Journal of Machine Learning Research 7 (2006) 2481-2514 Submitted 1/06; Revised 5/06; Published 11/06

Estimation of Gradients and Coordinate Covariation in Classification

Sayan Mukherjee SAYAN@STAT.DUKE.EDU
Qiang Wu QIANG@STAT.DUKE.EDU
Institute for Genome Sciences & Policy
Institute of Statistics and Decision Sciences
Department of Computer Science
Duke University
Durham, NC 27708, USA

Editor: Isabelle Guyon

Abstract

We introduce an algorithm that simultaneously estimates a classification function as well as its gradient in the supervised learning framework. The motivation for the algorithm is to find salient variables and estimate how they covary. An efficient implementation with respect to both memory and time is given. The utility of the algorithm is illustrated on simulated data as well as a gene expression data set. An error analysis is given for the convergence of the estimate of the classification function and its gradient to the true classification function and true gradient.

Keywords: Tikhnonov regularization, variable selection, reproducing kernel Hilbert space, generalization bounds, classification

1. Introduction

The advent of data sets with many variables or coordinates in the biological and physical sciences has driven the use of a variety of machine learning approaches based on Tikhonov regularization (global shrinkage estimators in the statistics literature) such as support vector machines (SVMs) (Vapnik, 1998) and regularized least square classification (Poggio and Girosi, 1990). These algorithms have been very successful in classification (binary regression) problems.

In a number of applications, such as the analysis of gene expression data, classical questions from statistical modeling of which variables are of relevance and how these variables interact arise. In the context of genomic data an objective of the analysis is to build an interpretable model of the biological process giving rise to the data. An example of this is that genes co-regulated by a biological pathway may be modeled as features that covary. Estimation of feature covariation is not considered in standard regression or classification methods that allow for variable selection: recursive feature elimination (RFE) (Guyon et al., 2002), least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), and basis pursuits denoising (Chen et al., 1999). Gradient information was used in Hermes and Buhmann (2000) and Evgeniou et al. (2000a) to select features via a sensitivity analysis on the gradient of the SVM solution. This approach does not directly estimate the gradient and its shortcomings will be described in Remark 3. Statistical models based on shrinkage or regularization were applied to the problem of learning coordinate covariation and relevance for regression problems in Mukherjee and Zhou (2006). We extend this approach to the

binary regression or classification setting by simultaneously estimating the classification function as well as its gradient.

1.1 Review on Convex Risk Minimization Approach for Classification

In this subsection we review the convex risk minimization approach.

Let X be a compact metric space and $Y = \{1,-1\}$. Let $\rho(x,y)$ be a probability distribution on $Z := X \times Y$ and $\mathbf{z} = \{(x_i,y_i)\}_{i=1}^m \in Z^m$ a random sample independently drawn according to $\rho(x,y)$.

Convex risk minimization methods, which include support vector machines (SVMs) and boosting as typical examples, have been successful in a variety of classification problems. This approach involves a convex loss function ϕ and learns a real-valued classification function from a given sample $\mathbf{z} = \{(x_i,y_i)\}_{i=1}^m$ by minimizing the convex empirical risk functional in a hypothesis space $\mathcal{H}$ (often with a regularization or penalty term):

$$f_{\mathbf{z}} = \arg\min_{f\in\mathcal{H}} \Big\{ \frac{1}{m}\sum_{i=1}^m \phi(y_i f(x_i)) \Big\}. \tag{1}$$

The loss function may take the form of the hinge loss $\phi(t) = (1-t)_+$ in SVMs and logistic loss $\phi(t) = \log(1+e^{-t})$ in boosting. Define the expected error of a function f as

$$\mathcal{R}(f) = \int \phi(yf(x))\, d\rho(x,y),$$

and the real-valued classification function as the function in $L^2_{\rho_X}$, where ρ_X is the marginal distribution on x, that minimizes

$$f_\phi = \arg\min_{f\in L^2_{\rho_X}} \mathcal{R}(f).$$

Under certain conditions (Vapnik, 1998; Bartlett et al., 2005) $\mathrm{sgn}[f_\phi]$ is a Bayes optimal classifier. Extensive investigation in learning theory (Cortes and Vapnik, 1995; Vapnik, 1998; Evgeniou et al., 2000b; Schoelkopf and Smola, 2001; Zhang, 2004; Bartlett et al., 2005; Wu and Zhou, 2005) has shown that $\mathcal{R}(f_{\mathbf{z}}) \to \mathcal{R}(f_\phi)$, which implies that the error of $\mathrm{sgn}(f_{\mathbf{z}})$ converges to the error of a Bayes optimal classifier with respect to the misclassification error:

$$\mathcal{C}(\mathrm{sgn}(f)) = \mathrm{Prob}\{\mathrm{sgn}(f(x)) \neq y\}.$$

This forms the theoretical foundation of the convex risk minimization method.

1.2 Learning the Classification Function and Gradient

In this paper we are interested in simultaneously learning f_ϕ and its gradient from the sample values, $\mathbf{z} = \{(x_i,y_i)\}_{i=1}^m$. Denote $x = (x^1,x^2,\ldots,x^n)^T \in \mathbb{R}^n$. The gradient of f_ϕ is the vector of functions (if the partial derivatives exist)

$$\nabla f_\phi = \left(\frac{\partial f_\phi}{\partial x^1}, \ldots, \frac{\partial f_\phi}{\partial x^n} \right)^T.$$

Note that gradient learning is meaningful for classification problems in this sense because f_ϕ is real-valued and may be smooth. For example, in the case of the logistic function (Hastie et al., 2001)

$$\phi(yf(x)) = \log\big(1+e^{-yf(x)}\big).$$

the classification function has a clear statistical interpretation (modeling the conditional probability $\rho(y|X)$ as a Bernoulli random variable)

$$\mathbf{P}\text{rob}(y = \pm 1|x) = \frac{1}{1+e^{-y f_\phi(x)}}.$$

In this case the classification function is

$$f_\phi(x) = \ln\left[\frac{\mathbf{P}\text{rob}(y=1|x)}{\mathbf{P}\text{rob}(y=-1|x)}\right]$$

and the gradient of f_ϕ exists under very mild conditions on the underlying distribution ρ. This is one of the reasons we use a logistic model rather than learning the gradient of a $\{-1,1\}$ classification function. In addition, the logistic model incorporates the uncertainty of the conditional probability at each x which the binary classification function does not.

The relevance of learning the gradient with respect to the problems of variable selection and estimating coordinate covariation is that the gradient provides the following information:
(a) variable selection: the norm of a partial derivative $\|\frac{\partial f_\phi}{\partial x^j}\|_{L^2_{\rho_X}}$ indicates the relevance of this variable, since a small norm implies a small change in the discriminative function f_ϕ with respect to the j-th coordinate,
(b) coordinate covariation: the inner product between partial derivatives $\left\langle \frac{\partial f_\phi}{\partial x^j}, \frac{\partial f_\phi}{\partial x^\ell} \right\rangle$ indicates the covariance of the j-th and ℓ-th coordinates with respect to variation in f_ϕ.

At first glance, the problem of estimating the gradient is equivalent to that of computing classical numerical derivatives in inverse problems. This is the case if we know the sample pair $\{(x_i, f_\phi(x_i)\}_{i=1}^m$. But we face the difficulty that what we have in hand is the set of samples $\mathbf{z}$ where $y_i \in \{\pm 1\}$ is not an approximation of the value $f_\phi(x_i)$ but only its sign. So the classical methods for numerical derivatives fail for learning gradients in the classification setting. Instead, we will motivate a new approach.

The derivation of our gradient learning algorithm can be motivated by the Taylor expansion of f_ϕ, assuming it exists:

$$f_\phi(x) \approx f_\phi(u) + \nabla f_\phi(x) \cdot (x-u), \qquad \text{for } x \approx u.$$

Our objective will be to estimate f_ϕ by a function g and its gradient ∇f_ϕ by a vector valued function $\vec{f} = (f_1, f_2, \ldots, f_n)^T : X \to \mathbb{R}^n$. If the estimates are accurate then the following should hold

$$f_\phi(x) \approx g(u) + \vec{f}(x) \cdot (x-u), \qquad \text{for } x \approx u.$$

The optimization given in (1) suggests a method for estimating g and $\vec{f}$: we minimize a quantity that is like the convex empirical risk but with $f(x_i)$ replaced by $g(u) + \vec{f}(x_i) \cdot (x_i - u)$ with some $u \approx x_i$. As for the choice of u, a natural idea is to set $u = x_j$ and take a weighted average with the weights being chosen to enforce the locality constraints $x_j \approx x_i$ implicit in the Taylor expansion. Various weights may play the same role whenever they satisfies $w_{i,j} \to 0$ as $|x_i - x_j| \to 0$. Throughout this paper we will use a Gaussian with variance s as our weight function:[1]

$$w_{i,j} = w_{i,j}^{(s)} = \frac{1}{s^{n+2}} e^{-\frac{|x_i - x_j|^2}{2s^2}} = w(x_i - x_j), \qquad i,j = 1,\ldots,m.$$

1. In standard problems such as density estimation the Gaussian is normalized by a term of the form $\frac{1}{s^n}$ since the following integral should be invariant with respect to dimension

$$\frac{1}{s^n}\int_{\mathbb{R}^n} e^{-|x|^2/s^2} dx = \text{constant}.$$

Other weight functions can be used as long as the bandwidth of the weight function decreases with the number of samples. Using the Gaussian weight function leads to the following empirical error functional.

