} \leq \frac{\gamma^k}{1-\gamma} \left[F_k - E'_k \right] (v_* - v_0), \tag{10}$$

$$v_* - v^{\pi_k} \leq \frac{\gamma^k}{1-\gamma} \left[E_k - E'_k \right] (Tv_0 - v_0), \text{ and} \tag{11}$$

$$v_* - v^{\pi_k} \leq \gamma^k \left[(P_*)^k \left((v_* - v_0) - \min_s [v_*(s) - v_0(s)] e \right) + \| v_* - v^{\pi_1} \|_\infty e \right] \tag{12}$$

where e is the vector of which all components are 1.

In order to derive (more interpretable) max norm bounds from the above component-wise bound, we rely on the following lemma, which for clarity of exposition is proved in appendix G.

13. See Footnote 11.

Lemma 7 *If for some non-negative vectors x and y, some constant $K \geq 0$, and some stochastic matrices X and X' we have*

$$x \leq K(X - X')y,$$

then

$$\|x\|_\infty \leq 2K \|y\|_\infty .$$

With this, the component-wise bounds of Lemma 6 become:

Proposition 8 (Non-asymptotic bounds for exact λ policy iteration)
For any $k > 0$,

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma} \|v_* - v_0\|_\infty , \tag{13}$$

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma} \|Tv_0 - v_0\|_\infty , \tag{14}$$

$$\textit{and} \quad \|v_* - v^{\pi_k}\|_\infty \leq \gamma^k \left(2\|v_* - v_0\|_\infty + \|v_* - v^{\pi_1}\|_\infty\right). \tag{15}$$

These *non-asymptotic* bounds supplement the *asymptotic* bound of Proposition 1 from Bertsekas and Ioffe (1996). Remarkably, these max-norm bounds show no dependence on the value λ. The bound of Equation 13 is expressed in terms of the initial distance between the value function and the optimal value function, and constitutes a generalization of the rate of convergence of value iteration by Puterman (1994) that we described in Equation 3 page 1184. The second inequality, Equation 14, is expressed in terms of the initial Bellman residual and is also well-known for value iteration (Puterman, 1994). The last inequality described in Equation 15 relies on the distance between the value function and the optimal value function and the value difference between the optimal policy and the first greedy policy; compared to the others, it has the advantage of not containing a $\frac{1}{1-\gamma}$ factor. To our knowledge, this bound is even new for the specific cases of value and policy iteration.

4.3 Performance Bounds For Approximate λ Policy Iteration

We now turn to the (slightly more involved) results on approximate λ policy iteration. We provide component-wise bounds of the loss $l_k = v_* - v^{\pi_k} \geq 0$ of using policy π_k instead of using the optimal policy, with respect to the approximation error ε_k, the policy Bellman residual $T_k v_k - v_k$ and the Bellman residual $Tv_k - v_k = T_{k+1}v_k - v_k$. Note the subtle difference between the two Bellman residuals: the policy Bellman residual says how much v_k differs from the value of π_k while the Bellman residual says how much v_k differs from the value of the policies π_{k+1} and π_*.

The core of our analysis, and the main contribution of this article, is described in the following lemma.

Lemma 9 (Component-Wise performance bounds for app. λ policy iteration)
For all $k \geq j \geq 0$, the following matrices

$$\begin{aligned}
B_{kj} &:= \frac{1-\gamma}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}(P_*)^{k-1-i}A_{i+1}A_i...A_{j+1}\right. \\
&\qquad\qquad \left. + \beta^{k-j}(I-\gamma P_k)^{-1}A_kA_{k-1}...A_{j+1}\right], \\
B'_{kj} &:= \gamma B_{kj}P_j + (1-\gamma)(P_*)^{k-j}, \\
C_{kj} &:= (1-\gamma)(P_*)^{k-j}(I-\gamma P_j)^{-1}, \\
C'_{kj} &:= (1-\gamma)(P_*)^{k-j-1}P_{j+1}(I-\gamma P_{j+1})^{-1}, \\
D &:= (1-\gamma)P_*(I-\gamma P_*)^{-1} \\
\text{and}\quad D'_k &:= (1-\gamma)P_k(I-\gamma P_k)^{-1}
\end{aligned}$$

are stochastic and for all k,

$$v_* - v^{\pi_k} \leq \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[B_{kj} - B'_{kj}\right]\varepsilon_j + O(\gamma^k), \tag{16}$$

$$v_* - v^{\pi_k} \leq \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[C_{kj} - C'_{kj}\right](T_jv_j - v_j) + O(\gamma^k), \tag{17}$$

$$\text{and}\quad v_* - v^{\pi_k} \leq \frac{\gamma}{1-\gamma}\left[D - D'_k\right](Tv_{k-1} - v_{k-1}). \tag{18}$$

The first relation (Equation 16) involves the errors (ε_k), is based on Lemmas 3-5 (presented in Section 4.1) and is proved in appendix D. The two other inequalities (the asymptotic performance of approximate λ policy iteration with respect to the Bellman residuals in Equations 17 and 18) are somewhat simpler and are proved independently in appendix E.

By taking the max norm in the above component-wise performance bounds, we obtain, for all k,

$$\begin{aligned}
\|v_* - v^{\pi_k}\|_\infty &\leq \frac{2}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left\|\varepsilon_j\right\|_\infty + O(\gamma^k), \\
\|v_* - v^{\pi_k}\|_\infty &\leq \frac{2}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left\|T_jv_j - v_j\right\|_\infty + O(\gamma^k), \\
\text{and}\quad \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{1-\gamma}\left\|Tv_{k-1} - v_{k-1}\right\|_\infty.
\end{aligned} \tag{19}$$

In the specific context of value and policy iteration, Munos (2003, 2007) has argued that most supervised learning algorithms (such as least squares regression) that are used in practice for approximating each iterate control the errors (ε_k) for some weighted L_p norm $\|\cdot\|_{p,\mu}$, defined for some distribution μ on the state space X as follows:

$$\|u\|_{\mu,p} = \left(\sum_x \mu(x)|u(x)|^p\right)^{1/p}.$$

As a consequence, Munos (2007, 2003) explained how to derive an analogue of the above result where the approximation error ε_k is expressed in terms of this L_p norm. Based on Munos' works, we provide below a useful technical lemma (proved in appendix G) that shows how the performance of approximate λ policy iteration can be translated into L_p norm bounds.

Lemma 10 *Let x_k, y_k be vectors and X_{kj}, X'_{kj} stochastic matrices satisfying for all k*

$$|x_k| \leq K \sum_{j=0}^{k-1} \xi_{k-j} (X_{kj} - X'_{kj}) y_j + O(\gamma^k),$$

where $(\xi_i)_{i \geq 1}$ is a sequence of non-negative weights satisfying

$$\sum_{i=1}^{\infty} \xi_i = K' < \infty.$$

Then, for all distribution μ,

$$\mu_{kj} := \frac{1}{2}(X_{kj} + X'_{kj})^T \mu$$

are distributions and

$$\limsup_{k \to \infty} \|x_k\|_{p,\mu} \leq 2KK' \lim_{l \to \infty} \left[\sup_{k \geq j \geq l} \|y_j\|_{p,\mu_{kj}} \right].$$

Thus, using this lemma and the fact that $\sum_{i=1}^{\infty} \gamma^i = \frac{\gamma}{1-\gamma}$, Lemma 9 can be turned into the following proposition.

Proposition 11 (L_p norm performance of approximate λPI)
With the notations of Lemma 9, for all p, $k \geq j \geq 0$ and all distribution μ,

$$\begin{aligned} \mu_{kj} &:= \frac{1}{2}\left(B_{kj} + B'_{kj}\right)^T \mu, \\ \mu'_{kj} &:= \frac{1}{2}\left(C_{kj} + C'_{kj}\right)^T \mu \\ \text{and} \quad \mu''_k &:= \frac{1}{2}\left(D + D'_k\right)^T \mu \end{aligned}$$

are distributions and the performance of the policies generated by λ policy iteration satisfies:

$$\begin{aligned} \limsup_{k \to \infty} \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{(1-\gamma)^2} \lim_{l \to \infty} \left[\sup_{k \geq j \geq l} \|\varepsilon_j\|_{p,\mu_{kj}} \right], \\ \limsup_{k \to \infty} \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{(1-\gamma)^2} \lim_{l \to \infty} \left[\sup_{k \geq j \geq l} \|T_j v_j - v_j\|_{p,\mu'_{kj}} \right], \\ \forall k, \quad \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{1-\gamma} \|T v_{k-1} - v_{k-1}\|_{p,\mu''_k}. \end{aligned}$$

Proposition 11 means that in order to control the performance loss (the left hand side) for some μ-weighted L_p norm, one needs to control the errors ε_j, the policy Bellman residual $T_j j_k - v_j$ or the Bellman residual $T v_{k-1} - v_{k-1}$ (the right hand sides) respectively for the norms μ_{kj}, μ'_{kj} and μ''_k.

Unfortunately, these distributions depend on unknown quantities (such as the stochastic matrix of the optimal policy, see the definitions in Lemma 9) and cannot be used in practice by the algorithm. To go round this issue, we follow Munos (2003, 2007) and introduce some assumption on the stochasticity of the MDP in terms of a so-called **concentrability coefficient**. Assume there exists a distribution ν and a real number $C(\nu)$ such that

$$C(\nu) := \max_{i,j,a} \frac{p_{ij}(a)}{\nu(j)}. \tag{20}$$

For instance, if one chooses the uniform law ν, then there always exists such a $C(\nu) \in (1,N)$ where N is the size of the state space. More generally, a small value of $C(\nu)$ requires that the underlying MDP has a significant amount of stochasticity; see (Munos, 2003, 2007) for more discussion on this coefficient. Given this definition, we have the following property.

Lemma 12 *Let X be a convex combination of products of stochastic matrices of the MDP. For any distribution μ, vector y, and p,*

$$\|y\|_{p,X^T\mu} \leq (C(\nu))^{1/p} \|y\|_{p,\nu}.$$

Proof It can be seen from the definition of the concentrability coefficient $C(\nu)$ that $\mu^T X \leq C(\nu)\nu^T$. Thus,

$$\begin{aligned}
\left(\|y\|_{p,X^T\mu}\right)^p &= \left(\|y\|_{p,X^T\mu}\right)^p \\
&= \mu^T X |y|^p \\
&\leq C(\nu)\nu^T |y|^p \\
&= C(\nu)\left(\|y\|_{p,\nu}\right)^p \\
&= C(\nu)\left(\|y\|_{p,\nu}\right)^p.
\end{aligned}$$

■

Using this lemma, and the fact that for any p, $\|x\|_\infty = \max_\mu \|x\|_{p,\mu}$, the L_p bounds of Proposition 11 lead to the following proposition.

Proposition 13 (L_∞/L_p norm performance of approximate λPI)
Let $C(\nu)$ be the concentrability coefficient defined in Equation 20. For all p,

$$\begin{aligned}
\limsup_{k\to\infty} \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{(1-\gamma)^2} [C(\nu)]^{1/p} \limsup_{k\to\infty} \|\varepsilon_k\|_{p,\nu}, \\
\limsup_{k\to\infty} \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{(1-\gamma)^2} [C(\nu)]^{1/p} \limsup_{k\to\infty} \|T_k v_k - v_k\|_{p,\nu}, \\
\textit{and } \forall k, \quad \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{1-\gamma} [C(\nu)]^{1/p} \|T v_{k-1} - v_{k-1}\|_{p,\nu}.
\end{aligned}$$

It is, once again, remarkable that these bounds do not explicitly depend on the value of λ. However, it should be clear that, with respect to the previous bounds, the influence of λ is now hidden in the concentrability coefficient $C(\nu)$. Furthermore, as it is the case in TD(λ) methods, and as will be

illustrated in the case study in Section 6, the value of λ will directly influence the errors $\left\|\varepsilon_j\right\|_{p,\nu}$ and the Bellman residual terms $\|T_k v_k - v_k\|_{p,\nu}$ and $\|T v_{k-1} - v_{k-1}\|_{p,\nu}$.

In general, one cannot give the guarantee that approximate λ policy iteration will converge. However, the performance bounds with respect to the approximation error can be improved if we observe empirically that the value or the policy converges. Note that the former condition implies the latter (while the opposite is not true: the policy may converge while the value still oscillates). Indeed, we have the following corollary (proved in appendix F).

Corollary 14 (L_∞/L_p norm performance of app. λPI in case of convergence)
If the value converges to some v, then the approximation error converges to some ε, and the corresponding greedy policy π satisfies

$$\|v_* - v^\pi\|_\infty \quad \leq \quad \frac{2\gamma}{1-\gamma} \left[C(\nu)\right]^{1/p} \|\varepsilon\|_{p,\nu} .$$

If the policy converges to some π, then

$$\|v_* - v^\pi\|_\infty \quad \leq \quad \frac{2\gamma(1-\lambda\gamma)}{(1-\gamma)^2} \left[C(\nu)\right]^{1/p} \limsup_{j\to\infty} \left\|\varepsilon_j\right\|_{p,\nu} .$$

It is interesting to notice that in the latter weaker situation where only the policy converges, the constant decreases from $\frac{1}{(1-\gamma)^2}$ to $\frac{1}{1-\gamma}$ when λ varies from 0 to 1; in other words, the closer to policy iteration, the better the bound in that situation.

4.4 Extension To The Undiscounted Case

The results we have described so far only apply to the situation where the discount factor γ is smaller than 1. Indeed, all our bounds involve terms of the form $\frac{1}{1-\gamma}$ that diverge to infinity as γ tends to 1. In this last subsection, we show how the component-wise analysis of Lemma 9 can be exploited to also cover the case where we have an undiscounted MDP ($\gamma = 1$), as for instance in the the case study on the Tetris domain presented in Section 6.

In undiscounted infinite horizon control problems, it is generally assumed that there exists a $N+1^{\text{th}}$ termination absorbing state 0. Once the system reaches this state, it remains there forever with no further reward, that is formally:

$$\forall a, \quad p_{00}(a) = 1 \text{ and } r(0,a,0) = 0.$$

In order to derive our results, we will introduce conditions that ensure that termination is guaranteed in finite time with probability 1 under any sequence of actions. Formally, we will assume that there exists an integer $n_0 \leq N$ and a real number $\alpha < 1$ such that for all initial distributions μ, all actions $a_0, a_1, ..., a_{n_0-1}$, the following relation

$$P\left[i_{n_0} \neq 0 | i_0 \sim \mu, a_0, ..., a_{n_0-1}\right] \leq \alpha \tag{21}$$

holds.[14] We can think of the MDP as only defined on the N non-terminal states, that is on $\{1, ...N\}$. Then, for any policy π, the matrix P_π is *sub-stochastic*, and the above assumption implies that for

14. In the literature, a stationary policy that reaches the terminal state in finite time with probability 1 is said to be *proper*. The usual assumptions in undiscounted infinite horizon control problems are: *(i)* there exists at least one proper policy and *(ii)* for every improper policy π, the corresponding value equals $-\infty$ for at least one state. The situation we consider here is simpler, since we assume that all (non-necessarily stationary nor deterministic) policies are proper.

all set of n_0 policies $\pi_1, \pi_2, \cdots, \pi_{n_0}$,

$$\left\| P_{\pi_1} P_{\pi_2} \cdots P_{\pi_{n_0}} \right\|_\infty \leq \alpha.$$

The component-wise analysis of λ policy iteration is here identical to what we have done before, except that we have[15] $\gamma = 1$ and $\beta = 1$. The matrix A_k that appeared recurrently in our analysis has the following special form:

$$A_k := (1-\lambda)(I - \lambda P_k)^{-1} P_k.$$

and is a sub-stochastic matrix. The first bound of the component-wise analysis of λ policy iteration (Lemma 9 page 1194) can be generalized as follows (see appendix H for details).

Lemma 15 (Component-Wise bounds in the undiscounted case)
Assume that there exist n_0 and α such that Equation 21 holds. Write $\eta := \frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}$. For all i, write

$$\delta_i := \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \left[\left(\frac{1-\lambda^{n_0}}{1-\lambda} \right) \left(\frac{\lambda}{1-\lambda^{n_0}\alpha} \right) \left(\frac{1-\eta^i}{1-\eta} \right) + \frac{n_0 \eta^i}{1-\alpha} \right].$$

For all $j < k$, the following matrices

$$\begin{aligned} G_{kj} &:= \frac{1}{\delta_{k-j}} \left[\frac{\lambda}{1-\lambda} \sum_{i=j}^{k-1} (P_*)^{k-1-i} A_{i+1} A_i \ldots A_{j+1} + (I - P_k)^{-1} A_k A_{k-1} \ldots A_{j+1} \right] \\ \text{and } G'_{kj} &:= \frac{1}{\delta_{k-j}} G_{kj} P_j \end{aligned}$$

are sub-stochastic and the performance of the policies generated by λ policy iteration satisfies

$$\forall k, \quad v_* - v^{\pi_k} \leq \sum_{j=0}^{k-1} \delta_{k-j} \left[G_{kj} - G'_{kj} \right] \varepsilon_j + O(\gamma^k). \tag{22}$$

By observing that $\eta \in (0,1)$, and that for all $x \in (0,1)$, $0 \leq \frac{1-x^{n_0}}{1-x} \leq n_0$, it can be seen that the coefficients δ_i are finite for all i. Furthermore, when $n_0 = 1$ (which matches the discounted case with $\alpha = \gamma$), one can observe that $\delta_i = \frac{\gamma^i}{1-\gamma}$ and that one recovers the result of Lemma 9.

This lemma can then be exploited to show that λ policy iteration enjoys an L_p norm guarantee. Indeed, an analogue of Proposition 11 (whose proof is detailed in appendix H) is the following proposition.

Proposition 16 (L_p norm bound in the undiscounted case)
Let $C(\nu)$ be the concentrability coefficient defined in Equation 20 page 1197. Let the notations and conditions of Lemma 15 hold. For all distribution μ on $(1, \cdots, N)$ and $k \geq j \geq 0$,

$$\mu_{kj} := \frac{1}{2} \left(G_{kj} + G'_{kj} \right)^T \mu$$

15. For simplicity in our discussion, we consider $\lambda < 1$ to avoid the special case $\lambda = 1$ for which β may be indefinite (see the definition of β in Equation 9 page 1189). The interested reader may however check that the results that we state are continuous in the neighborhood of $\lambda = 1$.

are non-negative vectors and

$$\tilde{\mu}_{kj} := \frac{\mu_{kj}}{\left\|\mu_{kj}\right\|_1}$$

are distributions on $(1,\cdots,N)$. *Then for all p, the loss of the policies generated by* λ *policy iteration satisfies*

$$\limsup_{k\to\infty} \left\|v_* - v^{\pi_k}\right\|_{p,\mu} \leq 2K(\lambda,n_0)(C(\nu))^{1/p} \lim_{j\to\infty} \left\|\varepsilon_j\right\|_{p,\nu}$$

where

$$K(\lambda,n_0) := \lambda f(\lambda)\frac{f(1)-f(\eta)}{1-\eta} + f(1)f(\eta) - f(1),$$
$$\forall x<1,\ f(x) := \frac{(1-x^{n_0})}{(1-x)(1-x^{n_0}\alpha)},\ \textit{and by continuity } f(1) := \frac{n_0}{1-\alpha}.$$

There are two main differences with respect to the results we have presented for the discounted case:

1. The fact that we considered the model (and thus the algorithm) only on the non-terminal states $(1,\cdots,N)$ means that we made the assumption that there is no error incurred in the terminal state 0. Note, however, that this is not a strong assumption since the value of the terminal state is necessarily 0.

2. The constant $K(\lambda,n_0)$ is dependent on λ. More precisely, it can be observed that:

$$\lim_{\lambda\to 0} K(\lambda,n_0) = \lim_{\lambda\to 1} K(\lambda,n_0) = \frac{{n_0}^2}{(1-\alpha)^2} - \frac{n_0}{1-\alpha}$$

   and that this is the minimal value of $\lambda \mapsto K(\lambda,n_0)$. Although we took particular care in deriving this bound, we leave for future work the question whether one could prove a similar result with the constant $\frac{{n_0}^2}{(1-\alpha)^2} - \frac{n_0}{1-\alpha}$ for all $\lambda \in (0,1)$. When $n_0 = 1$ (which matches the discounted case with $\alpha = \gamma$), $K(\lambda,1)$ does not depend anymore on λ and we recover, without surprise, the bound of Proposition 11 since

$$\forall\lambda,\ \ K(\lambda,1) = \frac{\alpha}{(1-\alpha)^2}.$$

5. Related Work

The study of approximate versions of value and policy iteration has been the topic of a rich literature (Bertsekas and Tsitsiklis, 1996), in particular in the discounted case on which we focus in what follows. The most well-known results, due to Bertsekas and Tsitsiklis (1996, pp. 332-333 for value iteration and Prop. 6.2 p. 276 for policy iteration), states that the performance loss due to using the policies π_k instead of the optimal policy π_* satisfies:

$$\limsup_{k\to\infty} \left\|v_* - v^{\pi_k}\right\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \sup_{k\geq 0} \left\|\varepsilon_k\right\|_\infty. \tag{23}$$

As mentioned earlier (after Equation 19 page 1195), Munos (2003, 2007) has argued that the above bound does not directly apply to practical implementations that usually control some L_p norm of the

errors. Munos extended the error analysis of Bertsekas and Tsitsiklis (1996) to this situation. His analysis begins by the following error propagation for value iteration—taken from (Munos, 2007, Lemma 4.1)—and for policy iteration—adapted from[16] (Munos, 2003, Lemma 4).

Lemma 17 (Asymptotic component-wise performance of AVI and API)
For all $k > j \geq 0$, the following matrices

$$\begin{aligned} Q_{kj} &:= (1-\gamma)(I-\gamma P_k)^{-1} P_k P_{k-1} \ldots P_{j+1}, \\ Q'_{kj} &:= (1-\gamma)(I-\gamma P_k)^{-1} (P_*)^{k-j}, \\ R_{kj} &:= (1-\gamma)(P_*)^{k-1-j} P_{j+1} (I-\gamma P_{j+1})^{-1}, \\ R'_{kj} &:= (1-\gamma)(P_*)^{k-1-j} \left(\gamma P_{j+1} (I-\gamma P_{j+1})^{-1} P_j + P_*\right) \\ \text{and} \quad R''_{kj} &:= (1-\gamma)(P_*)^{k-1} (I-\gamma P_j)^{-1} \end{aligned}$$

are stochastic. The asymptotic performance of the policies generated by approximate value iteration satisfies

$$\limsup_{k\to\infty} v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[Q_{kj} - Q'_{kj}\right] \varepsilon_j. \tag{24}$$

The asymptotic performance of the policies generated by approximate policy iteration satisfies

$$\limsup_{k\to\infty} v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[R_{kj} - R'_{kj}\right] \varepsilon_j \tag{25}$$

$$\text{and} \quad \limsup_{k\to\infty} v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[R''_{kj} - R_{kj}\right] (T^{\pi_k} v_k - v_k). \tag{26}$$

Then, introducing the concentrability coefficient $C(\nu)$ (Equation 20 page 1197) and using the techniques that we described through Lemmas 10 and 12, Munos (2003, 2007) turned these component-wise bounds into L_∞/L_p norm bounds that match those of our (more general) Proposition 13. In particular he obtains the following bound for both value and policy iteration,

$$\limsup_{k\to\infty} \|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \left[C(\nu)\right]^{1/p} \limsup_{k\to\infty} \|\varepsilon_k\|_{p,\nu},$$

that generalizes that of Bertsekas and Tsitsiklis (1996) (Equation 23) since $\left[C(\nu)\right]^{1/p}$ tends to 1 when p tends to infinity. Munos also provides some improved bounds when value iteration converges to some value (Munos, 2007, sections 5.2 and 5.3), or when policy iteration converges to some policy (Munos, 2003, Remark 4); similarly, these are special cases of our Corollary 14 page 1198.

At a somewhat more technical level, our key result on approximations, stated in Lemma 9 page 1194, gives a component-wise analysis for the whole family of algorithms λ policy iteration. It is thus natural to look at the relations between our bounds for general λ and the bounds derived separately by Munos for value iteration (Equation 24) and policy iteration (Equations 25 and 26).

16. We provide here a correction of the result stated by Munos (2003, Theorem 1) that is obtained by an inappropriate exchange of an expectation and a sup operator (Munos, 2003, Proofs of Corollaries 1 and 2). Note, however that the concentrability coefficient based results (Munos, 2003, Theorems 2 and 3) are not affected.

In the case where $\lambda = 0$ (and thus when λ policy iteration reduces to value iteration), consider the bound we gave in Equation 16. Since $\lambda = 0$, we have $\beta = \gamma$, $A_k = P_k$ and

$$B_{kj} = (1-\gamma)(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1}.$$

Our bound thus implies that

$$\begin{aligned} \limsup_{k\to\infty} v_* - v^{\pi_k} &\leq \limsup_{k\to\infty} \sum_{j=0}^{k-1} \gamma^{k-j} \Big[(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1} \\ &\quad - \Big(\gamma (I-\gamma P_k)^{-1} P_k P_{k-1} ... P_j + (P_*)^{k-j} \Big) \Big] \varepsilon_j. \end{aligned} \tag{27}$$

The bound derived by Munos for approximate value iteration (Equation 24) is

$$\begin{aligned} \limsup_{k\to\infty} v_* - v^{\pi_k} &\leq \limsup_{k\to\infty} (I-\gamma P_k)^{-1} \sum_{j=0}^{k-1} \gamma^{k-j} \Big[P_k P_{k-1} ... P_{j+1} - (P_*)^{k-j} \Big] \varepsilon_j \\ &= \limsup_{k\to\infty} \sum_{j=0}^{k-1} \gamma^{k-j} \Big[(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1} - (I-\gamma P_k)^{-1} (P_*)^{k-j} \Big] \varepsilon_j \\ &= \limsup_{k\to\infty} \sum_{j=0}^{k-1} \gamma^{k-j} \Big[(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1} \\ &\quad - \Big(\gamma (I-\gamma P_k)^{-1} P_k (P_*)^{k-j} + (P_*)^{k-j} \Big) \Big] \varepsilon_j. \end{aligned} \tag{28}$$

The above bounds are very close to each other: we can go from Equation 27 to Equation 28 by replacing $P_{k-1} ... P_j$ by $(P_*)^{k-j}$. Now, when $\lambda = 1$ (when λ policy iteration reduces to policy iteration), we have $\beta = 0$, $A_k = (1-\gamma)(I-\gamma P_k)^{-1} P_k$ and it is straightforward to see that $B_{kj} = R_{kj}$ and $B'_{kj} = R'_{kj}$, and the bound given in Equation 16 matches that of Munos in Equation 25. Finally, it can easily be observed that the stochastic matrices involved in Equation 26 (with the policy Bellman residual) match those of the one we gave in Equation 17: formally, we have $R''_{kj} = C_{kj}$ and $R_{kj} = C'_{kj}$. Thus, up to some little details, our component-wise analysis unifies those of Munos. It is not a surprise that we fall back on the result of Munos for approximate policy iteration because, as already mentioned at the end of Section 4.1, our proof is a generalization of his. If we do not exactly recover the component-wise analysis of Munos for approximate value iteration, this is not really fundamental as we saw that it does not affect the results once stated in terms of concentrability coefficients.

All our L_p norm bounds involve the use of some simple concentrability coefficient $C(\nu)$ (defined in Equation 20 page 1197). Munos (2007) introduced some concentrability coefficients that are finer than $C(\nu)$. In the same spirit, Farahmand et al. (2010) recently revisited the error propagation of Munos (2007, 2003) and improved (among other things) the constant in the bound related to these concentrability coefficients. In (Scherrer et al., 2012), we have further enhanced this constant by providing even finer coefficients, and provided a practical lemma (Scherrer et al., 2012, Lemma 3) to convert any component-wise bound into an L_p norm bound. Thus, rewriting our results for λ policy iteration with these refined coefficients is straightforward, and is not pursued here.

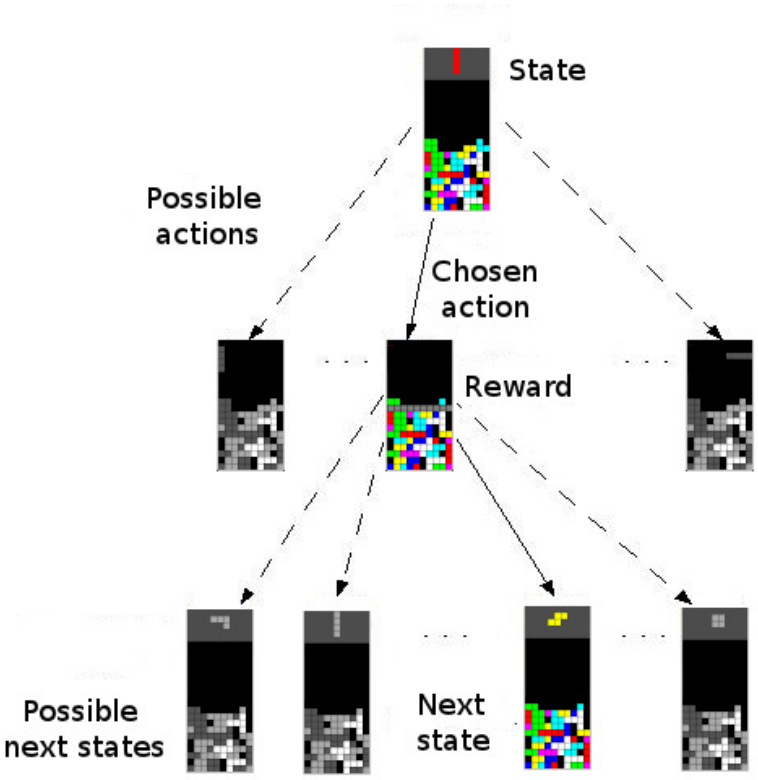


Figure 4: Modeling the Tetris game as an MDP

6. Application Of λ Policy Iteration To The Game Of Tetris

In the final part of this paper, we consider (and describe for the sake of keeping this paper self-contained) exactly the same application (Tetris) and implementation as Bertsekas and Ioffe (1996). Our main motivation here comes from the fact that we obtain empirical results that are different (and much less intriguing) than those of the original study. This gives us the opportunity to describe what we think are the reasons for such a difference. But before doing so, we begin by describing the Tetris domain.

6.1 The Game of Tetris And Its Model As An MDP

Tetris is a popular video game created in 1985 by Alexey Pajitnov. The game is played on a 10×20 grid where pieces of different shapes fall from the top. The player has to choose where each piece is added: he can move it horizontally and rotate it. When a row is filled, it is removed and all cells above it move one row downwards. The goal is to remove as many lines as possible before the game is over, that is when there is not enough space remaining on the top of the pile to put the current new piece.

Instead of mimicking the original game, precisely described by Fahey (2003), Bertsekas and Ioffe (1996) have focused on the main problem, that is choosing *where* and *in which orientation* to

drop each coming piece. The corresponding MDP model, illustrated in figure 4, is straightforward: the state consists of the wall configuration and the shape of the current piece. An action is the horizontal translation and the rotation which are applied to the piece before it is dropped on the wall. The reward is the number of lines which are removed after we have dropped the piece. As one considers the maximization of the score (the total number of lines removed during a game), the natural choice for the discount factor is $\gamma = 1$, that is we model Tetris as an undiscounted MDP, of which the terminal state corresponds to "game over".

In a bit more details, the dynamics of Tetris is made of two components: the place where one drops the current piece and the choice of a new piece. As the latter component is uncontrollable (a new piece is chosen with uniform probability), the value functions does not need to be computed for all wall-piece pairs configurations but only for all wall configurations (Bertsekas and Ioffe, 1996). Also considering that the first component of the dynamics is deterministic, the optimal value function satisfies the following Bellman equation:

$$\forall s \in S, \quad v_*(s) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \max_{a \in A(p)} r(s,p,a) + v_*(succ(s,p,a)), \tag{29}$$

where S is the set of wall configurations, $\mathcal{P}$ is the set of pieces, $A(p)$ is the set of translation-rotation pairs that can be applied to a piece p, $r(s,p,a)$ and $succ(s,p,a)$ are respectively the number of lines removed and the (deterministic) next wall configuration if one puts a piece p on the wall s in translation-orientation a. The only function that satisfies the above Bellman equation gives, for each wall configuration s, the average best score that can be achieved from s. If we know this function, a one step look-ahead strategy (that is a greedy policy) performs optimally.

6.2 An Instance Of Approximate λ Policy Iteration

For large scale problems, many approximate dynamic-programming algorithms are based on two complementary tricks:

- one uses samples to approximate the expectations such as that of Equation 8;
- one only looks for a linear approximation of the optimal value function:

  $$v^\theta(s) = \theta(0) + \sum_{k=1}^{K} \theta(k)\Phi_k(s)$$

  where $\theta = (\theta(0) \dots \theta(K))$ is the parameter vector and $\Phi_k(s)$ are some predefined feature functions on the state space. Thus, each value of θ characterizes a value function v^θ over the entire state space.

The instance of approximate λ policy iteration of Bertsekas and Ioffe (1996) follows these ideas. More specifically, this algorithm is devoted to MDPs which have a termination state, that has 0 reward and is absorbing. For this algorithm to be run, one must further assume that all policies are proper, which means that all policies reach the termination state with probability one in finite time.[17]

17. Bertsekas and Ioffe (1996) consider a weaker assumption for exact λ policy iteration and its analysis, namely that there exists at least *one* proper policy. However, this assumption is not sufficient for their approximate algorithm, because this builds sample trajectories that need to reach a termination state. If the terminal state were not reachable in finite time, this algorithm may not terminate in finite time.

This condition holds in the case of Tetris; in fact, Burgiel (1997) has shown that, whatever the strategy, some sequence of pieces (which necessarily occurs in finite time with probability 1) leads to game-over whatever the decisions taken. In particular, this implies that the condition required for our analysis (Equation 21 page 1198) holds.

Similarly to exact λ policy iteration, this approximate λ policy iteration maintains a *compact* value-policy pair (θ_t, π_t). Given θ_t, π_{t+1} is the greedy policy with respect to v^{θ_t}, and can easily be computed exactly in any given state as the argmax in Equation 29. This policy π_{t+1} is used to simulate a batch of M trajectories: for each trajectory m, $(s_{m,0}, s_{m,1}, \ldots, s_{m,N_m-1}, s_{m,N_m})$ denotes the sequence of states of the m^{th} trajectory, with s_{m,N_m} being the termination state. Then, for approximating the temporal-difference equation (Equation 8 page 1189), a reasonable choice for θ_{t+1} is one that satisfies:

$$\begin{aligned}
v^{\theta_{t+1}}(s_{m,N_m}) &\simeq 0, \\
v^{\theta_{t+1}}(s_{m,N_m-1}) &\simeq v^{\theta_t}(s_{m,N_m-1}) + \delta_t(s_{m,N_m-1}, s_{m,N_m}), \\
v^{\theta_{t+1}}(s_{m,N_m-2}) &\simeq v^{\theta_t}(s_{m,N_m-2}) + \delta_t(s_{m,N_m-2}, s_{m,N_m-1}) + \gamma\lambda\delta_t(s_{m,N_m-1}, s_{m,N_m}), \\
&\vdots \quad\quad \vdots \\
v^{\theta_{t+1}}(s_{m,k}) &\simeq v^{\theta_t}(s_{m,k}) + \sum_{j=k}^{N_{m-1}} (\gamma\lambda)^{s-k}\delta_t(s_{m,j}, s_{m,j+1}), \\
&\vdots \quad\quad \vdots \\
v^{\theta_{t+1}}(s_{m,0}) &\simeq v^{\theta_t}(s_{m,0}) + \sum_{j=0}^{N_{m-1}} (\gamma\lambda)^{s}\delta_t(s_{m,j}, s_{m,j+1})
\end{aligned} \tag{30}$$

for all trajectories m, where

$$\delta_t(s_{m,N_m-1}, s_{m,N_m}) = r(s_{m,N_m-1}, \pi_{t+1}(s_{m,N_m-1}), s_{m,N_m}) - v^{\theta_t}(s_{m,N_m-1}), \tag{31}$$

and for all $j < N_m - 1$,

$$\delta_t(s_{m,j}, s_{m,j+1}) = r(s_{m,j}, \pi_{t+1}(s_{m,j}), s_{m,j+1}) + \gamma v^{\theta_t}(s_{m,j+1}) - v^{\theta_t}(s_{m,j})$$

are the temporal differences. Note that Equations 30 and 31 correspond to the terminal states after which there is no subsequent reward. A standard and efficient solution to this problem consists in minimizing the least-squares error, that is to choose θ_{t+1} as follows:

$$\theta_{t+1} = \arg\min_{\theta} \sum_{m=1}^{M} \sum_{k=0}^{N_m} \left(v^{\theta}(s_{m,k}) - v^{\theta_t}(s_{m,k}) - \sum_{j=k}^{N_{m-1}} (\gamma\lambda)^{j-k}\delta_t(s_{m,j}, s_{m,j+1}) \right)^2.$$

This approximate version of λ policy iteration generalizes well-known algorithms. When $\lambda = 0$, the generic term becomes a sample of $[T^{\pi_{k+1}} v](s_{m,k})$:

$$\begin{aligned}
v^{\theta_{t+1}}(s_{m,k}) &\simeq v^{\theta_t}(s_{m,k}) + \delta_t(s_{m,k}, s_{m,k+1}) \\
&= r(s_{m,k}, \pi_{t+1}(s_{m,k}), s_{m,k+1}) + \gamma v^{\theta_t}(s_{m,k+1}).
\end{aligned} \tag{32}$$

When $\lambda = 1$, the generic term becomes the sampled discounted return from $s_{m,k}$ until the end of the trajectory:

$$v^{\theta_{t+1}}(s_{m,k}) \simeq v^{\theta_t}(s_{m,k}) + \sum_{j=k}^{N_{m-1}} \gamma^{s-k} \delta_t(s_{m,j}, s_{m,j+1})$$
$$= \sum_{j=k}^{N_{m-1}} \gamma^{j-k} r(s_{m,j}, \pi_{t+1}(s_{m,j}), s_{m,j+1}). \quad (33)$$

In other words, for these limit values of λ, the algorithms correspond to approximate versions of value and policy iteration as described by Bertsekas and Tsitsiklis (1996). Also, as explained by Bertsekas and Ioffe (1996) and already mentioned in the introduction, the TD(λ) algorithm with linear features described by Sutton and Barto (1998, chapter 8.2) matches the algorithm we have just described when the above fitting problem is *approximated* using gradient iterations after each sample.

We follow the same protocol as originally proposed by Bertsekas and Ioffe (1996). Let $w = 10$ be the width of the board. We consider approximating the value function as a linear combination of $2w+2 = 22$ feature functions:

$$v^\theta(s) = \theta(0) + \sum_{k=1}^{w} \theta(k) h_k + \sum_{k=1}^{w-1} \theta(k+w) \Delta h_k + \theta(2w) H + \theta(2w+1) L,$$

where

- for all $k \in \{1, 2, \cdots, w\}$, h_k is the *height* of the k^{th} column of the wall;
- for all $k \in \{1, 2, \cdots, w-1\}$, Δh_k is the *height difference* $|h_k - h_{k+1}|$ between columns k and $k+1$;
- H is the *maximum wall height*, that is $\max_k h_k$;
- L is the number of *holes* (the number of empty cells covered by at least one full cell).

We started our experiments with the initial following vector: $\theta(2w) = -10$, $\theta(2w+1) = -1$ and $\theta(k) = 0$ for all $k < 2w$, so that the initial greedy policy scores in the low tens (Bertsekas and Ioffe, 1996). We used $M = 100$ training games for each policy update. As this implementation of λ policy iteration is stochastic, we ran each experiment 10 times. figure 5 displays the learning curves. The left graph shows the 10 runs (each point is the average score computed with the $M = 100$ games) and the corresponding point-wise average for a single value of λ, while the right graph shows such point-wise average curves for different values of λ: 0.0, 0.3, 0.5, 0.7 and 0.9. We chose to display on the left graph the runs corresponding to the value of $\lambda = 0.9$ that seemed to be the best on the right graph.

We can make the following observations.

- Although we initialized with not so bad a policy (the first value is around 30), the performance first drops to 0 and it *really* starts improving after a few iterations (typically around ten). This is due to the fact that the initial value function is really bad: with the given parameters, the initial value is negative whereas it is clear that the optimal value function (the average best score) is positive. Further experiments showed that the overall behavior of the algorithm was not affected by the weight initialization.

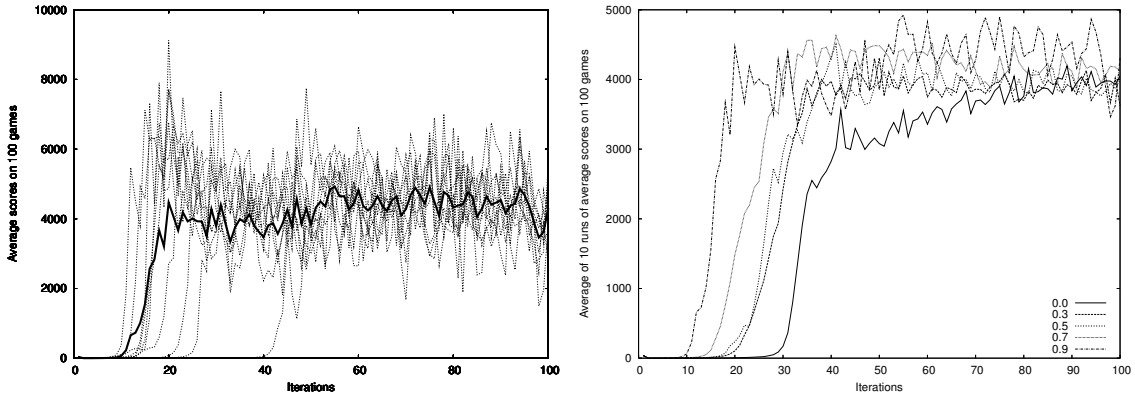


Figure 5: Average Score versus the number of iterations. **Left**: 10 runs of λ policy iteration with $\lambda = 0.9$. Each point of each run is the average score computed with $M = 100$ games. The dark curve is a point-wise average of the 10 runs. **Right**: Point-wise average of 10 runs of λ policy iteration for different values of λ; the curve which appears to be the best ($\lambda = 0.9$) is the same as the bold curve of the left graph.

- The rise of performance globally happens sooner for larger values of λ, that is for values that make the algorithm closer to policy iteration. This is not surprising as it complies with the fact that λ modulates the speed at which the value estimate tracks the real value of the current policy. However, the performance did not rise for $\lambda = 1$ (when it is equivalent to approximate policy iteration). We believe this is due to the fact that the variance of the value update is too high.

- Quantitatively, the scores reach an overall level of 4000 lines per games for a big range of values of λ.

The empirical results we have just described qualitatively and quantitatively differ from those of Bertsekas and Ioffe (1996), even though it is the exact same experimental setup. About their results, the authors wrote: "*An interesting and somewhat paradoxical observation is that a high performance is achieved after relatively few policy iterations, but the performance gradually drops significantly. We have no explanation for this intriguing phenomenon, which occurred with all of the successful methods that we tried*". As we explain now, we believe that the "intriguing" character of the results of Bertsekas and Ioffe (1996) might be related to a subtle implementation difference. Indeed, we can reproduce learning curves that are similar to those of Bertsekas and Ioffe (1996) with a little modification in our implementation of λ policy iteration, that removes the special treatments for the terminal states done through Equations 30 and 31. More precisely, if we replace them by the following equations:

$$v^{\theta_{t+1}}(s_{m,N_m}) \simeq v^{\theta_t}(s_{m,N_m}), \tag{34}$$

$$\delta_t(s_{m,N_m-1}, s_{m,N_m}) = r(s_{m,N_m-1}, \pi_{t+1}(s_{m,N_m-1})) + \gamma v^{\theta_t}(s_{m,N_m}) - v^{\theta_t}(s_{m,N_m-1}), \tag{35}$$

that is if we replace the terminal value 0 by the value $v^{\theta_t}(s_{m,N_m})$ which is computed through the features of the terminal wall configuration s_{m,N_m}, then we get the performance shown in figure 6.

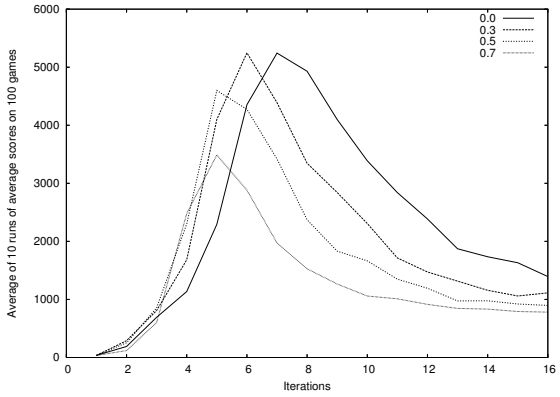


Figure 6: Average score versus the number of iterations of λ policy iteration, *modified* so that it resembles the results of Bertsekas and Ioffe (1996) (see text for details).

We observe that the performance evolution qualitatively matches the performance curves published in Bertsekas and Ioffe (1996) and illustrates the above quotation describing the "intriguing phenomenon."[18]

In such a modified form, the approximate λ policy iteration algorithm makes much less sense. In particular, it is not true anymore that it reduces to approximate value iteration and approximate policy iteration when $\lambda = 0$ and $\lambda = 1$ respectively: Equations 34 and 35 induce a bias so that we cannot recover the identities of Equations 32 and 33. A closer examination of these experiments showed that the weights (θ_k) were diverging. This is not a surprise, since the use of Equations 34 and 35 violates the condition (expressed at the end of Section 4.4) that there should be no error in the terminal state.

7. Conclusion And Future Work

We have considered the λ policy iteration algorithm introduced by Bertsekas and Ioffe (1996) that generalizes the standard algorithms value and policy iteration. We have extended the preliminary analysis of this algorithm provided by Bertsekas and Ioffe (1996) in various ways:

1. We have derived non-asymptotic convergence rates for its exact version. In particular, one such rate (Equation 13 page 1194) generalizes that for value iteration by Puterman (1994), and another one (Equation 15) is to our knowledge new even when λ policy iteration reduces to value or policy iteration.

18. A watchful reader may have noticed that the performance that we obtain is about twice that of Bertsekas and Ioffe (1996). A close inspection of the Tetris domain description given by Bertsekas and Ioffe (1996) shows that the authors consider the game of Tetris on a 10×19 board instead of our 10×20 setting, and as argued in a recent review on Tetris (Thiéry and Scherrer, 2009), this small difference is sufficient for explaining such a big performance difference.

2. We have provided asymptotic performance bounds when the algorithm is run with approximation, that generalize those made *separately* for value iteration (Munos, 2007) and policy iteration (Munos, 2003).

3. Furthermore, under assumptions ensuring that a terminal is reached in finite time with probability 1, we have extended our bounds to the undiscounted situation.

More generally, we believe that an important contribution of this paper is of conceptual nature: we have provided a unified view on some of the main approximate dynamic programming algorithms. Though the usual contraction or monotonicity arguments do not apply anymore, we explained in Section 4.1 how series of component-wise inequalities on objects we called the *value*, the *distance*, the *shift* and the *Bellman residuals* could lead to bounds on the performance loss. This line of analysis has recently been reused in variations of λ policy iteration. In (Scherrer et al., 2012), this has allowed us to provide an L_p norm performance bound for the modified policy iteration family of algorithms (Puterman and Shin, 1978). In (Thiéry and Scherrer, 2010; Scherrer and Thiéry, 2010), we have given L_∞ norm performance bounds[19] of an algorithm, named optimistic policy iteration, that makes any convex combination of the modified policy iteration possible updates, and thus generalizes both λ policy iteration and modified policy iteration. We hope that this original line of analysis will be useful for the study of other dynamic-programming/reinforcement-learning algorithms in the future.

Regarding λ policy iteration, an important research direction would be to study the implications of the choice of the parameter λ, as for instance is done by Singh and Dayan (1998) for the value estimation problem. On this matter, the original analysis by Bertsekas and Ioffe (1996) shows how one can concretely implement λ policy iteration. Each iteration requires the computation of the fixed point of the β-contracting operator M_k (see Equation 5 page 1186). We plan to study the trade-off between the ease for computing this fixed point (the smaller β, the faster) and the time for λ policy iteration to converge to the optimal policy (the bigger β, the faster). Although the reader might have noticed that most of our bounds have no explicit dependence on λ, the algorithm implicitly depends on λ through the stochastic matrices that are involved along the iterations, and the variance of the error terms. Understanding better the influence of this main parameter constitutes interesting future work.

Last but not least, we should insist on the fact that the implementation that we have described in Section 6.2, and which is borrowed from Bertsekas and Ioffe (1996), is just one possible instance of λ policy iteration. In the case of linear approximation architectures, Thiéry and Scherrer (2010) have proposed an implementation of λ policy iteration that is based on LSPI (Lagoudakis and Parr, 2003), in which the fixed point of M_k is approximated using LSTD(0) (Bradtke and Barto, 1996). Recently, Bertsekas (2011) proposed to compute this very fixed point with a variation of LSPE(λ′) (Bertsekas and Ioffe, 1996; Nedić and Bertsekas, 2003) for some λ′ potentially different from λ. Because of their very close structure, any existing implementation of approximate policy iteration may probably be turned into some implementation of λ policy iteration. Proposing such implementations and assessing their relative merits constitutes interesting future research. This may in particular be done through some finite sample analysis, as recently done for approximate value and policy iteration implementations (Antos et al., 2007, 2008; Munos and Szepesvári, 2008; Lazaric et al., 2010).

19. The extension to L_p norm is straightforward.

Acknowledgments

The author would like to thank Christophe Thiéry for contributing to the code and the illustration for Tetris, and the two anonymous reviewers for providing many comments that helped improve the presentation of the paper. Preliminary versions of this paper appeared as technical reports on http://hal.inria.fr/inria-00185271/en.

Appendix A.

The following appendices contain all the proofs concerning the analysis of λ policy iteration. We write $P_k = P^{\pi_k}$ for the stochastic matrix corresponding to the policy π_k which is greedy with respect to v_{k-1}, P_* for the stochastic matrix corresponding to the optimal policy π_*. Similarly we write T_k and T for the associated Bellman operators.

The proof techniques we have developed are inspired by those of Munos (2003, 2007). Most of the inequalities appear from the definition of the greedy operator:

$$\pi = \text{greedy}(v) \Leftrightarrow \forall \pi', T^{\pi'} v \leq T^{\pi} v.$$

We often use the property that a convex combination of stochastic matrices is also a stochastic matrix. A recurrent instance of this property is: if P is some stochastic matrix, then the geometric average

$$(1-\alpha)\sum_{i=0}^{\infty}(\alpha P)^i = (1-\alpha)(I-\alpha P)^{-1}$$

with $0 \leq \alpha < 1$ is also a stochastic matrix. We use the property that if some vectors x and y are such that $x \leq y$, then $Px \leq Py$ for any stochastic matrix P. Eventually, we will use the following equivalent forms of the operator T_λ^π (three of them were introduced in page 1187): for any value v and any policy π, we have

$$\begin{aligned} T_\lambda^\pi v &:= v + (I - \lambda\gamma P^\pi)^{-1}(T^\pi v - v) \\ &= (I - \lambda\gamma P^\pi)^{-1}(T^\pi v - \lambda\gamma P^\pi v) \qquad (36) \\ &= (I - \lambda\gamma P^\pi)^{-1}(r^\pi + (1-\lambda)\gamma P^\pi v) \qquad (37) \\ &= (I - \lambda\gamma P^\pi)^{-1}(\lambda r^\pi + (1-\lambda) T^\pi v). \end{aligned}$$

Appendix B. Proofs Of Lemmas 3-5 (Core Lemmas Of The Error Propagation)

In this section, we prove the series of Lemmas that are at the heart of our analysis of the error propagation of λ policy iteration.

B.1 Proof Of Lemma 3 (A Relation Between The Shift And the Bellman Residual)

Using the definition of $w_k = T_\lambda^{\pi_k} v_{k-1}$ and the formulation of Equation 37, we can see that we have:

$$\begin{aligned}
(I-\gamma P_k)s_k &= (I-\gamma P_k)(w_k - v^{\pi_k}) \\
&= (I-\gamma P_k)w_k - r_k \\
&= (I-\lambda\gamma P_k + \lambda\gamma P_k - \gamma P_k)w_k - r_k \\
&= (I-\lambda\gamma P_k)w_k + (\lambda\gamma P_k - \gamma P_k)w_k - r_k \\
&= r_k + (1-\lambda)\gamma P_k v_{k-1} + (\lambda-1)\gamma P_k w_k - r_k \\
&= (1-\lambda)\gamma P_k (v_{k-1} - w_k) \\
&= (1-\lambda)\gamma P_k (I-\lambda\gamma P_k)^{-1}(v_{k-1} - T_k v_{k-1}) \\
&= (1-\lambda)\gamma P_k (I-\lambda\gamma P_k)^{-1}(-b_{k-1}).
\end{aligned}$$

Therefore

$$s_k = \beta(I-\gamma P_k)^{-1} A_k (-b_{k-1})$$

with

$$A_k := (1-\lambda\gamma)P_k(I-\lambda\gamma P_k)^{-1}.$$

Suppose that we have a lower bound of the Bellman residual: $b_{k-1} \geq \underline{b_{k-1}}$ (we shall derive one soon). Since $(I-\gamma P_k)^{-1}A_k$ only has non-negative elements then

$$s_k \leq \beta(I-\gamma P_k)^{-1} A_k (-\underline{b_{k-1}}) := \overline{s_k}.$$

B.2 Proof Of Lemma 4 (A Lower Bound On The Bellman Residual)

From the definition of the algorithm, and using the fact that $T_k v^{\pi_k} = v^{\pi_k}$, we see that:

$$\begin{aligned}
b_k &= T_{k+1}v_k - v_k \\
&= T_{k+1}v_k - T_k v_k + T_k v_k - v_k \\
&\geq T_k v_k - v_k \\
&= T_k v_k - T_k v^{\pi_k} + v^{\pi_k} - v_k \\
&= \gamma P_k (v_k - v^{\pi_k}) + v^{\pi_k} - v_k \\
&= (\gamma P_k - I)(s_k + \varepsilon_k). \\
&= \beta A_k b_{k-1} + (\gamma P_k - I)\varepsilon_k
\end{aligned}$$

where we eventually used the relation between s_k and b_k (Lemma 3). In other words:

$$b_{k+1} \geq \beta A_{k+1} b_k + x_{k+1}$$

with

$$x_k := (\gamma P_k - I)\varepsilon_k.$$

Since A_k is a stochastic matrix and $\beta \geq 0$, we get by induction:

$$b_k \geq \sum_{j=1}^{k} \beta^{k-j} (A_k A_{k-1} \ldots A_{j+1}) x_j + \beta^k (A_k A_{k-1} \ldots A_1) b_0 := \underline{b_k}.$$

B.3 Proof Of Lemma 5 (An Upper Bound On The Distance)

Given that $T_* v_* = v_*$, we have

$$\begin{aligned} v_* &= v_* + (I - \lambda\gamma P_{k+1})^{-1}(T_* v_* - v_*) \\ &= (I - \lambda\gamma P_{k+1})^{-1}(T_* v_* - \lambda\gamma P_{k+1} v_*). \end{aligned}$$

Using the definition of $w_{k+1} = T_\lambda^{\pi_{k+1}} v_k$ and the formulation of Equation 36, one can see that the distance satisfies:

$$\begin{aligned} d_{k+1} &= v_* - w_{k+1} \\ &= (I - \lambda\gamma P_{k+1})^{-1}[(T_* v_* - \lambda\gamma P_{k+1} v_*) - (T_{k+1} v_k - \lambda\gamma P_{k+1} v_k)] \\ &= (I - \lambda\gamma P_{k+1})^{-1}[T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(v_k - v_*)] \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(v_k - v_*) \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(w_k + \varepsilon_k - v_*) \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(\varepsilon_k - d_k) \\ &= T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(\varepsilon_k + d_{k+1} - d_k). \end{aligned}$$

Since π^{k+1} is greedy with respect to v_k, we have $T_{k+1} v_k \geq T_* v_k$ and therefore:

$$\begin{aligned} T_* v_* - T_{k+1} v_k &= T_* v_* - T_* v_k + T_* v_k - T_{k+1} v_k \\ &\leq T_* v_* - T_* v_k \\ &= \gamma P_* (v_* - v_k) \\ &= \gamma P_* (v_* - (w_k + \varepsilon_k)) \\ &= \gamma P_* d_k - \gamma P_* \varepsilon_k. \end{aligned}$$

As a consequence, the distance satisfies:

$$d_{k+1} \leq \gamma P_* d_k + \lambda\gamma P_{k+1}(\varepsilon_k + d_{k+1} - d_k) - \gamma P_* \varepsilon_k.$$

Noticing that:

$$\begin{aligned} \varepsilon_k + d_{k+1} - d_k &= \varepsilon_k + w_k - w_{k+1} \\ &= v_k - w_{k+1} \\ &= -(I - \lambda\gamma P_{k+1})^{-1}(T_{k+1} v_k - v_k) \\ &= (I - \lambda\gamma P_{k+1})^{-1}(-b_k) \\ &\leq (I - \lambda\gamma P_{k+1})^{-1}(-\underline{b_k}), \end{aligned}$$

we get:

$$d_{k+1} \leq \gamma P_* d_k + y_k$$

where

$$y_k := \frac{\lambda\gamma}{1 - \lambda\gamma} A_{k+1}(-\underline{b_k}) - \gamma P_* \varepsilon_k.$$

Since P_* is a stochastic matrix and $\gamma \geq 0$, we have by induction:

$$d_k \leq \sum_{j=0}^{k-1} \gamma^{k-1-j} (P_*)^{k-1-j} y_j + \gamma^k (P_*)^k d_0 = \overline{d_k}.$$

Appendix C. Proof Of Lemma 6 (Performance Of Exact λ Policy Iteration

We here derive the convergence rate bounds for exact λ policy iteration (as expressed in Lemma 6 page 1193). We rely on the loss bound analysis of appendix B with $\varepsilon_k = 0$. In this specific case, we know that the loss $l_k \leq \overline{d_k} + \overline{s_k}$ where

$$\begin{aligned} -\underline{b_k} &= \beta^k A_k A_{k-1} \ldots A_1 (-b_0), \\ \overline{d_k} &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} (P_*)^{k-1-j} A_{j+1} (-\underline{b_j}) + \gamma^k (P_*)^k d_0, \\ \text{and } \overline{s_k} &= \beta (I - \gamma P_k)^{-1} A_k (-\underline{b_{k-1}}). \end{aligned}$$

Introducing the following stochastic matrices:

$$\begin{aligned} X_{i,k} &:= (P_*)^{k-1-i} A_{i+1} A_i \ldots A_1 \\ \text{and } Y_k &:= (1-\gamma)(I - \gamma P_k)^{-1} A_k A_{k-1} \ldots A_1, \end{aligned}$$

we have

$$\overline{d_k} = \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j X_{j,k} (-b_0) + \gamma^k (P_*)^k d_0$$

and

$$\overline{s_k} = \frac{\beta^k}{1-\gamma} Y_k (-b_0).$$

Therefore the loss satisfies:

$$\begin{aligned} l_k &\leq \overline{d_k} + \overline{s_k} \\ &\leq \left(\frac{\gamma^k}{1-\gamma} \right) E'_k (-b_0) + \gamma^k (P_*)^k d_0 \end{aligned} \tag{38}$$

with

$$E'_k := \left(\frac{1-\gamma}{\gamma^k} \right) \left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j X_{j,k} + \frac{\beta^k}{1-\gamma} Y_k \right).$$

To end the proof, we simply need to prove the following lemma:

Lemma 18 *E'_k is a stochastic matrix.*

Proof Using the facts that $\frac{\lambda\gamma}{\gamma-\beta} = \frac{1}{1-\beta}$ and $(1-\beta)(1-\lambda\gamma) = 1-\gamma$, one can observe that

$$\begin{aligned} \frac{1-\gamma}{\gamma^k} \left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j + \frac{\beta^k}{1-\gamma} \right) &= \frac{1-\gamma}{\gamma^k} \left(\frac{\lambda\gamma}{1-\lambda\gamma} \frac{\gamma^k - \beta^k}{\gamma-\beta} + \frac{\beta^k}{1-\gamma} \right) \\ &= \frac{1-\gamma}{\gamma^k} \left(\frac{\gamma^k - \beta^k}{1-\gamma} + \frac{\beta^k}{1-\gamma} \right) \\ &= 1 \end{aligned}$$

and deduce that E'_k is a stochastic matrix, since it is a convex combination of stochastic matrices. ■

C.1 Proof Of Equation 11 (A Bound With Respect To The Bellman Residual)

We first need the following lemma:

Lemma 19 *The bias and the distance are related as follows:*

$$b_k \geq (I - \gamma P_*) d_k.$$

Proof Since π_{k+1} is greedy with respect to v_k, $T_{k+1} v_k \geq T_* v_k$ and

$$\begin{aligned} b_k &= T_{k+1} v_k - v_k \\ &= T_{k+1} v_k - T_* v_k + T_* v_k - T_* v_* + v_* - v_k \\ &\geq \gamma P_* (v_k - v_*) + v_* - v_k \\ &= (I - \gamma P_*) d_k. \end{aligned}$$

■

We thus have:

$$d_0 \leq (I - \gamma P_*)^{-1} b_0.$$

Then Equation 38 becomes

$$\begin{aligned} l_k &\leq \left[\gamma^k (P_*)^k (I - \gamma P_*)^{-1} - \left(\frac{\gamma^k}{1-\gamma} \right) E'_k \right] b_0 \\ &= \frac{\gamma^k}{1-\gamma} \left[E_k - E'_k \right] b_0 \end{aligned}$$

where:

$$E_k := (1-\gamma)(P_*)^k (I - \gamma P_*)^{-1}$$

is a stochastic matrix.

C.2 Proof Of Equation 10 (A Bound With Respect To The Distance)

From Lemma 19, we know that

$$-b_0 \leq (I - \gamma P_*)(-d_0).$$

Then, Equation 38 becomes

$$\begin{aligned} l_k &\leq \left[\gamma^k (P_*)^k - \left(\frac{\gamma^k}{1-\gamma} \right) E'_k (I - \gamma P_*) \right] d_0 \\ &= \frac{\gamma^k}{1-\gamma} \left[F_k - E'_k \right] d_0 \end{aligned}$$

where

$$F_k := (1-\gamma) P_*^k + \gamma E'_k P_*$$

is a stochastic matrix.

C.3 Proof Of Equation 12 (A Bound With Respect To The Distance And The Loss Of The Greedy Policy)

Define $\hat{v}_0 := v_0 - Ke$ where K is some constant and e denotes the vector of which all components are 1. The following statements are equivalent:

$$\begin{aligned} \hat{b}_0 &\geq 0, \\ T_1\hat{v}_0 &\geq \hat{v}_0, \\ r_1 + \gamma P_1(v_0 - Ke) &\geq v_0 - Ke, \\ (I - \gamma P_1)Ke &\geq -r_1 + (I - \gamma P_1)v_0, \\ Ke &\geq (I - \gamma P_1)^{-1}(-r_1) + v_0, \\ Ke &\geq v_0 - v^{\pi_1}. \end{aligned}$$

The minimal K for which $\hat{b}_0 \geq 0$ is thus $K := \max_s[v_0(s) - v^{\pi_1}(s)]$. As $\hat{v}_0$ and v_0 only differ by a constant vector, they generate the same sequence of policies $\pi_1, \pi_2 ...$ Then, as $\hat{b}_0 \geq 0$, Equation 38 implies that

$$\begin{aligned} \|v_* - v^{\pi_k}\|_\infty &\leq \gamma^k \|v_* - \hat{v}_0\|_\infty \\ &\leq \gamma^k (\|v_* - v_0\|_\infty + K). \end{aligned}$$

The result is obtained by noticing that

$$\begin{aligned} K &= \max_s[v_0(s) - v_*(s) + v_*(s) - v^{\pi_1}(s)] \\ &\leq \|v_* - v_0\|_\infty + \|v_* - v^{\pi_1}\|_\infty. \end{aligned}$$

Appendix D. Proof Of Equation 16 In Lemma 9 (Component-Wise Bounds On The Error Propagation)

We here use the loss bound analysis of appendix B to derive an asymptotic analysis of approximate λ policy iteration with respect to the approximation error. The results stated here constitute a proof of the first inequality of Lemma 9 page 1194.

D.1 Proof Of Equation 16

Since the loss satisfies

$$l_k = d_k + s_k \leq \overline{d_k} + \overline{s_k}, \tag{39}$$

an upper bound of the loss can be derived from the upper bound of the distance and the shift.

Let us first concentrate on the bound $\overline{d_k}$ of the distance. Lemmas 4 and 5 imply that:

$$\begin{aligned} \overline{d_k} &= \sum_{i=0}^{k-1} \gamma^{k-1-i} (P_*)^{k-1-i} y_i + O(\gamma^k), \\ y_i &= \frac{\lambda\gamma}{1-\lambda\gamma} A_{i+1}(-\underline{b_i}) - \gamma P_* \varepsilon_i, \\ -\underline{b_i} &= \sum_{j=0}^{i} \beta^{i-j} (A_i A_{i-1} ... A_{j+1})(-x_j) + O(\beta^i), \\ \text{and } -x_j &= (I - \gamma P_j)\varepsilon_j. \end{aligned} \tag{40}$$

Writing

$$X_{i,j,k} := (P_*)^{k-1-i} A_{i+1} A_i ... A_{j+1}$$

and putting all things together, we see that:

$$\begin{aligned}
\overline{d_k} &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \gamma^{k-1-i} \left(\sum_{j=0}^{i} \beta^{i-j} X_{i,j,k} (I - \gamma P_j) \varepsilon_j + O(\beta^i) \right) \\
&\qquad - \sum_{i=0}^{k-1} \gamma^{k-i} (P_*)^{k-i} \varepsilon_i + O(\gamma^k) \\
&= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \sum_{j=0}^{i} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k} (I - \gamma P_j) \varepsilon_j - \sum_{i=0}^{k-1} \gamma^{k-i} (P_*)^{k-i} \varepsilon_i + O(\gamma^k) \\
&= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \sum_{i=j}^{k-1} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k} (I - \gamma P_j) \varepsilon_j - \sum_{j=0}^{k-1} \gamma^{k-j} (P_*)^{k-j} \varepsilon_j + O(\gamma^k) \\
&= \sum_{j=0}^{k-1} \left[\left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=j}^{k-1} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k} (I - \gamma P_j) \right) - \gamma^{k-j} (P_*)^{k-j} \right] \varepsilon_j + O(\gamma^k) \qquad (41)
\end{aligned}$$

where between the first two lines, we used the fact that:

$$\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \gamma^{k-1-i} \beta^i = \frac{\lambda\gamma}{1-\lambda\gamma} \frac{\gamma^k - \beta^k}{\gamma - \beta} = \frac{\gamma^k - \beta^k}{1-\gamma} = O(\gamma^k)$$

using the identities $\lambda\gamma = \frac{\gamma-\beta}{1-\beta}$ and $1 - \gamma\lambda = \frac{1-\gamma}{1-\beta}$.

Let us now consider the bound $\overline{s_k}$ of the shift. From Lemma 3 and the bound on b_k in Equation 40, we have

$$\begin{aligned}
\overline{s_k} &= \beta (I - \gamma P_k)^{-1} A_k (-\underline{b_{k-1}}) \\
&= \beta (I - \gamma P_k)^{-1} A_k \left[\left(\sum_{j=0}^{k-1} \beta^{k-1-j} (A_{k-1} A_{k-2} ... A_{j+1}) (-x_j) \right) + O(\gamma^k) \right] \\
&= \sum_{j=0}^{k-1} \frac{\beta^{k-j}}{1-\gamma} Y_{j,k} (I - \gamma P_j) \varepsilon_j + O(\gamma^k) \qquad (42)
\end{aligned}$$

with

$$Y_{j,k} := (1-\gamma)(I - \gamma P_k)^{-1} A_k A_{k-1} ... A_{j+1}.$$

Eventually, from Equations 39, 41 and 42 we get:

$$\begin{aligned}
l_k \leq \sum_{j=0}^{k-1} \left[\left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=j}^{k-1} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k} + \frac{\beta^{k-j}}{1-\gamma} Y_{j,k} \right) (I - \gamma P_j) - \gamma^{k-j} (P_*)^{k-j} \right] \varepsilon_j \\
+ O(\gamma^k). \qquad (43)
\end{aligned}$$

Introduce the following matrices:

$$\begin{aligned}
B_{kj} &:= \frac{1-\gamma}{\gamma^{k-j}} \left[\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=j}^{k-1} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k} + \frac{\beta^{k-j}}{1-\gamma} Y_{j,k} \right] \\
B'_{kj} &:= \gamma B_{kj} P_j + (1-\gamma)(P_*)^{k-j}.
\end{aligned}$$

Lemma 20 *B_{kj} and B'_{kj} are stochastic matrices.*

Proof Using the identities: $\lambda\gamma = \frac{\gamma-\beta}{1-\beta}$ and $(1-\beta)(1-\gamma\lambda) = 1-\gamma$, one can see that

$$\begin{aligned}
\frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}+\frac{\beta^{k-j}}{1-\gamma}\right] &= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\frac{\gamma^{k-j}-\beta^{k-j}}{\gamma-\beta}+\frac{\beta^{k-j}}{1-\gamma}\right] \\
&= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\gamma^{k-j}-\beta^{k-j}}{(1-\lambda\gamma)(1-\beta)}+\frac{\beta^{k-j}}{1-\gamma}\right] \\
&= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\gamma^{k-j}-\beta^{k-j}}{1-\gamma}+\frac{\beta^{k-j}}{1-\gamma}\right] \\
&= 1
\end{aligned}$$

and deduce that B_{kj} is a stochastic, since it is a convex combination of stochastic matrices. Then it is also clear that B'_{kj} is a stochastic matrix. ■

Thus, Equation 43 can be rewritten as follows:

$$\begin{aligned}
l_k &\leq \sum_{j=0}^{k-1}\left[\frac{\gamma^{k-j}}{1-\gamma}B_{kj}(I-\gamma P_j)-\gamma^{k-j}(P_*)^{k-j}\right]\varepsilon_j + O(\gamma^k) \\
&= \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[B_{kj}-B'_{kj}\right]\varepsilon_j + O(\gamma^k).
\end{aligned}$$

Appendix E. Proofs Of Equations 17-18 In Lemma 9 (Component-Wise Bounds With Respect To The Bellman Residuals)

In this section, we study the loss

$$l_k := v_* - v^{\pi_k}$$

with respect to the two following **Bellman residuals**:

$$b'_k := T_k v_k - v_k$$

$$\text{and } \; b_k := T_{k+1}v_k - v_k = Tv_k - v_k.$$

The term b'_k says how much v_k differs from the value of π_k while b_k says how much v_k differs from the value of the policies π_{k+1} and π_*. The results stated here prove the last two inequalities of Lemma 9 page 1194.