Definition 1 *Given a sample $\mathbf{z} \in Z^m$, a function $g : X \to \mathbb{R}$, and a vector-valued function $\vec{f} : X \to \mathbb{R}^n$, we define the empirical error as follows:*

$$\mathcal{E}_{\mathbf{z}}(g, \vec{f}) = \frac{1}{m^2} \sum_{i,j=1}^{m} w_{i,j}^{(s)} \phi\big(y_i(g(x_j) + \vec{f}(x_i) \cdot (x_i - x_j))\big).$$

We may expect that minimizing this error functional using functions in a hypothesis space $\mathcal{H}^{n+1}$ leads to $g_{\mathbf{z}}$ and $\vec{f}_{\mathbf{z}}$ such that

$$g_{\mathbf{z}}(u) + \vec{f}_{\mathbf{z}}(x) \cdot (x-u) \approx f_{\mathbf{z}}(x) \approx f_\phi(x) \approx f_\phi(u) + \nabla f_\phi(x) \cdot (x-u), \qquad \text{for } x \approx u.$$

This in general leads to $g_{\mathbf{z}} \approx f_\phi$ and $\vec{f}_{\mathbf{z}} \approx \nabla f_\phi$.

To formulate the algorithm, we need to specify the hypothesis space. In this paper we will restrict $\mathcal{H}$ to be a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_K$ with an associated Mercer kernel $K : X \times X \to \mathbb{R}$ that is continuous, symmetric and positive semidefinite. The RKHS is defined (Aronszajn, 1950) to be the completion of the linear span of the set of functions $\{K_x := K(\cdot, x) : x \in X\}$ with the inner product $\langle \cdot, \cdot \rangle_K$ satisfying $\langle K_u, K_v \rangle_K = K(u,v)$. The reproducing property of $\mathcal{H}_K$ is

$$\langle K_x, f \rangle_K = f(x), \qquad \forall x \in X, f \in \mathcal{H}_K. \tag{2}$$

This implies that every function $f \in \mathcal{H}_K$ is continuous and bounded. Hence $\mathcal{H}_K \subset C(X) \subset L^2_{\rho_X}(X)$.

Regularizing or shrinking the empirical error $\mathcal{E}_{\mathbf{z}}(g, \vec{f})$ with respect to the RKHS norm defines the following optimization problem.

Definition 2 *Given a sample $\mathbf{z} \in Z^m$ we can estimate the classification function, $g_{\mathbf{z}}$, and its gradient, $\vec{f}_{\mathbf{z}}$, as follows:*

$$(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) = \underset{(g,\vec{f}) \in \mathcal{H}_K^{n+1}}{\arg\min} \left\{ \mathcal{E}_{\mathbf{z}}(g, \vec{f}) + \frac{\lambda}{2}(\|g\|_K^2 + \|\vec{f}\|_K^2) \right\}, \tag{3}$$

where $s, \lambda > 0$ are the regularization parameters and, for $\vec{f} = (f^1, \ldots, f^n) \in \mathcal{H}_K^n$, $\|\vec{f}\|_K^2 = \sum_{i=1}^n \|f^i\|_K^2$.

The immediate advantages of this technique are preventing overfitting and easy computability due to a representer theorem (see Section 2). Another advantage of our method is the derived functions are already approximations of the partial derivatives and they have RKHS inner products which are computed in the estimation process. The inner products reflect the nature of the measure, which is often on a low dimensional manifold embedded in a high dimensional space.

In our paper for technical reasons that will become apparent in the proofs the following quantity should be invariant with respect to dimension

$$\frac{1}{s^{n+2}} \int_{\mathbb{R}^n} e^{-|x|^2/s^2} |x|^2 dx = \text{constant}.$$

Remark 3 *One may consider a natural approach of finding an approximation of f_ϕ (for example by* (1)*) and then computing partial derivatives. But recall our aim is feature (or variable) selection. The problem with this approach is that the partial derivatives may no longer be in the RKHS of the classification function. This leaves us with the problem of not having a norm or computable metric to work with.*

The hypothesis space $\mathcal{H}_K^n$ in the optimization problem (3) may be replaced by some other space of vector-valued functions (Micchelli and Pontil, 2005) in order to learn the gradients.

The distance between points in the Taylor expansion as well as in the weighting function are in the input space and not the feature space of the kernel. This is a natural formulation and an argument for this formulation is that with this distance the algorithms can be extended to a manifold setting without any changes (Mukherjee et al., 2006).

1.3 Overview

In Section 2, we show that the minimizer of the optimization problem (3) satisfies a representer theorem and then provide a procedure to compute the parameters. In Section 3, we prove the convergence of our estimate of the gradient, $\vec{f}_{\mathbf{z}}$, to the true gradient of the classification function, ∇f_ϕ. The utility of the algorithm is illustrated in Section 4 on simulated data as well as gene expression data. We close with a brief discussion in Section 5.

2. Representer Theorem and Parameter Computation

The optimization problem defined by Equation (3) is a convex optimization problem because $\phi(\cdot)$, $\|g\|_K^2$, and $\|\vec{f}\|_K^2$ are all convex functionals. Denote $\mathbb{R}^{p\times q}$ as the space of $p \times q$ matrices. The algorithm that implements the optimization procedure is given in Section 2.1.

The following theorem is an analog of the standard representer theorem (Wahba, 1990; Schoelkopf and Smola, 2001) that states the minimizer of the optimization problem defined by Equation (3) has a finite dimensional representation.

Proposition 4 *Given a sample $\mathbf{z} \in Z^m$ the solution of (3) exists and takes the form*

$$g_{\mathbf{z}}(x) = \sum_{i=1}^m \alpha_{i,\mathbf{z}} K(x,x_i) \quad \textit{and} \quad \vec{f}_{\mathbf{z}}(x) = \sum_{i=1}^m c_{i,\mathbf{z}} K(x,x_i) \tag{4}$$

with $c_{\mathbf{z}} = (c_{1,\mathbf{z}},\ldots,c_{m,\mathbf{z}}) \in \mathbb{R}^{n\times m}$ and $\alpha_{\mathbf{z}} = (\alpha_{1,\mathbf{z}},...,\alpha_{m,\mathbf{z}})^T \in \mathbb{R}^m$.

Proof The existence follows from the convexity of ϕ and functionals $\|g\|_K^2$ and $\|\vec{f}\|_K^2$. Suppose $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ is a minimizer. We can write functions $g_{\mathbf{z}} \in \mathcal{H}_K$ and $\vec{f}_{\mathbf{z}} \in \mathcal{H}_K^n$ as

$$g_{\mathbf{z}} = g_\| + g_\perp \ \text{ and } \ \vec{f}_{\mathbf{z}} = \vec{f}_\| + \vec{f}_\perp$$

where $g_\|$ and each element of $\vec{f}_\|$ is in the span of $\{K_{x_1},,...,K_{x_m}\}$, and $g_\perp$ and $\vec{f}_\perp$ are functions in the orthogonal complement. By the reproducing property $g_{\mathbf{z}}(x_i) = g_\|(x_i)$ and $\vec{f}(x_i) = \vec{f}_\|(x_i)$ for all x_i. So the functions $g_\perp$ and $\vec{f}_\perp$ do not have an effect on $\mathcal{E}_{\mathbf{z}}(g,\vec{f})$. But $\|g_{\mathbf{z}}\|_K^2 = \|g_\| + g_\perp\|_K^2 > \|g_\|\|_K^2$ and $\|\vec{f}_{\mathbf{z}}\|_K^2 = \|\vec{f}_\| + \vec{f}_\perp\|_K^2 > \|\vec{f}_\|\|_K^2$ unless $g_\perp, \vec{f}_\perp = 0$. This implies that $g_{\mathbf{z}} = g_\|$ and $\vec{f}_{\mathbf{z}} = \vec{f}_\|$. This results in the representations in Equation (4). ■

The optimization in Equation (3) can be written in terms of the coefficients $c_{\mathbf{z}}$ and $\alpha_{\mathbf{z}}$. We define a matrix $C = (c_1, c_2, \ldots, c_m) \in \mathbb{R}^{n\times m}$ (when optimized these will be the coefficients $c_{\mathbf{z}}$ in the gradient expansion) and the vector $\alpha \in \mathbb{R}^m$ (when optimized the vector will be $\alpha_{\mathbf{z}}$). We denote the kernel matrix K where $K_{ij} = K(x_i, x_j)$ for $i, j = 1, ..., m$ and the i-th row of the matrix as k_i. The optimization function can be written in matrix form as

$$\Phi(C,\alpha) = \frac{1}{m^2}\sum_{i,j=1}^{m} w_{i,j}\phi\big(y_i(k_j\alpha + k_iC^T(x_i - x_j))\big) + \frac{\lambda}{2}\big(\alpha^T K\alpha + \mathrm{Tr}(CKC^T)\big), \tag{5}$$

where $\mathrm{Tr}(M)$ is the trace of a matrix M.