E.1 Proof Of Equation 17 (Bounds With Respect To The Policy Bellman Residual)

Our analysis relies on the following lemma

Lemma 21 *Suppose that we have a policy π, a function v that is an approximation of the value v^π of π in the sense that its residual $b' := T^\pi v - v$ is small. Taking the greedy policy π' with respect to v reduces the loss as follows:*

$$v_* - v^{\pi'} \leq \gamma P_*(v_* - v^\pi) + \left(\gamma P_*(I-\gamma P)^{-1} - \gamma P'(I-\gamma P')^{-1}\right) b'$$

where P and P' are the stochastic matrices which correspond to π and π'.

Proof We have:

$$\begin{aligned} v_* - v^{\pi'} &= T_* v_* - T^{\pi'} v^{\pi'} \\ &= T_* v_* - T_* v^{\pi} + T_* v^{\pi} - T_* v + T_* v - T^{\pi'} v + T^{\pi'} v - T^{\pi'} v^{\pi'} \\ &\leq \gamma P_* (v_* - v^{\pi}) + \gamma P_* (v^{\pi} - v) + \gamma P' (v - v^{\pi'}) \end{aligned} \tag{44}$$

where we used the fact that $T_* v \leq T^{\pi'} v$. One can see that:

$$\begin{aligned} v^{\pi} - v &= T^{\pi} v^{\pi} - v \\ &= T^{\pi} v^{\pi} - T^{\pi} v + T^{\pi} v - v \\ &= \gamma P (v^{\pi} - v) + b' \\ &= (I - \gamma P)^{-1} b' \end{aligned} \tag{45}$$

and that

$$\begin{aligned} v - v^{\pi'} &= v - T^{\pi'} v^{\pi'} \\ &= v - T^{\pi} v + T^{\pi} v - T^{\pi'} v + T^{\pi'} v - T^{\pi'} v^{\pi'} \\ &\leq -b' + \gamma P' (v - v^{\pi'}) \\ &\leq (I - \gamma P')^{-1} (-b'). \end{aligned} \tag{46}$$

where we used the fact that $T^{\pi} v \leq T^{\pi'} v$. We get the result by putting back Equations 45 and 46 into Equation 44. ■

To derive a bound for λ policy iteration, we simply apply the above lemma to $\pi = \pi_k$, $v = v_k$ and $\pi' = \pi_{k+1}$. We thus get:

$$l_{k+1} \leq \gamma P_* l_k + \left(\gamma P_* (I - \gamma P_k)^{-1} - \gamma P_{k+1} (I - \gamma P_{k+1})^{-1} \right) b'_k.$$

By induction, we obtain for all k,

$$l_k \leq \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[C_{kj} - C'_{kj} \right] b'_j + O(\gamma^k)$$

where we have defined the following stochastic matrices:

$$\begin{aligned} C_{kj} &:= (1-\gamma)(P_*)^{k-j} (I - \gamma P_j)^{-1} \\ C'_{kj} &:= (1-\gamma)(P_*)^{k-j-1} P_{j+1} (I - \gamma P_{j+1})^{-1}. \end{aligned}$$

E.2 Proof Of Equation 18 (Bounds With Respect To The Bellman Residual)

We rely on the following lemma, that is for instance proved by Munos (2007).

Lemma 22 *Suppose that we have a function v. Let π be the greedy policy with respect to v. Then*

$$v_* - v^{\pi} \leq \gamma \left[P_* (I - \gamma P_*)^{-1} - P^{\pi} (I - \gamma P^{\pi})^{-1} \right] (T^{\pi} v - v).$$

We provide a proof for the sake of completeness:

Proof Using the fact that $T_* v \leq T^\pi v$, we see that

$$\begin{aligned} v_* - v^\pi &= T_* v_* - T^\pi v^\pi \\ &= T_* v_* - T_* v + T_* v - T^\pi v + T^\pi v - T^\pi v^\pi \\ &\leq T_* v_* - T_* v + T^\pi v - T^\pi v^\pi \\ &= \gamma P_* (v_* - v) + \gamma P^\pi (v - v^\pi) \\ &= \gamma P_* (v_* - v^\pi) + \gamma P_* (v^\pi - v) \gamma P^\pi (v - v^\pi) \\ &\leq (I - \gamma P_*)^{-1} (\gamma P_* - \gamma P^\pi)(v^\pi - v). \end{aligned}$$

Using Equation 45 we see that:

$$v^\pi - v = (I - \gamma P^\pi)^{-1} (T^\pi v - v).$$

Thus,

$$\begin{aligned} v_* - v^\pi &\leq (I - \gamma P_*)^{-1} (\gamma P_* - \gamma P^\pi)(I - \gamma P^\pi)^{-1} (T^\pi v - v) \\ &= (I - \gamma P_*)^{-1} (\gamma P_* - I + I - \gamma P^\pi)(I - \gamma P^\pi)^{-1} (T^\pi v - v) \\ &= \left[(I - \gamma P_*)^{-1} - (I - \gamma P^\pi)^{-1} \right] (T^\pi v - v) \\ &= \gamma \left[P_* (I - \gamma P_*)^{-1} - P^\pi (I - \gamma P^\pi)^{-1} \right] (T^\pi v - v). \end{aligned}$$

■

To derive a bound for λ policy iteration, we simply apply the above lemma to $v = v_{k-1}$ and $\pi = \pi_k$. We thus get:

$$l_k \leq \frac{\gamma}{1 - \gamma} \left[D - D'_k \right] b_{k-1} \tag{47}$$

where

$$\begin{aligned} D &:= (1 - \gamma) P_* (I - \gamma P_*)^{-1} \\ \text{and } D'_k &:= (1 - \gamma) P_k (I - \gamma P_k)^{-1} \end{aligned}$$

are stochastic matrices.

Appendix F. Proofs Of Corollary 14

This section provides a proof of Corollary 14 page 1198, in which we refine the bounds when the value or the policy converges.

F.1 Proof Of The First Inequality Of Corollary 14 (When The Value Converges)

Suppose that λ policy iteration converges to some value v. Let policy π be the corresponding greedy policy, with stochastic matrix P. Let b be the Bellman residual of v. It is also clear that the approximation error also converges to some ε. Indeed from Algorithm 3 and Equation 6, we get:

$$b = Tv - v = (I - \lambda \gamma P)(-\varepsilon).$$

From the bound with respect to the Bellman residual (Equation 47 page 1219), we can see that:

$$\begin{aligned}
v_* - v^\pi &\leq \left[(I-\gamma P_*)^{-1} - (I-\gamma P)^{-1}\right] b \\
&= \left[(I-\gamma P)^{-1} - (I-\gamma P_*)^{-1}\right](I-\lambda\gamma P)\varepsilon \\
&= \left[(I-\gamma P)^{-1}(I-\lambda\gamma P) - (I-\gamma P_*)^{-1}(I-\lambda\gamma P)\right]\varepsilon \\
&= \left[(I-\gamma P)^{-1}(I-\gamma P+\gamma P-\lambda\gamma P) - (I-\gamma P_*)^{-1}(I-\lambda\gamma P)\right]\varepsilon \\
&= \left[\left(I+(1-\lambda)(I-\gamma P)^{-1}\gamma P+\lambda(I-\gamma P_*)^{-1}\gamma P\right) - (I-\gamma P_*)^{-1}\right]\varepsilon \\
&= \left[\left((1-\lambda)(I-\gamma P)^{-1}\gamma P+\lambda(I-\gamma P_*)^{-1}\gamma P\right) - (I-\gamma P_*)^{-1}\gamma P_*\right]\varepsilon \\
&= \frac{\gamma}{1-\gamma}\left[B_v - D\right]\varepsilon.
\end{aligned}$$

where

$$\begin{aligned}
B_v &:= (1-\gamma)\left((1-\lambda)(I-\gamma P)^{-1}P+\lambda(I-\gamma P_*)^{-1}P\right) \\
D &:= (1-\gamma)P_*(I-\gamma P_*)^{-1}.
\end{aligned}$$

Lemma 23 *B_v and D are stochastic matrices.*

Proof It is clear that D is a stochastic matrix. For B_v, we simply observe that

$$\begin{aligned}
(1-\gamma)\left(1+\frac{(1-\lambda)\gamma}{1-\gamma}+\frac{\lambda\gamma}{1-\gamma}\right) &= (1-\gamma)\left(1+\frac{\gamma}{1-\gamma}\right) \\
&= 1
\end{aligned}$$

and deduce that B_v is a stochastic matrix, as a convex combination of stochastic matrices. Then, the first bound of Corollary 14 follows from the application of Lemmas 10 and 12. ■

F.2 Proof Of The Second Inequality Of Corollary 14 (When The Policy Converges)

Suppose that λ policy iteration converges to some policy π. Write P the corresponding stochastic matrix and

$$A^\pi := (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}.$$

Then for some big enough k_0, we have:

$$l_k \leq \sum_{j=0}^{k-1}\left[\frac{\gamma^{k-j}}{1-\gamma}A^\pi_{kj}A^\pi(I-\gamma P) - \gamma^{k-j}(P_*)^{k-j}\right]\varepsilon_j + O(\gamma^k)$$

where

$$A^\pi_{kj} := \frac{1-\gamma}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}(P_*)^{k-1-i}(A^\pi)^{i-j} + \beta^{k-j}(I-\gamma P)^{-1}(A^\pi)^{k-1-j}\right]$$

is a stochastic matrix (for the same reasons why B_{kj} is a stochastic matrix in Lemma 20). Noticing that

$$\begin{aligned}
A^\pi(I-\gamma P) &= (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}(I-\gamma P) \\
&= (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}(I-\lambda\gamma P+\lambda\gamma P-\gamma P) \\
&= (1-\lambda\gamma)P(I-(1-\lambda)(I-\lambda\gamma P)^{-1}\gamma P) \\
&= (1-\lambda\gamma)P-\gamma(1-\lambda)A^\pi P
\end{aligned}$$

we can deduce that

$$\begin{aligned} l_k &\leq \sum_{j=0}^{k-1} \left[\frac{\gamma^{k-j}}{1-\gamma} A_{kj}^{\pi} [(1-\lambda\gamma)P - \gamma(1-\lambda)A^{\pi}P] - \gamma^{k-j}(P_*)^{k-j} \right] \varepsilon_j + O(\gamma^k) \\ &= \sum_{j=0}^{k-1} \gamma^{k-j} \left[\frac{1-\lambda\gamma}{1-\gamma} A_{kj}^{\pi} P - \left[\frac{\gamma(1-\lambda)}{1-\gamma} A_{kj}^{\pi} A^{\pi} P + (P_*)^{k-j} \right] \right] \varepsilon_j + O(\gamma^k) \\ &= \frac{1-\lambda\gamma}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[B_{kj}^{\pi} - B_{kj}^{\prime\pi} \right] \varepsilon_j + O(\gamma^k) \end{aligned}$$

where

$$\begin{aligned} B_{kj}^{\pi} &:= A_{kj}^{\pi} P \\ B_{kj}^{\prime\pi} &:= \frac{1-\gamma}{1-\lambda\gamma} \left[\frac{\gamma(1-\lambda)}{1-\gamma} A_{kj}^{\pi} A^{\pi} P + (P_*)^{k-j} \right]. \end{aligned}$$

Lemma 24 *B_{kj}^{π} and $B_{kj}^{\prime\pi}$ are stochastic matrices.*

Proof It is clear that B_{kj}^{π} is a stochastic matrix. Also, since

$$\begin{aligned} \frac{1-\gamma}{1-\lambda\gamma} \left(1 + \frac{\gamma(1-\lambda)}{1-\gamma} \right) &= \frac{1-\gamma}{1-\lambda\gamma} \frac{1-\gamma+\gamma-\lambda\gamma}{1-\gamma} \\ &= 1, \end{aligned}$$

$B_{kj}^{\prime\pi}$ is a convex combination of stochastic matrices, and thus a stochastic matrix. Then, the second bound of Corollary 14 follows from the application of Lemmas 10 and 12. ■

Appendix G. Proofs Of Lemmas 7 And 10 (From Component-Wise Bounds To L_p Norm Bounds)

This section contains the proofs of Lemmas 7 (page 1194) and 10 (page 1196) that enable us to derive L_p norm performance bounds from component-wise bounds. It is easy to see that Lemma 7 is a special case of Lemma 10, so we only prove the latter.

Consider the notations of Lemma 10. We have for all k,

$$|x_k| \leq K \sum_{j=0}^{k-1} \xi_{k-j} (X_{kj} - X'_{kj}) y_j + O(\gamma^k).$$

By taking the absolute value and using the fact that X_{kj} and X'_{kj} are stochastic matrices, we get for all k,

$$|x_k| \leq K \sum_{j=0}^{k-1} \xi_{k-j} (X_{kj} + X'_{kj}) |y_j| + O(\gamma^k).$$

It can then be seen that

$$\begin{aligned}
\limsup_{k\to\infty} \left(\|x_k\|_{p,\mu}\right)^p &= K^p \limsup_{k\to\infty} \mu^T (|x_k|)^p \\
&\leq K^p \limsup_{k\to\infty} \mu^T \left[\sum_{j=0}^{k-1} \xi_{k-j}(X_{kj}+X'_{kj})|y_j|\right]^p \\
&= K^p \limsup_{k\to\infty} \mu^T \left[\frac{\left(\sum_{j=0}^{k-1} \xi_{k-j} \frac{1}{2}(X_{kj}+X'_{kj})2|y_j|\right)}{\sum_{j=0}^{k-1}\xi_{k-j}}\right]^p \left(\sum_{j=0}^{k-1}\xi_{k-j}\right)^p .
\end{aligned}$$

By using Jensen's inequality (with the convex function $x \mapsto x^p$), we get:

$$\begin{aligned}
\limsup_{k\to\infty} \left(\|x_k\|_{p,\mu}\right)^p &\leq K^p \limsup_{k\to\infty} \mu^T \frac{\sum_{j=0}^{k-1}\xi_{k-j}\frac{1}{2}(X_{kj}+X'_{kj})(2|y_j|)^p}{\sum_{j=0}^{k-1}\xi_{k-j}} \left(\sum_{j'=0}^{k-1}\xi_{k-j'}\right)^p \\
&= K^p \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} {\mu_{kj}}^T (2|y_j|)^p \left(\sum_{j'=0}^{k-1}\xi_{k-j'}\right)^{p-1} \\
&\leq K^p \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(2\left\|y_j\right\|_{p,\mu_{kj}}\right)^p K'^{p-1} \\
&= 2^p K^p K'^{p-1} \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(\left\|y_j\right\|_{p,\mu_{kj}}\right)^p \\
&\leq 2^p K^p K'^{p-1} \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(\sup_{k'\geq j'\geq 0} \left\|y_{j'}\right\|_{p,\mu_{k'j'}}\right)^p \\
&= 2^p K^p K'^{p-1} K' \left(\sup_{k'\geq j'\geq 0} \left\|y_{j'}\right\|_{p,\mu_{k'j'}}\right)^p \\
&= 2^p K^p K'^{p} \left(\sup_{k'\geq j'\geq 0} \left\|y_{j'}\right\|_{p,\mu_{k'j'}}\right)^p
\end{aligned}$$

where we used $\sum_{j=0}^{k-1}\xi_{k-j} \leq K'$. We can apply the exact same analysis to any starting index l (instead of 0) and since the function $l \mapsto \sup_{k'\geq j'\geq l} \left\|y_{j'}\right\|_{p,\mu_{k'j'}}$ is non-decreasing, we deduce that:

$$\limsup_{k\to\infty} \left(\|x_k\|_{p,\mu}\right)^p \leq 2^p K^p K'^p \lim_{l\to\infty} \left(\sup_{k'\geq j'\geq l} \left\|y_{j'}\right\|_{p,\mu_{k'j'}}\right)^p$$

and the result follows. ■

Appendix H. Proofs Of Lemma 15 And Proposition 16 (Analysis Of The Undiscounted Case)

This last section contains the proofs of Lemma 15 and Proposition 16 that provide the analysis of an undiscounted problem.

H.1 Proof Of Lemma 15 (Component-Wise Bound)

First of all, we recall the relation expressed in Equation 22 page 1199 between the loss and the stochastic matrices:

$$\forall k_0, \quad \limsup_{k\to\infty} v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \sum_{j=0}^{k-1} \delta_{k-j} \left[G_{kj} - G'_{kj}\right] \varepsilon_j.$$

It is obtained by simply rewriting the first inequality of Lemma 9 with $\gamma = 1$ and $\beta = 1$ (note in particular that the terms δ_{k-j} collapse through the definition of G_{kj} and G'_{kj}).

To complete the proof of the lemma, we need to show that the matrices G_{kj} and G'_{kj} are sub-stochastic matrices. By construction, these matrices are sum of non-negative matrices so we only need to show that their max norm is smaller than or equal to 1.

For all n, write $\mathcal{M}_n$ the set of matrices that is defined as follows:

- for all sets of n policies $(\pi_1, \pi_2, \cdots, \pi_n)$, $\quad P_{\pi_1} P_{\pi_2} \cdots P_{\pi_n} \in \mathcal{M}_n$;
- for all $\eta \in (0,1)$, and $(P,Q) \in \mathcal{M}_n \times \mathcal{M}_n$, $\quad \eta P + (1-\eta) Q \in \mathcal{M}_n$.

The motivation for introducing this set is that we have the following properties: For all n, $P \in \mathcal{M}_n$ is a sub-stochastic matrix such that $\|P\|_\infty \leq \alpha^{\left\lfloor \frac{n}{n_0} \right\rfloor}$. We use the somewhat abusive notation Π_n for denoting any element of $\mathcal{M}_n$. For instance, for some matrix P, writing $P = a\Pi_i + b\Pi_j\Pi_k = a\Pi_i + b\Pi_{j+k}$ should be read as follows: there exist $P_1 \in \mathcal{M}_i$, $P_2 \in \mathcal{M}_j$, $P_3 \in \mathcal{M}_k$ and $P_4 \in \mathcal{M}_{k+j}$ such that $P = aP_1 + bP_2P_3 = aP_1 + bP_4$.

Recall the definition of the sub-stochastic matrix

$$A_k = (1-\lambda)(I - \lambda P_k)^{-1} P_k = (1-\lambda) \sum_{i=0}^{\infty} \lambda^i \Pi_{i+1}.$$

Let $i \leq j < k$. It can be seen that

$$\begin{aligned}
(P_*)^{k-1-i} A_{i+1} A_i \ldots A_{j+1} &= \Pi_{k-1-i} \underbrace{\left((1-\lambda)\sum_{i=0}^{\infty} \lambda^i \Pi_{i+1}\right) \cdots \left((1-\lambda)\sum_{i=0}^{\infty} \lambda^i \Pi_{i+1}\right)}_{i-j+1 \text{ terms}} \\
&= \Pi_{k-j} \underbrace{\left((1-\lambda)\sum_{i=0}^{\infty} \lambda^i \Pi_{i}\right) \cdots \left((1-\lambda)\sum_{i=0}^{\infty} \lambda^i \Pi_{i}\right)}_{i-j+1 \text{ terms}}. \qquad (48)
\end{aligned}$$

Now, observe that

$$\begin{aligned}\left\|\sum_{i=0}^{\infty}\lambda^i\Pi_i\right\|_\infty &\leq \sum_{i=0}^{\infty}\lambda^i\left\|\Pi_i\right\|_\infty \\ &\leq \sum_{i=0}^{\infty}\lambda^i\alpha^{\left\lfloor\frac{i}{n_0}\right\rfloor} \\ &= \sum_{j=0}^{\infty}\sum_{i=0}^{n_0-1}\lambda^{jn_0+i}\alpha^j \\ &= \sum_{j=0}^{\infty}(\lambda^{n_0}\alpha)^j\sum_{i=0}^{n_0-1}\lambda^i \\ &= \frac{1-\lambda^{n_0}}{(1-\lambda^{n_0}\alpha)(1-\lambda)}. \end{aligned} \tag{49}$$

As a consequence, writing $\eta := \frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}$, we see from Equation 48 that

$$\left\|(P_*)^{k-1-i}A_{i+1}A_i...A_{j+1}\right\|_\infty \leq \alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}\eta^{i-j+1}.$$

Similarly, by using Equation 49 and noticing that $\frac{1-\lambda^{n_0}}{1-\lambda} \xrightarrow{\lambda\to 1} n_0$, it can be seen that

$$\left\|(I-P_k)^{-1}A_kA_{k-1}\cdots A_{j+1}\right\|_\infty \leq \frac{n_0}{1-\alpha}\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}\eta^{k-j}.$$

We are ready to bound the norm of the matrix G_{kj}:

$$\begin{aligned}\left\|G_{kj}\right\|_\infty &\leq \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\frac{\lambda}{1-\lambda}\sum_{i=j}^{k-1}\eta^{i-j+1}+\frac{n_0\eta^{k-j}}{1-\alpha}\right] \\ &= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{\lambda}{1-\lambda}\right)\eta\left(\frac{1-\eta^{k-j}}{1-\eta}\right)+\frac{n_0\eta^{k-j}}{1-\alpha}\right] \\ &= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{\lambda}{1-\lambda}\right)\left(\frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^{k-j}}{1-\eta}\right)+\frac{n_0\eta^{k-j}}{1-\alpha}\right] \\ &= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^{k-j}}{1-\eta}\right)+\frac{n_0\eta^{k-j}}{1-\alpha}\right] \\ &= 1.\end{aligned}$$

where we used the definition of η. Therefore G_{kj} is a sub-stochastic matrix. It trivially follows that G'_{kj} is also a sub-stochastic matrix.

H.2 Proof Of Proposition 16 (L_p Norm Bound)

In order to prove the L_p norm bound of Proposition 16, we rely on the following variation of Lemma 10.

Lemma 25 *If x_k and y_k are sequences of vectors and X_{kj}, X'_{kj} sequences of sub-stochastic matrices satisfying*

$$\forall k, \quad |x_k| \leq K \sum_{j=0}^{k-1} \xi_{k-j}(X_{kj} - X'_{kj}) y_j + O(\gamma^k),$$

where $(\xi_i)_{i\geq 1}$ is a sequence of non-negative weights satisfying

$$\sum_{i=1}^{\infty} \xi_i = K' < \infty,$$

then, for all distribution μ,

$$\mu_{kj} := \frac{1}{2}(X_{kj} + X'_{kj})^T \mu$$

is a non-negative vector and $\tilde{\mu}_{kj} := \frac{\mu_{kj}}{\|\mu_{kj}\|_1}$ is a distribution, and

$$\limsup_{k\to\infty} \|x_k\|_{p,\mu} \leq 2KK' \lim_{l\to\infty} \left[\sup_{k\geq j\geq l} \|y_j\|_{p,\tilde{\mu}_{kj}} \right].$$

Proof The proof follows the lines of that of Lemma 10 in appendix G. The only difference is that in order to express the bound in terms of the distributions $\tilde{\mu}_{kj}$, we use the fact that $\mu_{kj} \leq \tilde{\mu}_{kj}$ which derives from $\|\mu_{kj}\|_1 \leq 1$ since X_{kj} and X'_{kj} are sub-stochastic matrices. ■

Proposition 16 is obtained by applying this Lemma and an analogue of Lemma 12 for L_p norm on the component-wise bound (Lemma 15, see previous subsection). The only remaining thing that needs to be checked is that $\sum_{i=1}^{\infty} \delta_i$ has the right value. This is what we do now.

Similarly to Equation 49, one can see that:

$$\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \eta^i = \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}$$

and

$$\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} (1-\eta^i) = \frac{n_0}{1-\alpha} - \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}.$$

As a consequence,

$$\begin{aligned}
\sum_{i=0}^{\infty} \delta_i &= \sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^i}{1-\eta}\right) + \frac{n_0\eta^i}{1-\alpha} \\
&= \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor}(1-\eta^i)}{1-\eta}\right) + \frac{n_0 \sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \eta^i}{1-\alpha} \\
&= \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1}{1-\eta}\right)\left(\frac{n_0}{1-\alpha} - \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}\right) \\
&\qquad + \left(\frac{n_0}{1-\alpha}\right)\left(\frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}\right) \\
&= \lambda f(\lambda) \frac{1}{1-\eta}(f(1) - f(\eta)) + f(1)f(\eta)
\end{aligned}$$

where for all x, $f(x) := \frac{(1-x^{n_0})}{(1-x)(1-x^{n_0}\alpha)}$ and $f(1) = \frac{n_0}{1-\alpha}$ by continuity. Now, we can conclude by noticing that

$$\sum_{i=1}^{\infty} \delta_i = \sum_{i=0}^{\infty} \delta_i - \delta_0$$

and $\delta_0 = \frac{n_0}{1-\alpha} = f(1)$.

References

A. Antos, Cs. Szepesvári, and R. Munos. Value-iteration based fitted policy iteration: Learning with a single trajectory. In *ADPRL*, pages 330–337. IEEE, 2007.

A. Antos, Cs. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning Journal*, 71:89–129, 2008.

D. Bertsekas. Lambda policy iteration: A review and a new implementation. Technical Report LIDS-2874, MIT, 2011.

D. Bertsekas and S. Ioffe. Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical Report LIDS-P-2349, MIT, 1996.

D.P. Bertsekas and J.N. Tsitsiklis. *Neurodynamic Programming*. Athena Scientific, 1996.

S.J. Bradtke and A.G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

H. Burgiel. How to Lose at Tetris. *Mathematical Gazette*, 81:194–200, 1997.

C. P. Fahey. Tetris AI, computer plays tetris. `http://colinfahey.com/tetris/tetris_en.html`, 2003.

A.M. Farahmand, R. Munos, and Cs. Szepesvári. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 2010.

M.G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Analysis of a classification-based policy iteration algorithm. In *International Conference on Machine Learning (ICML 2010)*, pages 607–614, 2010.

R. Munos. Error bounds for approximate policy iteration. In *International Conference on Machine Learning (ICML 2003)*, pages 560–567, 2003.

R. Munos. Performance bounds in L_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*, 46(2):541–561, 2007.

R. Munos and Cs. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

A. Nedić and D. P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13:79–110, 2003.

M. Puterman. *Markov Decision Processes*. Wiley, New York, 1994.

M. Puterman and M. Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11), 1978.

B. Scherrer and C. Thiéry. Performance bound for approximate optimistic policy iteration. Technical report, INRIA, 2010.

B. Scherrer, M. Ghavamzadeh, V. Gabillon, and M. Geist. Approximate modified policy iteration. In *International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland, 2012.

S. Singh and P. Dayan. Analytical mean squared error curves for temporal difference learning. *Machine Learning Journal*, 32(1):5–40, 1998.

R.S. Sutton and A.G. Barto. *Reinforcement Learning, An introduction*. BradFord Book. The MIT Press, 1998.

C. Thiéry and B. Scherrer. Improvements on learning Tetris with cross entropy. *International Computer Games Association Journal*, 32, 2009.

C. Thiéry and B. Scherrer. Least-squares λ policy iteration: Bias-variance trade-off in control problems. In *International Conference on Machine Learning (ICML 2010)*, Haifa, Israël, 2010.

Journal of Machine Learning Research 14 (2013) 1229-1250 Submitted 4/08; Revised 3/10; Published 5/13

Manifold Regularization and Semi-supervised Learning: Some Theoretical Analyses

Partha Niyogi[*]
Departments of Computer Science, Statistics
University of Chicago
1100 E. 58th Street, Ryerson 167
Hyde Park, Chicago, IL 60637, USA

Editor: Zoubin Ghahramani

Abstract

Manifold regularization (Belkin et al., 2006) is a geometrically motivated framework for machine learning within which several semi-supervised algorithms have been constructed. Here we try to provide some theoretical understanding of this approach. Our main result is to expose the natural structure of a class of problems on which manifold regularization methods are helpful. We show that for such problems, no supervised learner can learn effectively. On the other hand, a manifold based learner (that knows the manifold or "learns" it from unlabeled examples) can learn with relatively few labeled examples. Our analysis follows a minimax style with an emphasis on finite sample results (in terms of n: the number of labeled examples). These results allow us to properly interpret manifold regularization and related spectral and geometric algorithms in terms of their potential use in semi-supervised learning.

Keywords: semi-supervised learning, manifold regularization, graph Laplacian, minimax rates

1. Introduction

The last decade has seen a flurry of activity within machine learning on two topics that are the subject of this paper: *manifold method* and *semi-supervised learning*. While manifold methods are generally applicable to a variety of problems, the framework of manifold regularization (Belkin et al., 2006) is especially suitable for semi-supervised applications.

Manifold regularization provides a framework within which many graph based algorithms for semi-supervised learning have been derived (see Zhu, 2008, for a survey). There are many things that are poorly understood about this framework. *First*, manifold regularization is not a single algorithm but rather a collection of algorithms. So what exactly is "manifold regularization"? *Second*, while many semi-supervised algorithms have been derived from this perspective and many have enjoyed empirical success, there are few theoretical analyses that characterize the class of problems on which manifold regularization approaches are likely to work. In particular, there is some confusion on a seemingly fundamental point. Even when the data might have a manifold structure, it is not clear whether learning the manifold is *necessary* for good performance. For example, recent results (Bickel and Li, 2007; Lafferty and Wasserman, 2007) suggest that when data lives on a low dimensional manifold, it may be possible to obtain good rates of learning using classical methods suitably

∗. This article had been accepted subject to minor revisions by JMLR at the time the author sadly passed away. JMLR thanks Mikhail Belkin and Richard Maclin for their help in preparing the final version.

adapted without knowing very much about the manifold in question beyond its dimension. This has led some people (e.g., Lafferty and Wasserman, 2007) to suggest that manifold regularization does not provide any particular advantage.

What is particularly missing in the prior research so far is a crisp theoretical statement which shows the benefits of manifold regularization techniques quite clearly. This paper provides such a theoretical analysis, and explicates the nature of manifold regularization in the context of semi-supervised learning. Our main theorems (Theorems 2 and 4) show that there can be classes of learning problems on which (i) a learner that knows the manifold (alternatively learns it from large (infinite) unlabeled data via manifold regularization) obtains a fast rate of convergence (upper bound) while (ii) without knowledge of the manifold (via oracle access or manifold learning), *no learning scheme* exists that is guaranteed to converge to the target function (lower bound). This provides for the first time a clear separation between a manifold method and alternatives for a suitably chosen class of problems (problems that have intrinsic manifold structure). To illustrate this conceptual point, we have defined a simple class of problems where the support of the data is simply a one dimensional manifold (the circle) embedded in an ambient Euclidean space. Our result is the first of this kind. However, it is worth emphasizing that this conceptual point may also obtain in far more general manifold settings. The discussion of Section 2.3 and the theorems of Section 3.2 provide pointers to these more general results that may cover cases of greater practical relevance.

The plan of the paper: Against this backdrop, the rest of the paper is structured as follows. In Section 1.1, we develop the basic minimax framework of analysis that allows us to compare the rates of learning for manifold based semi-supervised learners and fully supervised learners. Following this in Section 2, we demonstrate a separation between the two kinds of learners by proving an upper bound on the manifold based learner and a lower bound on any alternative learner. In Section 3, we take a broader look at manifold learning and regularization in order to expose some subtle issues around these subjects that have not been carefully considered by the machine learning community. This section also includes generalizations of our main theorems of Section 2. In Section 4, we consider the general structure that learning problems must have for semi-supervised approaches to be viable. We show how both the classical results of Castelli and Cover (1996, one of the earliest known examples of the power of semi-supervised learning) and the recent results of manifold regularization relate to this general structure. Finally, in Section 5 we reiterate our main conclusions.

1.1 A Minimax Framework for Analysis

A learning problem is specified by a probability distribution p on $X \times Y$ according to which labelled examples $z_i = (x_i, y_i)$ pairs are drawn and presented to a learning algorithm (estimation procedure). We are interested in an understanding of the case in which $X = \mathbb{R}^D$, $Y \subset \mathbb{R}$ but p_X (the marginal distribution of p on X) is supported on some submanifold $\mathcal{M} \subset X$. In particular, we are interested in understanding how knowledge of this submanifold may potentially help a learning algorithm. To this end, we will consider two kinds of learning algorithms:

1. Algorithms that have no knowledge of the submanifold $\mathcal{M}$ but learn from (x_i, y_i) pairs in a purely supervised way.

2. Algorithms that have perfect knowledge of the submanifold. This knowledge may be acquired by a manifold learning procedure through unlabeled examples x_i's and having access to an

essentially infinite number of them. Such a learner may be viewed as a semi-supervised learner.

Our main result is to elucidate the structure of a class of problems on which there is a difference in the performance of algorithms of Type 1 and 2.

Let $\mathcal{P}$ be a collection of probability distributions p and thus denote a class of learning problems. For simplicity and ease of comparison with other classical results, we place some regularity conditions on $\mathcal{P}$. Every $p \in \mathcal{P}$ is such that its marginal p_X has support on a k-dimensional manifold $\mathcal{M} \subset X$. Different p's may have different supports. For simplicity, we will consider the case where p_X is uniform on $\mathcal{M}$: this corresponds to a situation in which the marginal is the most regular.

Given such a $\mathcal{P}$ we can naturally define the class $\mathcal{P}_{\mathcal{M}}$ to be

$$\mathcal{P}_{\mathcal{M}} = \{p \in P | p_X \text{ is uniform on } \mathcal{M}\}.$$

Clearly, we have

$$\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}}.$$

Consider $p \in \mathcal{P}_{\mathcal{M}}$. This denotes a learning problem and the regression function m_p is defined as

$$m_p(x) = E[y|x] \text{ when } x \in \mathcal{M}.$$

Note that $m_p(x)$ is not defined outside of $\mathcal{M}$. We will be interested in cases when m_p belongs to some restricted family of functions $H_{\mathcal{M}}$ (for example, a Sobolev space). Thus assuming a family $H_{\mathcal{M}}$ is equivalent to assuming a restriction on the class of conditional probability distributions $p(y|x)$ where $p \in \mathcal{P}$. For simplicity, we will assume the noiseless case where $p(y|x)$ is either 0 or 1 for every x and every y, that is, there is no noise in the Y space.

Since $X \setminus \mathcal{M}$ has measure zero (with respect to p_X), we can define $m_p(x)$ to be anything we want when $x \in X \setminus \mathcal{M}$. We define $m_p(x) = 0$ when $x \notin \mathcal{M}$.

For a learning problem p, the learner is presented with a collection of labeled examples $\{z_i = (x_i, y_i), i = 1, \ldots, n\}$ where each z_i is drawn i.i.d. according to p. A learning algorithm A maps the collection of data $\bar{z} = (z_1, \ldots, z_n)$ into a function $A(\bar{z})$. Now we can define the following minimax rate (for the class $\mathcal{P}$) as

$$R(n, \mathcal{P}) = \inf_A \sup_{p \in \mathcal{P}} E_{\bar{z}} ||A(\bar{z}) - m_p||_{L^2(p_X)}.$$

This is the best possible rate achieved by any learner that has *no knowledge of the manifold* $\mathcal{M}$. We will contrast it with a learner that has oracle access endowing it with knowledge of the manifold. To begin, note that since $\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}}$, we see that

$$R(n, \mathcal{P}) = \inf_A \sup_{\mathcal{M}} \sup_{p \in \mathcal{P}_{\mathcal{M}}} E_{\bar{z}} ||A(\bar{z}) - m_p||_{L^2(p_X)}.$$

Now a manifold based learner A' is given a collection of labeled examples $\bar{z} = (z_1, \ldots, z_n)$ just like the supervised learner. However, in addition, it also has knowledge of $\mathcal{M}$ (the support of the unlabeled data). It might acquire this knowledge through manifold learning or through oracle access (the limit of infinite amounts of unlabeled data). Thus A' maps $(\bar{z}, \mathcal{M})$ into a function denoted by $A'(\bar{z}, \mathcal{M})$. The minimax rate for such a manifold based learner for the class $\mathcal{P}_{\mathcal{M}}$ is given by

$$\inf_{A'} \sup_{p \in \mathcal{P}_{\mathcal{M}}} E_{\bar{z}} ||A'(\bar{z}, \mathcal{M}) - m_p||_{L^2(p_X)}.$$

Taking the supremum over all possible manifolds (just as in the supervised case), we have

$$Q(n,\mathcal{P}) = \sup_{\mathcal{M}} \inf_{A'} \sup_{p\in\mathcal{P}_{\mathcal{M}}} E_{\bar{z}}||A' - m_p||_{L^2(p_X)}.$$

1.2 The Manifold Assumption for Semi-supervised Learning

So the question at hand is: for what class of problems $\mathcal{P}$ with the structure as described above, might one expect a gap between $R(n,\mathcal{P})$ and $Q(n,\mathcal{P})$. This is a class of problems for which knowing the manifold confers an advantage to the learner.

There are two main assumptions behind the manifold based approach to semi-supervised learning. First, one assumes that the support of the probability distribution is on some low dimensional manifold. The motivation behind this assumption comes from the intuition that although natural data in its surface form lives in a high dimensional space (speech, image, text, etc.), they are often generated by systems with much fewer underlying degrees of freedom and therefore have lower intrinsic dimensionality. This assumption and its corresponding motivation has been articulated many times in papers on manifold methods (see Roweis and Saul, 2000, for example). Second, one assumes that the underlying target function one is trying to learn (for prediction) is smooth with respect to this underlying manifold. A smoothness assumption lies at the heart of many machine learning methods including especially splines (Wahba, 1990), regularization networks (Evgeniou et al., 2000), and kernel based methods (using regularization in reproducing kernel Hilbert spaces; Schölkopf and Smola, 2002). However, smoothness in these approaches is typically measured in the ambient Euclidean space. In manifold regularization, a geometric smoothness penalty is instead imposed.

Thus, for a manifold M, let $\phi_1, \phi_2, \ldots,$ be the eigenfunctions of the manifold Laplacian (ordered by frequency). Then, $m_p(x)$ may be expressed in this basis as $m_p = \sum_i \alpha_i \phi_i$ or

$$m_p = \text{sign}(\sum_i \alpha_i \phi_i)$$

where the α_i's have a sharp decay to zero.

Against this backdrop, one might now consider manifold regularization to get some better understanding of when and why it might be expected to provide good semi-supervised learning. First off, it is worthwhile to clarify what is meant by manifold regularization. The term "manifold regularization" was introduced by Belkin et al. (2006) to describe a class of algorithms in which geometrically motivated regularization penalties were used. One unifying framework adopts a setting of Tikhonov regularization over a Reproducing Kernel Hilbert Space of functions to yield algorithms that arise as special cases of the following:

$$\hat{f} = \arg\min_{f\in H_K} \frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \gamma_A||f||_K^2 + \gamma_I||f||_I^2. \tag{1}$$

Here $K : X \times X \to \mathbb{R}$ is a p.d. kernel that defines a suitable RKHS (H_K) of functions that are ambiently defined. The ambient RKHS norm $||f||_K$ and an "intrinsic norm" $||f||_I$ are traded-off against each other. Intuitively the intrinsic norm $||f||_I$ penalizes functions by considering only $f_{\mathcal{M}}$ the restriction of f to $\mathcal{M}$ and essentially considering various smoothness functionals. Since the eigenfunctions of the Laplacian provide a basis for L^2 functions intrinsically defined on $\mathcal{M}$, one might

express $f_{\mathcal{M}} = \sum_i \alpha_i \phi_i$ in this basis and consider constraints on the coefficients.

Remarks:

1. Various choices of $||f||_I^2$ include: (i) *iterated* Laplacian given by $\int_{\mathcal{M}} f(\Delta^i f) = \sum_j \alpha_j^2 \lambda_j^i$, (ii) *heat kernel* given by $\sum_j e^{t\lambda_j} \alpha_j^2$, and (iii)*band limiting* given by $||f||_I^2 = \sum_i \mu_i \alpha_i^2$ where $\mu_i = \infty$ for all $i > p$.

2. The loss function V can vary from squared loss to hinge loss to the 0–1 loss for classification giving rise to different kinds of algorithmic procedures.

3. While Equation 1 is regularization in the Tikhonov form, one could consider other kinds of model selection principles that are in the spirit of manifold regularization. For example, the method of Belkin and Niyogi (2003) is a version of the method of sieves that may be interpreted as manifold regularization with bandlimited functions where one allows the bandwidth to grow as more and more data becomes available.

4. The formalism provides a class of algorithms A' that have access to labeled examples $\bar{z}$ and the manifold $\mathcal{M}$ from which all the terms in the optimization of Equation 1 can be computed. Thus $A'(\bar{z}, \mathcal{M}) = \hat{f}$.

5. Finally it is worth noting that in practice when the manifold is unknown, the quantity $||f||_I^2 = \int_{\mathcal{M}} f(\Delta^i f)$ is approximated by collecting unlabeled points $x_i \in \mathcal{M}$, making a suitable nearest neighbor graph with the vertices identified with the unlabeled points, and regularizing the function using the graph Laplacian. The graph is viewed as a proxy for the manifold and in this sense, many graph based approaches to semi-supervised learning (see Zhu, 2008, for review) may be accommodated within the purview of manifold regularization.

The point of these remarks is that manifold regularization combines the perspective of kernel based methods with the perspective of manifold and graph based methods. It admits a variety of different algorithms that incorporate a geometrically motivated complexity penalty. We will later demonstrate (in Section 3) one such canonical algorithm for the class of learning problems considered in Section 2 of this paper.

2. A Prototypical Example: Embeddings of the Circle into Euclidean Space

In this section, we will construct a class of learning problems $\mathcal{P}$ that have manifold structure $\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}}$ and demonstrate a separation between $R(n, \mathcal{P})$ and $Q(n, \mathcal{P})$. For simplicity, we will show a specific construction where every $\mathcal{M}$ considered is a different embedding of the circle into Euclidean space. In particular, we will see that $R(n) = \Omega(1)$ while $lim_{n\to\infty} Q(n) = 0$ at a fast rate. Thus the learner with knowledge of the manifold learns easily while the learner with no such knowledge cannot learn at all.

Let $\phi : S^1 \to X$ be an isometric embedding of the circle into a Euclidean space. Now consider the family of such isometric embeddings and let this be the family of one-dimensional submanifolds that we will deal with. Thus each $\mathcal{M} \subset X$ is of the form $\mathcal{M} = \phi(S^1)$ for some ϕ.

Let H_{S^1} be the set of functions defined on the circle that take the value $+1$ on half the circle and -1 on the other half. Thus in local coordinates (θ denoting the coordinate of a point in S^1), we can write the class H_{S^1} as

$$H_{S^1} = \{h_\alpha : S^1 \to \mathbb{R} | h_\alpha(\theta) = \text{sign}(\sin(\theta + \alpha)); \alpha \in [0, 2\pi)\}.$$

Now for each $\mathcal{M} = \theta(S^1)$ we can define the class $H_\mathcal{M}$ as

$$H_\mathcal{M} = \{h : \mathcal{M} \to \mathbb{R} | h(x) = h_\alpha(\phi^{-1}(x)) \text{ for some } h_\alpha \in H_{S^1}\}. \quad (2)$$

This defines a class of regression functions (also classification functions) for our setting. Correspondingly, in our noiseless setting, we can now define $\mathcal{P}_\mathcal{M}$ as follows. For each, $h \in H_\mathcal{M}$, we can define the probability distribution $p^{(h)}$ on $X \times Y$ by letting the marginal $p_X^{(h)}$ be uniform on $\mathcal{M}$ and the conditional $p^{(h)}(y|x)$ be a probability mass function concentrated on two points $y = +1$ and $y = -1$ such that

$$p^{(h)}(y = +1|x) = 1 \Longleftrightarrow h(x) = +1$$

Thus

$$\mathcal{P}_\mathcal{M} = \{p^{(h)} | h \in H_\mathcal{M}\}$$

In our setting, we can therefore interpret the learning problem as an instantiation either of regression or of classification based on our interest.

Now that $\mathcal{P}_\mathcal{M}$ is defined, the set $\mathcal{P} = \cup_\mathcal{M} \mathcal{P}_\mathcal{M}$ follows naturally. A picture of the situation is shown in Figure 1.

Remark 1 *Recall that many machine learning methods (notably splines and kernel methods) construct classifiers from spaces of smooth functions. The Sobolev spaces (see Adams and Fournier, 2003) are spaces of functions whose derivatives up to a certain order are square integrable. These spaces arise in theoretical analysis of such machine learning methods and it is often the case that predictors are chosen from such spaces or regression functions are assumed to be in such spaces depending on the context of the work. For example, Lafferty and Wasserman (2007) make precisely such an assumption. In our setting, note that H_{S^1} and correspondingly $H_\mathcal{M}$ as defined above is not itself a Sobolev space. However, it is obtained by thresholding functions in a Sobolev space. In particular, we can write*

$$H_{S^1} = \{sign(h) | h = \alpha\phi + \beta\psi\}$$

where $\phi(\theta) = sin(\theta)$ and $\psi(\theta) = cos(\theta)$ are eigenfunctions of the Laplacian Δ_{S^1} on the circle. These are the eigenfunctions corresponding to $\lambda = 1$ and define the corresponding two dimensional eigenspace. More generally one could consider a family of functions obtained by thresholding functions in a Sobolev space of any chosen order and clearly H_{S^1} is contained in any such family. Finally it is worth noting that the arguments presented below do not depend on thresholding and would work with functions that are bandlimited or in a Sobolev space just as well.

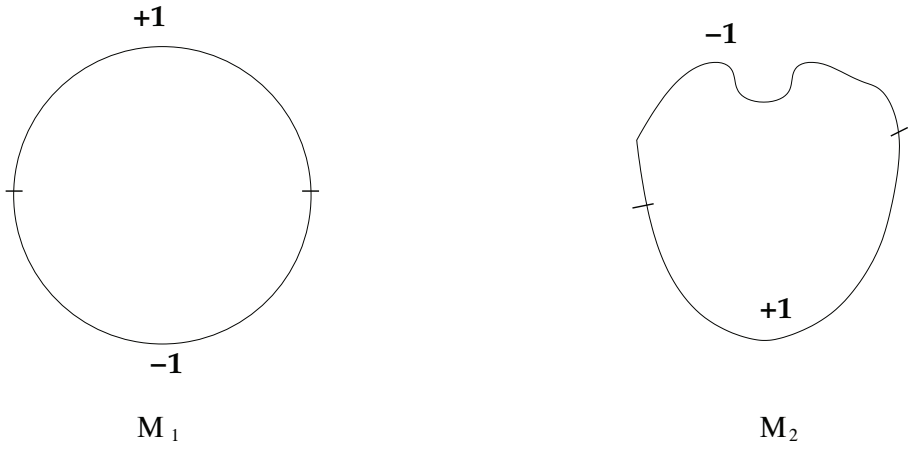


Figure 1: Shown are two embeddings of M_1 and M_2 of the circle in Euclidean space (the plane in this case). The two functions, one from $M_1 \to \mathbb{R}$ and the other from $M_2 \to \mathbb{R}$ are denoted by labelings $+1, -1$ correspond to half circles shown.

2.1 Upper Bound on $Q(n, \mathcal{P})$

Let us begin by noting that if the manifold $\mathcal{M}$ is known, the learner knows the class $\mathcal{P}_{\mathcal{M}}$. The learner merely needs to approximate one of the target functions in $H_{\mathcal{M}}$. It is clear that the space $H_{\mathcal{M}}$ is a family of $0-1$ valued functions whose VC-dimension is 2. Therefore, an algorithm that does empirical risk minimization over the class $H_{\mathcal{M}}$ will yield an upperbound of $O(\sqrt{\frac{\log(n)}{n}})$ by the usual arguments. Therefore the following theorem is obtained.

Theorem 2 *Following the notation of Section 1, let $H_{\mathcal{M}}$ be the family of functions defined by Equation 2 and $\mathcal{P}$ be the corresponding family of learning problems. Then the learner with knowledge of the manifold converges at a fast rate given by*

$$Q(n, \mathcal{P}) \leq 2\sqrt{\frac{3\log(n)}{n}}$$

and this rate is optimal. Thus every problem in this class P can be learned efficiently.

Remark 3 *If the class $H_{\mathcal{M}}$ is a a parametric family of the form $\sum_{i=1}^{p} \alpha_i \phi_i$ where ϕ_i are the eigenfunctions of the Laplacian, one obtains the same parametric rate. Similarly, if the class $H_{\mathcal{M}}$ is a ball in a Sobolev space of appropriate order, suitable rates on the family may be obtained by the usual arguments.*

2.2 Lower Bound on $R(n, \mathcal{P})$

We now prove the following.

Theorem 4 *Let $\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}}$ where each $\mathcal{M} = \phi(S^1)$ is an isometric embedding of the circle into X as shown. For each $p \in \mathcal{P}$, the marginal p_X is uniform on some $\mathcal{M}$ and the conditional $p(y|x)$ is*

given by the construction in the previous section. Then

$$R(n,\mathcal{P}) = \inf_A \sup_{p\in\mathcal{P}} E_{\bar{z}}||A(\bar{z}) - m_p||_{L^2(p_X)} = \Omega(1)$$

Thus, it is not the case that every problem in the class $\mathcal{P}$ can be learned efficiently. In other words, for every n, there exists a problem in $\mathcal{P}$ that requires more than n examples.

We provide the proof below. A specific role in the proof is played by a construction (Construction 1 later in the proof) that is used to show the existence of a family of geometrically structured learning problems (probability measures) that will end up becoming unlearnable as a family.

Proof Given n, choose a number $d = 2n$. Following Construction 1, there exist a set (denoted by $\mathcal{P}_d \subset \mathcal{P}$) of 2^d probability distributions that may be defined. Our proof uses the probabilistic method. We show that there exists a universal constant K (independent of n) such that

$$\forall A, \frac{1}{2^d}\sum_{p\in\mathcal{P}_d} E_{\bar{z}}||A(\bar{z}) - m_p||_{L^2(p_X)} \geq K$$

from which we conclude that

$$\forall A, \sup_{p\in\mathcal{P}_d} E_{\bar{z}}||A(\bar{z}) - m_p||_{L^2(p_X)} \geq K$$

Since $\mathcal{P}_d \subset \mathcal{P}$, the result follows.

To begin, consider a $p \in \mathcal{P}$. Let $\bar{z} = (z_1,\ldots,z_n)$ be a set of i.i.d. examples drawn according to p. Note that this is equivalent to drawing $\bar{x} = (x_1,\ldots,x_n)$ i.i.d. according to p_X and for each x_i, drawing y_i according to $p(y|x_i)$. Since the conditional $p(y|x)$ is concentrated on one point, the y_i's are deterministically assigned. Accordingly, we can denote this dependence by writing $\bar{z} = \bar{z}_p(\bar{x})$.

Now consider

$$E_{\bar{z}}||A(\bar{z}) - m_p||_{L^2(p_X)}.$$

This is equal to

$$\int_{Z^n} dP(\bar{z})||A(\bar{z}) - m_p||_{L^2(p_X)} = \int_{X^n} dp_X^n(\bar{x})||A(\bar{z}_p(\bar{x})) - m_p||_{L^2(p_X)}.$$

(To clarify notation, we observe that dp_X^n is the singular measure on X^n with support on $\mathcal{M}^n$ which is the natural product measure corresponding to the distribution of n data points $x_1,\ldots,x_n$ drawn i.i.d. with each x_i distributed according to p_X.) The above in turn is lowerbounded by

$$\geq \sum_{l=0}^{n}\int_{\bar{x}\in S_l} dp_X^n(\bar{x})||A(\bar{z}_p(\bar{x})) - m_p||_{L^2(p_X)}$$

where

$$S_l = \{\bar{x}\in X^n | \text{ exactly } l \text{ segments contain data and links do not }\}.$$

More formally,

$$S_l = \{\bar{x}\in X^n | \bar{x}\cap c_i \neq \phi \text{ for exactly } l \text{ segments } c_i \text{ and } \bar{x}\cap B = \emptyset\}.$$

Now we concentrate on lowerbounding $\int_{\bar{x}\in S_l} dp_X^n(\bar{x})||A(\bar{z}_p(\bar{x}))-m_p||_{L^2(p_X)}$. Using the fact that p_X is uniform, we have that $dp_X^n(\bar{x}) = cd(\bar{x})$ (where c is a normalizing constant and $d(\bar{x})$ is the Lebesgue measure or volume form on the associated product space) and therefore

$$\int_{\bar{x}\in S_l} dp_X^n(\bar{x})||A(\bar{z}_p(\bar{x}))-m_p||_{L^2(p_X)} = \int_{\bar{x}\in S_l} cd(\bar{x})||A(\bar{z}_p(\bar{x}))-m_p||_{L^2(p_X)}.$$

Thus, we have

$$E_{\bar{z}}||A(\bar{z})-m_p||_{L^2(p_X)} \geq \sum_{l=0}^{n}\int_{\bar{x}\in S_l} cd(\bar{x})||A(\bar{z}_p(\bar{x}))-m_p||_{L^2(p_X)}. \tag{3}$$

Now we see that

$$\begin{aligned}[l]\frac{1}{2^d}\sum_{p\in\mathcal{P}_d} E_{\bar{z}}||A(\bar{z}_p(\bar{x}))-m_p||_{L^2(p_X)} &\geq \frac{1}{2^d}\sum_{p\in\mathcal{P}_d}\sum_{l=0}^{n} c\int_{\bar{x}\in S_l} d(\bar{x})||A-m_p|| \\ &\geq \sum_{l=0}^{n} c\int_{\bar{x}\in S_l}\left(\frac{1}{2^d}\sum_p ||A-m_p||\right) d(\bar{x}).\end{aligned}$$

By Lemma 5, we see that for each $\bar{x}\in S_l$, we have

$$\frac{1}{2^d}\sum_p ||A-m_p|| \geq (1-\alpha-\beta)\frac{d-n}{4d}$$

from which we conclude that

$$\frac{1}{2^d}\sum_p E_{\bar{z}}||A(\bar{z})-m_p||_{L^2(p_X)} \geq (1-\alpha-\beta)\frac{d-n}{4d}\sum_{l=0}^{n}\int_{\bar{x}\in S_l} cd(\bar{x}).$$

Now we note that

$$\sum_{l=0}^{n}\int_{\bar{x}\in S_l} cd(\bar{x}) = \text{Prob}(\bar{x}\cap B=\emptyset) \geq (1-\beta)^n.$$

Therefore,

$$\sup_p E_{\bar{z}}||A(\bar{z})-m_p||_{L^2(p_X)} \geq (1-\alpha-\beta)\frac{d-n}{4d}(1-\beta)^n \geq (1-\alpha-\beta)\frac{1}{8}(1-\beta)^n. \tag{4}$$

Since α and β (and for that matter, d) are in our control, we can choose them to make the right-hand side of Inequality 4 greater than some constant. This proves our theorem. ■

We now construct a family of intersecting manifolds such that given two points on any manifold in this family, it is difficult to judge (without knowing the manifold) whether these points are near or far in geodesic distance. The class of learning problems consists of probability distributions p such that p_X is supported on some manifold in this class. This construction plays a central role in the proof of the lower bound.

Construction 1. Consider a set of 2^d manifolds where each manifold has a structure shown in Figure 2. Each manifold has three disjoint subsets: A (loops), B (links), and C (chain) such that

$$\mathcal{M} = A\cup B\cup C.$$

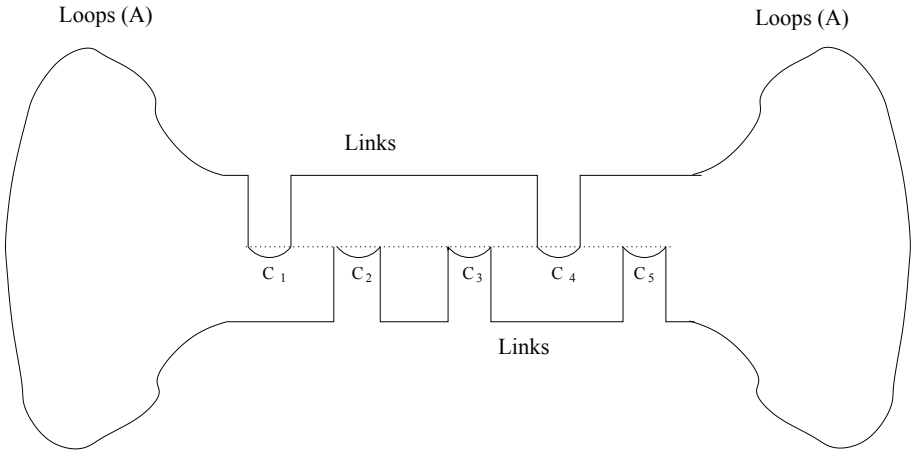


Figure 2: Figure accompanying Construction 1.

The chain C consists of d segments denoted by $C_1, C_2, \ldots, C_d$ such that $C = \cup C_i$. The links connect the loops to the segments as shown in Figure 2 so that one obtains a closed curve corresponding to an embedding of the circle into $\mathbb{R}^D$. For each choice $S \subset \{1, \ldots, d\}$ one constructs a manifold (we can denote this by $\mathcal{M}_S$) such that the links connect C_i (for $i \in S$) to the "upper half" of the loop and they connect C_j (for $j \in \{1, \ldots, d\} \setminus S$) to the "bottom half" of the loop as indicated in the figure. Thus there are 2^d manifolds altogether where each $\mathcal{M}_S$ differs from the others in the link structure but the loops and chain are common to all, that is,

$$A \cup C \subset \cap_S \mathcal{M}_S.$$

For manifold $\mathcal{M}_S$, let

$$\frac{l(A)}{l(\mathcal{M}_S)} = \int_A p_X^{(S)}(x)dx = \alpha_S$$

where $p_X^{(S)}$ is the probability density function on the manifold $\mathcal{M}_S$. Similarly

$$\frac{l(B)}{l(\mathcal{M}_S)} = \int_B p_X^{(S)}(x)dx = \beta_S$$

and

$$\frac{l(C)}{l(\mathcal{M}_S)} = \int_C p_X^{(S)}(x)dx = \gamma_S.$$

It is easy to check that one can construct these manifolds so that

$$\beta_S \leq \beta; \gamma_S \geq \gamma.$$

Thus for each manifold $\mathcal{M}_S$, we have the associated class of probability distributions $\mathcal{P}_{\mathcal{M}_S}$. These are used in the construction of the lower bound. Now for each such manifold $\mathcal{M}_S$, we pick one probability distribution $p^{(S)} \in \mathcal{P}_{\mathcal{M}_S}$ such that for every $k \in S$, we have

$$\text{For all } k \in S, p^{(S)}(y = +1|x) = 1 \text{ for all } x \in C_k$$

and for every $k \in \{1,\ldots,d\} \setminus S$, we have

$$\text{For all } k \in \{1,\ldots,d\} \setminus S, p^{(S)}(y=-1|x) = 1 \text{ for all } x \in C_k.$$

Furthermore, the $p^{(S)}$ are all chosen so that the associated conditionals $p^{(S)}(y=+1|x)$ agree on the loops, that is, for any $S, S\prime \in \{1,\ldots,d\}$,

$$p^{(S)}(y=+1|x) = p^{(S\prime)}(y=+1|x) \text{ for all } x \in A$$

This defines 2^d different probability distributions that satisfy for each p: (i) the support of the marginal p_X includes $A \cup C$, (ii) the support of p_X for different p have different link structures (iii) the conditionals $p(y|x)$ disagree on the the chain. We now prove the following technical lemma that proves an inequality that holds when the data only lives on the segments and not on the links that constitute the embedded circle of Construction 1. This inequality is used in the proof of Theorem 4.

Lemma 5 *Let $\bar{x} \in S_l$ be a collection of n points such that no point belongs to the links and exactly l segments contain at least one point.*

$$\frac{1}{2^d} \sum_{p \in \mathcal{P}_d} ||A(\bar{z}_p(\bar{x})) - m_p||_{L^2(p_X)} \geq (1-\alpha-\beta)\frac{d-n}{4d}.$$

Proof Since $\bar{x} \in S_l$, there are $d-l$ segments of the chain C such that no data is seen from them. We let $A(\bar{z}_p(\bar{x}))$ be the function hypothesized by the learner on receiving the data set $\bar{z}_p(\bar{x})$. We begin by noting that the family $\mathcal{P}_d$ may be naturally divided into 2^l subsets in the following way. Following the notation of Construction 1, recall that every element of $\mathcal{P}_d$ may be identified with a set $S \subset \{1,\ldots,d\}$. We denote this element by $p^{(S)}$. Now let L denote the set of indices of the segments C_i that contain data, that is,

$$L = \{i | C_i \cap \bar{x} \neq \emptyset\}.$$

Then for every subset $D \subset L$, we have

$$\mathcal{P}_D = \{p^{(S)} \in \mathcal{P}_d | S \cap L = D\}.$$

Thus all the elements of $\mathcal{P}_D$ agree in their labelling of the segments containing data but disagree in their labelling of segments not containing data. Clearly there are 2^l possible choices for D and each such choice leads to a family containing 2^{dl} probability distributions. Let us denote these 2^l families by $\mathcal{P}_1$ through $\mathcal{P}_{2^l}$.

Consider $\mathcal{P}_i$. By construction, for all probability distributions $p, q \in \mathcal{P}_i$, we have that $\bar{z}_p(\bar{x}) = \bar{z}_q(\bar{x})$. Let us denote this by $\bar{z}_i(\bar{x})$, that is, $\bar{z}_i(\bar{x}) = \bar{z}_p(\bar{x})$ for all $p \in \mathcal{P}_i$.

Now $f = A(\bar{z}_i(\bar{x}))$ is the function hypothesized by the learner on receiving the data set $\bar{z}_i(\bar{x})$. For any $p \in \mathcal{P}$ and any segment c_k, we say that p "disagrees" with f on c_k if $|f(x)m_p(x)| \geq 1$ on a majority of c_k, that is,

$$\int_A p_X(x) \geq \int_{c_k \setminus A} p_X(x)$$

where $A = \{x \in c_k | |f(x)m_p(x)| \geq 1\}$. Therefore, if f and p disagree on c_k, we have

$$\int_{c_k} (f(x) - m_p(x))^2 p_X(x) \geq \frac{1}{2}\int_{c_k} p_X(x) \geq \frac{1}{2d}(1-\alpha-\beta).$$

It is easy to check that for every choice of j unseen segments, there exists a $p \in \mathcal{P}_i$ such that p disagrees with f on each of the chosen segments. Therefore, for such a p, we have

$$||(A(\bar{z}_p(\bar{x})) - m_p||^2_{L^2(p_X)} \geq \frac{1}{2}\frac{j}{d}(1-\alpha-\beta).$$

Counting all the 2^{dl} elements of $\mathcal{P}_i$ based on the combinatorics of unseen segments, we see (using the fact that $||A(\bar{z}_p(\bar{x})) - m_p|| \geq \sqrt{\frac{1}{2}\frac{j}{d}(1-\alpha-\beta)} \geq \frac{1}{2}\frac{j}{d}(1-\alpha-\beta)$)

$$\sum_{p\in\mathcal{P}_i} ||A(\bar{x}_p(\bar{x}) - m_p|| \geq \sum_{j=0}^{d-l} \left(\begin{array}{c} d-l \\ j \end{array} \right) \frac{1}{2}\frac{j}{d}(1-\alpha-\beta) = 2^{d-l}(1-\alpha-\beta)\frac{d-l}{4d}.$$

Therefore, since $l \leq n$, we have

$$\sum_{i=1}^{2^l} \sum_{p\in\mathcal{P}_i} ||A(\bar{x}_p(\bar{x})) - m_p|| \geq 2^d(1-\alpha-\beta)\frac{d-n}{4d}.$$

■

2.3 Discussion

Thus we see that knowledge of the manifold can have profound consequences for learning. The proof of the lower bound reflects the intuition that has always been at the root of manifold based methods for semi-supervised learning. Following Figure 2, if one knows the manifold, one sees that C_1 and C_4 are "close" while C_1 and C_3 are "far." But this is only one step of the argument. We must further have the prior knowledge that the target function varies smoothly along the manifold and so "closeness on the manifold" translates to similarity in function values (or label probabilities). However, this closeness is not obvious from the ambient distances alone. This makes the task of the learner who does not know the manifold difficult: in fact impossible in the sense described in Theorem 4.

Some further remarks are in order. These provide an idea of the ways in which our main theorems can be extended. Thus we may appreciate the more general circumstances under which we might see a separation between manifold methods and alternative methods.

1. While we provide a detailed construction for the case of different embeddings of the circle into $\mathbb{R}^N$, it is clear that the argument is general and similar constructions can be made for many different classes of k-manifolds. Thus if $\mathcal{M}$ is taken to be a k-dimensional submanifold of $\mathbb{R}^N$, then one could let $\mathcal{M}$ be a family of k-dimensional submanifolds of $\mathbb{R}^N$ and let $\mathcal{P}$ be the naturally associated family of probability distributions that define a collection of learning problems. Our proof of Theorem 4 can be naturally adapted to such a setting.

2. Our example explicitly considers a class $H_{\mathcal{M}}$ that consists of a one-parameter family of functions. It is important to reiterate that many different choices of $H_{\mathcal{M}}$ would provide the same result. For one, thresholding is not necessary, and if the class $H_{\mathcal{M}}$ was simply defined as bandlimited functions, that is, consisting of functions of the form $\sum_{i=1}^{p} \alpha_i \phi_i$ (where ϕ_i are the eigenfunctions of the Laplacian of $\mathcal{M}$), the result of Theorem 4 holds as well. Similarly Sobolev spaces (constructed from functions $f = \sum_i \alpha_i \phi_i$ where $\alpha_i^2 \lambda_i^s < \infty$) also work with and without thresholding.

3. We have considered the simplest case where there is no noise in the Y-direction, that is, the conditional $p(y|x)$ is concentrated at one point $m_p(x)$ for each x. Considering a more general setting with noise does not change the import of our results. The upper bound of Theorem 2 makes use of the fact that m_p belongs to a restricted (uniformly Glivenko-Cantelli) family $H_{\mathcal{M}}$. With a $0-1$ loss function defined as $V(h,z) = 1_{[y \neq h(x)]}$, the rate may be as good as $O^*(\frac{1}{n})$ in the noise-free case but drops to $O^*(\frac{1}{\sqrt{n}})$ in the noisy case. The lower bound of Theorem 4 for the noiseless case also holds for the noisy case by immediate implication. Both upper and lower bounds are valid also for arbitrary marginal distributions p_X (not just uniform) that have support on some manifold $\mathcal{M}$.

4. Finally, one can consider a variety of loss functions other than the L_2 loss function considered here. The natural $0-1$-valued loss function (which for the special case of binary valued functions coincides with the the L_2 loss) can be interpreted as the probability of error of the classifier in the classification setting.

3. Manifold Learning and Manifold Regularization

3.1 Knowing the Manifold and Learning It

In the discussion so far, we have implicitly assumed that an oracle can provide perfect information about the manifold in whatever form we choose. We see that access to such an oracle can provide great power in learning from labeled examples for classes of problems that have a suitable structure.

Yet, the whole issue of *knowing the manifold* is considerably more subtle than appears at first blush and in fact has never been carefully considered by the machine learning community. For example, consider the following oracles that all provide knowledge of the manifold but in different forms.

1. One could know $\mathcal{M}$ as a set through some kind of set-membership oracle. For example, a membership oracle that makes sense is of the following sort: given a point x and a number $r > 0$, the oracle tells us whether x is in a tubular neighborhood of radius r around the manifold.

2. One could know a system of coordinate charts on the manifold. For example, maps of the form $\psi_i : U_i \to \mathbb{R}^D$ where $U_i \subset \mathbb{R}^k$ is an open set.

3. One could know in some explicit form the harmonic functions on the manifold, the Laplacian $\Delta_{\mathcal{M}}$, and the Heat Kernel $H_t(p,q)$ on the manifold.

4. One could know the manifold up to some geometric or topological invariants. For example, one might know just the dimension of the manifold. Alternatively, one might know the homology, the homeomorphism or diffeomorphism type, etc. of the manifold.

5. One could have metric information on the manifold. One might know the metric tensor at points on the manifold, one might know the geodesic distances between points on the manifold, or one might know the heat kernel from which various derived distances (such as diffusion distance) are obtained.

Depending upon the kind of oracle access we have, the task of the learner might vary from simple to impossible. For example, in the problem described in Section 2 of this paper, the natural

algorithm that realizes the upper bound of Theorem 2 performs empirical risk minimization over the class $H_{\mathcal{M}}$. To do this it needs, of course, to be able to represent $H_{\mathcal{M}}$ in a computationally efficient manner. In order to do this, it needs to know the eigenfunctions (in the specific example, only the first two, but in general some arbitrary number depending on the choice of $H_{\mathcal{M}}$) of the Laplacian on the $\mathcal{M}$. This is immediately accessible from Oracle 3. It can be computed from Oracles 1, 2, and 5 but this computation is intractable in general. From Oracle 4, it cannot be computed at all.

The next question one needs to address is: In the absence of an oracle but given random samples of example points on the manifold, can one *learn the manifold*? In particular, can one learn it in a form that is suitable for further processing. In the context of this paper, the answer is yes.