Proposition 5 *If ϕ is differentiable, then the coefficients $c_{\mathbf{z}}$ and $\alpha_{\mathbf{z}}$ can be computed from the equation $\nabla\Phi(\alpha, C) = 0$.*

We can optimize (5) by using Newton's method to solve $\nabla\Phi(\alpha, C) = 0$. The matrix C however is an $n \times m$ matrix and optimizing in $\mathbb{R}^{mn}$ is problematic for applications where $n \gg m$. We will show that the coefficients can be computed by the optimization of an $m \times d$ matrix, where typically $d \ll m$. We will then apply Newton's method in this reduced space.

Define a vector-valued function

$$h = ((h^0)^T, (h_1)^T, \ldots, (h_m)^T)^T : \mathbb{R}^{(n+1)m} \to \mathbb{R}^{(n+1)m}$$

with

$$h^0 = (h_1^0, \ldots, h_m^0)^T, \quad h_j^0(\alpha, C) = \frac{1}{m^2}\sum_{i=1}^{m} w_{i,j}\phi'\big(y_i(k_j\alpha + k_iC^T(x_j - x_i))\big)y_i + \lambda\alpha_j$$

and, for $i = 1, \ldots, m$,

$$h_i(\alpha, C) = \frac{1}{m^2}\sum_{j=1}^{m} w_{i,j}\phi'\big(y_i(k_j\alpha + k_iC^T(x_j - x_i))\big)y_i(x_i - x_j) + \lambda c_i.$$

By direct computation, we have

$$\nabla\Phi(\alpha, C) = \begin{pmatrix} K & 0 \\ 0 & K \otimes I_n \end{pmatrix} h(\alpha, C) \tag{6}$$

where I_n is the $n \times n$ identity matrix. Solving for the coefficients will give us the following proposition.

Proposition 6 *If the solution to the equation $h(\alpha, C) = 0$ exists, then the coefficients $c_{\mathbf{z}}$ in the representation of $\vec{f}_{\mathbf{z}}$ satisfy the constraint for every $i = 1, \ldots, m$ $c_{i,\mathbf{z}} \in V_{\mathbf{x}} = \mathrm{span}\left\{x_i - x_j : i, j = 1, \ldots, m\right\}$.*

Proof By the assumption, there exists $(\alpha_{\mathbf{z}}, c_{\mathbf{z}})$ solving the equation $h(\alpha, C) = 0$. So $\nabla\Phi(\alpha_{\mathbf{z}}, c_{\mathbf{z}}) = 0$ and $(\alpha_{\mathbf{z}}, c_{\mathbf{z}})$ gives the representation of $g_{\mathbf{z}}$ and $\vec{f}_{\mathbf{z}}$. By the definition of h, we have $h_i(\alpha_{\mathbf{z}}, c_{\mathbf{z}}) = 0$ which implies $c_{i,\mathbf{z}} \in V_{\mathbf{x}}$. This proves the proposition. ■

Remark 7 *We know the solution $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ exists and even is unique. This implies the existence of the solution to $\nabla\Phi(\alpha, C) = 0$. But the existence of the solution to $h(\alpha, C) = 0$ is not clear. In fact, this may not be always the case when K is not invertible.*

Proposition 6 states that the coefficients $c_{i,\mathbf{z}}$ are in the span of the pairwise differences of the input data, which is a low dimensional subspace of $\mathbb{R}^n$. This allows us to reduce the dimension of the optimization problem of solving for the coefficients $c_{\mathbf{z}}$. We apply the well known approach of singular value decomposition to the matrix involving the data $\mathbf{x}$ given by

$$M_{\mathbf{x}} = (x_1 - x_m, x_2 - x_m, \ldots, x_{m-1} - x_m, x_m - x_m) \in \mathbb{R}^{n \times m}.$$

Assume the rank of $M_{\mathbf{x}}$ is d. The theory of singular value decomposition tells us that there exists an $n \times n$ orthogonal matrix $V = (V_1, V_2, \ldots, V_n)$ and an $m \times m$ orthogonal matrix $U = (U_1, U_2, \ldots, U_m)$ such that

$$M_{\mathbf{x}} = V \Sigma U^T = (V_1\ V_2\ \cdots\ V_n) \begin{pmatrix} \operatorname{diag}\{\sigma_1, \sigma_2, \cdots, \sigma_d\} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} U_1^T \\ U_2^T \\ \vdots \\ U_m^T \end{pmatrix}.$$

Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d > \sigma_{d+1} = \ldots = \sigma_{\min\{m,n\}} = 0$ are the singular values of $M_{\mathbf{x}}$. The matrix Σ is $n \times m$ and has entries of zero except for $\Sigma_{i,i} = \sigma_i$ for $i = 1, \ldots, d$. The vectors $\{V_i\}_{i=1}^d$ form an orthonormal basis for $V_{\mathbf{x}}$ and denote $V = (V_1, \ldots, V_d)$.

Set $\beta_i \in \mathbb{R}^d$ to satisfy $x_i - x_m = V\beta_i$ for $i = 1, \ldots, m$. For $\gamma^0 \in \mathbb{R}^m$ and $\gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{R}^{d \times m}$, define the vector-valued function

$$u = ((u^0)^T, (u_1)^T, \ldots, (u_m)^T)^T : \mathbb{R}^{m(d+1)} \to \mathbb{R}^{m(d+1)}$$

by

$$u^0 = (u_1^0, \ldots, u_m^0)^T, \quad u_j^0(\gamma^0, \gamma) = \frac{1}{m^2} \sum_{i=1}^m w_{i,j} \phi'\big(y_i(k_j\gamma^0 + k_i\gamma^T(\beta_i - \beta_j))\big) y_i + \lambda\gamma_j^0,$$

and, for $i = 1, \ldots, m$,

$$u_i(\gamma^0, \gamma) = \frac{1}{m^2} \sum_{j=1}^m w_{i,j} \phi'\big(y_i(k_j\gamma^0 + k_i\gamma^T(\beta_i - \beta_j))\big) y_i(\beta_i - \beta_j) + \lambda\gamma_i.$$

Proposition 8 *If $\gamma_{\mathbf{z}}^0 \in \mathbb{R}^m$ and $\gamma_{\mathbf{z}} = (\gamma_{1,\mathbf{z}}, \ldots, \gamma_{m,\mathbf{z}}) \in \mathbb{R}^{d \times m}$ are solutions of the equation $u(\gamma^0, \gamma) = 0$, then $c_{\mathbf{z}}$ and $\alpha_{\mathbf{z}}$ defined by*

$$\alpha_{\mathbf{z}} = \gamma_{\mathbf{z}}^0, \qquad c_{i,\mathbf{z}} = V\gamma_{i,\mathbf{z}} \ \text{ for } \ i = 1, \ldots, m,$$

solve $\nabla\Phi(\alpha, C) = 0$ and hence yield a representation of $g_{\mathbf{z}}$ and $\vec{f}_{\mathbf{z}}$ respectively.