Let us recall the following fundamental fact from Belkin and Niyogi (2005) that has some significance for the problem in this paper.

Let $\mathcal{M}$ be a compact, Riemannian submanifold (without boundary) of $\mathbb{R}^N$ and let $\Delta_{\mathcal{M}}$ be the Laplace operator (on functions) on this manifold. Let $\bar{x} = \{x_1, \ldots, x_m\}$ be a collection of m points sampled in i.i.d. fashion according to the uniform probability distribution on $\mathcal{M}$. Then one may define the point cloud Laplace operator L_m^t as follows:

$$L_m^t f(x) = \frac{1}{t} \frac{1}{(4\pi t)^{d/2}} \frac{1}{m} \sum_{i=1}^{m} (f(x) - f(x_i)) e^{-\frac{||x-x_i||^2}{4t}}$$

The point cloud Laplacian is a random operator that is the natural extension of the graph Laplacian operator to the whole space. For any thrice differentiable function $f : \mathcal{M} \to \mathbb{R}$, we have

Theorem 6

$$\lim_{t \to 0, m \to \infty} L_m^t f(x) = \Delta_{\mathcal{M}} f(x).$$

Some remarks are in order:

1. Given $\bar{x} \in \mathcal{M}$ as above, consider the graph with vertices (let V be the vertex set) identified with the points in $\bar{x}$ and adjacency matrix $W_{ij} = \frac{1}{mt} \frac{1}{(4\pi t)^{d/2}} e^{-\frac{||x-x_i||^2}{4t}}$. Given $f : \mathcal{M} \to \mathbb{R}$, the restriction $f_V : \bar{x} \to \mathbb{R}$ is a function defined on the vertices of this graph. Correspondingly, the graph Laplacian $L = (D - W)$ acts on f_V and it is easy to check that

   $$(L_m^t f)|_{x_i} = (L f_V)|_{x_i}.$$

   In other words, the point cloud Laplacian and graph Laplacian agree on the data. However, the point cloud Laplacian is defined everywhere while the graph Laplacian is only defined on the data.

2. The quantity t (similar to a bandwidth) needs to go to zero at a suitable rate ($tm^{d+2} \to \infty$) so there exists a sequence t_m such that the point cloud Laplacian converges to the manifold Laplacian as $m \to \infty$.

3. It is possible to show (see Belkin and Niyogi, 2005; Coifman and Lafon, 2006; Giné and Koltchinskii, 2006; Hein et al., 2005) that this basic convergence is true for arbitrary probability distributions (not just the uniform distribution as stated in the above theorem) in which case the point cloud Laplacian converges to an operator of the Laplace type that may be related to the weighted Laplacian (Grigoryan, 2006).

4. While the above convergence is pointwise, it also holds uniformly over classes of functions with suitable conditions on their derivatives (Belkin and Niyogi, 2008; Giné and Koltchinskii, 2006).

5. Finally, and most crucially (see Belkin and Niyogi, 2006), if $\lambda_m^{(i)}$ and $\phi_m^{(i)}$ are the ith (in increasing order) eigenvalue and corresponding eigenfunction respectively of the operator $L_m^{t_m}$, then with probability 1, as m goes to infinity,

$$lim_{m\to\infty}|\lambda_i - \lambda_m^{(i)}| = 0$$

and

$$lim_{m\to\infty}|\phi_m^{(i)} - \phi_i|_{L_2(\mathcal{M})} = 0.$$

In other words, the eigenvalues and eigenfunctions of the point cloud Laplacian converge to those of the manifold Laplacian as the number of data points m go to infinity.

These results enable us to present a semi-supervised algorithm that learns the manifold from unlabeled data and uses this knowledge to realize the upper bound of Theorem 2.

3.2 A Manifold Regularization Algorithm For Semi-supervised Learning

Let $\bar{z} = (z_1, \ldots, z_n)$ be a set of n i.i.d. labeled examples drawn according to p and $\bar{x} = (x_1, \ldots, x_m)$ be a set of m i.i.d. unlabeled examples drawn according to p_X. Then a semi-supervised learner's estimate may be denoted by $A(\bar{z}, \bar{x})$. Let us consider the following kind of manifold regularization based semi-supervised learner.

1. Construct the point cloud Laplacian operator $L_m^{t_m}$ from the *unlabeled* data $\bar{x}$.

2. Solve for the eigenfunctions of $L_m^{t_m}$ and take the first two (orthogonal to the constant unction). Let these be ϕ_m and ψ_m respectively.

3. Perform empirical risk minimization with the empirical eigenfunctions by minimizing

$$\hat{f}_m = \arg\min_{f=\alpha\phi_m+\beta\psi_m} \frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

subject to $\alpha_i^2 + \beta_i^2 = 1$. Here $V(f(x), y) = \frac{1}{4}|y - \text{ sign}(f(x))|^2$ is the $0-1$ loss. This is equivalent to Ivanov regularization with an intrinsic norm that forces candidate hypothesis functions to be bandlimited.

Note that if the empirical risk minimization was performed with the true eigenfunctions (ϕ and ψ respectively), then the resulting algorithm achieves the rate of Theorem 2. Since for large m, the empirical eigenfunctions are close to the true ones by the result in Belkin and Niyogi (2006), we may expect the above algorithm to perform well. Thus we may compare the two manifold regularization algorithms (an empirical one with unlabeled data and an oracle one that knows the manifold):

$$A(\bar{z}, \bar{x}) = \text{ sign}(\hat{f}_m) = \text{ sign}(\hat{\alpha}_m\phi_m + \hat{\beta}_m\psi_m)$$

and

$$A_{oracle}(\bar{z}, \mathcal{M}) = \text{ sign}(\hat{f}) = \text{ sign}(\hat{\alpha}\phi + \hat{\beta}\psi).$$

We can now state the following:

Theorem 7 *For any $\varepsilon > 0$, we have*

$$\sup_{p \in \mathcal{P}_{\mathcal{M}}} E_{\bar{z}} ||m_p - A||^2_{L^2(p_X)} \leq \frac{4}{2\pi}(\arcsin(\varepsilon)) + \frac{1}{\varepsilon^2}(||\phi - \phi_m||^2 + ||\psi - \psi_m||^2) + 3\sqrt{\frac{3\log(n)}{n}}.$$

Proof Consider $p \in \mathcal{P}$ and let $m_p = \text{ sign}(\alpha_p \phi + \beta_p \psi)$. Let $g_m = \alpha_p \phi_m + \beta_p \psi_m$. Now, first note that by the fact of empirical risk minimization, we have

$$\frac{1}{n}\sum_{z \in \bar{z}} V(A(x), y) \leq \frac{1}{n}\sum_{z \in \bar{z}} V(\text{sign}(g_m(x)), y).$$

Second, note that the set of functions $\mathcal{F} = \{\text{sign}(f) | f = \alpha\phi_m + \beta\psi_m\}$ has VC dimension equal to 2. Therefore the empirical risk converges to the true risk uniformly over this class so that with probability $> 1 - \delta$, we have

$$[l] E_z[V(A(x), y)] - \sqrt{\frac{2\log(n) + \log(1/\delta)}{n}} \leq \frac{1}{n}\sum_{z \in \bar{z}} V(A(x), y)$$

$$\leq \frac{1}{n}\sum_{z \in \bar{z}} V(\text{sign}(g_m(x)), y) \leq E_z[V(\text{sign}(g_m(x)), y)] + \sqrt{\frac{2\log(n) + \log(1/\delta)}{n}}.$$

Using the fact that $V(h(x), y) = \frac{1}{4}(y - h)^2$, we have in general for any h

$$E_z[V(h(x), y)] = \frac{1}{4} E_z(y - m_p)^2 + \frac{1}{4} ||m_p - h||^2_{L(p_X)}$$

from which we obtain with probability $> 1 - \delta$ over choices of labeled training sets $\bar{z}$,

$$||m_p - A||^2 \leq ||m_p - \text{ sign}(g_m)||^2 + 2\sqrt{\frac{2\log(n) + \log(1/\delta)}{n}}.$$

Setting $\delta = \frac{1}{n}$ and noting that $||m_p - A||^2 \leq 1$, we have after some straightforward manipulations,

$$E_{\bar{z}} ||m_p - A||^2 \leq ||m_p - \text{ sign}(g_m)||^2 + 3\sqrt{\frac{3\log(n)}{n}}.$$

Using Lemma 8, we get for any $\varepsilon > 0$,

$$\sup_{p \in \mathcal{P}_{\mathcal{M}}} E_{\bar{z}} ||m_p - A||^2 \leq \frac{4}{2\pi}(\arcsin(\varepsilon)) + \frac{1}{\varepsilon^2}(||\phi - \phi_m||^2 + ||\psi - \psi_m||^2) + 3\sqrt{\frac{3\log(n)}{n}}.$$

■

Lemma 8 *Let f, g be any two functions. Then for any $\varepsilon > 0$,*

$$||sign(f) - sign(g)||^2_{L^2(p_X)} \leq \mu(X_{\varepsilon, f}) + \frac{1}{\varepsilon^2} ||f - g||^2_{L^2(p_X)}$$

where $X_{\varepsilon,f} = \{x \mid |f(x)| \leq \varepsilon\}$ *and* μ *is the measure corresponding the the marginal distribution* p_X.

Further, if $f = \alpha\phi + \beta\psi$ $(\alpha^2 + \beta^2 = 1)$ *and* $g = \alpha\phi_m + \beta\psi_m$ *where* ϕ, ψ *are eigenfunctions of the Laplacian on* $\mathcal{M}$ *while* ϕ_m, ψ_m *are eigenfunctions of point cloud Laplacian as defined in the previous developments. Then for any* $\varepsilon > 0$

$$||sign(f) - sign(g)||^2_{L^2(p_X)} \leq \frac{4}{2\pi}(\arcsin(\varepsilon)) + \frac{1}{\varepsilon^2}(||\phi - \phi_m||^2 + ||\psi - \psi_m||^2).$$

Proof We see that

$$\begin{aligned}||\text{sign}(f) - \text{sign}(g)||^2_{L^2(p_X)} &= \int_{X_{\varepsilon,f}} |\text{sign}(f(x)) - \text{sign}(g(x))|^2 + \int_{\mathcal{M}\setminus X_{\varepsilon,f}} |\text{sign}(f(x)) - \text{sign}(g(x))|^2 \\ &\leq 4\mu(X_{\varepsilon,f}) + \int_{\mathcal{M}\setminus X_{\varepsilon,f}} |\text{sign}(f(x)) - \text{sign}(g(x))|^2.\end{aligned}$$

Note that if $x \in \mathcal{M}\setminus X_{\varepsilon,f}$, we have that

$$|\text{sign}(f(x)) - \text{sign}(g(x))| \leq \frac{2}{\varepsilon}|f(x) - g(x)|.$$

Therefore,

$$\int_{\mathcal{M}\setminus X_{\varepsilon,f}} |\text{sign}(f(x)) - \text{sign}(g(x))|^2 \leq \frac{4}{\varepsilon^2}\int_{\mathcal{M}\setminus X_{\varepsilon,f}} |f(x) - g(x)|^2 \leq \frac{4}{\varepsilon^2}||f - g||^2_{L^2(p_X)}.$$

This proves the first part. The second part follows by a straightforward calculation on the circle. ■

Some remarks:

1. While we have stated the above theorem for our running example of embeddings of the circle into $\mathbb{R}^D$, it is clear that the results can be generalized to cover arbitrary k-manifolds, more general classes of functions $H_{\mathcal{M}}$, noise, and loss functions V. Many of these extensions are already implicit in the proof and associated technical discussions.

2. A corollary of the above theorem is relevant for the $(m = \infty)$ case that has been covered in Castelli and Cover (1996). We will discuss this in the next section. The corollary is

   Corollary 9 *Let* $\mathcal{P} = \cup_{\mathcal{M}}\mathcal{P}_{\mathcal{M}}$ *be a collection of learning problems with the structure described in Section 2, that is, each* $p \in \mathcal{P}$ *is such that the marginal* p_X *has support on a submanifold* $\mathcal{M}$ *of* $\mathbb{R}^D$ *which corresponds to a particular isometric embedding of the circle into Euclidean space. For each such p, the regression function* $m_p = E[p(y|x)]$ *belongs to a class of functions* $H_{\mathcal{M}}$ *which consists of thresholding bandlimited functions on* $\mathcal{M}$. *Then no supervised learning algorithm exists that is guaranteed to converge for every problem in* $\mathcal{P}$ *(Theorem 4). Yet the semi-supervised manifold regularization algorithm described above (with infinite amount of unlabeled data) converges at a fast rate as a function of labelled data. In other words,*

$$\sup_{\mathcal{M}} \lim_{m\to\infty} \sup_{\mathcal{P}_{\mathcal{M}}} ||m_p - A(\bar{z},\bar{x})||^2_{L^2(p_X)} = 3\sqrt{\frac{3\log(n)}{n}}.$$

3. It is natural to ask what it would take to move the limit $m \to \infty$ outside. In order to do this, one will need to put additional constraints on the class of possible manifolds $\mathcal{M}$ that we are allowed to consider. But putting such constraints we can construct classes of learning problems where for any realistic number of labeled examples n, there is a gap between the performance of a supervised learner and the manifold based semi-supervised learner. An example of such a theorem is:

Theorem 10 *Fix any number N. Then there exists a class of learning problems $\mathcal{P}_N$ such that for all $n < N$*

$$R(n, \mathcal{P}_N) = \inf_A \sup_{p \in \mathcal{P}_N} E_{\bar{z}} ||m_p - A(\bar{z})|| \geq 1/100$$

while

$$Q(n, \mathcal{P}_N) = \lim_{m \to \infty} \sup_{p \in \mathcal{P}_N} ||m_p - A_{manreg}(\bar{z}, \bar{x})||^2 \leq \sqrt{\frac{3\log(n)}{n}}.$$

Proof We provide only a sketch of the argument and avoid technical details. We begin by choosing a family of submanifolds of $[-M, M]^D$ with a uniform bound on their curvature. One form such a bound can take is the following: Let τ be the largest number such that the open normal bundle of radius r about $\mathcal{M}$ is an imbedding for any $r < \tau$. This provides a bound on the norm of the second fundamental form (curvature) and nearness to self intersection of the submanifold. Now $\mathcal{P}_N$ will contain probability distributions p such that p_X is supported on some $\mathcal{M}$ with a τ curvature bound and $p(y|x)$ is 0 or 1 for every x, y such that the regression function $m_p = E[y|x]$ belongs to $H_{\mathcal{M}}$. As before, we choose $H_{\mathcal{M}}$ to be the span of the first K eigenfunctions of the Laplacian Δ on $\mathcal{M}$. For a lower bound $R(n)$, we follow Construction 1 and choose $d = 2N$ (from the proof of the lower bound of Theorem 4). Following Construction 1, the circle can be embedded in $[-M, M]^D$ by twisting in all directions. Let l be the length of a single segment of the chain. Since the τ condition needs to be respected for every embedding, the circle cannot twist too much and come too close to self intersection. In particular, this will imply that $2NlV_\tau < M^D$ where V_τ is the volume of the $D-1$ dimensional ball of radius τ. For an upper bound $Q(n)$, we follow the the manifold regularization algorithm of the previous section and note that eigenfunctions of the Laplacian can be estimated for compact manifolds with a curvature bound. ■

However, asymptotically, $R(n)$ and $Q(n)$ have the same rate for $n >> N$. Since N can be arbitrarily chosen to be astronomically large, this asymptotic rate is of little consequence in practical learning situations. This suggests the limitations of asymptotic analysis without a careful consideration of the finite sample situation.

4. The Structure of Semi-supervised Learning

It is worthwhile to reflect on why the manifold regularization algorithm is able to display improved performance in semi-supervised learning. The manifold assumption is a device that allows us to link the marginal p_X with the conditional $p(y|x)$. Through unlabeled data $\bar{x}$, we can learn the manifold $\mathcal{M}$ thereby greatly reducing the class of possible conditionals $p(y|x)$ that we need to consider. More

generally, semi-supervised learning will be feasible only if such a link is made. To clarify the structure of problems on which semi-supervised learning is likely to be meaningful, let us define a map $\pi : p \to p_X$ that takes any probability distribution p on $X \times Y$ and maps it to the marginal p_X.

Given any collection of learning problems $\mathcal{P}$, we have

$$\pi : \mathcal{P} \to \mathcal{P}_X$$

where $\mathcal{P}_X = \{p_X | p \in P\}$. Consider the case in which the structure of $\mathcal{P}$ is such that for any $q \in \mathcal{P}_X$, the family of conditionals $\pi^{-1}(q) = \{p \in \mathcal{P} | p_X = q\}$ is "small." For a situation like this, knowing the marginal tells us a lot about the conditional and therefore unlabeled data can be useful.

4.1 Castelli and Cover Interpreted

Let us consider the structure of the class of learning problems considered by Castelli and Cover (1996). They consider a two-class problem with the following structure. The class of learning problems $\mathcal{P}$ is such that for each $p \in \mathcal{P}$, the marginal $q = p_X$ can be uniquely expressed as

$$q = \mu f + (1-\mu)g$$

where $0 \leq \mu \leq 1$ and f, g belong to some class G of possible probability distributions. In other words, the marginal is always a mixture (identifiable) of two distributions. Furthermore, the class $\mathcal{P}$ of possible probability distributions is such that there are precisely two probability distributions $p_1, p_2 \in \mathcal{P}$ such that their marginals are equal to q. In other words,

$$\pi^{-1}(q) = \{p_1, p_2\}$$

where $p_1(y = 1|x) = \frac{\mu f(x)}{q(x)}$ and $p_2(y = 1|x) = \frac{(1-\mu)g(x)}{q(x)}$.

In this setting, unlabeled data allows the learner to estimate the marginal q. Once the marginal is obtained, the class of possible conditionals is reduced to *exactly two functions*. Castelli and Cover (1996) show that the risk now converges to the Bayes' risk exponentially as a function of labeled data (i.e., the analog of an upper bound on $Q(n, \mathcal{P})$ is approximately e^{-n}). The reason semi-supervised learning is successful in this setting is that the marginal q tells us a great deal about the class of possible conditionals. It seems that a precise lower bound on purely supervised learning (the analog of $R(n, \mathcal{P})$) has never been clearly stated in that setting.

4.2 Manifold Regularization Interpreted

In its most general form, manifold regularization encompasses a class of geometrically motivated approaches to learning. Spectral geometry provides the unifying point of view and the spectral analysis of a suitable geometrically motivated operator yields a "distinguished basis." Since (i) only unlabeled examples are needed for the spectral analysis and the learning of this basis, and (ii) the target function is assumed to be compactly representable in this basis, the idea has the possibility to succeed in semi-supervised learning. Indeed, the previous theorems clarify the theoretical basis of this approach. This, together with the empirical success of algorithms based on these intuitions suggest there is some merit in this point of view.

In general, let q be a probability density function on $X = \mathbb{R}^D$. The support of q may be a submanifold of X (with possibly many connected components). Alternatively, it may lie close to a submanifold, it may be all of X, or it may be a subset of X. As long as q is far from uniform, that

is, it has a "shape," one may consider the following "weighted Laplacian" (see Grigoryan, 2006) defined as

$$\Delta_q f(x) = \frac{1}{q(x)} \mathrm{div}(q \mathrm{grad} f)$$

where the gradient (grad) and divergence (div) are with respect to the support of q (which may simply be all of X).

The heat kernel associated with this weighted Laplacian (essentially the Fokker-Planck operator) is given by $e^{-t\Delta_q}$. Laplacian eigenmaps and Diffusion maps are thus defined in this more general setting.

If $\phi_1, \phi_2, \ldots$ represent an eigenbasis for this operator, then, one may consider the regression function m_q to belong to the family (parameterized by $s = (s_1, s_2, \ldots)$) where each $s_i \in \mathbb{R} \cup \{\infty\}$.

$$H_q^S = \{h : X \to \mathbb{R} \text{ such that } h = \sum_i \alpha_i \phi_i \text{ and } \sum_i \alpha_i^2 s_i < \infty\}.$$

Some natural choices of s are (i) $\forall i > p, s_i = \infty$: this gives us bandlimited functions (ii) $s_i = \lambda_i^t$ is the ith eigenvalue of Δ_q : this gives us spaces of Sobolev type (iii) $\forall i \in A, s_i = 1$, else $s_i = \infty$ where A is a finite set: this gives us functions that are sparse in that basis.

The class of learning problems $\mathcal{P}(s)$ may then be factored as

$$P^{(s)} = \cup_q \mathcal{P}_q^{(s)}$$

where

$$\mathcal{P}_q^{(s)} = \{p | p_x = q \text{ and } m_p \in H_q^S\}.$$

The logic of the geometric approach to semi-supervised learning is as follows:

1. Unlabeled data allow us to approximate q, the eigenvalues and eigenfunctions of Δ_q, and therefore the space H^s.

2. If s is such that $\pi^{-1}(q)$ is "small" for every q, then a small number of labeled examples suffice to learn the regression function m_q.

In problems that have this general structure, we expect manifold regularization and related algorithms (that use the graph Laplacian or a suitable spectral approximation) to work well. Precise theorems showing the correctness of these algorithms for a variety of choices of s remains part of future work. The theorems in this paper establish results for some choices of s and are a step in a broader understanding of this question.

5. Conclusions

We have considered a minimax style framework within which we have investigated the potential role of manifold learning in learning from labeled and unlabeled examples. We demonstrated the natural structure of a class of problems on which knowing the manifold makes a big difference. On such problems, we see that manifold regularization is provably better than any supervised learning algorithm.

Our proof clarifies a potential source of confusion in the literature on manifold learning. We see that if data lives on an underlying manifold but this manifold is *unknown* and belongs to a class

of possible smooth manifolds, it is possible that supervised learning (classification and regression problems) may be ineffective, even impossible. In contrast, if the manifold is fixed though unknown, it may be possible to (e.g., Bickel and Li, 2007) learn effectively by a classical method suitably modified. In between these two cases lie various situations that need to be properly explored for a greater understanding of the potential benefits and limitations of manifold methods and the need for manifold learning.

Our analysis allows us to see the role of manifold regularization in semi-supervised learning in a clear way. Several algorithms using manifold and associated graph-based methods have seen some empirical success recently. Our paper provides a framework within which we may be able to analyze and possibly motivate or justify such algorithms.

Acknowledgments

I would like to thank Misha Belkin for wide ranging discussions on the themes of this paper and Andrea Caponnetto for discussions leading to the proof of Theorem 4.

References

Robert A. Adams and John J.F. Fournier. *Sobolev Spaces*, volume 140. Academic press, 2003.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. In *Eighteenth Annual Conference on Learning Theory*, pages 486–500. Springer, Bertinoro, Italy, 2005.

Mikhail Belkin and Partha Niyogi. Convergence of Laplacian eigenmaps. In *NIPS*, pages 129–136, 2006.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Peter J Bickel and Bo Li. Local polynomial regression on unknown manifolds. *IMS Lecture Notes-Monograph Series*, pages 177–186, 2007.

Vittorio Castelli and Thomas M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *Information Theory, IEEE Transactions on*, 42(6):2102–2117, 1996.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

Evarist Giné and Vladimir Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: Large sample results. *Lecture Notes-Monograph Series*, pages 238–259, 2006.

Alexander Grigoryan. Heat kernels on weighted manifolds and applications. *Contemporary Mathematics*, 398:93–191, 2006.

Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. In *COLT*, pages 470–485, 2005.

John Lafferty and Larry Wasserman. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems*. NIPS Foundation, 2007.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. the MIT Press, 2002.

Grace Wahba. *Spline Models for Observational Data*, volume 59. Society for industrial and applied mathematics, 1990.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report TR 1530, University of Wisconsin–Madison, Computer Sciences Department, 2008.

Journal of Machine Learning Research 14 (2013) 1251-1284 Submitted 10/11; Revised 9/12; Published 5/13

Random Spanning Trees and the Prediction of Weighted Graphs

Nicolò Cesa-Bianchi NICOLO.CESA-BIANCHI@UNIMI.IT
Dipartimento di Informatica
Università degli Studi di Milano
via Comelico 39
20135 - Milano, Italy

Claudio Gentile CLAUDIO.GENTILE@UNINSUBRIA.IT
DiSTA
Università dell'Insubria
via Mazzini 5
21100 - Varese, Italy

Fabio Vitale FABIO.VITALE@UNIMI.IT
Dipartimento di Informatica
Università degli Studi di Milano
via Comelico 39
20135 - Milano, Italy

Giovanni Zappella GIOVANNI.ZAPPELLA@UNIMI.IT
Dipartimento di Matematica
Università degli Studi di Milano
via Saldini 50
20133 - Milano, Italy

Editor: Shie Mannor

Abstract

We investigate the problem of sequentially predicting the binary labels on the nodes of an arbitrary weighted graph. We show that, under a suitable parametrization of the problem, the optimal number of prediction mistakes can be characterized (up to logarithmic factors) by the cutsize of a random spanning tree of the graph. The cutsize is induced by the unknown adversarial labeling of the graph nodes. In deriving our characterization, we obtain a simple randomized algorithm achieving in expectation the optimal mistake bound on any polynomially connected weighted graph. Our algorithm draws a random spanning tree of the original graph and then predicts the nodes of this tree in constant expected amortized time and linear space. Experiments on real-world data sets show that our method compares well to both global (Perceptron) and local (label propagation) methods, while being generally faster in practice.

Keywords: online learning, learning on graphs, graph prediction, random spanning trees

1. Introduction

A widespread approach to the solution of classification problems is representing data sets through a weighted graph where nodes are the data items and edge weights quantify the similarity between pairs of data items. This technique for coding input data has been applied to several domains, including Web spam detection (Herbster et al., 2009b), classification of genomic data (Tsuda and

Schölkopf, 2009), face recognition (Chang and Yeung, 2006), and text categorization (Goldberg and Zhu, 2004). In many applications, edge weights are computed through a complex data-modeling process and typically convey information that is relevant to the task of classifying the nodes.

In the sequential version of this problem, nodes are presented in an arbitrary (possibly adversarial) order, and the learner must predict the binary label of each node before observing its true value. Since real-world applications typically involve large data sets (i.e., large graphs), online learning methods play an important role because of their good scaling properties. An interesting special case of the online problem is the so-called transductive setting, where the entire graph structure (including edge weights) is known in advance. The transductive setting is interesting in that the learner has the chance of reconfiguring the graph before learning starts, so as to make the problem look easier. This data preprocessing can be viewed as a kind of regularization in the context of graph prediction.

When the graph is unweighted (i.e., when all edges have the same common weight), it was found in previous works (Herbster et al., 2005; Herbster and Pontil, 2007; Herbster, 2008; Herbster and Lever, 2009) that a key parameter to control the number of online prediction mistakes is the size of the cut induced by the unknown adversarial labeling of the nodes, that is, the number of edges in the graph whose endpoints are assigned disagreeing labels. However, while the number of mistakes is obviously bounded by the number of nodes, the cutsize scales with the number of edges. This naturally led to the idea of solving the prediction problem on a spanning tree of the graph (Cesa-Bianchi et al., 2009; Herbster et al., 2009a,b), whose number of edges is exactly equal to the number of nodes minus one. Now, since the cutsize of the spanning tree is smaller than that of the original graph, the number of mistakes in predicting the nodes is more tightly controlled. In light of the previous discussion, we can also view the spanning tree as a "maximally regularized" version of the original graph.

Since a graph has up to exponentially many spanning trees, which one should be used to maximize the predictive performance? This question can be answered by recalling the adversarial nature of the online setting, where the presentation of nodes and the assignment of labels to them are both arbitrary. This suggests to pick a tree at random among all spanning trees of the graph so as to prevent the adversary from concentrating the cutsize on the chosen tree (Cesa-Bianchi et al., 2009). Kirchoff's equivalence between the effective resistance of an edge and its probability of being included in a random spanning tree allows to express the expected cutsize of a random spanning tree in a simple form. Namely, as the sum of resistances over all edges in the cut of G induced by the adversarial label assignment.

Although the results of Cesa-Bianchi et al. (2009) yield a mistake bound for arbitrary unweighted graphs in terms of the cutsize of a random spanning tree, no general lower bounds are known for online unweighted graph prediction. The scenario gets even more uncertain in the case of weighted graphs, where the only previous papers we are aware of Herbster and Pontil (2007), Herbster (2008), and Herbster and Lever (2009) essentially contain only upper bounds. In this paper we fill this gap, and show that the expected cutsize of a random spanning tree of the graph delivers a convenient parametrization[1] that captures the hardness of the graph learning problem in the general weighted case. Given any weighted graph, we prove that any online prediction algorithm must err on a number of nodes which is at least as big as the expected cutsize of the graph's random spanning tree (which is defined in terms of the graph weights). Moreover, we exhibit a simple randomized algorithm achieving in expectation the optimal mistake bound to within logarithmic

1. Different parametrizations of the node prediction problem exist that lead to bounds which are incomparable to ours—see Section 2.

factors. This bound applies to any sufficiently connected weighted graph whose weighted cutsize is not an overwhelming fraction of the total weight.

Following the ideas of Cesa-Bianchi et al. (2009), our algorithm first extracts a random spanning tree of the original graph. Then, it predicts all nodes of this tree using a generalization of the method proposed by Herbster et al. (2009a). Our tree prediction procedure is extremely efficient: it only requires *constant* amortized time per prediction and space *linear in the number of nodes*. Again, we would like to stress that computational efficiency is a central issue in practical applications where the involved data sets can be very large. In such contexts, learning algorithms whose computation time scales quadratically, or slower, in the number of data points should be considered impractical.

As in the work by Herbster et al. (2009a), our algorithm first linearizes the tree, and then operates on the resulting line graph via a nearest neighbor rule. We show that, besides running time, this linearization step brings further benefits to the overall prediction process. In particular, similar to Herbster and Pontil (2007, Theorem 4.2), the algorithm turns out to be resilient to perturbations of the labeling, a clearly desirable feature from a practical standpoint.

In order to provide convincing empirical evidence, we also present an experimental evaluation of our method compared to other algorithms recently proposed in the literature on graph prediction. In particular, we test our algorithm against the Perceptron algorithm with Laplacian kernel by Herbster and Pontil (2007); Herbster et al. (2009b), and against a version of the label propagation algorithm by Zhu et al. (2003). These two baselines can viewed as representatives of global (Perceptron) and local (label propagation) learning methods on graphs. The experiments have been carried out on five medium-sized real-world data sets. The two tree-based algorithms (ours and the Perceptron algorithm) have been tested using spanning trees generated in various ways, including committees of spanning trees aggregated by majority votes. In a nutshell, our experimental comparison shows that predictors based on our online algorithm compare well to all baselines while being very efficient in most cases.

The paper is organized as follows. Next, we recall preliminaries and introduce our basic notation. Section 2 surveys related work in the literature. In Section 3 we prove the general lower bound relating the mistakes of any prediction algorithm to the expected cutsize of a random spanning tree of the weighted graph. In the subsequent section, we present our prediction algorithm WTA (Weighted Tree Algorithm), along with a detailed mistake bound analysis restricted to weighted trees. This analysis is extended to weighted graphs in Section 5, where we provide an upper bound matching the lower bound up to log factors on any sufficiently connected graph. In Section 6, we quantify the robustness of our algorithm to label perturbation. In Section 7, we provide the constant amortized time implementation of WTA. Based on this implementation, in Section 8 we present the experimental results. Section 9 is devoted to conclusive remarks.

1.1 Preliminaries and Basic Notation

Let $G = (V, E, W)$ be an undirected, connected, and weighted graph with n nodes and positive edge weights $w_{i,j} > 0$ for $(i, j) \in E$. A labeling of G is any assignment $y = (y_1, \ldots, y_n) \in \{-1, +1\}^n$ of binary labels to its nodes. We use (G, y) to denote the resulting labeled weighted graph.

The online learning protocol for predicting (G, y) can be defined as the following game between a (possibly randomized) learner and an adversary. The game is parameterized by the graph $G = (V, E, W)$. Preliminarily, and hidden to the learner, the adversary chooses a labeling y of G. Then the nodes of G are presented to the learner one by one, according to a permutation of V, which

is adaptively selected by the adversary. More precisely, at each time step $t = 1, \ldots, n$ the adversary chooses the next node i_t in the permutation of V, and presents it to the learner for the prediction of the associated label y_{i_t}. Then y_{i_t} is revealed, disclosing whether a mistake occurred. The learner's goal is to minimize the total number of prediction mistakes. Note that while the adversarial choice of the permutation can depend on the algorithm's randomization, the choice of the labeling is oblivious to it. In other words, the learner uses randomization to fend off the adversarial choice of labels, whereas it is fully deterministic against the adversarial choice of the permutation. The requirement that the adversary is fully oblivious when choosing labels is then dictated by the fact that the randomized learners considered in this paper make all their random choices at the beginning of the prediction process (i.e., before seeing the labels).

Now, it is reasonable to expect that prediction performance degrades with the increase of "randomness" in the labeling. For this reason, our analysis of graph prediction algorithms bounds from above the number of prediction mistakes in terms of appropriate notions of graph label *regularity*. A standard notion of label regularity is the cutsize of a labeled graph, defined as follows. A ϕ-edge of a labeled graph (G,y) is any edge (i,j) such that $y_i \neq y_j$. Similarly, an edge (i,j) is ϕ-free if $y_i = y_j$. Let $E^\phi \subseteq E$ be the set of ϕ-edges in (G,y). The quantity $\Phi_G(y) = \left|E^\phi\right|$ is the *cutsize* of (G,y), that is, the number of ϕ-edges in E^ϕ (independent of the edge weights). The *weighted cutsize* of (G,y) is defined by

$$\Phi_G^W(y) = \sum_{(i,j)\in E^\phi} w_{i,j} \ .$$

For a fixed (G,y), we denote by $r_{i,j}^W$ the effective resistance between nodes i and j of G. In the interpretation of the graph as an electric network, where the weights $w_{i,j}$ are the edge conductances, the effective resistance $r_{i,j}^W$ is the voltage between i and j when a unit current flow is maintained through them. For $(i,j) \in E$, let also $p_{i,j} = w_{i,j} r_{i,j}^W$ be the probability that (i,j) belongs to a random spanning tree T—see, for example, the monograph of Lyons and Peres (2009). Then we have

$$\mathbb{E}\,\Phi_T(y) = \sum_{(i,j)\in E^\phi} p_{i,j} = \sum_{(i,j)\in E^\phi} w_{i,j} r_{i,j}^W \ , \tag{1}$$

where the expectation $\mathbb{E}$ is over the random choice of spanning tree T. Observe the natural weight-scale independence properties of (1). A uniform rescaling of the edge weights $w_{i,j}$ cannot have an influence on the probabilities $p_{i,j}$, thereby making each product $w_{i,j} r_{i,j}^W$ scale independent. In addition, since $\sum_{(i,j)\in E} p_{i,j}$ is equal to $n-1$, irrespective of the edge weighting, we have $0 \leq \mathbb{E}\,\Phi_T(y) \leq n-1$. Hence the ratio $\frac{1}{n-1}\mathbb{E}\,\Phi_T(y) \in [0,1]$ provides a *density-independent* measure of the cutsize in G, and even allows to compare labelings on different graphs.

Now contrast $\mathbb{E}\,\Phi_T(y)$ to the more standard weighted cutsize measure $\Phi_G^W(y)$. First, $\Phi_G^W(y)$ is clearly weight-scale dependent. Second, it can be much larger than n on dense graphs, even in the unweighted $w_{i,j} = 1$ case. Third, it strongly depends on the density of G, which is generally related to $\sum_{(i,j)\in E} w_{i,j}$. In fact, $\mathbb{E}\,\Phi_T(y)$ can be much smaller than $\Phi_G^W(y)$ when there are strongly connected regions in G contributing prominently to the weighted cutsize. To see this, consider the following scenario: If $(i,j) \in E^\phi$ and $w_{i,j}$ is large, then (i,j) gives a big contribution to $\Phi_G^W(y)$ (it is easy to see that in such cases $\Phi_G^W(y)$ can be much larger than n). However, this does not necessarily happen with $\mathbb{E}\,\Phi_T(y)$. In fact, if i and j are strongly connected (i.e., if there are many disjoint paths connecting them), then $r_{i,j}^W$ is very small and so are the terms $w_{i,j} r_{i,j}^W$ in (1). Therefore, the effect of the large weight $w_{i,j}$ may often be compensated by the small probability of including (i,j) in the random spanning tree. See Figure 1 for an example.

A different way of taking into account graph connectivity is provided by the covering ball approach taken by Herbster (2008) and Herbster and Lever (2009)—see the next section.

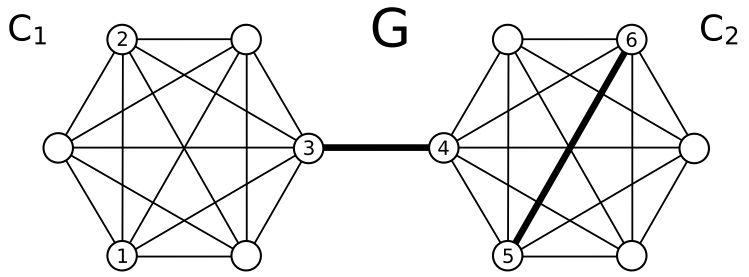


Figure 1: A barbell graph. The weight of the two thick black edges is equal to $\sqrt{V}$, all the other edges have unit weight. If the two labels y_1 and y_2 are such that $y_1 \neq y_2$, then the contribution of the edges on the left clique C_1 to the cutsizes $\Phi_G(y)$ and $\Phi_G^W(y)$ must be large. However, since the probability of including each edge of C_1 in a random spanning tree T is $O(1/|V|)$, C_1's contribution to $\mathbb{E}\,\Phi_T(y)$ is $|V|$ times smaller than $\Phi_{C_1}(y) = \Phi_{C_1}^W(y)$. If $y_3 \neq y_4$, then the contribution of edge (3,4) to $\Phi_G^W(y)$ is large. Because this edge is a bridge, the probability of including it in T is one, independent of $w_{3,4}$. Indeed, we have $p_{3,4} = w_{3,4}\, r_{3,4}^W = w_{3,4}/w_{3,4} = 1$. If $y_5 \neq y_6$, then the contribution of the right clique C_2 to $\Phi_G^W(y)$ is large. On the other hand, the probability of including edge $(5,6)$ in T is equal to $p_{5,6} = w_{5,6}\, r_{5,6}^W = O(1/\sqrt{|V|})$. Hence, the contribution of $(5,6)$ to $\mathbb{E}\,\Phi_T(y)$ is small because the large weight of $(5,6)$ is offset by the fact that nodes 5 and 6 are strongly connected (i.e., there are many different paths among them). Finally, note that $p_{i,j} = O(1/|V|)$ holds for all edges (i,j) in C_2, implying (similar to clique C_1) that C_2's contribution to $\mathbb{E}\,\Phi_T(y)$ is $|V|$ times smaller than $\Phi_{C_2}^W(y)$.

2. Related Work

With the above notation and preliminaries in hand, we now briefly survey the results in the existing literature which are most closely related to this paper. Further comments are made at the end of Section 5.

Standard online linear learners, such as the Perceptron algorithm, are applied to the general (weighted) graph prediction problem by embedding the n vertices of the graph in $\mathbb{R}^n$ through a map $i \mapsto K^{-1/2} e_i$, where $e_i \in \mathbb{R}^n$ is the i-th vector in the canonical basis of $\mathbb{R}^n$, and K is a positive definite $n \times n$ matrix. The graph Perceptron algorithm (Herbster et al., 2005; Herbster and Pontil, 2007) uses $K = L_G + 1\,1^\top$, where L_G is the (weighted) Laplacian of G and $1 = (1,\ldots,1)$. The resulting mistake bound is of the form $\Phi_G^W(y) D_G^W$, where $D_G^W = \max_{i,j} r_{i,j}^W$ is the resistance diameter of G. As expected, this bound is weight-scale independent, but the interplay between the two factors in it may lead to a vacuous result. At a given scale for the weights $w_{i,j}$, if G is dense, then we may have $D_G^W = O(1)$ while $\Phi_G^W(y)$ is of the order of n^2. If G is sparse, then $\Phi_G^W(y) = O(n)$ but then D_G^W may become as large as n.

The idea of using a spanning tree to reduce the cutsize of G has been investigated by Herbster et al. (2009b), where the graph Perceptron algorithm is applied to a spanning tree T of G. The

resulting mistake bound is of the form $\Phi_T^W(y)D_T^W$, that is, the graph Perceptron bound applied to tree T. Since $\Phi_T^W(y) \le \Phi_G^W(y)$ this bound has a smaller cutsize than the previous one. On the other hand, D_T^W can be much larger than D_G^W because removing edges may increase the resistance. Hence the two bounds are generally incomparable.

Herbster et al. (2009b) suggest to apply the graph Perceptron algorithm to the spanning tree T with smallest geodesic diameter. The geodesic diameter of a weighted graph G is defined by

$$\Delta_G^W = \max_{i,j} \min_{\Pi_{i,j}} \sum_{(r,s)\in\Pi_{i,j}} \frac{1}{w_{i,j}} ,$$

where the minimum is over all paths $\Pi_{i,j}$ between i and j. The reason behind this choice of T is that, for the spanning tree T with smallest geodesic diameter, it holds that $D_T^W \le 2\Delta_G^W$. However, one the one hand $D_G^W \le \Delta_G^W$, so there is no guarantee that $D_T^W = O\big(D_G^W\big)$, and on the other hand the adversary may still concentrate all ϕ-edges on the chosen tree T, so there is no guarantee that $\Phi_T^W(y)$ remains small either.

Herbster et al. (2009a) introduce a different technique showing its application to the case of unweighted graphs. After reducing the graph to a spanning tree T, the tree is linearized via a depth-first visit. This gives a line graph S (the so-called *spine* of G) such that $\Phi_S(y) \le 2\,\Phi_T(y)$. By running a Nearest Neighbor (NN) predictor on S, Herbster et al. (2009a) prove a mistake bound of the form $\Phi_S(y)\log\big(n/\Phi_S(y)\big) + \Phi_S(y)$. As observed by Fakcharoenphol and Kijsirikul (2008), similar techniques have been developed to solve low-congestion routing problems.

Another natural parametrization for the labels of a weighted graph that takes the graph structure into account is *clusterability*, that is, the extent to which the graph nodes can be covered by a few balls of small resistance diameter. With this inductive bias in mind, Herbster (2008) developed the Pounce algorithm, which can be seen as a combination of graph Perceptron and NN prediction. The number of mistakes has a bound of the form

$$\min_{\rho>0}\big(\mathcal{N}(G,\rho) + \Phi_G^W(y)\rho\big) , \tag{2}$$

where $\mathcal{N}(G,\rho)$ is the smallest number of balls of resistance diameter ρ it takes to cover the nodes of G. Note that the graph Perceptron bound is recovered when $\rho = D_G^W$. Moreover, observe that, unlike graph Perceptron's, bound (2) is never vacuous, as it holds uniformly for all covers of G (even the one made up of singletons, corresponding to $\rho \to 0$). A further trick for the unweighted case proposed by Herbster et al. (2009a) is to take advantage of both previous approaches (graph Perceptron and NN on line graphs) by building a binary tree on G. This "support tree" helps in keeping the diameter of G as small as possible, for example, logarithmic in the number of nodes n. The resulting prediction algorithm is again a combination of a Perceptron-like algorithm and NN, and the corresponding number of mistakes is the minimum over two earlier bounds: a NN-based bound of the form $\Phi_G(y)(\log n)^2$ and an unweighted version of bound (2).

Generally speaking, clusterability and resistance-weighted cutsize $\mathbb{E}\,\Phi_T(y)$ exploit the graph structure in different ways. Consider, for instance, a barbell graph made up of two m-cliques joined by k unweighted ϕ-edges with no endpoints in common (hence $k \le m$). This is one of the examples considered by Herbster and Lever (2009). If m is much larger than k, then bound (2) scales linearly with k (the two balls in the cover correspond to the two m-cliques). On the other hand, $\mathbb{E}\,\Phi_T(y)$ tends to be constant: Because m is much larger than k, the probability of including any ϕ-edge in T tends to $1/k$, as m increases and k stays constant. On the other hand, if k gets close to m the

resistance diameter of the graph decreases, and (2) becomes a constant. In fact, one can show that when $k = m$ even $\mathbb{E}\,\Phi_T(y)$ is a constant, independent of m. In particular, the probability that a ϕ-edge is included in the random spanning tree T is upper bounded by $\frac{3m-1}{m(m+1)}$, that is, $\mathbb{E}\,\Phi_T(y) \to 3$ when m grows large. This can be shown by computing the effective resistance of ϕ-edge (i, j) as the minimum, over all unit-strength flow functions with i as source and j as sink, of the squared flow values summed over all edges, see, for example, Lyons and Peres (2009).

When the graph at hand has a large diameter, for example, an m-line graph connected to an m-clique (this is sometimes called a "lollipop" graph) the gap between the covering-based bound (2) and $\mathbb{E}\,\Phi_T(y)$ is magnified. Yet, it is fair to say that the bounds we are about to prove for our algorithm have an extra factor, beyond $\mathbb{E}\,\Phi_T(y)$, which is logarithmic in m. A similar logarithmic factor is achieved by the combined algorithm proposed by Herbster et al. (2009a).

An even more refined way of exploiting cluster structure and connectivity in graphs is contained in the paper of Herbster and Lever (2009), where the authors provide a comprehensive study of the application of dual-norm techniques to the prediction of weighted graphs, again with the goal of obtaining logarithmic performance guarantees on large diameter graphs. In order to trade-off the contribution of cutsize Φ_G^W and resistance diameter D_G^W, the authors develop a notion of p-norm resistance. The obtained bounds are dual norm versions of the covering ball bound (2). Roughly speaking, one can select the dual norm parameter of the algorithm to obtain a logarithmic contribution from the resistance diameter at the cost of squaring the contribution due to the cutsize. This quadratic term can be further reduced if the graph is well connected. For instance, in the unweighted barbell graph mentioned above, selecting the norm appropriately leads to a bound which is constant even when $k \ll m$.

Further comments on the comparison between the results presented by Herbster and Lever (2009) and the ones in our paper are postponed to the end of Section 5.

Departing from the online learning scenario, it is worth mentioning the significantly large literature on the general problem of learning the nodes of a graph in the train/test transductive setting: Many algorithms have been proposed, including the label-consistent mincut approach of Blum and Chawla (2001), Blum et al. (2004) and a number of other "energy minimization" methods—for example, the ones by Zhu et al. (2003) and Belkin et al. (2004) of which label propagation is an instance. See the work of Bengio et al. (2006) for a relatively recent survey on this subject.

Our graph prediction algorithm is based on a random spanning tree of the original graph. The problem of drawing a random spanning tree of an arbitrary graph has a long history—see, for example, the monograph by Lyons and Peres (2009). In the unweighted case, a random spanning tree can be sampled with a random walk in expected time $O(n \ln n)$ for "most" graphs, as shown by Broder (1989). Using the beautiful algorithm of Wilson (1996), the expected time reduces to $O(n)$—see also the work of Alon et al. (2008). However, all known techniques take expected time $\Theta(n^3)$ on certain pathological graphs. In the weighted case, the above methods can take longer due to the hardness of reaching, via a random walk, portions of the graph which are connected only via light-weighted edges. To sidestep this issue, in our experiments we tested a viable fast approximation where weights are disregarded when building the spanning tree, and only used at prediction time. Finally, the space complexity for generating a random spanning tree is always linear in the graph size.

To conclude this section, it is worth mentioning that, although we exploit random spanning trees to reduce the cutsize, similar approaches can also be used to approximate the cutsize of a weighted graph by sparsification—see, for example, the work of Spielman and Srivastava (2008). However,

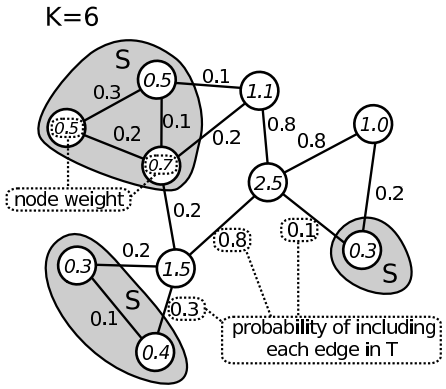


Figure 2: The adversarial strategy. Numbers on edges are the probabilities $p_{i,j}$ of those edges being included in a random spanning tree for the weighted graph under consideration. Numbers within nodes denote the weight of that node based on the $p_{i,j}$—see main text. We set the budget K to 6, hence the subset S contains the 6 nodes having smallest weight. The adversary assigns a random label to each node in S thus forcing $|S|/2$ mistakes in expectation. Then, it labels all nodes in $V \setminus S$ with a unique label, chosen in such a way as to minimize the cutsize consistent with the labels previously assigned to the nodes of S.

because the resulting graphs are not as sparse as spanning trees, we do not currently see how to use those results.

3. A General Lower Bound

This section contains our general lower bound. We show that any prediction algorithm must err at least $\frac{1}{2}\mathbb{E}\,\Phi_T(y)$ times on any weighted graph.

Theorem 1 *Let $G=(V,E,W)$ be a weighted undirected graph with n nodes and weights $w_{i,j}>0$ for $(i,j)\in E$. Then for all $K\le n$ there exists a randomized labeling y of G such that for all (deterministic or randomized) algorithms A, the expected number of prediction mistakes made by A is at least $K/2$, while $\mathbb{E}\,\Phi_T(y)<K$.*

Proof The adversary uses the weighting P induced by W and defined by $p_{i,j}=w_{i,j}r^W_{i,j}$. By (1), $p_{i,j}$ is the probability that edge (i,j) belongs to a random spanning tree T of G. Let $P_i=\sum_j p_{i,j}$ be the sum over the induced weights of all edges incident to node i. We call P_i the *weight* of node i. Let $S\subseteq V$ be the set of K nodes i in G having the smallest weight P_i. The adversary assigns a random label to each node $i\in S$. This guarantees that, no matter what, the algorithm A will make on average $K/2$ mistakes on the nodes in S. The labels of the remaining nodes in $V\setminus S$ are set either all $+1$ or all -1, depending on which one of the two choices yields the smaller $\Phi^P_G(y)$. See Figure 2 for an illustrative example. We now show that the weighted cutsize $\Phi^P_G(y)$ of this labeling y is less than K, *independent of* the labels of the nodes in S.

Since the nodes in $V\setminus S$ have all the same label, the ϕ-edges induced by this labeling can only connect either two nodes in S or one node in S and one node in $V\setminus S$. Hence $\Phi^P_G(y)$ can be written as

$$\Phi^P_G(y)=\Phi^{P,\mathrm{int}}_G(y)+\Phi^{P,\mathrm{ext}}_G(y)\ ,$$

where $\Phi_G^{P,int}(y)$ is the cutsize contribution within S, and $\Phi_G^{P,ext}(y)$ is the one from edges between S and $V \setminus S$. We can now bound these two terms by combining the definition of S with the equality $\sum_{(i,j)\in E} p_{i,j} = n-1$ as in the sequel. Let

$$P_S^{\text{int}} = \sum_{(i,j)\in E\,:\,i,j\in S} p_{i,j} \qquad \text{and} \qquad P_S^{\text{ext}} = \sum_{(i,j)\in E\,:\,i\in S,\,j\in V\setminus S} p_{i,j}\ .$$

From the very definition of P_S^{int} and $\Phi_G^{P,\text{int}}(y)$ we have $\Phi_G^{P,\text{int}}(y) \le P_S^{\text{int}}$. Moreover, from the way the labels of nodes in $V \setminus S$ are selected, it follows that $\Phi_G^{P,\text{ext}}(y) \le P_S^{\text{ext}}/2$. Finally,

$$\sum_{i\in S} P_i = 2P_S^{\text{int}} + P_S^{\text{ext}}$$

holds, since each edge connecting nodes in S is counted twice in the sum $\sum_{i\in S} P_i$. Putting everything together we obtain

$$2P_S^{\text{int}} + P_S^{\text{ext}} = \sum_{i\in S} P_i \le \frac{K}{n}\sum_{i\in V} P_i = \frac{2K}{n}\sum_{(i,j)\in E} p_{i,j} = \frac{2K(n-1)}{n}\ ,$$

the inequality following from the definition of S. Hence

$$\mathbb{E}\,\Phi_T(y) = \Phi_G^P(y) = \Phi_G^{P,\text{int}}(y) + \Phi_G^{P,\text{ext}}(y) \le P_S^{\text{int}} + \frac{P_S^{\text{ext}}}{2} \le \frac{K(n-1)}{n} < K$$

concluding the proof. ■

4. The Weighted Tree Algorithm

We now describe the Weighted Tree Algorithm (WTA) for predicting the labels of a weighted tree. In Section 5 we show how to apply WTA to the more general weighted graph prediction problem. WTA first transforms the tree into a line graph (i.e., a list), then runs a fast nearest neighbor method to predict the labels of each node in the line. Though this technique is similar to that one used by Herbster et al. (2009a), the fact that the tree is weighted makes the analysis significantly more difficult, and the practical scope of our algorithm significantly wider. Our experimental comparison in Section 8 confirms that exploiting the weight information is often beneficial in real-world graph prediction problem.

Given a labeled weighted tree (T,y), the algorithm initially creates a weighted line graph L' containing some duplicates of the nodes in T. Then, each duplicate node (together with its incident edges) is replaced by a single edge with a suitably chosen weight. This results in the final weighted line graph L which is then used for prediction. In order to create L from T, WTA performs the following *tree linearization* steps:

1. An arbitrary node r of T is chosen, and a line L' containing only r is created.

2. Starting from r, a depth-first visit of T is performed. Each time an edge (i,j) is traversed (even in a backtracking step) from i to j, the edge is appended to L' with its weight $w_{i,j}$, and j becomes the current terminal node of L'. Note that backtracking steps can create in L' at most one duplicate of each edge in T, while nodes in T may be duplicated several times in L'.

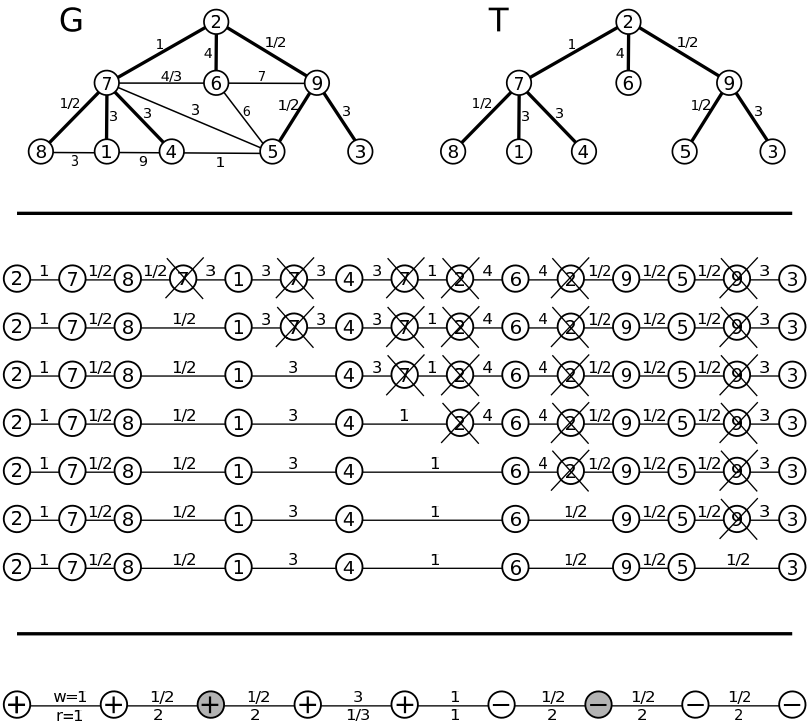


Figure 3: **Top:** A weighted graph G with 9 nodes. Initially, WTA extracts a random spanning tree T out of G. The weights on the edges in T are the same as those of G. **Middle:** The spanning tree T is linearized through a depth-first traversal starting from an arbitrary node (node 2 in this figure). For simplicity, we assume the traversal visits the siblings from left to right. As soon as a node is visited it gets stored in a line graph L' (first line graph from top). Backtracking steps produce duplicates in L' of some of the nodes in T. For instance, node 7 is the first node to be duplicated when the visit backtracks from node 8. The duplicated nodes are progressively eliminated from L' in the order of their insertion in L'. Several iterations of this node elimination process are displayed from the top to the bottom, showing how L' is progressively shrunk to the final line L (bottom line). Each line represents the elimination of a single duplicated node. The crossed nodes in each line are the nodes which are scheduled to be eliminated. Each time a new node j is eliminated, its two adjacent nodes i and k are connected by the lighter of the two edges (i, j) and (j,k). For instance: the left-most duplicated 7 is dropped by directly connecting the two adjacent nodes 8 and 1 by an edge with weight $1/2$; the right-most node 2 is eliminated by directly connecting node 6 to node 9 with an edge with weight $1/2$, and so on. Observe that this elimination procedure can be carried out *in any order* without changing the resulting list L. **Bottom:** We show WTA's prediction on the line L so obtained. In this figure, the numbers above the edges denote the edge weights, the ones below are the resistors, that is, weight reciprocals. We are at time step $t = 3$ where two labels have so far been revealed (gray nodes). WTA predicts on the remaining nodes according to a nearest neighbor rule on L, based on the resistance distance metric. All possible predictions made by WTA at this time step are shown.

3. L' is traversed once, starting from terminal r. During this traversal, duplicate nodes are eliminated as soon as they are encountered. This works as follows. Let j be a duplicate node, and (j',j) and (j,j'') be the two incident edges. The two edges are replaced by a new edge (j',j'') having weight $w_{j',j''} = \min\{w_{j',j}, w_{j,j''}\}$.[2] Let L be the resulting line.

The analysis of Section 4.1 shows that this choice of $w_{j',j''}$ guarantees that the weighted cutsize of L is smaller than twice the weighted cutsize of T.

Once L is created from T, the algorithm predicts the label of each node i_t using a nearest-neighbor rule operating on L with a *resistance distance* metric. That is, the prediction on i_t is the label of i_{s^*}, being $s^* = \operatorname{argmin}_{s<t} d(i_s, i_t)$ the previously revealed node closest to i_t, and $d(i,j) = \sum_{s=1}^{k} 1/w_{v_s,v_{s+1}}$ is the sum of the resistors (i.e., reciprocals of edge weights) along the unique path $i = v_1 \to v_2 \to \cdots \to v_{k+1} = j$ connecting node i to node j. Figure 3 gives an example of WTA at work.

4.1 Analysis of WTA

The following lemma gives a mistake bound on WTA run on any weighted line graph. Given any labeled graph (G,y), we denote by R_G^W the sum of resistors of ϕ-free edges in G,

$$R_G^W = \sum_{(i,j)\in E\setminus E^\phi} \frac{1}{w_{i,j}} \,.$$

Also, given any ϕ-free edge subset $E' \subset E \setminus E^\phi$, we define $R_G^W(\neg E')$ as the sum of the resistors of all ϕ-free edges in $E \setminus (E^\phi \cup E')$,

$$R_G^W(\neg E') = \sum_{(i,j)\in E\setminus(E^\phi\cup E')} \frac{1}{w_{i,j}} \,.$$

Note that $R_G^W(\neg E') \le R_G^W$, since we drop some edges from the sum in the defining formula.

Finally, we use $f \overset{O}{=} g$ as shorthand for $f = O(g)$. The following lemma is the starting point of our theoretical investigation—please see Appendix A for proofs.

Lemma 2 *If* WTA *is run on a labeled weighted line graph (L,y), then the total number m_L of mistakes satisfies*

$$m_L \overset{O}{=} \Phi_L(y)\left(1+\log\left(1+\frac{R_L^W(\neg E')\,\Phi_L^W(y)}{\Phi_L(y)}\right)\right) + |E'|$$

for all subsets E' of $E \setminus E^\phi$.

Note that the bound of Lemma 2 implies that, for any $K = |E'| \ge 0$, one can drop from the bound the contribution of any set of K resistors in R_L^W at the cost of adding K extra mistakes. We now provide an upper bound on the number of mistakes made by WTA on any weighted tree $T = (V,E,W)$ in terms of the number of ϕ-edges, the weighted cutsize, and R_T^W.

2. By iterating this elimination procedure, it might happen that more than two adjacent nodes get eliminated. In this case, the two surviving terminal nodes are connected in L by the lightest edge among the eliminated ones in L'.

Theorem 3 *If* WTA *is run on a labeled weighted tree (T,y), then the total number m_T of mistakes satisfies*

$$m_T \stackrel{O}{=} \Phi_T(y)\left(1+\log\left(1+\frac{R_T^W(\neg E')\,\Phi_T^W(y)}{\Phi_T(y)}\right)\right)+|E'|$$

for all subsets E' of $E\setminus E^{\phi}$.

The logarithmic factor in the above bound shows that the algorithm takes advantage of labelings such that the weights of ϕ-edges are small (thus making $\Phi_T^W(y)$ small) and the weights of ϕ-free edges are high (thus making R_T^W small). This matches the intuition behind WTA's nearest-neighbor rule according to which nodes that are close to each other are expected to have the same label. In particular, observe that the way the above quantities are combined makes the bound independent of rescaling of the edge weights. Again, this has to be expected, since WTA's prediction is scale insensitive. On the other hand, it may appear less natural that the mistake bound also depends linearly on the cutsize $\Phi_T(y)$, *independent of the edge weights*. The specialization to trees of our lower bound (Theorem 1 in Section 3) implies that this linear dependence of mistakes on the unweighted cutsize is necessary whenever the adversarial labeling is chosen from a set of labelings with bounded $\Phi_T(y)$.

5. Predicting a Weighted Graph

In order to solve the more general problem of predicting the labels of a weighted graph G, one can first generate a spanning tree T of G and then run WTA directly on T. In this case, it is possible to rephrase Theorem 3 in terms of the properties of G. Note that for each spanning tree T of G, $\Phi_T^W(y)\le\Phi_G^W(y)$ and $\Phi_T(y)\le\Phi_G(y)$. Specific choices of the spanning tree T control in different ways the quantities in the mistake bound of Theorem 3. For example, a minimum spanning tree tends to reduce the value of $\widetilde{R}_T^W$, betting on the fact that ϕ-edges are light. The next theorem relies on *random* spanning trees.

Theorem 4 *If* WTA *is run on a random spanning tree T of a labeled weighted graph (G,y), then the total number m_G of mistakes satisfies*

$$\mathbb{E}\,m_G \stackrel{O}{=} \mathbb{E}\big[\Phi_T(y)\big]\left(1+\log\left(1+w_{\max}^{\phi}\,\mathbb{E}\big[R_T^W\big]\right)\right)\ , \tag{3}$$

where $w_{\max}^{\phi}=\max_{(i,j)\in E^{\phi}} w_{i,j}$.

Note that the mistake bound in (3) is scale-invariant, since $\mathbb{E}\big[\Phi_T(y)\big]=\sum_{(i,j)\in E^{\phi}} w_{i,j}r_{i,j}^W$ cannot be affected by a uniform rescaling of the edge weights (as we said in Section 1.1), and so is the product $w_{\max}^{\phi}\mathbb{E}\big[R_T^W\big]=w_{\max}^{\phi}\sum_{(i,j)\in E\setminus E^{\phi}} r_{i,j}^W$.

We now compare the mistake bound (3) to the lower bound stated in Theorem 1. In particular, we prove that WTA is optimal (up to $\log n$ factors) on every weighted connected graph in which the ϕ-edge weights are not "superpolynomially overloaded" w.r.t. the ϕ-free edge weights. In order to rule out pathological cases, when the weighted graph is nearly disconnected, we impose the following mild assumption on the graphs being considered.

We say that a graph is *polynomially connected* if the ratio of any pair of effective resistances (even those between nonadjacent nodes) in the graph is polynomial in the total number of nodes

n. This definition essentially states that a weighted graph can be considered connected if no pair of nodes can be found which is substantially less connected than any other pair of nodes. Again, as one would naturally expect, this definition is independent of uniform weight rescaling. The following corollary shows that if WTA is not optimal on a polynomially connected graph, then the labeling must be so irregular that the total weight of ϕ-edges is an overwhelming fraction of the overall weight.

Corollary 5 *Pick any polynomially connected weighted graph G with n nodes. If the ratio of the total weight of ϕ-edges to the total weight of ϕ-free edges is bounded by a polynomial in n, then the total number of mistakes m_G made by* WTA *when run on a random spanning tree T of G satisfies*

$$\mathbb{E}\, m_G \stackrel{O}{=} \mathbb{E}\big[\Phi_T(y)\big] \log n \,.$$

Note that when the hypothesis of this corollary is not satisfied the bound of WTA is not necessarily vacuous. For example, $\mathbb{E}\big[R_T^W\big] w_{\max}^{\phi} = n^{\mathrm{polylog}(n)}$ implies an upper bound which is optimal up to $\mathrm{polylog}(n)$ factors. In particular, having a constant number of ϕ-free edges with exponentially large resistance contradicts the assumption of polynomial connectivity, but it need not lead to a vacuous bound in Theorem 4. In fact, one can use Lemma 2 to drop from the mistake bound of Theorem 4 the contribution of any set of $O(1)$ resistances in $\mathbb{E}\big[R_T^W\big] = \sum_{(i,j)\in E\setminus E^{\phi}} r_{i,j}^W$ at the cost of adding just $O(1)$ extra mistakes. This could be seen as a robustness property of WTA's bound against graphs that do not fully satisfy the connectedness assumption.

We further elaborate on the robustness properties of WTA in Section 6. In the meanwhile, note how Corollary 5 compares to the expected mistake bound of algorithms like graph Perceptron (see Section 2) on the same random spanning tree. This bound depends on the expectation of the product $\Phi_T^W(y) D_T^W$, where D_T^W is the diameter of T in the resistance distance metric. Recall from the discussion in Section 2 that these two factors are negatively correlated because $\Phi_T^W(y)$ depends linearly on the edge weights, while D_T^W depends linearly on the reciprocal of these weights. Moreover, for any given scale of the edge weights, D_T^W can be linear in the number n of nodes.

Another interesting comparison is to the covering ball bounds of Herbster (2008) and Herbster and Lever (2009). Consider the case when G is an unweighted tree with diameter D. Whereas the dual norm approach of Herbster and Lever (2009) gives a mistake bound of the form $\Phi_G(y)^2 \log D$, our approach, as well as the one by Herbster et al. (2009a), yields $\Phi_G(y) \log n$. Namely, the dependence on $\Phi_G(y)$ becomes linear rather than quadratic, but the diameter D gets replaced by n, the number of nodes in G. Replacing n by D seems to be a benefit brought by the covering ball approach.[3] More generally, one can say that the covering ball approach seems to allow to replace the extra $\log n$ term contained in Corollary 5 by more refined structural parameters of the graph (like its diameter D), but it does so at the cost of squaring the dependence on the cutsize. A typical (and unsurprising) example where the dual-norm covering ball bounds are better then the one in Corollary 5 is when the labeled graph is well-clustered. One such example we already mentioned in Section 2: On the unweighted barbell graph made up of m-cliques connected by $k \ll m$ ϕ-edges, the algorithm of Herbster and Lever (2009) has a *constant* bound on the number of mistakes (i.e., independent of both m and k), the Pounce algorithm has a *linear* bound in k, while Corollary 5 delivers a *logarithmic* bound in $m+k$. Yet, it is fair to point out that the bounds of Herbster (2008)

3. As a matter of fact, a bound of the form $\Phi_G(y) \log D$ on unweighted trees is also achieved by the direct analysis of Cesa-Bianchi et al. (2009).

and Herbster and Lever (2009) refer to computationally heavier algorithms than WTA: Pounce has a deterministic initialization step that computes the inverse Laplacian matrix of the graph (this is cubic in n, or quadratic in the case of trees), the minimum (Ψ, p)-seminorm interpolation algorithm of Herbster and Lever (2009) has no initialization, but each step requires the solution of a constrained convex optimization problem (whose time complexity was not quantified by the authors). Further comments on the time complexity of our algorithm are given in Section 7.

6. The Robustness of WTA to Label Perturbation

In this section we show that WTA is tolerant to noise, that is, the number of mistakes made by WTA on most labeled graphs (G,y) does not significantly change if a small number of labels are perturbed before running the algorithm. This is especially the case if the input graph G is polynomially connected (see Section 5 for a definition).

As in previous sections, we start off from the case when the input graph is a tree, and then we extend the result to general graphs using random spanning trees.

Suppose that the labels y in the tree (T,y) used as input to the algorithm have actually been obtained from another labeling y' of T through the perturbation (flipping) of some of its labels. As explained at the beginning of Section 4, WTA operates on a line graph L obtained through the linearization process of the input tree T. The following theorem shows that, whereas the cutsize differences $|\Phi_T^W(y) - \Phi_T^W(y')|$ and $|\Phi_T(y) - \Phi_T(y')|$ on tree T can in principle be very large, the cutsize differences $|\Phi_L^W(y) - \Phi_L^W(y')|$ and $|\Phi_L(y) - \Phi_L(y')|$ on the line graph L built by WTA are always small.

In order to quantify the above differences, we need a couple of ancillary definitions. Given a labeled tree (T,y), define $\zeta_T(K)$ to be the sum of the weights of the K heaviest edges in T,

$$\zeta_T(K) = \max_{E' \subseteq E \,:\, |E'|=K} \sum_{(i,j)\in E'} w_{i,j} \ .$$

If T is unweighted we clearly have $\zeta_T(K) = K$. Moreover, given any two labelings y and y' of T's nodes, we let $\delta(y,y')$ be the number of nodes for which the two labelings differ, that is, $\delta(y,y') = \big|\{i = 1,\ldots,n : y_i \neq y'_i\}\big|$.

Theorem 6 *On any given labeled tree (T,y) the tree linearization step of* WTA *generates a line graph L such that:*

1. $\Phi_L^W(y) \leq \min_{y' \in \{-1,+1\}^n} 2\Big(\Phi_T^W(y') + \zeta_T\big(\delta(y,y')\big)\Big)$;
2. $\Phi_L(y) \leq \min_{y' \in \{-1,+1\}^n} 2\big(\Phi_T(y') + \delta(y,y')\big)$.