Proof By the facts that $c_i = V\gamma_i$ for $i = 1, \ldots, m$ defines a one-to-one mapping from $V_{\mathbf{x}}$ onto $\mathbb{R}^d$ and $V^TV = I_d$ the d-dimensional identity matrix, direct computation shows that $u(\gamma_{\mathbf{z}}^0, \gamma_{\mathbf{z}}) = 0$ implies $h(\alpha_{\mathbf{z}}, c_{\mathbf{z}}) = 0$. Then the conclusion follows from Proposition 5 and Equation (6). ■

We now use Proposition 8 to derive the update rule in Newton's method to optimize the coefficients γ^0 and γ. Let $\eta = ((\gamma^0)^T, (\gamma_1)^T, ..., (\gamma_m)^T)^T \in \mathbb{R}^{m(d+1)}$ and consider the map $u(\eta)$ on $\mathbb{R}^{m(d+1)}$ defined as $u = ((u^0)^T, (u_1)^T, ..., (u_m)^T)^T$. When ϕ is twice differentiable, we can use Newton's method to solve the equation $u(\eta) = 0$ by the following iterative update rule

$$\eta_{t+1} = \eta_t - (\nabla u(\eta_t))^{-1} u(\eta_t).$$

2.1 The Optimization Algorithm

The results of the previous section are summarized here to formulate the algorithm that implements the optimization procedure. Before we state the algorithm we restate the matrices and vectors involved in the optimization:

1. the input data $(x_i)_{i=1}^m$
2. the kernel matrix $\mathbf{K}$ given the kernel function
$$\mathbf{K}_{i,j} = K(x_i, x_j) \text{ for } i, j = 1, .., m$$
3. the elements of the weight matrix given the parameter s
$$w_{i,j} = \exp(-\|x_i - x_j\|^2/2s^2) \text{ for } i, j = 1, .., m$$
4. the label vector computed from the output variables $\mathbf{y} = (y_1, \ldots, y_m)^T$
5. $M_{\mathbf{x}} = [x_1 - x_m, x_2 - x_m, \ldots, x_{m-1} - x_m, x_m - x_m] \in \mathbb{R}^{n \times m}$
6. $V = (v_1, v_2, \ldots, v_d)$ the d left eigenvectors of $M_{\mathbf{x}}^T M_{\mathbf{x}}$
7. $\beta_i = V^T(x_i - x_m)$ for $i = 1$ to m
8. at iteration t we have the vector $\eta_t = ((\gamma^0)^T, (\gamma_1)^T, ..., (\gamma_m)^T)^T \in \mathbb{R}^{m(d+1)}$ with $\gamma^0 := \eta_t(1:m)$, $\gamma_i := \eta_t(m + (i-1)d + 1 : m + id)$, and $\gamma := (\gamma_1, \ldots, \gamma_m)$
9. at each iteration the matrix $\mathbf{a} \in \mathbb{R}^{m \times m}$ is defined by its components
$$\mathbf{a}_{i,j} = w_{i,j} \phi'\big(y_i(k_j \gamma^0 + k_i \gamma^T(\beta_i - \beta_j))\big)$$
where k_i is the i-th column of the kernel matrix
10. at each iteration the matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ is defined by its components
$$\mathbf{A}_{i,j} = w_{i,j} \phi''\big(y_i(k_j \gamma^0 + k_i \gamma^T(\beta_i - \beta_j))\big)$$
11. given the matrix $\mathbf{a}$ we define the vectors $b_0 = \mathbf{a}^T \mathbf{y}$ and
$$b_i = y_i \sum_{j=1}^m \mathbf{a}_{i,j}(\beta_i - \beta_j) \text{ where } i = 1, \ldots, m$$

12. given the matrix $\mathbf{A}$ we define the $m \times m$ matrix

$$K_0 = \text{diag}(\mathbf{A}\mathbf{1}_m)\mathbf{K} \text{ where } \mathbf{1}_m = (1,1,\ldots,1)^T$$

13. from the matrices

$$K_1(j,\ell) = \sum_{i=1}^{m} \mathbf{A}_{i,j} K(x_i, x_\ell)(\beta_i - \beta_j)^T \text{ where } j,\ell = 1,..,m$$

construct the matrix

$$\tilde{K_1} = \begin{pmatrix} K_1(1,1) & \ldots & K_1(1,m) \\ \vdots & \ddots & \vdots \\ K_1(m,1) & \ldots & K_1(m,m) \end{pmatrix}$$

14. from the matrices

$$K_2(i,\ell) = \sum_{j=1}^{m} \mathbf{A}_{i,j} K(x_j, x_\ell)(\beta_i - \beta_j) \text{ where } i,\ell = 1,..,m$$

construct the matrix

$$\tilde{K_2} = \begin{pmatrix} K_2(1,1) & \ldots & K_2(1,m) \\ \vdots & \ddots & \vdots \\ K_2(m,1) & \ldots & K_2(m,m) \end{pmatrix}$$

15. from the matrices

$$B_i = \sum_{j=1}^{m} \mathbf{A}_{i,j}(\beta_i - \beta_j)(\beta_i - \beta_j)^T \text{ where } i = 1,..,m$$

construct the matrix

$$\tilde{K_3} = \begin{pmatrix} B_1K(x_1,x_1) & B_1K(x_1,x_2) & \ldots & B_1K(x_1,x_m) \\ B_2K(x_2,x_1) & B_2K(x_2,x_2) & \ldots & B_2K(x_2,x_m) \\ \vdots & \vdots & \ddots & \vdots \\ B_mK(x_m,x_1) & B_mK(x_m,x_2) & \ldots & B_mK(x_m,x_m) \end{pmatrix}.$$

16. the coefficents of the classification function estimate, $\alpha_{\mathbf{z}} \in \mathbb{R}^m$ and $g_{\mathbf{z}} = \sum_{i=1}^{m} \alpha_{i,\mathbf{z}} K(x, x_i)$

17. the coefficents of the gradient estimate $(c_{i,\mathbf{z}})^T \in \mathbb{R}^p$ for $i = 1,\ldots,m$ and $\vec{f}_{\mathbf{z}} = \sum_{i=1}^{m} c_{i,\mathbf{z}} K(x, x_i)$.

Given the above quantities we now state the algorithm for solving the optimization problem for learning gradients.

Algorithm 1: Algorithm for computing $g_{\mathbf{z}}$ and $\vec{f}_{\mathbf{z}}$

input : inputs $\mathbf{x} = (x_i, \ldots, x_m)$, labels $\mathbf{y} = (y_1, \ldots, y_m)^T$, kernel K, weights $(w_{i,j})$, regularization parameter s, $\lambda > 0$ and threshold $\varepsilon > 0$

return: coefficients $\alpha_{\mathbf{z}}$ and $(c_{i,\mathbf{z}})_{i=1}^m$

$M_{\mathbf{x}} = [x_1 - x_m, x_2 - x_m, \ldots, x_{m-1} - x_m, x_m - x_m]$;
$[V, \Sigma, U] = \text{svd}(M_{\mathbf{x}})$;

$\eta_0 = \mathbf{0}$; stop $\leftarrow$ false; $t \leftarrow 0$;
repeat
  $u(\eta_t) = \frac{1}{m^2}(b_0^T, b_1^T, \ldots, b_m^T)^T + \lambda \eta_t$;
  $\nabla u(\eta_t) = \lambda I_{m(d+1)} + \frac{1}{m^2}\begin{pmatrix} K_0 & \tilde{K_1} \\ \tilde{K_2} & \tilde{K_3} \end{pmatrix}$;
  $\Delta \eta_t = (\nabla u(\eta_t))^{-1} u(\eta_t)$;
  $\eta_{t+1} = \eta_t - \Delta \eta_t$;
  $t \leftarrow t + 1$
  If $\|\Delta \eta_t\| \le \varepsilon$ stop $\leftarrow$ true
until *stop=true* ;

$\alpha_{\mathbf{z}} = \eta_t(1 : m)$;
$\gamma_{i,\mathbf{z}} = \eta_t(m + (i-1)d + 1 : m + id)$ for $i = 1, \ldots, m$;
$c_{i,\mathbf{z}} = V \gamma^{i,\mathbf{z}}$ for $i = 1, \ldots, m$;

3. Error Analysis

In this section, we investigate the statistical performance of the algorithm. We will show that under certain conditions, $g_{\mathbf{z}} \to f_\phi$ and $\vec{f}_{\mathbf{z}} \to \nabla f_\phi$ as $\lambda, s \to 0$. Let us first illustrate this by a specific case where $\phi(\cdot)$ is the logistic loss and $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$ (this case corresponds to the realizable setting in the PAC learning paradigm). Denote as ∂X the boundary of X and $d(x, \partial X)$ the distance of $x \in X$ from ∂X.