In order to highlight the consequences of WTA's linearization step contained in Theorem 6, consider as a simple example an unweighted star graph (T,y) where all labels are $+1$ except for the central node c whose label is -1. We have $\Phi_T(y) = n-1$, but flipping the sign of y_c we would obtain the star graph (T,y') with $\Phi_T(y') = 0$. Using Theorem 6 (Item 2) we get $\Phi_L(y) \leq 2$. Hence, on this star graph WTA's linearization step generates a line graph with a constant number of ϕ-edges even if the input tree T has no ϕ-free edges. Because flipping the labels of a few nodes (in this case the label of c) we obtain a tree with a much more regular labeling, the labels of those nodes can naturally be seen as corrupted by noise.

The following theorem quantifies to what extent the mistake bound of WTA on trees can take advantage of the tolerance to label perturbation contained in Theorem 6. Introducing shorthands for the right-hand side expressions in Theorem 6,

$$\widetilde{\Phi}^W_T(y) = \min_{y' \in \{-1,+1\}^n} 2\Big(\Phi^W_T(y') + \zeta_T\big(\delta(y,y')\big)\Big)$$

and

$$\widetilde{\Phi}_T(y) = \min_{y' \in \{-1,+1\}^n} 2\left(\Phi_T(y') + \delta(y,y')\right) ,$$

we have the following robust version of Theorem 3.

Theorem 7 *If* WTA *is run on a weighted and labeled tree (T,y), then the total number m_T of mistakes satisfies*

$$m_T \stackrel{O}{=} \widetilde{\Phi}_T(y)\left(1+\log\left(1+\frac{R^W_T(\neg E')\,\widetilde{\Phi}^W_T(y)}{\widetilde{\Phi}_T(y)}\right)\right) + \Phi_T(y) + |E'|$$

for all subsets E' of $E \setminus E^{\phi}$.

As a simple consequence, we have the following corollary.

Corollary 8 *If* WTA *is run on a weighted and polynomially connected labeled tree (T,y), then the total number m_T of mistakes satisfies*

$$m_T \stackrel{O}{=} \widetilde{\Phi}_T(y)\log n .$$

Theorem 7 combines the result of Theorem 3 with the robustness to label perturbation of WTA's tree linearization procedure. Comparing the two theorems, we see that the main advantage of the tree linearization lies in the mistake bound dependence on the logarithmic factors occurring in the formulas: Theorem 7 shows that, when $\widetilde{\Phi}_T(y) \ll \Phi_T(y)$, then the performance of WTA can be just linear in $\Phi_T(y)$. Theorem 3 shows instead that the dependence on $\Phi_T(y)$ is in general superlinear even in cases when flipping few labels of y makes the cutsize $\Phi_T(y)$ decrease in a substantial way. In many cases, the tolerance to noise allows us to achieve even better results: Corollary 8 states that, if T is polynomially connected and there exists a labeling y' with small $\delta(y,y')$ such that $\Phi_T(y')$ is much smaller than $\Phi_T(y)$, then the performance of WTA is about the same as if the algorithm were run on (T,y'). In fact, from Lemma 2 we know that when T is polynomially connected the mistake bound of WTA mainly depends on the number of ϕ-edges in (L,y), which can often be much smaller than those in (T,y). As a simple example, let T be an unweighted star graph with a labeling y and z be the difference between the number of $+1$ and the number of -1 in y. Then the mistake bound of WTA is linear in $z\log n$ irrespective of $\Phi_T(y)$ and, specifically, irrespective of the label assigned to the central node of the star, which can greatly affect the actual value of $\Phi_T(y)$.

We are now ready to extend the above results to the case when WTA operates on a general weighted graph (G,y) via a uniformly generated random spanning tree T. As before, we need some shorthand notation. Define $\Phi^*_G(y)$ as

$$\Phi^*_G(y) = \min_{y' \in \{-1,+1\}^n}\Big(\mathbb{E}\big[\Phi_T(y')\big] + \delta(y,y')\Big) ,$$

where the expectation is over the random draw of a spanning tree T of G. The following are the robust versions of Theorem 4 and Corollary 5.

Theorem 9 *If* WTA *is run on a random spanning tree T of a labeled weighted graph (G,y), then the total number m_G of mistakes satisfies*

$$\mathbb{E}\,m_G \stackrel{O}{=} \Phi^*_G(y)\left(1+\log\left(1+w^{\phi}_{\max}\mathbb{E}\left[R^W_T\right]\right)\right)+\mathbb{E}\left[\Phi_T(y)\right]\,,$$

where $w^{\phi}_{\max} = \max_{(i,j)\in E^{\phi}} w_{i,j}$.

Corollary 10 *If* WTA *is run on a random spanning tree T of a labeled weighted graph (G,y) and the ratio of the weights of each pair of edges of G is polynomial in n, then the total number m_G of mistakes satisfies*

$$\mathbb{E}\,m_G \stackrel{O}{=} \Phi^*_G(y)\log n\,.$$

The relationship between Theorem 9 and Theorem 4 is similar to the one between Theorem 7 and Theorem 3. When there exists a labeling y' such that $\delta(y,y')$ is small and $\mathbb{E}\left[\Phi_T(y')\right] \ll \mathbb{E}\left[\Phi_T(y)\right]$, then Theorem 9 allows a linear dependence on $\mathbb{E}\left[\Phi_T(y)\right]$. Finally, Corollary 10 quantifies the advantages of WTA's noise tolerance under a similar (but stricter) assumption as the one contained in Corollary 5.

7. Implementation

As explained in Section 4, WTA runs in two phases: (i) a random spanning tree is drawn; (ii) the tree is linearized and labels are sequentially predicted. As discussed in Section 1.1, Wilson's algorithm can draw a random spanning tree of "most" unweighted graphs in expected time $O(n)$. The analysis of running times on weighted graphs is significantly more complex, and outside the scope of this paper. A naive implementation of WTA's second phase runs in time $O(n\log n)$ and requires linear memory space when operating on a tree with n nodes. We now describe how to implement the second phase to run in time $O(n)$, that is, in *constant* amortized time per prediction step.

Once the given tree T is linearized into an n-node line L, we initially traverse L from left to right. Call j_0 the left-most terminal node of L. During this traversal, the resistance distance $d(j_0,i)$ is incrementally computed for each node i in L. This makes it possible to calculate $d(i,j)$ in constant time for any pair of nodes, since $d(i,j) = |d(j_0,i)-d(j_0,j)|$ for all $i,j\in L$. On top of L, a complete binary tree T' with $2^{\lceil\log_2 n\rceil}$ leaves is constructed.[4] The k-th leftmost leaf (in the usual tree representation) of T' is the k-th node in L (numbering the nodes of L from left to right). The algorithm maintains this data-structure in such a way that at time t: (i) the subsequence of leaves whose labels are revealed at time t are connected through a (bidirectional) list B, and (ii) all the ancestors in T' of the leaves of B are marked. See Figure 4.

When WTA is required to predict the label y_{i_t}, the algorithm looks for the two closest revealed leaves i' and i'' oppositely located in L with respect to i_t. The above data structure supports this operation as follows. WTA starts from i_t and goes upwards in T' until the first marked ancestor $\mathrm{anc}(i_t)$ of i_t is reached. During this upward traversal, the algorithm marks each internal node of T' on the path connecting i_t to $\mathrm{anc}(i_t)$. Then, WTA starts from $\mathrm{anc}(i_t)$ and goes downwards in order to find the leaf $i'\in B$ closest to i_t. Note how the algorithm uses node marks for finding its way down: For instance, in Figure 4 the algorithm goes left since $\mathrm{anc}(i_t)$ was reached from below through the

4. For simplicity, this description assumes n is a power of 2. If this is not the case, we could add dummy nodes to L before building T'.

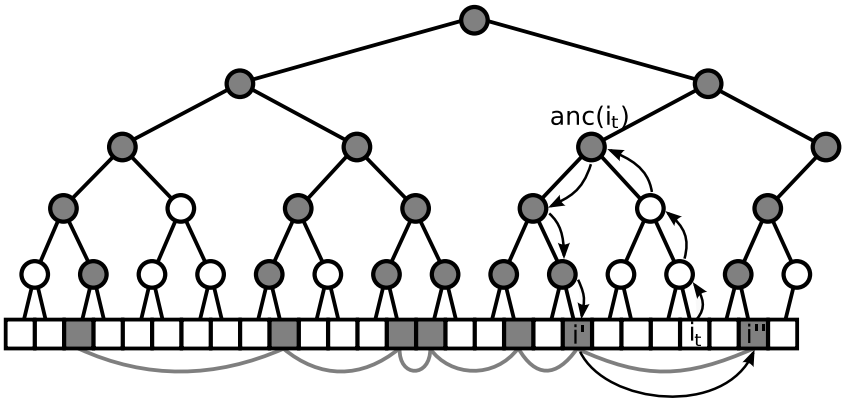


Figure 4: Constant amortized time implementation of WTA. The line L has $n = 27$ nodes (the adjacent squares at the bottom). Shaded squares are the revealed nodes, connected through a dark grey doubly-linked list B. The depicted tree T' has both unmarked (white) and marked (shaded) nodes. The arrows indicate the traversal operations performed by WTA when predicting the label of node i_t: The upward traversal stops as soon as a marked ancestor $\mathrm{anc}(i_t)$ is found, and then a downward traversal begins. Note that WTA first descends to the left, and then keeps going right all the way down. Once i' is determined, a single step within B suffices to determine i''.

right child node, and then keeps right all the way down to i'. Node i'' (if present) is then identified via the links in B. The two distances $d(i_t, i')$ and $d(i_t, i'')$ are compared, and the closest node to i_t within B is then determined. Finally, WTA updates the links of B by inserting i_t between i' and i''.

In order to quantify the amortized time per trial, the key observation is that each internal node k of T' gets visited only twice during *upward* traversals over the n trials: The first visit takes place when k gets marked for the first time, the second visit of k occurs when a subsequent upward visit also marks the other (unmarked) child of k. Once both of k's children are marked, we are guaranteed that no further upward visits to k will be performed. Since the preprocessing operations take $O(n)$, this shows that the total running time over the n trials is linear in n, as anticipated. Note, however, that the worst-case time per trial is $O(\log n)$. For instance, on the very first trial T' has to be traversed all the way up and down.

This is the way we implemented WTA on the experiments described in the next section.

8. Experiments

We now present the results of an experimental comparison on a number of real-world weighted graphs from different domains: text categorization, optical character recognition, spam detection and bioinformatics. Although our theoretical analysis is for the sequential prediction model, all experiments are carried out using a more standard train-test scenario. This makes it easy to compare WTA against popular non-sequential baselines, such as Label Propagation.

We compare our algorithm to the following other methods, intended as representatives of two different ways of coping with the graph prediction problem: global vs. local prediction.

Perceptron with Laplacian kernel: Introduced by Herbster and Pontil (2007) and here abbreviated as GPA (graph Perceptron algorithm). This algorithm sequentially predicts the nodes of a weighted graph $G = (V,E)$ after mapping V via the linear kernel based on $L_G^+ + 1\,1^\top$, where L_G is the Laplacian matrix of G. Following Herbster et al. (2009b), we run GPA on a spanning tree T of the original graph. This is because a careful computation of the Laplacian pseudoinverse of a n-node tree takes time $\Theta(n+m^2+mD)$ where m is the number of training examples plus the number of test examples (labels to predict), and D is the tree diameter—see the work of Herbster et al. (2009b) for a proof of this fact. However, in most of our experiments $m = n$, implying a running time of $\Theta(n^2)$ for GPA.

Note that GPA is a global approach, in that the graph topology affects, via the inverse Laplacian, the prediction on all nodes.

Weighted Majority Vote: Introduced here and abbreviated as WMV. Since the common underlying assumption to graph prediction algorithms is that adjacent nodes are labeled similarly, a very intuitive and fast algorithm for predicting the label of a node i is via a weighted majority vote on the available labels of the adjacent nodes. More precisely, WMV predicts using the sign of

$$\sum_{j:(i,j)\in E} y_j w_{i,j} \ ,$$

where $y_j = 0$ if node j is not available in the training set. The overall time and space requirements are both of order $\Theta(|E|)$, since we need to read (at least once) the weights of all edges. WMV is also a local approach, in the sense that prediction at each node is only affected by the labels of adjacent nodes.

Label Propagation: Introduced by Zhu et al. (2003) and here abbreviated as LABPROP. This is a batch transductive learning method based on solving a (possibly sparse) linear system of equations which requires $\Theta(|E||V|)$ time. This bad scalability prevented us from carrying out comparative experiments on larger graphs of 10^6 or more nodes. Note that WMV can be viewed as a fast approximation of LABPROP.

In our experiments, we combined WTA and GPA with spanning trees generated in different ways (note that WMV and LABPROP do not use spanning trees).

Random Spanning Tree (RST). Each spanning tree is taken with probability proportional to the product of its edge weights—see, for example, the monograph by Lyons and Peres (2009, Chapter 4). In addition, we also tested WTA combined with RST generated by ignoring the edge weights (which were then restored before running WTA). This second approach gives a prediction algorithm whose total expected running time, including the generation of the spanning tree, is $\Theta(|V|)$ on most graphs. We abbreviate this spanning tree as NWRST (non-weighted RST).

Depth-first spanning tree (DFST). This spanning tree is created via the following randomized depth-first visit: A root is selected at random, then each newly visited node is chosen with probability proportional to the weights of the edges connecting the current vertex with the adjacent nodes that have not been visited yet. This spanning tree is faster to generate than RST, and can be viewed as an approximate version of RST.

Minimum Spanning Tree (MST). The spanning tree minimizing the sum of the resistors of all edges. This is the tree whose Laplacian best approximates the Laplacian of G according to the trace norm criterion—see, for example, the paper of Herbster et al. (2009b). Note that the expected running time for the creation of a MST is $O(|E|)$, see the work by Karger et al. (1995), while the

worst-case time is $O\big(|E|\alpha(|E|,|V|)\big)$ where α is the inverse of the Ackermann function (Chazelle, 2000).

Shortest Path Spanning Tree (SPST). Herbster et al. (2009b) use the shortest path tree because it has a small diameter (at most twice the diameter of G). This allows them to better control the theoretical performance of GPA. We generated several shortest path spanning trees by choosing the root node at random, and then took the one with minimum diameter.

In order to check whether the information carried by the edge weight has predictive value for a nearest neighbor rule like WTA, we also performed a test by ignoring the edge weights during both the generation of the spanning tree and the running of WTA's nearest neighbor rule. This is essentially the algorithm analyzed by Herbster et al. (2009a), and we denote it by NWWTA (non-weighted WTA). We combined NWWTA with weighted and unweighted spanning trees. So, for instance, NWWTA+RST runs a 1-NN rule (NWWTA) that does not take edge weights into account (i.e., pretending that all weights are unitary) on a random spanning tree generated according to the actual edge weights. NWWTA+NWRST runs NWWTA on a random spanning tree that also disregards edge weights.

Finally, in order to make the classifications based on RST's more robust with respect to the variance associated with the random generation of the spanning tree, we also tested committees of RST's. For example, K*WTA+RST denotes the classifier obtained by drawing K RST's, running WTA on each one of them, and then aggregating the predictions of the K resulting classifiers via a majority vote. For our experiments we chose $K = 7, 11, 17$.

We ran our experiments on five real-world data sets.

RCV1: The first 10,000 documents (in chronological order) of Reuters Corpus Volume 1, with TF-IDF preprocessing and Euclidean normalization. This data set is available at `trec.nist.gov/data/reuters/reuters.html`.

USPS: The USPS data set with features normalized into $[0,2]$. This data set is available at `www-i6.informatik.rwth-aachen.de/~keysers/usps.html`.

KROGAN: This is a high-throughput protein-protein interaction network for budding yeast. It has been used by Krogan et al. (2006) and Pandey et al. (2007).

COMBINED: A second data set from the work of Pandey et al. (2007). It is a combination of three data sets: Gavin et al.'s (2002), Ito et al.'s (2001), and Uetz et al.'s (2000).

WEBSPAM: A large data set (110,900 nodes and 1,836,136 edges) of inter-host links created for the Web Spam Challenge 2008 (Yahoo Research and Univ. of Milan, 2007). The data set is available at `barcelona.research.yahoo.net/webspam/datasets/`. This is a weighted graph with binary labels and a pre-defined train/test split: 3,897 training nodes and 1,993 test nodes (the remaining ones being unlabeled).[5]

We created graphs from RCV1 and USPS with as many nodes as the total number of examples (x_i, y_i) in the data sets. That is, 10,000 nodes for RCV1 and 7291+2007 = 9298 for USPS. Following previous experimental settings (Zhu et al., 2003; Belkin et al., 2004), the graphs were constructed using k-NN based on the standard Euclidean distance $\|x_i - x_j\|$ between node i and node j. The weight $w_{i,j}$ was set to $w_{i,j} = \exp\big(-\|x_i - x_j\|^2 / \sigma_{i,j}^2\big)$, if j is one of the k nearest neighbors of i, and 0 otherwise. To set $\sigma_{i,j}^2$, we first computed the average square distance between i and its k nearest neighbors (call it σ_i^2), then we computed σ_j^2 in the same way, and finally set $\sigma_{i,j}^2 = \big(\sigma_i^2 + \sigma_j^2\big)/2$. We

5. We do not compare our results to those obtained in the challenge since we are only exploiting the graph (weighted) topology here, disregarding content features.

generated two graphs for each data set by running k-NN with $k = 10$ (RCV1-10 and USPS-10) and $k = 100$ (RCV1-100 and USPS-100). The labels were set using the four most frequent categories in RCV1 and all 10 categories in USPS.

In KROGAN and COMBINED we only considered the biggest connected components of both data sets, obtaining 2,169 nodes and 6,102 edges for KROGAN, and 2,871 nodes and 6,407 edges for COMBINED. In these graphs, each node belongs to one or more classes, each class representing a gene function. We selected the set of functional labels at depth one in the FunCat classification scheme of the MIPS database (Ruepp, 2004), resulting in seventeen classes per data set.

In order to associate binary classification tasks with the six non-binary data sets/graphs (RCV1-10, RCV1-100, USPS-10, USPS-100, KROGAN, COMBINED) we binarized the corresponding multiclass problems via a standard one-vs-rest scheme. We thus obtained: four binary classification tasks for RCV1-10 and RCV1-100, ten binary tasks for USPS-10 and USPS-100, seventeen binary tasks for both KROGAN and COMBINED. For a given a binary task and data set, we tried different proportions of training set and test set sizes. In particular, we used training sets of size 5%, 10%, 25% and 50%. For any given size, the training sets were randomly selected.

We report error rates and F-measures on the test set, after macro-averaging over the binary tasks. The results are contained in Tables 1–7 (Appendix B) and in Figures 5–6. Specifically, Tables 1–6 contain results for all combinations of algorithms and train/test split for the first six data sets (i.e., all but WEBSPAM).

The WEBSPAM data set is very large, and requires us a lot of computational resources in order to run experiments on this graph. Moreover, GPA has always shown inferior accuracy performance than the corresponding version of WTA (i.e., the one using the same kind of spanning tree) on all other data sets. Hence we decided not to go on any further with the refined implementation of GPA on trees we mentioned above. In Table 7 we only report test error results on the four algorithms WTA, WMV, LABPROP, and WTA with a committee of seven (nonweighted) random spanning trees.

In our experimental setup we tried to control the sources of variance in the first six data sets as follows:

1. We first generated ten random permutations of the node indices for each one of the six graphs/data sets;

2. on each permutation we generated the training/test splits;

3. we computed MST and SPST for each graph and made (for WTA, GPA, WMV, and LABPROP) one run per permutation on each of the 4+4+10+10+17+17 = 62 binary problems, averaging results over permutations and splits;

4. for each graph, we generated ten random instances for each one of RST, NWRST, DFST, and then operated as in step 2, with a further averaging over the randomness in the tree generation.

Figure 5 extracts from Tables 1–6 the error levels of the best spanning tree performers, and compared them to WMV and LABPROP. For comparison purposes, we also displayed the error levels achieved by WTA operating on a committee of seventeen random spanning trees (see below). Figure 6 (left) contains the error level on WEBSPAM reported in Table 7. Finally, Figure 6 (right) is meant to emphasize the error rate differences between RST and NWRST run with WTA.

Several interesting observations and conclusions can be drawn from our experiments.

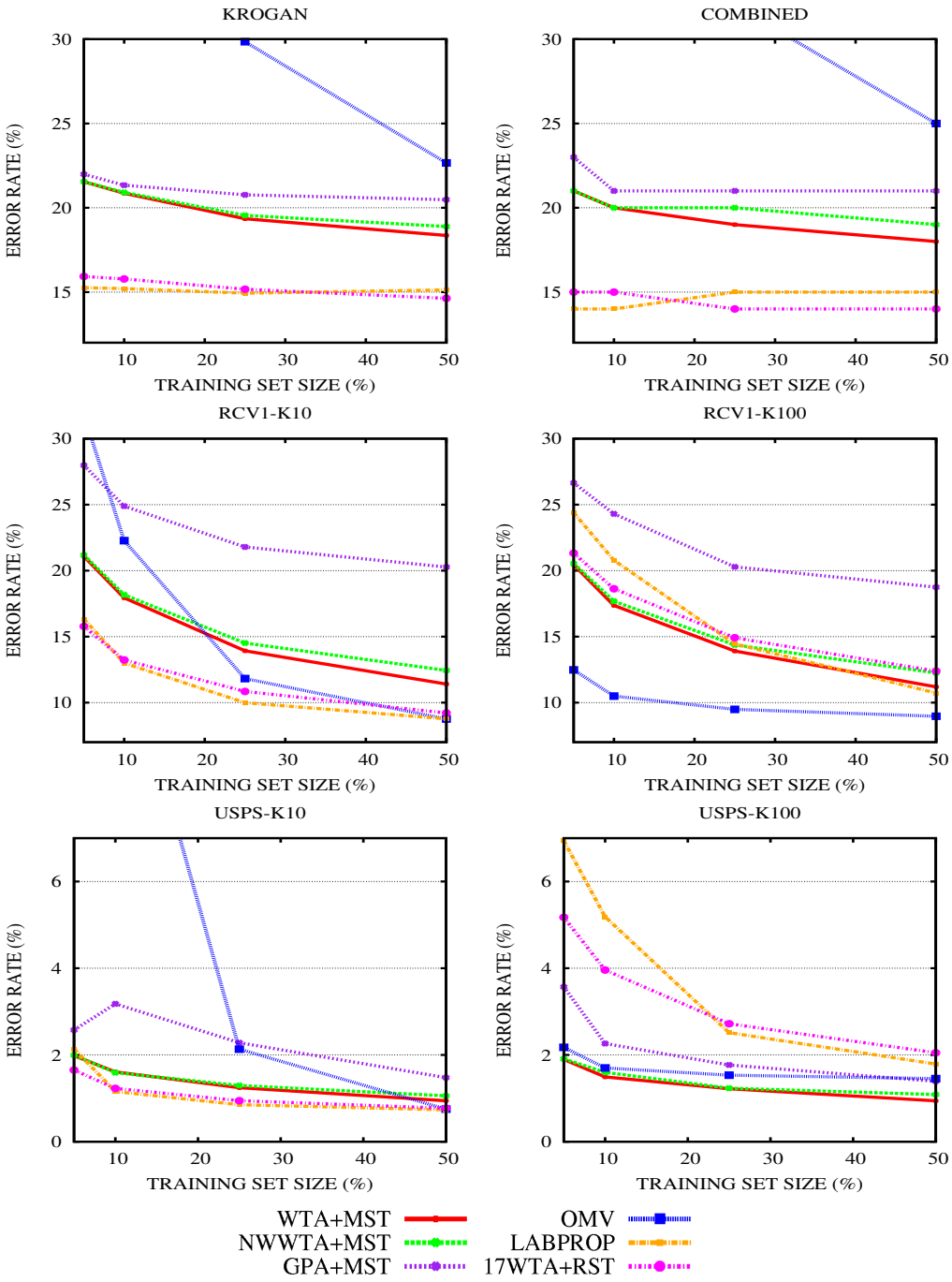


Figure 5: Macroaveraged test error rates on the first six data sets as a function of the training set size. The results are extracted from Tables 1–6 in Appendix B. Only the best performing spanning tree (i.e., MST) is shown for the algorithms that use spanning trees. These results are compared to WMV, LABPROP, and 17*WTA+RST.

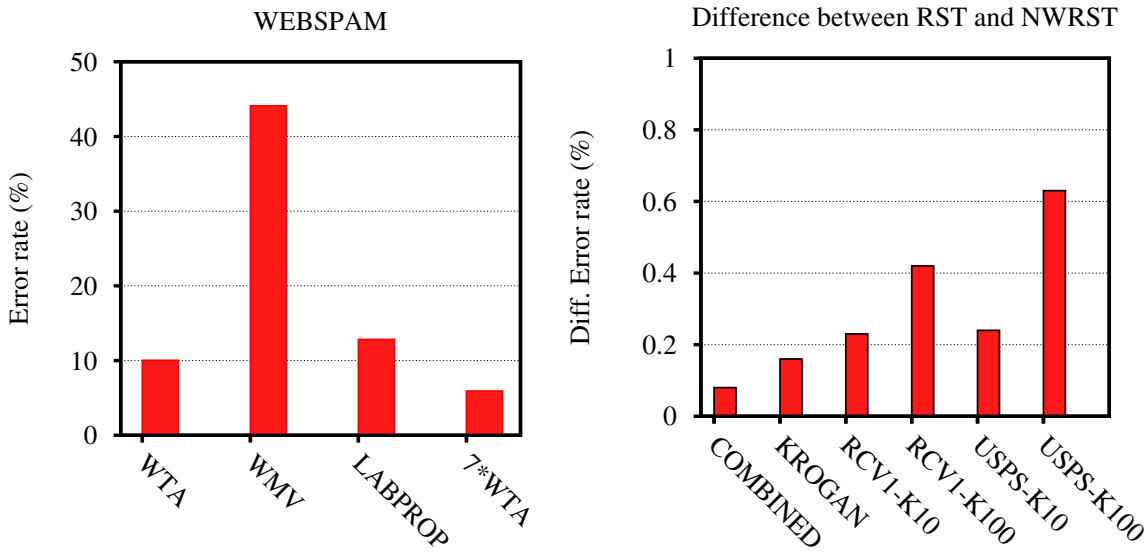


Figure 6: **Left:** Error rate levels on WEBSPAM taken from Table 7 in Appendix B. **Right:** Average error rate difference across data sets when using WTA+NWRST rather than WTA+RST.

1. WTA outperforms GPA on all data sets and with all spanning tree combinations. In particular, though we only reported aggregated results, the same relative performance pattern among the two algorithms repeats systematically over all binary classification problems. In addition, WTA runs significantly faster than GPA, requires less memory storage (linear in $|V|$, rather than quadratic), and is also fairly easy to implement.

2. By comparing NWWTA to WTA, we see that the edge weight information in the nearest neighbor rule increases accuracy, though only by a small amount.

3. WMV is a fast and accurate approximation to LABPROP when either the graph is dense (RCV1-100, and USPS-100) or the training set is comparatively large (25%–50%), although neither of the two situations often occurs in real-world applications.

4. The best performing spanning tree for both WTA and GPA is MST. This might be explained by the fact that MST tends to select light ϕ-edges of the original graph.

5. NWRST and DFST are fast approximations to RST. Though the use of NWRST and DFST does not provide theoretical performance guarantees as for RST, in our experiments they do actually perform comparably. Hence, in practice, NWRST and DFST might be viewed as fast and practical ways to generate spanning trees for WTA.

6. The prediction performance of WTA+MST is sometimes slightly inferior to LABPROP's. However, it should be stressed that LABPROP takes time $\Theta(|E||V|)$, whereas a single sweep of WTA+MST over the graph just takes time $O(|E|\alpha(|E|,|V|))$, where α is the inverse of the Ackermann function (Chazelle, 2000). Committees of spanning trees are a simple way to make WTA approach, and sometimes surpass, the performance of LABPROP. One can see

that on sparse graphs using committees gives a good performances improvement. In particular, committees of WTA can reach the same performances of LABPROP while adding just a constant factor to their (linear) time complexity.

9. Conclusions and Open Questions

We introduced and analyzed WTA, a randomized online prediction algorithm for weighted graph prediction. The algorithm uses random spanning trees and has nearly optimal performance guarantees in terms of expected prediction accuracy. The expected running time of WTA is optimal when the random spanning tree is drawn ignoring edge weights. Thanks to its linearization phase, the algorithm is also provably robust to label noise.

Our experimental evaluation shows that WTA outperforms other previously proposed online predictors. Moreover, when combined with an aggregation of random spanning trees, WTA also tends to beat standard batch predictors, such as label propagation. These features make WTA (and its combinations) suitable to large scale applications.

There are two main directions in which this work can improved. First, previous analyses (Cesa-Bianchi et al., 2009) reveal that WTA's analysis is loose, at least when the input graph is an unweighted tree with small diameter. This is the main source of the $\Omega(\ln|V|)$ slack between WTA upper bound and the general lower bound of Theorem 1. So we ask whether, at least in certain cases, this slack could be reduced. Second, in our analysis we express our upper and lower bounds in terms of the cutsize. One may object that a more natural quantity for our setting is the weighted cutsize, as this better reflects the assumption that ϕ-edges tend to be light, a natural notion of bias for weighted graphs. In more generality, we ask what are other criteria that make a notion of bias better than another one. For example, we may prefer a bias which is robust to small perturbations of the problem instance. In this sense Φ^*_G, the cutsize robust to label perturbation introduced in Section 6, is a better bias than $\mathbb{E}\,\Phi_T$. We thus ask whether there is a notion of bias, more natural and robust than $\mathbb{E}\,\Phi_T$, which captures as tightly as possible the optimal number of online mistakes on general weighted graphs. A partial answer to this question is provided by the recent work of Vitale et al. (2012). It would also be nice to tie this machinery with recent results in the active node classification setting on trees developed by Cesa-Bianchi et al. (2010).

Acknowledgments

We would like to thank the anonymous reviewers whose comments helped us to improve the presentation of this paper, and to better put it in the context of the existing literature. This work was supported in part by Google Inc. through a Google Research Award, and by the PASCAL2 Network of Excellence under EC grant 216886. This publication only reflects the authors views.

Appendix A.

This appendix contains the proofs of Lemma 2, Theorem 3, Theorem 4, Corollary 5, Theorem 6, Theorem 7, Corollary 8, Theorem 9, and Corollary 10. Notation and references are as in the main text. We start by proving Lemma 2.

Proof [Lemma 2] Let a *cluster* be any maximal sub-line of L whose edges are all ϕ-free. Then L contains exactly $\Phi_L(y)+1$ clusters, which we number consecutively, starting from one of the two

terminal nodes. Consider the k-th cluster c_k. Let v_0 be the first node of c_k whose label is predicted by WTA. After y_{v_0} is revealed, the cluster splits into two edge-disjoint sub-lines c'_k and c''_k, both having v_0 as terminal node.[6] Let v'_k and v''_k be the closest nodes to v_0 such that (i) $y_{v'_k} = y_{v''_k} \neq y_{v_0}$ and (ii) v'_k is adjacent to a terminal node of c'_k, and v''_k is adjacent to a terminal node of c''_k. The nearest neighbor prediction rule of WTA guarantees that the first mistake made on c'_k (respectively, c''_k) must occur on a node v_1 such that $d(v_0,v_1) \geq d(v_1,v'_k)$ (respectively, $d(v_0,v_1) \geq d(v_1,v''_k)$). By iterating this argument for the subsequent mistakes we see that the total number of mistakes made on cluster c_k is bounded by

$$1 + \left\lfloor \log_2 \frac{R'_k + (w'_k)^{-1}}{(w'_k)^{-1}} \right\rfloor + \left\lfloor \log_2 \frac{R''_k + (w''_k)^{-1}}{(w''_k)^{-1}} \right\rfloor$$

where R'_k is the resistance diameter of sub-line c'_k, and w'_k is the weight of the ϕ-edge between v'_k and the terminal node of c'_k closest to it (R''_k and w''_k are defined similarly). Hence, summing the above displayed expression over clusters $k = 1,\ldots,\Phi_L(y)+1$ we obtain

$$\begin{aligned}
m_L &\stackrel{O}{=} \Phi_L(y) + \sum_k \Big(\log\big(1 + R'_k w'_k\big) + \log\big(1 + R'_k w''_k\big) \Big) \\
&\stackrel{O}{=} \Phi_L(y) \left(1 + \log\left(1 + \frac{1}{\Phi_L(y)} \sum_k R'_k w'_k \right) + \log\left(1 + \frac{1}{\Phi_L(y)} \sum_k R''_k w''_k \right) \right) \\
&\stackrel{O}{=} \Phi_L(y) \left(1 + \log\left(1 + \frac{R^W_L \Phi^W_L(y)}{\Phi_L(y)} \right) \right) ,
\end{aligned}$$

where in the second step we used Jensen's inequality and in the last one the fact that $\sum_k (R'_k + R''_k) = R^W_L$ and $\max_k w'_k \stackrel{O}{=} \Phi^W_L(y)$, $\max_k w''_k \stackrel{O}{=} \Phi^W_L(y)$. This proves the lemma in the case $E' \equiv \emptyset$.

In order to conclude the proof, observe that if we take any semi-cluster c'_k (obtained, as before, by splitting cluster c_k, being $v_0 \in c_k$ the first node whose label is predicted by WTA), and pretend to split it into two sub-clusters connected by a ϕ-free edge, we could repeat the previous dichotomic argument almost verbatim on the two sub-clusters at the cost of adding an extra mistake. We now make this intuitive argument more precise. Let (i,j) be a ϕ-free edge belonging to semi-cluster c'_k, and suppose without loss of generality that i is closer to v_0 than to j. If we remove edge (i,j) then c'_k splits into two subclusters: $c'_k(v_0)$ and $c'_k(j)$, containing node v_0 and j, respectively (see Figure 7). Let $m_{c'_k}$, $m_{c'_k(v_0)}$ and $m_{c'_k(j)}$ be the number of mistakes made on c'_k, $c'_k(v_0)$ and $c'_k(j)$, respectively. We clearly have $m_{c'_k} = m_{c'_k(v_0)} + m_{c'_k(j)}$.

Let now γ'_k be the semi-cluster obtained from c'_k by contracting edge (i,j) so as to make i coincide with j (we sometimes write $i \equiv j$). Cluster γ'_k can be split into two parts which overlap only at node $i \equiv j$: $\gamma'_k(v_0)$, with terminal nodes v_0 and i (coinciding with node j), and $\gamma'_k(j)$. In a similar fashion, let $m_{\gamma'_k}$, $m_{\gamma'_k(v_0)}$, and $m_{\gamma'_k(j)}$ be the number of mistakes made on γ'_k, $\gamma'_k(v_0)$ and $\gamma'_k(j)$, respectively. We have $m_{\gamma'_k} = m_{\gamma'_k(v_0)} + m_{\gamma'_k(j)} - 1$, where the -1 takes into account that $\gamma'_k(v_0)$ and $\gamma'_k(j)$ overlap at node $i \equiv j$.

Observing now that, for each node v belonging to $c'_k(v_0)$ (and $\gamma'_k(v_0)$), the distance $d(v,v'_k)$ is smaller on γ_k than on c'_k, we can apply the above-mentioned dichotomic argument to bound the mistakes made on c'_k, obtaining $m_{\gamma'_k(v_0)} \leq m_{c'_k(v_0)}$. Since $m_{c'_k(j)} = m_{\gamma'_k(j)}$, we can finally write $m_{c'_k} = m_{c'_k(v_0)} + m_{c'_k(j)} \leq m_{\gamma'_k(v_0)} + m_{\gamma'_k(j)} = m_{\gamma'_k} + 1$. Iterating this argument for all edges in E' concludes the

6. With no loss of generality, we assume that neither of the two sub-lines is empty, so that v_0 is not a terminal node of c_k.

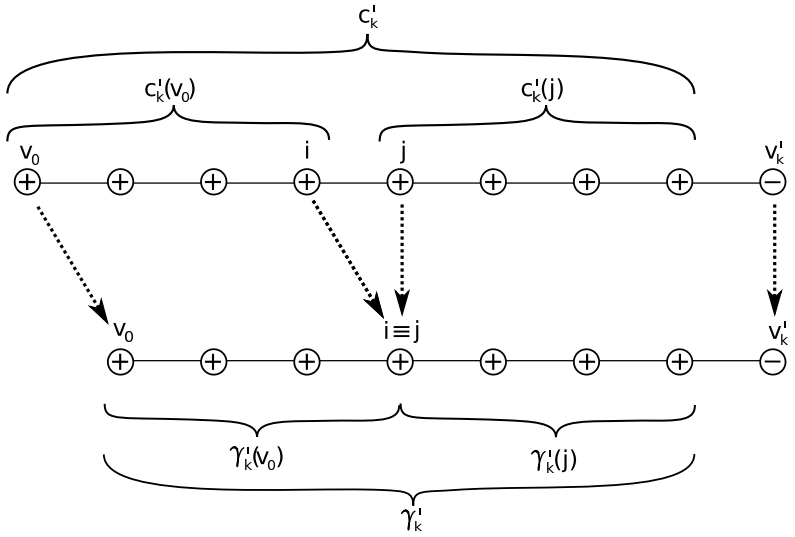


Figure 7: We illustrate the way we bound the number of mistakes on semi-cluster c'_k by dropping the resistance contribution of any (possibly very light) edge (i, j), at the cost of increasing the mistake bound on c'_k by 1. The removal of (i, j) makes c'_k split into subclusters $c'_k(v_0)$ and $c'_k(j)$. We can then drop edge (i, j) by making node i coincide with node j. The resulting semi-cluster is denoted γ'_k. This shortened version of c'_k can be viewed as split into subcluster $\gamma'_k(v_0)$ and subcluster $\gamma'_k(j)$, corresponding to $c'_k(v_0)$ and $c'_k(j)$, respectively. Now, the number of mistakes made on $c'_k(v_0)$ and $c'_k(j)$ can be bounded by those made on $\gamma'_k(v_0)$ and $\gamma'_k(j)$. Hence, we can bound the mistakes on c'_k through the ones made on γ'_k, with the addition of a single mistake, rather than two, due to the double node $i \equiv j$ of γ'_k.

proof. ■

In view of proving Theorem 3, we now prove the following two lemmas.

Lemma 11 *Given any tree T, let $E(T)$ be the edge set of T, and let $E(L')$ and $E(L)$ be the edge sets of line graphs L' and L obtained via* WTA*'s tree linearization of T. Then the following holds.*

1. *There exists a partition $\mathcal{P}_{L'}$ of $E(L')$ in pairs and a bijective mapping $\mu_{L'} : \mathcal{P}_{L'} \to E(T)$ such that the weight of both edges in each pair $S' \in \mathcal{P}_{L'}$ is equal to the weight of the edge $\mu_{L'}(S')$.*

2. *There exists a partition $\mathcal{P}_L$ of $E(L)$ in sets S such that $|S| \leq 2$, and there exists an injective mapping $\mu_L : \mathcal{P}_L \to E(T)$ such that the weight of the edges in each pair $S \in \mathcal{P}_L$ is equal to the weight of the edge $\mu_L(S)$.*

Proof We start by defining the bijective mapping $\mu_{L'} : \mathcal{P}_{L'} \to E(T)$. Since each edge (i, j) of T is traversed exactly twice in the depth-first visit that generates L',[7] once in a forward step and once in a backward step, we partition $E(L')$ in pairs S' such that $\mu_{L'}(S') = (i, j)$ if and only if S' contains the pair of distinct edges created in L' by the two traversals of (i, j). By construction, the edges in each

7. For the sake of simplicity, we are assuming here that the depth-first visit of T terminates by backtracking over all nodes on the path between the last node visited in a forward step and the root.

pair S' have weight equal to $\mu_{L'}(S')$. Moreover, this mapping is clearly bijective, since any edge of L' is created by a single traversal of an edge in T. The second mapping $\mu_L : \mathcal{P}(L) \to E(T)$ is created as follows. $\mathcal{P}_L$ is created from $\mathcal{P}_{L'}$ by removing from each $S' \in \mathcal{P}_{L'}$ the edges that are eliminated when L' is transformed into L. Note that we have $|\mathcal{P}_L| \leq |\mathcal{P}_{L'}|$ and for any $S \in \mathcal{P}_L$ there is a unique $S' \in \mathcal{P}_{L'}$ such that $S \subseteq S'$. Now, for each $S \in \mathcal{P}_L$ let $\mu_L(S) = \mu_{L'}(S')$, where S' is such that $S \subseteq S'$. Since $\mu_{L'}$ is bijective, μ_L is injective. Moreover, since the edges in S' have the same weight as the edge $\mu_{L'}(S')$, the same property holds for μ_L. ■

Lemma 12 *Let (T,y) be a labeled tree, let (L,y) be the linearization of T, and let L' be the line graph with duplicates (as described above). Then the following holds.*[8]

1. $\Phi^W_L(y) \leq \Phi^W_{L'}(y) \leq 2\Phi^W_T(y)$;
2. $\Phi_L(y) \leq \Phi_{L'}(y) \leq 2\Phi_T(y)$.

Proof From Lemma 11 (Item 1) we know that L' contains a duplicated edge for each edge of T. This immediately implies $\Phi_{L'}(y) \leq 2\Phi_T(y)$ and $\Phi^W_{L'}(y) \leq 2\Phi^W_T(y)$.

To prove the remaining inequalities, note that from the description of WTA in Section 4 (Step 3), we see that when L' is transformed into L the pair of edges (j',j) and (j,j'') of L', which are incident to a duplicate node j, gets replaced in L (together with j) by a single edge (j',j''). Now each such edge (j',j'') cannot be a ϕ-edge in L unless either (j,j') or (j,j'') is a ϕ-edge in L', and this establishes $\Phi_L(y) \leq \Phi_{L'}(y)$. Finally, if (j',j'') is a ϕ-edge in L, then its weight is not larger than the weight of the associated ϕ-edge in L' (Step 3 of WTA), and this establishes $\Phi^W_L(y) \leq \Phi^W_{L'}(y)$. ■

Recall that, given a labeled graph $G = (V,E)$ and any ϕ-free edge subset $E' \subset E \setminus E^\phi$, the quantity $R^W_G(\neg E')$ is the sum of the resistors of all ϕ-free edges in $E \setminus (E^\phi \cup E')$.

Lemma 13 *If* WTA *is run on a weighted line graph (L,y) obtained through the linearization of a given labeled tree (T,y) with edge set E, then the total number m_T of mistakes satisfies*

$$m_T \stackrel{O}{=} \Phi_L(y)\left(1+\log_2\left(1+\frac{R^W_T(\neg E')\,\Phi^W_L(y)}{\Phi_L(y)}\right)\right)+\Phi_T(y)+|E'|\,,$$

where E' is an arbitrary subset of $E \setminus E^\phi$.

Proof Lemma 11 (Item 2), exhibits an injective mapping $\mu_L : \mathcal{P} \to E$, where $\mathcal{P}$ is a partition of the edge set $E(L)$ of L, such that every $S \in \mathcal{P}$ satisfies $|S| \leq 2$. Hence, we have $|E'(L)| \leq 2|E'|$, where $E'(L)$ is the union of the pre-images of edges in E' according to μ_L—note that some edge in E' might not have a pre-image in $E(L)$. By the same argument, we also establish $|E_0(L)| \leq 2\Phi_T$, where $E_0(L)$ is the set of ϕ-free edges of L that belong to elements S of the partition $\mathcal{P}_L$ such that $\mu_L(S) \in E^\phi$.

Since the edges of L that are neither in $E_0(L)$ nor in $E'(L)$ are partitioned by $\mathcal{P}_L$ in edge sets having cardinality at most two, which in turn can be injectively mapped via μ_L to $E \setminus (E^\phi \cup E')$, we have $R^W_L\Big(\neg\big(E'(L) \cup E_0(L)\big)\Big) \leq 2R^W_T(\neg E')$. Finally, we use $|E'(L)| \leq 2|E'|$ and $|E_0(L)| \leq 2\Phi_T(y)$ (which we just established) and apply Lemma 2 with $E' \equiv E'(L) \cup E_0(L)$. This concludes the proof.

8. Item 2 in this lemma is essentially contained in the paper by Herbster et al. (2009a).

■

Proof [of Theorem 3] We use Lemma 12 to establish $\Phi_L(y) \leq 2\Phi_T(y)$ and $\Phi_L^W(y) \leq 2\Phi_T^W(y)$. We then conclude with an application of Lemma 13. ■

Lemma 14 *If* WTA *is run on a weighted line graph (L,y) obtained through the linearization of random spanning tree T of a labeled weighted graph (G,y), then the total number m_G of mistakes satisfies*

$$\mathbb{E}\, m_G \overset{O}{=} \mathbb{E}\big[\Phi_L(y)\big]\left(1+\log\left(1+w_{\max}^{\phi}\mathbb{E}\big[R_T^W\big]\right)+\mathbb{E}\big[\Phi_T(y)\big]\right) ,$$

where $w_{\max}^{\phi} = \max_{(i,j)\in E^{\phi}} w_{i,j}$.

Proof Using Lemma 13 with $E' \equiv \emptyset$ we can write

$$\begin{aligned}
\mathbb{E}\, m_G &\overset{O}{=} \mathbb{E}\left[\Phi_L(y)\left(1+\log\left(1+\frac{R_T^W\Phi_L^W(y)}{\Phi_L(y)}\right)\right)+\Phi_T\right] \\
&\overset{O}{=} \mathbb{E}\Big[\Phi_L(y)\Big(1+\log\Big(1+R_T^W w_{\max}^{\phi}\Big)\Big)+\Phi_T\Big] \\
&\overset{O}{=} \mathbb{E}\big[\Phi_L(y)\big]\Big(1+\log\Big(1+\mathbb{E}\big[R_T^W\big]w_{\max}^{\phi}\Big)\Big)+\mathbb{E}\big[\Phi_T(y)\big] ,
\end{aligned}$$

where the second equality follows from the fact that $\Phi_L^W(y) \leq \Phi_L(y) w_{\max}^{\phi}$, which in turn follows from Lemma 11, and the third one follows from Jensen's inequality applied to the concave function $(x,y) \mapsto x\Big(1+\log\Big(1+y w_{\max}^{\phi}\Big)\Big)$ for $x,y \geq 0$. ■

Proof [Theorem 4] We apply Lemma 14 and then Lemma 12 to get $\Phi_L(y) \leq 2\Phi_T(y)$. ■

Proof [Corollary 5] Let $f > \text{poly(n)}$ denote a function growing faster than any polynomial in n. Choose a polynomially connected graph G and a labeling y. For the sake of contradiction, assume that WTA makes more than $O(\mathbb{E}\big[\Phi_T(y)\big]\log n)$ mistakes on (G,y). Then Theorem 4 implies $w_{\max}^{\phi}\mathbb{E}\big[R_T^W\big] > \text{poly}(n)$. Since $\mathbb{E}\big[R_T^W\big] = \sum_{(i,j)\in E\setminus E^{\phi}} r_{i,j}^W$, we have that $w_{\max}^{\phi}\max_{(i,j)\in E\setminus E^{\phi}} r_{i,j}^W > \text{poly}(n)$. Together with the assumption of polynomial connectivity for G, this implies $w_{\max}^{\phi} r_{i,j}^W > \text{poly}(n)$ for all ϕ-free edges (i,j). By definition of effective resistance, $w_{i,j} r_{i,j}^W \leq 1$ for all $(i,j) \in E$. This gives $w_{\max}^{\phi}/w_{i,j} > \text{poly}(n)$ for all ϕ-free edges (i,j), which in turn implies

$$\frac{\sum_{(i,j)\in E^{\phi}} w_{i,j}}{\sum_{(i,j)\in E\setminus E^{\phi}} w_{i,j}} > \text{poly}(n) .$$

As this contradicts our hypothesis, the proof is concluded. ■

Proof [Theorem 6] We only prove the first part of the theorem. The proof of the second part corresponds to the special case when all weights are equal to 1.

Let $\Delta(y,y') \subseteq V$ be the set of nodes i such that $y_i \neq y'_i$. We therefore have $\delta(y,y') = |\Delta(y,y')|$. Since in a line graph each node is adjacent to at most two other nodes, the label flip of any node $j \in \Delta(y,y')$ can cause an increase of the weighted cutsize of L by at most $w_{i',j} + w_{j,i''}$, where i' and i'' are the two nodes adjacent to j in L (in the special case when j is terminal node we can set $w_{j,i''} = 0$). Hence, flipping the labels of all nodes in $\Delta(y,y')$, we have that the total cutsize increase is bounded by the sum of the weights of the $2\delta(y,y')$ heaviest edges in L, which implies

$$\Phi_L^W(y) \leq \Phi_L^W(y') + \zeta_L\big(2\delta(y,y')\big) \ .$$

By Lemma 12, $\Phi_L^W(u) \leq 2\Phi_T^W(u)$. Moreover, Lemma 11 gives an injective mapping $\mu_L : \mathcal{P}_L \to E$ (E is the edge set of T) such that the elements of $\mathcal{P}$ have cardinality at most two, and the weight of each edge $\mu_L(S)$ is the same as the weights of the edges in S. Hence, the total weight of the $2\delta(y,y')$ heaviest edges in L is at most twice the total weight of the $\delta(y,y')$ heaviest edges in T. Therefore $\zeta_L\big(2\delta(y,y')\big) \leq 2\zeta_T\big(\delta(y,y')\big)$. Hence, we have obtained

$$\Phi_L^W(y) \leq 2\Phi_T^W(y') + 2\zeta_T\big(\delta(y,y')\big) \ ,$$

concluding the proof. ■

Proof [Theorem 7] We use Theorem 6 to bound $\Phi_L(y)$ and $\Phi_L^W(y)$ in the mistake bound of Lemma 13. ■

Proof [Corollary 8] Recall that the resistance between two nodes i and j of any tree is simply the sum of the inverse weights over all edges on the path connecting the two nodes. Since T is polynomially connected, we know that the ratio of any pair of edge weights is polynomial in n. This implies that $R_L^W \Phi_L^W(y)$ is polynomial in n, too. We apply Theorem 6 to bound $\Phi_L(y)$ in the mistake bound of Lemma 2 with $E' = \emptyset$. This concludes the proof. ■

Lemma 15 *If* WTA *is run on a line graph L obtained by linearizing a random spanning tree T of a labeled and weighted graph (G,y), then we have*

$$\mathbb{E}\big[\Phi_L(y)\big] \stackrel{O}{=} \Phi_G^*(y) \ .$$

Proof Recall that Theorem 6 holds for any spanning tree T of G. Thus it suffices to apply part 2 of Theorem 6 and use $\mathbb{E}\big[\min X\big] \leq \min \mathbb{E}[X]$. ■

Proof [Theorem 9] We apply Lemma 15 to bound $\mathbb{E}\big[\Phi_L(y)\big]$ in Lemma 14. ■

Proof [Corollary 10] We can use Lemma 2 with the setting $E' \equiv \emptyset$, and bound $\mathbb{E}\big[\Phi_L(y)\big]$ via Lemma 15. To conclude, observe that since the ratio of the weights of any pair of edges in G is polynomial in n, then $R_L^W \Phi_L^W(y)$ is polynomial in n, too. ■

Appendix B.

This appendix summarizes all our experimental results. For each combination of data set, algorithm, and train/test split, we provide macro-averaged error rates and F-measures on the test set. The algorithms are WTA, NWWTA, and GPA (all combined with various spanning trees), WMV, LABPROP, and WTA run with committees of random spanning trees. WEBSPAM was too large a data set to perform as thorough an investigation. Hence we only report test error results on the four algorithms WTA, WMV, LABPROP, and WTA with a committee of 7 (nonweighted) random spanning trees.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	25.54	0.81	22.67	0.84	19.06	0.86	16.57	0.88
WTA+NWRST	25.81	0.81	22.70	0.83	19.24	0.86	17.00	0.87
WTA+MST	21.09	0.84	17.94	0.87	13.93	0.90	11.40	0.91
WTA+SPST	25.47	0.81	22.65	0.83	19.31	0.86	17.24	0.87
WTA+DFST	26.02	0.81	22.34	0.84	17.73	0.87	14.89	0.89
NWWTA+RST	25.28	0.81	22.45	0.84	19.12	0.86	17.16	0.87
NWWTA+NWRST	25.97	0.81	23.14	0.83	19.54	0.86	17.84	0.87
NWWTA+MST	21.18	0.84	18.17	0.87	14.51	0.89	12.44	0.91
NWWTA+SPST	25.49	0.81	22.81	0.83	19.64	0.86	17.55	0.87
NWWTA+DFST	26.08	0.81	22.82	0.83	17.93	0.87	15.64	0.88
GPA+RST	32.75	0.75	29.85	0.78	27.67	0.80	24.44	0.82
GPA+NWRST	34.27	0.74	30.36	0.78	28.90	0.79	25.99	0.81
GPA+MST	27.98	0.79	24.89	0.82	21.80	0.84	20.27	0.85
GPA+SPST	27.18	0.79	25.13	0.82	22.20	0.84	20.27	0.85
GPA+DFST	47.11	0.61	45.65	0.64	43.08	0.66	38.20	0.71
7*WTA+RST	17.40	0.87	14.85	0.90	12.15	0.91	10.39	0.92
7*WTA+NWRST	17.81	0.87	15.15	0.89	12.51	0.91	10.92	0.92
11*WTA+RST	16.40	0.88	13.86	0.90	11.38	0.92	9.71	0.93
11*WTA+NWRST	16.78	0.88	14.22	0.90	11.73	0.92	10.20	0.93
17*WTA+RST	15.78	0.89	13.23	0.91	10.85	0.92	9.22	0.94
17*WTA+NWRST	16.07	0.89	13.55	0.90	11.18	0.92	9.65	0.93
WMV	31.82	0.76	22.27	0.84	11.82	0.91	8.76	0.93
LABPROP	16.33	0.89	13.00	0.91	10.00	0.93	8.77	0.94

Table 1: RCV1-10. Average error rate and F-measure on 4 classes.

References

N. Alon, C. Avin, M. Koucký, G. Kozma, Z. Lotker, and M.R. Tuttle. Many random walks are faster than one. In *Proc. of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 119–128. Springer, 2008.

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Proc. of the 17th Annual Conference on Learning Theory*, pages 624–638. Springer, 2004.

Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	32.03	0.77	29.36	0.79	26.09	0.81	23.25	0.83
WTA+NWRST	32.05	0.77	29.89	0.78	26.65	0.80	23.82	0.83
WTA+MST	20.45	0.85	17.36	0.87	13.91	0.90	11.19	0.92
WTA+SPST	29.26	0.79	27.06	0.80	24.96	0.82	23.17	0.83
WTA+DFST	32.03	0.77	28.89	0.79	24.18	0.82	20.57	0.85
NWWTA+RST	31.72	0.77	29.46	0.78	26.20	0.81	24.04	0.82
NWWTA+NWRST	32.52	0.76	29.95	0.78	26.88	0.80	24.84	0.82
NWWTA+MST	20.54	0.85	17.68	0.87	14.37	0.89	12.25	0.91
NWWTA+SPST	29.28	0.79	27.13	0.80	25.16	0.82	23.72	0.83
NWWTA+DFST	32.05	0.77	28.81	0.79	24.14	0.82	21.28	0.84
GPA+RST	36.47	0.73	35.33	0.74	33.81	0.75	32.32	0.76
GPA+NWRST	38.26	0.72	35.91	0.73	35.20	0.74	32.73	0.76
GPA+MST	26.65	0.81	24.30	0.82	20.29	0.85	18.75	0.86
GPA+SPST	32.43	0.74	28.00	0.78	26.61	0.79	25.77	0.80
GPA+DFST	48.35	0.61	47.85	0.61	44.78	0.65	41.12	0.68
7*WTA+RST	23.30	0.84	20.55	0.86	16.87	0.88	14.34	0.90
7*WTA+NWRST	23.64	0.84	20.77	0.86	17.27	0.88	14.81	0.90
11*WTA+RST	22.06	0.85	19.39	0.87	15.63	0.89	13.20	0.91
11*WTA+NWRST	22.29	0.85	19.54	0.87	16.09	0.89	13.61	0.91
17*WTA+RST	21.33	0.86	18.62	0.88	14.91	0.90	12.39	0.92
17*WTA+NWRST	21.49	0.86	18.86	0.87	15.29	0.89	12.78	0.91
WMV	12.48	0.91	10.50	0.93	9.49	0.93	8.96	0.94
LABPROP	24.39	0.85	20.78	0.87	14.45	0.91	10.73	0.93

Table 2: RCV1-100. Average error rate and F-measure on 4 classes.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. of the 18th International Conference on Machine Learning*, pages 19–26. Morgan Kaufmann, 2001.

A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *Proc. of the 21st International Conference on Machine Learning*, pages 97–104, 2004.

A. Broder. Generating random spanning trees. In *Proc. of the 30th Annual Symposium on Foundations of Computer Science*, pages 442–447. IEEE Press, 1989.

N. Cesa-Bianchi, C. Gentile, and F. Vitale. Fast and optimal prediction of a labeled tree. In *Proceedings of the 22nd Annual Conference on Learning Theory*. Omnipress, 2009.

N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. Active learning on trees and graphs. In *Proceedings of the 23rd Annual Conference on Learning Theory*. Omnipress, 2010.

H. Chang and D.Y. Yeung. Graph Laplacian kernels for object classification from a single example. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2011–2016. IEEE Press, 2006.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	5.32	0.97	4.28	0.98	3.08	0.98	2.36	0.99
WTA+NWRST	5.65	0.97	4.51	0.97	3.29	0.98	2.56	0.98
WTA+MST	1.98	0.99	1.61	0.99	1.24	0.99	0.94	0.99
WTA+SPST	6.25	0.97	4.72	0.97	3.37	0.98	2.60	0.99
WTA+DFST	6.43	0.96	4.60	0.97	2.92	0.98	2.04	0.99
NWWTA+RST	5.31	0.97	4.25	0.98	3.19	0.98	2.70	0.99
NWWTA+NWRST	5.95	0.97	4.65	0.97	3.45	0.98	2.92	0.98
NWWTA+MST	1.99	0.99	1.59	0.99	1.29	0.99	1.06	0.99
NWWTA+SPST	6.30	0.96	4.83	0.97	3.50	0.98	2.84	0.98
NWWTA+DFST	6.49	0.96	4.59	0.97	3.09	0.98	2.35	0.99
GPA+RST	12.64	0.93	8.53	0.95	6.65	0.96	5.65	0.97
GPA+NWRST	12.53	0.93	9.05	0.95	6.90	0.96	5.19	0.97
GPA+MST	2.58	0.99	3.18	0.98	2.28	0.99	1.48	0.99
GPA+SPST	7.64	0.96	6.26	0.96	4.13	0.98	3.55	0.98
GPA+DFST	42.77	0.70	39.39	0.73	32.38	0.79	20.53	0.87
7*WTA+RST	2.09	0.99	1.56	0.99	1.14	0.99	0.90	0.99
7*WTA+NWRST	2.35	0.99	1.75	0.99	1.26	0.99	1.02	0.99
11*WTA+RST	1.84	0.99	1.35	0.99	1.01	0.99	0.82	1.00
11*WTA+NWRST	2.05	0.99	1.53	0.99	1.14	0.99	0.91	0.99
17*WTA+RST	1.65	0.99	1.23	0.99	0.95	0.99	0.77	1.00
17*WTA+NWRST	1.87	0.99	1.39	0.99	1.06	0.99	0.85	1.00
WMV	24.84	0.85	12.28	0.93	2.13	0.99	0.75	1.00
LABPROP	2.14	0.99	1.16	0.99	0.85	0.99	0.73	1.00

Table 3: USPS-10. Average error rate and F-measure on 10 classes.

B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. In *Journal of the ACM*, 47(6):1028–1047, 2000.

J. Fakcharoenphol and B. Kijsirikul. Low congestion online routing and an improved mistake bound for online prediction of graph labeling. *CoRR*, abs/0809.2075, 2008.

A.-C. Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.

A. Goldberg and X. Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based algorithms for Natural Language Processing*, 2004.

M. Herbster. Exploiting cluster-structure to predict the labeling of a graph. In *Proc. of the 19th International Conference on Algorithmic Learning Theory*, pages 54–69. Springer, 2008.

M. Herbster and G. Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proc. of the 22nd Annual Conference on Learning Theory*. Omnipress, 2009.

M. Herbster and M. Pontil. Prediction on a graph with the Perceptron. In *Advances in Neural Information Processing Systems 21*, pages 577–584. MIT Press, 2007.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	9.62	0.95	8.29	0.95	6.55	0.96	5.36	0.97
WTA+NWRST	10.32	0.94	9.00	0.95	7.17	0.96	5.83	0.97
WTA+MST	1.90	0.99	1.49	0.99	1.22	0.99	0.94	0.99
WTA+SPST	8.68	0.95	7.27	0.96	5.78	0.97	4.88	0.97
WTA+DFST	10.36	0.94	8.13	0.96	5.62	0.97	4.21	0.98
NWWTA+RST	9.71	0.95	8.38	0.95	6.78	0.96	5.89	0.97
NWWTA+NWRST	10.39	0.94	9.08	0.95	7.46	0.96	6.45	0.96
NWWTA+MST	1.91	0.99	1.60	0.99	1.23	0.99	1.09	0.99
NWWTA+SPST	8.76	0.95	7.46	0.96	5.94	0.97	5.28	0.97
NWWTA+DFST	10.46	0.94	8.30	0.95	6.00	0.97	4.65	0.97
GPA+RST	14.81	0.91	13.38	0.92	11.94	0.93	9.81	0.94
GPA+NWRST	17.34	0.90	13.68	0.92	11.39	0.94	11.46	0.94
GPA+MST	3.57	0.98	2.26	0.99	1.77	0.99	1.39	0.99
GPA+SPST	8.42	0.95	7.94	0.95	7.20	0.96	5.71	0.97
GPA+DFST	46.09	0.67	42.59	0.71	37.66	0.75	28.45	0.82
7*WTA+RST	5.28	0.97	4.24	0.98	3.05	0.98	2.37	0.99
7*WTA+NWRST	5.82	0.97	4.73	0.97	3.48	0.98	2.69	0.98
11*WTA+RST	5.07	0.97	3.96	0.98	2.76	0.99	2.11	0.99
11*WTA+NWRST	5.55	0.97	4.38	0.98	3.14	0.98	2.40	0.99
17*WTA+RST	5.17	0.97	3.96	0.98	2.72	0.99	2.05	0.99
17*WTA+NWRST	7.60	0.96	6.38	0.97	4.68	0.97	3.32	0.98
WMV	2.17	0.99	1.70	0.99	1.53	0.99	1.45	0.99
LABPROP	6.94	0.96	5.19	0.97	2.51	0.99	1.79	0.99

Table 4: USPS-100. Average error rate and F-measure on 10 classes.

M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *Proc. of the 22nd International Conference on Machine Learning*, pages 305–312. ACM Press, 2005.

M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In *Advances in Neural Information Processing Systems 22*, pages 649–656. MIT Press, 2009a.

M. Herbster, M. Pontil, and S. Rojas-Galeano. Fast prediction on a tree. In *Advances in Neural Information Processing Systems 22*, pages 657–664. MIT Press, 2009b.

T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569–4574, 2001.

D. Karger, P. Klein, and R. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. In *Journal of the ACM*, 42: 321–328. ACM, 1995b.

N.J. Krogan et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440:637–643, 2006.

R. Lyons and Y. Peres. Probability on trees and networks. Manuscript, 2009.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	21.73	0.86	21.37	0.86	19.89	0.87	19.09	0.88
WTA+NWRST	21.86	0.86	21.50	0.86	20.03	0.87	19.33	0.88
WTA+MST	21.55	0.86	20.86	0.87	19.35	0.88	18.36	0.88
WTA+SPST	21.86	0.86	21.58	0.86	20.38	0.87	19.40	0.88
WTA+DFST	21.78	0.86	21.22	0.86	19.88	0.87	18.60	0.88
NWWTA+RST	21.83	0.86	21.43	0.86	20.08	0.87	19.64	0.88
NWWTA+NWRST	21.98	0.86	21.55	0.86	20.26	0.87	19.75	0.87
NWWTA+MST	21.55	0.86	20.91	0.87	19.55	0.88	18.89	0.88
NWWTA+SPST	21.86	0.86	21.57	0.86	20.50	0.87	19.81	0.87
NWWTA+DFST	21.79	0.86	21.33	0.86	20.00	0.87	19.09	0.88
GPA+RST	22.70	0.85	22.75	0.85	22.14	0.86	21.28	0.86
GPA+NWRST	23.83	0.84	23.28	0.85	22.48	0.85	21.53	0.86
GPA+MST	21.99	0.86	21.34	0.86	20.77	0.86	20.48	0.87
GPA+SPST	22.33	0.84	21.34	0.86	20.71	0.86	20.74	0.86
GPA+DFST	39.77	0.72	31.93	0.78	25.70	0.83	24.09	0.84
7*WTA+RST	16.83	0.90	16.63	0.90	15.78	0.90	15.29	0.90
7*WTA+NWRST	16.85	0.90	16.60	0.90	15.89	0.90	15.41	0.90
11*WTA+RST	16.28	0.90	16.11	0.90	15.36	0.91	14.92	0.91
11*WTA+NWRST	16.28	0.90	16.08	0.90	15.55	0.90	14.99	0.91
17*WTA+RST	15.93	0.90	15.78	0.90	15.17	0.91	14.63	0.91
17*WTA+NWRST	15.98	0.90	15.69	0.91	15.23	0.91	14.68	0.91
WMV	42.98	0.70	38.88	0.73	29.85	0.80	22.66	0.85
LABPROP	15.26	0.91	15.21	0.91	14.94	0.91	15.13	0.91

Table 5: KROGAN. Average error rate and F-measure on 17 classes.

G. Pandey, M. Steinbach, R. Gupta, T. Garg, and V. Kumar. Association analysis-based transformations for protein interaction networks: a function prediction case study. In *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 540–549. ACM Press, 2007.

Yahoo! Research (Barcelona) and Laboratory of Web Algorithmics (Univ. of Milan). Web Spam Collection. URL `barcelona.research.yahoo.net/webspam/datasets/`.

A. Ruepp. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proc. of the 40th annual ACM symposium on Theory of computing*, pages 563–568. ACM Press, 2008.

H. Shin K. Tsuda and B. Schölkopf. Protein functional class prediction with a combined graph. *Expert Systems with Applications*, 36:3284–3292, 2009.

P. Uetz et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 6770(403):623–627, 2000.

Train/test split	5%		10%		25%		50%	
Predictors	Error	F	Error	F	Error	F	Error	F
WTA+RST	21.68	0.86	21.05	0.87	20.08	0.87	18.99	0.88
WTA+NWRST	21.47	0.87	21.29	0.86	20.18	0.87	19.17	0.88
WTA+MST	21.57	0.86	20.63	0.87	19.61	0.88	18.37	0.88
WTA+SPST	21.39	0.87	21.34	0.86	20.52	0.87	19.57	0.88
WTA+DFST	21.88	0.86	21.09	0.87	19.82	0.87	18.83	0.88
NWWTA+RST	21.50	0.87	21.15	0.87	20.43	0.87	19.95	0.87
NWWTA+NWRST	21.61	0.86	21.26	0.87	20.52	0.87	20.09	0.87
NWWTA+MST	21.53	0.86	20.95	0.87	20.35	0.87	19.81	0.88
NWWTA+SPST	21.37	0.87	21.06	0.87	20.55	0.87	20.06	0.87
NWWTA+DFST	21.88	0.86	21.05	0.87	20.50	0.87	19.74	0.88
GPA+RST	23.56	0.85	22.27	0.86	21.86	0.86	21.68	0.86
GPA+NWRST	23.91	0.85	23.11	0.85	22.47	0.86	21.30	0.86
GPA+MST	23.32	0.85	21.60	0.86	21.77	0.86	21.67	0.86
GPA+SPST	22.55	0.85	21.89	0.85	21.64	0.85	21.70	0.85
GPA+DFST	41.69	0.71	30.82	0.79	26.75	0.82	23.56	0.84
7*WTA+RST	16.39	0.90	16.09	0.90	15.77	0.91	15.29	0.91
7*WTA+NWRST	16.35	0.90	16.10	0.90	15.77	0.90	15.47	0.91
11*WTA+RST	15.89	0.91	15.61	0.91	15.32	0.91	14.84	0.91
11*WTA+NWRST	15.82	0.91	15.57	0.91	15.34	0.91	14.98	0.91
17*WTA+RST	15.54	0.91	15.31	0.91	14.97	0.91	14.55	0.91
17*WTA+NWRST	15.45	0.91	15.29	0.91	15.05	0.91	14.66	0.91
WMV	44.74	0.68	40.75	0.72	32.97	0.78	25.28	0.84
LABPROP	14.93	0.91	14.98	0.91	15.23	0.91	15.31	0.90

Table 6: COMBINED. Average error rate and F-measure on 17 classes.

Predictors	Error	F
WTA+NWRST	10.03	0.95
3*WTA+NWRST	6.44	0.97
7*WTA+NWRST	5.91	0.97
WMV	44.1	0.71
LABPROP	12.84	0.93

Table 7: WEBSPAM. Test set error rate and F-measure. WTA operates only on NWRST.

F. Vitale, N. Cesa-Bianchi, C. Gentile, and G. Zappella. See the tree through the lines: the Shazoo algorithm. In *Proc. of the 25th Annual Conference on Neural Information Processing Systems*, pages 1584-1592. Curran Associates, 2012.

D.B. Wilson. Generating random spanning trees more quickly than the cover time. In *Proc. of the 28th ACM Symposium on the Theory of Computing*, pages 296–303. ACM Press, 1996.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.