Theorem 9 *Let ϕ be the logistic loss. Assume that for some constants $c_\rho > 0$ and $0 < \theta \le 1$ the marginal distribution ρ_X satisfies*

$$\rho_X\left(\{x \in X : d(x, \partial X) < s\}\right) \le c_\rho s, \tag{7}$$

and the density $p(x)$ of ρ_X exists and satisfies

$$\sup_{x \in X} p(x) \le c_\rho \quad \textit{and} \quad |p(x) - p(u)| \le c_\rho |x - u|^\theta, \ \forall u, x \in X. \tag{8}$$

Suppose that $K \in C^2$ *and* $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$. *Choose* $\lambda = \lambda(m) = m^{-\frac{2\theta}{3(n+2+2\theta)}}$ *and* $s = s(m) = m^{-\frac{1}{3(n+2+2\theta)}}$. *Then there exists a constant* $C > 0$ *such that for any* $0 < \eta < 1$ *with confidence* $1-\eta$

$$\begin{aligned} \|g_{\mathbf{z}} - f_\phi\|_{L^2_{\rho_X}} &\leq C \log\frac{4}{\eta}\left(\frac{1}{m}\right)^{\frac{\theta}{6(n+2+2\theta)}}, \\ \|\vec{f}_{\mathbf{z}} - \nabla f_\phi\|_{L^2_{\rho_X}} &\leq C \log\frac{4}{\eta}\left(\frac{1}{m}\right)^{\frac{\theta}{6(n+2+2\theta)}}. \end{aligned}$$

Condition (8) means the density of the marginal distribution is Hölder θ. Condition (7) is about the behavior of ρ_X near the boundary of X. When the boundary is piecewise smooth, (8) implies (7).

Theorem 9 is a consequence of the more general Theorem 10 which we prove in Section A.3.

We first define two quantities that will be used extensively.

$$\kappa = \sup_{x\in X}\sqrt{K(x,x)}; \qquad D = \max_{x,u\in X}|x-u|.$$

Note that the reproducing property (2) of the RKHS $\mathcal{H}_K$ implies $\|f\|_\infty \leq \kappa\|f\|_K$ for $f \in \mathcal{H}_K$. This will be used constantly in the following.

For a convex loss function ϕ and $r > 0$, define

$$\begin{aligned} L_r &= \max\left\{|\phi'(\kappa(1+D)r)|, |\phi'(-\kappa(1+D)r)|\right\}, \\ M_r &= \max\left\{\phi(\kappa(1+D)r), \phi(-\kappa(1+D)r)\right\}. \end{aligned}$$

By convexity of ϕ both L_r and M_r increase with r.

Theorem 10 *Let the convex loss function* ϕ *be twice differentiable and satisfy*

$$q_1(T) = \inf_{|t|\leq T}\phi''(t) > 0, \qquad q_2(T) = \sup_{|t|\leq T}\phi''(t) < \infty.$$

Assume ρ *satisfies* (7) *and* (8), $K \in C^2$, $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$. *Then there exists a constant* $\tilde{C}$ *such that for* $0 < \delta < 1/2$, $0 < s, \lambda \leq 1$ *with probability at least* $1-2\delta$

$$\max\left\{\|g_{\mathbf{z}} - f_\phi\|^2_{L^2_{\rho_X}}, \|\vec{f}_{\mathbf{z}} - \nabla f_\phi\|^2_{L^2_{\rho_X}}\right\} \leq \tilde{C}\left\{r^2 s^\theta + B_r\left(\frac{L_r r + M_r \log\frac{2}{\delta}}{\sqrt{m}s^{n+2}} + s^2 + \lambda\right)s^{-\theta}\right\},$$

where

$$r = \tilde{c}\left\{1 + \frac{s^2}{\lambda} + \left(\frac{L_{\lambda,s}}{\sqrt{\lambda}s^{n+2}} + M_{\lambda,s}\log\frac{2}{\delta}\right)\frac{1}{\sqrt{m}\lambda s^{n+2}}\right\}^{1/2} \tag{9}$$

with some $\tilde{c} \geq 1$, $L_{\lambda,s} = L_{\sqrt{2\phi(0)/\lambda s^{n+2}}}$, *and* $M_{\lambda,s} = M_{\sqrt{2\phi(0)/\lambda s^{n+2}}}$ *and* $B_r = \min\left\{\frac{1}{q_1(c_0 r)}, r\right\}$ *with some* $c_0 > 0$.

Remark 11 *Theorem 10 applies only to the loss functions satisfying* $\phi''(t) > 0$ *because of the requirements on* q_1, *which excludes the SVM hinge loss. As for the quantity* B_r, *we note that it does not increase very quickly with* r. *One can take* $B_r = r$ *for logistic loss and exponential loss where* $q_1(T)$ *decays exponentially fast with* T. *While for the square loss,* $B_r = 1$ *for* $q_1(T) \equiv 1$.

Since the entire proof is rather complicated it has been postponed to the appendix. We now prove Theorem 9 using Theorem 10.

Proof of Theorem 9. Note that for logistic loss, $\phi'(t) = \frac{-e^{-t}}{1+e^{-t}} \in (-1,1)$. So $L_r \leq 1$ and $L_{\lambda,s} \leq 1$. Since $\phi(t) \leq \phi(0) + |t| < 1 + |t|$, we have $M_r \leq (1+\kappa(1+D))r$ when $r \geq 1$ and so $M_{\lambda,s} \leq 2(1+\kappa(1+D))(\lambda s^{n+2})^{-1}$. Also, $\phi''(t) = \frac{2e^{-t}}{(1+e^{-t})^2}$ implies $\frac{1}{q_1(r)} \geq c_0 r$, $q_2(c_0 r) \leq 1/2$. Substitute $L_{\lambda,s}$ and $M_{\lambda,s}$ into (9). The choice of λ, s ensures that $r \leq r_0$ with $r_0 > 1$ an absolute constant. Since B_r, L_r, M_r are increasing with respect to r, so is the upper bound in Theorem 10. Substituting r_0 into this upper bound and by the choice of λ, s, we obtain with confidence at least $1-2\delta$

$$\max\left\{ \|g_{\mathbf{z}} - f_\phi\|^2_{L^2_{\rho_X}}, \|\vec{f}_{\mathbf{z}} - \nabla f_\phi\|^2_{L^2_{\rho_X}} \right\} \leq C \log\frac{2}{\delta} \left(\frac{1}{m}\right)^{\frac{\theta}{3(n+2+2\theta)}},$$

where

$$C = \tilde{C}\left((r_0)^2 + B_{r_0}(L_{r_0} r_0 + M_{r_0} + 2)\right).$$

Setting $\delta = \frac{\eta}{2}$ finishes the proof. ■

Remark 12 *In order to calculate the learning rate, we have imposed a rigid assumption on f_ϕ: both f_ϕ and each element of ∇f_ϕ are in $\mathcal{H}_K$. But the convergence may hold under milder conditions, say, they lie in the closure of $\mathcal{H}_K$ in $L^2_{\rho_X}$. This is in general true if $\mathcal{H}_K$ is dense in $L^2_{\rho_X}$, for example the case of a Gaussian kernel.*

4. Simulated Data and Gene Expression Data

In this section we apply the gradient learning algorithm (3) to the problem of estimating a classification function and simultaneously selecting relevant variables and measuring their covariance. The idea is to rank the importance of variables according to the norm of their partial derivatives $\|\frac{\partial f_\phi}{\partial x^\ell}\|$, since a small norm implies small changes of the classification function with respect to this variable. By our error analysis, we expect $\vec{f}_{\mathbf{z}} \approx \nabla f_\phi$. So we shall use the norms of the components of $\vec{f}_{\mathbf{z}}$ to rank the variables.

Definition 13 *The relative magnitude of the norm for the variables is defined as*

$$s^\phi_\ell = \frac{\|(\vec{f}_{\mathbf{z}})_\ell\|_K}{\left(\sum_{j=1}^n \|(\vec{f}_{\mathbf{z}})_j\|^2_K\right)^{1/2}}, \qquad \ell = 1, \ldots, n.$$

In the same way, we can study coordinate covariances by an empirical matrix.

Definition 14 *The* **empirical gradient matrix** *(EGM), $F_{\mathbf{z}}$, is the $n \times m$ matrix whose columns are $\vec{f}_{\mathbf{z}}(x_j)$ with $j = 1, \ldots, m$. The* **empirical covariance matrix** *(ECM), $\Xi_{\mathbf{z}}$, is the $n \times n$ matrix of inner products of the directional derivative of two coordinates*

$$Cov(\vec{f}_{\mathbf{z}}) := \left[\langle (\vec{f}_{\mathbf{z}})_p, (\vec{f}_{\mathbf{z}})_q \rangle_K\right]^n_{p,q=1} = c_{\mathbf{z}} K c^T_{\mathbf{z}} = \sum_{i,j=1}^m c_{i,\mathbf{z}} c^T_{j,\mathbf{z}} K(x_i, x_j).$$

The ECM gives us the covariance between the coordinates while the EGM gives us information as how the variables differ over different sections of the space.

We apply our idea to three data sets. The first two data sets are artificial ones which we use to illustrate the procedure. The third is a cancer classification problem that has been well studied and serves as further confirmation of the utility of the method. For all three the parameter s of the Gaussian was set as the median of all pairwise distances between points in the data. More experiments including data sets for more challenging classification problems can be found in Mukherjee et al. (2006) where we also developed a novel feature selection procedure via learning gradients.

4.1 Linearly Separable Simulation

Linearly separable data is drawn from two classes in an $n = 80$ dimensional space. Samples from class -1 were drawn from

$$\begin{aligned} x^j &\sim \mathbf{No}(1.5,1), \text{ for } j = 1,\ldots,10, \\ x^j &\sim \mathbf{No}(-3,1), \text{ for } j = 11,\ldots,20, \\ x^j &\sim \mathbf{No}(0,\sigma_{\text{noise}}), \text{ for } j = 21,\ldots,80, \end{aligned}$$

where $\mathbf{No}(\mu,\sigma)$ is the normal distribution with mean μ and standard deviation σ. Samples from class $+1$ were drawn from

$$\begin{aligned} x^j &\sim \mathbf{No}(1.5,1), \text{ for } j = 41,\ldots,50, \\ x^j &\sim \mathbf{No}(-3,1), \text{ for } j = 51,\ldots,60, \\ x^j &\sim \mathbf{No}(0,\sigma_{\text{noise}})), \text{ for } j = 1,\ldots,40,61,\ldots,80. \end{aligned}$$

We ran our algorithm on draws of the above data using a linear kernel and report both the results of the gradient estimate as well as the classification function passed through a logistic function.

Drawing twenty samples from the two respective classes results in a design matrix $\mathbf{x}$ that is 80×40 where the first twenty samples belong to class -1 and the remaining to class $+1$. Figure 1 contains results for data where we set $\sigma_{\text{noise}} = .1$. A draw of this matrix is displayed in Figure 1a. In Figure 1d we display the conditional likelihoods obtained by the classification function on the training data. A leave-one-out analysis yields similar results. For Figure 2 the data was generated with $\sigma_{\text{noise}} = 1$. Note that in the this case standard methods such as PCA would not find the correct features since the variance in all dimensions is equal. The plots corresponding to Figures 2a-d are analogous to those in Figure 1.

In Figures 1b and 2b we plot the norm of each component of the estimate of the gradient, $\{\|(\vec{f}_{\mathbf{z}})_\ell\|_K\}_{\ell=1}^{80}$. The norm of each component gives an indication of the importance of a variable and variables with small norms can be eliminated. Note that the coordinates with large norms are the ones we expect, $\ell = 1,\ldots,20,41,\ldots,60$. Figures 1c and 2c display the empirical covariance matrix. The blocking structure of this matrix indicates the covariance of coordinates.

4.2 Nonlinearly Separable Simulation

Data is drawn from two classes in an $n = 200$ dimensional space that are nonlinearly separable in the first two dimensions. Samples from class $+1$ were drawn from

$$\begin{aligned} (x^1,x^2) &= (r\sin(\theta), r\cos(\theta)), \text{ where } r \sim U[0,1] \text{ and } \theta \sim U[0,2\pi], \\ x^j &\sim \mathbf{No}(0.0,.2), \text{ for } j = 3,\ldots,200, \end{aligned}$$

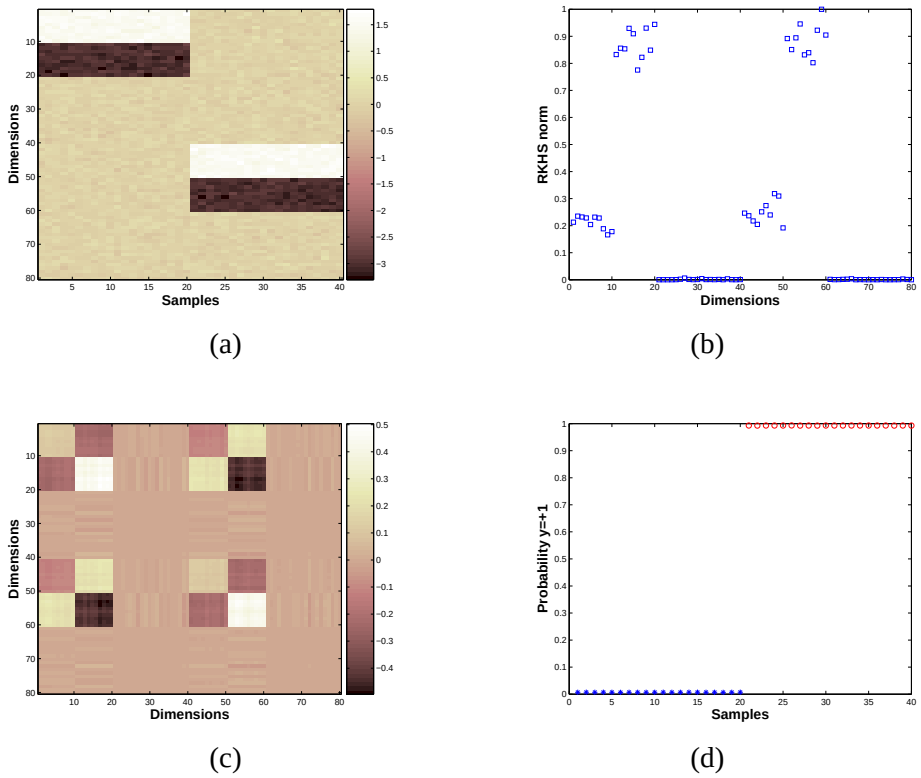

Figure 1: a) The data matrix $\mathbf{x}$ where each sample corresponds to a column and the first twenty samples correspond to class -1 and the second twenty to class $+1$, b) the RKHS norm for each dimension, c) the empirical covariance matrix, d) the predicted class probabilities on the training data.

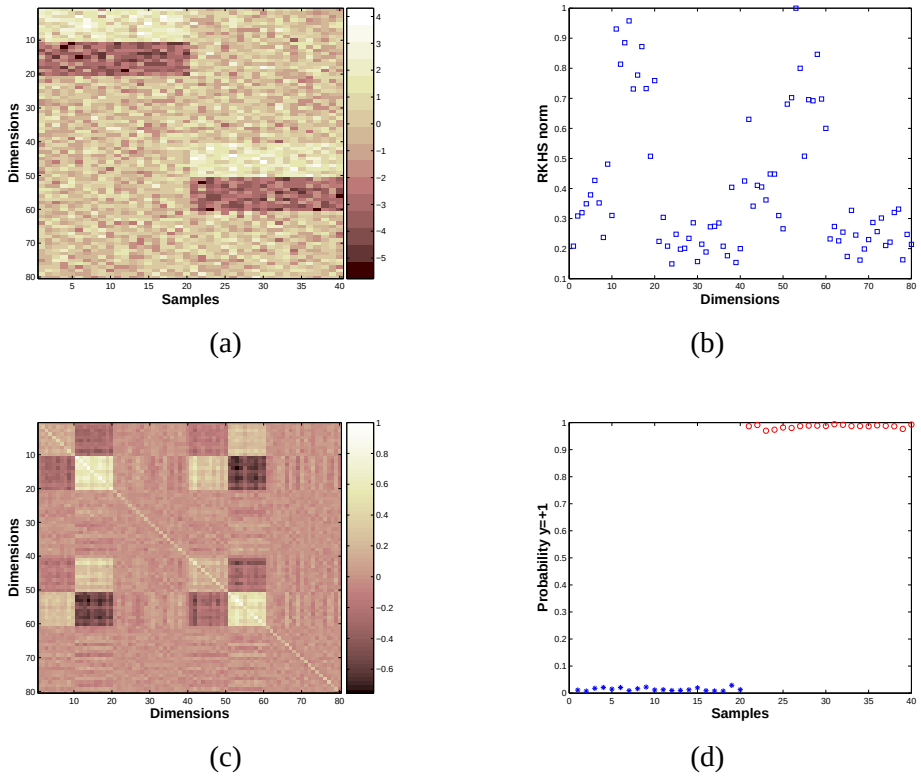

Figure 2: a) The data matrix $\mathbf{x}$ where each sample corresponds to a column and the first twenty samples correspond to class -1 and the second twenty to class $+1$, b) the RKHS norm for each dimension, c) the empirical covariance matrix, d) the predicted class probabilities on the training data.

where $U[a,b]$ is the uniform distribution with support on the interval $[a,b]$. Samples from class -1 were drawn from

$$\begin{aligned}(x^1,x^2) &= (r\sin(\theta), r\cos(\theta)), \text{ where } r \sim U[2,3] \text{ and } \theta \sim U[0,2\pi],\\ x^j &\sim \mathbf{No}(0.0,.2), \text{ for } j = 3,\ldots,200.\end{aligned}$$

Note that the data can be separated by a circle in the first two dimensions.

Drawing thirty samples from the two respective classes results in a design matrix $\mathbf{x}$ that is 200×60 where the first thirty samples belong to class -1 and the remaining to class $+1$. A draw of the first two dimensions of the data is displayed in Figure 3a. Since a linear function cannot accurately classify the data we used a Gaussian kernel

$$K(u,v) = e^{-|u-v|^2/2\sigma^2},$$

where σ was set to the median pairwise distances between points. In the following we report both the results of the gradient estimate as well as the classification function passed through a logistic

function. In Figure 3c,d we plot the norm of each component of the estimate of the gradient. The norm of first two coordinates are much larger than the norm of any of the other coordinates,

$$\frac{\min_{i=1,2} \| \left(\vec{f}_{\mathbf{z}}\right)_i \|_K}{\max_{i=3,\ldots,200} \| \left(\vec{f}_{\mathbf{z}}\right)_i \|_K} > 90.$$

In Figure 3b we plot the ECM. The blocking structure of the ECM indicates the covariance of the first two coordinates. In Figure 3e we display the conditional likelihoods obtained by the classification function on the training data without any feature selection. The classification accuracy improves when we rerun our algorithm using only the dimensions with nonzero norms (3f). The classification results are comparable to what would be obtained by using regularized logistic regression.

4.3 Gene Expression Data

In computational biology, specifically in the subfield of gene expression analysis variable selection and estimation of covariation is of fundamental importance. Microarray technologies enable experimenters to measure the expression level of thousands of genes, the entire genome, at once. The expression level of a gene is proportional to the number of copies of mRNA transcribed by that gene. This readout of gene expression is considered a proxy of the state of the cell. The goals of gene expression analysis include using the expression level of the genes to predict classes, for example tissue morphology or treatment outcome, or real-valued quantities such as drug toxicity or sensitivity. Fundamental to understanding the biology giving rise to the outcome or toxicity is determining which genes are most relevant for the prediction.

4.4 Leukemia Classification

We apply our procedure to a well studied expression data set. The data set is a result of a study using expression data to discriminate acute myeloid leukemia (AML) from acute lymphoblastic leukemia (ALL) (Golub et al., 1999; Slonim et al., 2000) and estimating the genes most relevant to this discrimination. The data set contains 48 samples of AML and 25 samples of ALL. Expression levels of $n = 7,129$ genes and expressed sequence tags (ESTs) were measured via an oligonucleotide microarray for each sample. This data set was split into a training set of 38 samples and a test set of 35 samples.

Various variable selection algorithms have been applied to this data set by using the training set specified in Golub et al. (1999) to select variables and build a classification model and then compute the classification error on the test set. In the same spirit as recursive feature elimination (RFE) we iteratively run our procedure on the training data and remove all variables except for the $\mathcal{S}$ with the largest norm, s_ℓ^ϕ. In Table 1 we report test errors for various values of $\mathcal{S}$ that result from the following procedure:

1. given training data $\mathbf{z}_{7129}$ and test data $\mathbf{tz}_{7129}$ compute the number of errors on the test data $\text{ter}_{7129}(\mathbf{tz}_{7129}) = |\text{sign}[g_{\mathbf{z}_{7129}}(\mathbf{tz}_{7129})] \neq ty|$ and the vector of norms $\{s_\ell^\phi\}_{\ell=1}^{7129}$

2. for $\mathcal{S} = 3000, 1000, 500, 400, 300, 200, 100, 50$ repeat steps 3,4

3. project the test and training data into the dimensions corresponding to the top $\mathcal{S}$ values of $\{s_\ell^\phi\} : \mathbf{z}_\mathcal{S}$ and $\mathbf{tz}_\mathcal{S}$

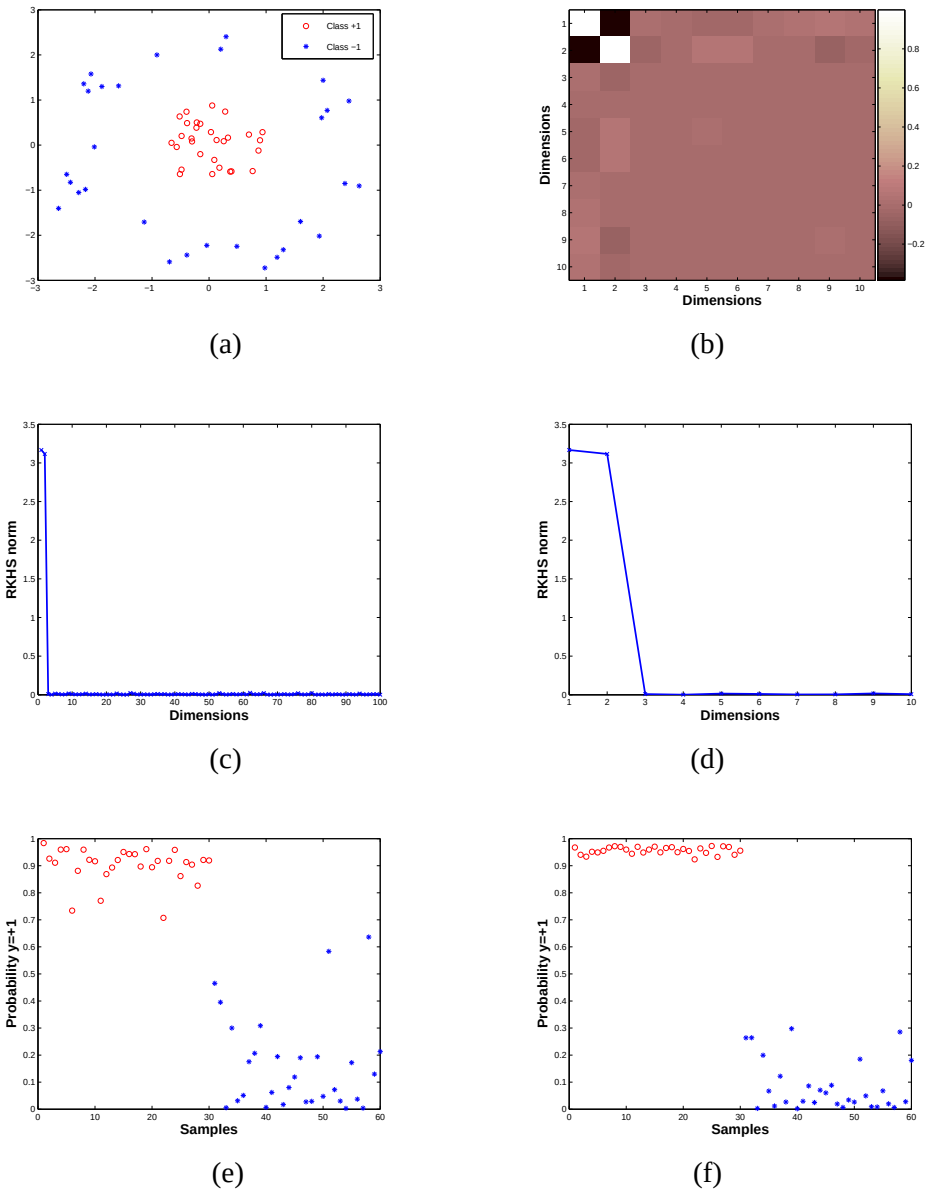

Figure 3: a) The first two dimensions of the data matrix class $+1$ are the circles and class -1 are the stars, b) the empirical covariance matrix for the first 10 dimensions, c) the RKHS norm for the first 100 dimensions , d) the RKHS norm for the first 10 dimensions, e) the predicted class probabilities on the training data with no feature selection again circles are class $+1$ and stars are class -1, f) the predicted class probabilities on the training data with feature selection.

4. given the training data $\mathbf{z}_{\mathcal{S}}$ and test data $\mathbf{tz}_{\mathcal{S}}$ compute the number of errors on the test data $\mathrm{ter}_{\mathcal{S}}(\mathbf{tz}_{\mathcal{S}}) = |\mathrm{sign}[g_{\mathbf{z}_{\mathcal{S}}}(\mathbf{tz}_{\mathcal{S}})] \neq ty|$ and the vector of norms $\{s_\ell^\phi\}$.

The classification accuracy is very similar to other feature selection algorithms such as recursive feature elimination (RFE) (Guyon et al., 2002; Lee et al., 2004) and radius-margin bound (RMB) (Chapelle et al., 2002) both of which were developed specifically for SVMs. In this context we are doing as well as state of the art methods. However, it is important to note that many methods will do very well on this data set and the previously mentioned methods cannot address the issue of covariation.

genes (S)	50	100	200	300	400	500	1,000	3,000	7,129
test errors	2	1	1	1	1	1	1	1	2

Table 1: Number of errors in classification for various values of $\mathcal{S}$ using the genes corresponding to dimensions with the largest norms. The predictions were made using the sign of the classification function output by our method evaluated at each test sample.

In Figure 4a-d we plot the relative magnitude sequence s_ℓ^ϕ for the genes. On this data set the decay in the ranked scores $s_{(\ell)}^\phi$ is steeper than that for most statistics that have been previously used on this data. To illustrate this we compared the gradient score to the Fisher score (Slonim et al., 2000) for each gene

$$t_\ell = \frac{|\hat{\mu}_\ell^{\mathrm{AML}} - \hat{\mu}_\ell^{\mathrm{ALL}}|}{\hat{\sigma}_\ell^{\mathrm{AML}} + \hat{\sigma}_\ell^{\mathrm{ALL}}},$$

where $\hat{\mu}_\ell^{\mathrm{AML}}$ is the mean expression level for the AML samples in the ℓ-th gene, $\hat{\mu}_\ell^{\mathrm{ALL}}$ is the mean expression level for the ALL samples in the ℓ-th gene, $\hat{\sigma}_\ell^{\mathrm{AML}}$ is the standard deviation of the expression level for the AML samples in the ℓ-th gene, and $\hat{\sigma}_\ell^{\mathrm{ALL}}$ is the standard deviation of the expression level for the ALL samples in the ℓ-th gene. We then normalize these scores

$$s_\ell^F = \frac{t_\ell}{\left(\sum_{p=1}^n t_p^2\right)^{1/2}}.$$

Figure 4a-d displays the relative decay of $s_{(\ell)}^\phi$ and $s_{(\ell)}^F$ over various numbers of dimensions. In all plots it is apparent that the decay rate of $s_{(\ell)}^\phi$ is much steeper. Plotting the decay of the elements for the normalized hyperplane $\frac{w^0}{\|w^0\|}$ that is the solution of a linear SVM or the solution of regularized linear logistic regression results in a plot much more like that of the Fisher score than the gradient statistic. Whether and how this steepness (sparsity) has an implication on the generalization error is an open question.

We can also examine the EGM and the ECM. The EGM in this case is a $7,129 \times 38$ matrix and the ECM is $7,129 \times 7,129$ matrix. In Figure 5 we plot the ECM for the 50 dimensions that resulted from the iterative procedure outlined above. This matrix indicates how the dimensions covary and can be used to construct clusters of genes.

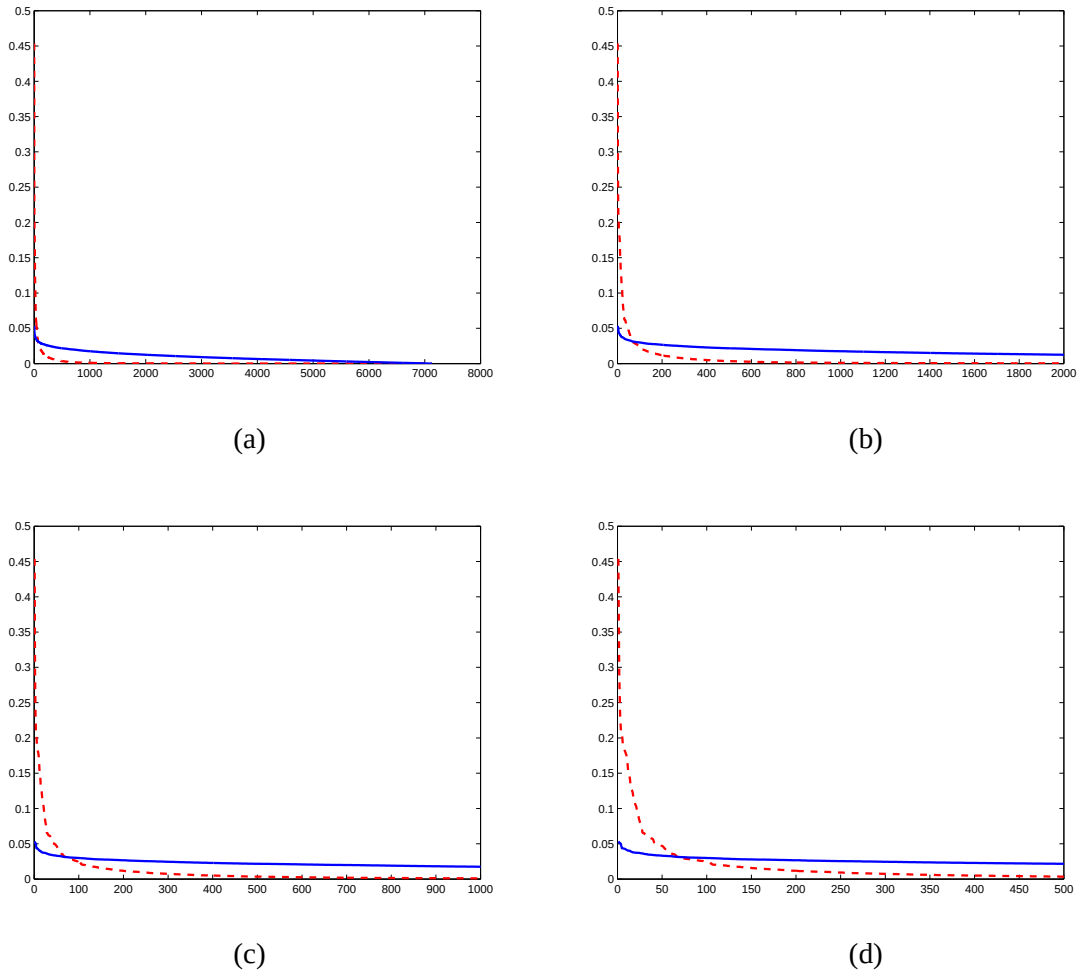

Figure 4: The decay of $s^{\phi}_{(\ell)}$ (dashed line) and $s^{F}_{(\ell)}$ (solid line) over: a) all the genes/dimensions, b) the top 3000 genes/dimensions, c) the top 1000 genes/dimensions, d) the top 500 genes/dimensions.

5. Discussion

We introduce an algorithm that learns a classification function and its gradient from sample data in the logistic regression framework. The relevance of this method for variable selection is motivated. An error analysis is given for the convergence of the estimated classification function and gradient to the true ones respectively. This method also places the problem of variable selection into the powerful framework of Tikhonov regularization. There are many extensions and refinements and open questions regarding this method which we discuss below:

1. Accuracy of classification function: It seems intuitive that the classification function obtained by our method should be strictly worse than that obtained by standard regularized logistic

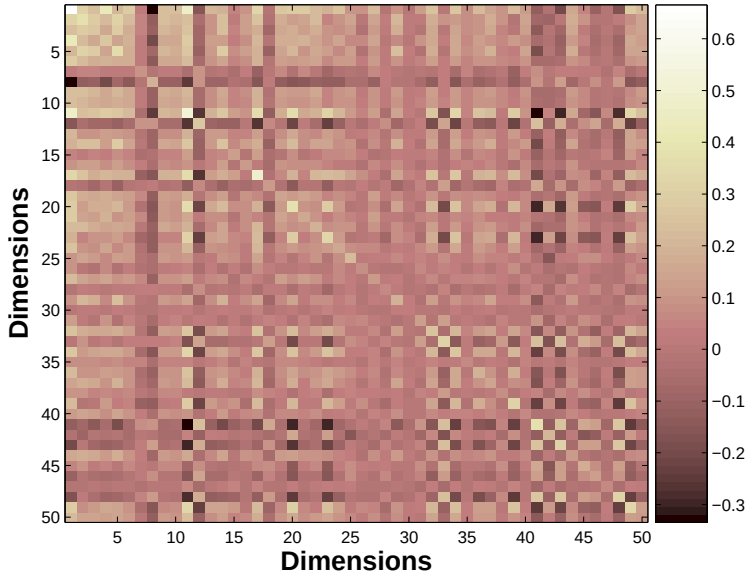

Figure 5: The ECM for the top 50 dimensions.

regression. This is simply a corollary of very useful dictum proposed by Vladimir Vapnik (Vapnik, 1998), "When solving a a given problem, try to avoid solving a more general problem as an intermediate step." Although we strongly expect our classification function to be less accurate than that provided by regularized logistic regression we need to do more empirical work to confirm this.

2. Logistic regression models: An alternative optimization problem was proposed in Mukherjee and Zhou (2006) for estimating the gradient $\vec{f}_{\mathbf{z}}$ in the binary regression problem

$$\vec{f}_{\mathbf{z},\lambda} = \arg\min_{\vec{f}\in\mathcal{H}_K^n} \left\{ \frac{1}{m^2} \sum_{i,j=1}^{m} w_{i,j}^{(s)} \phi\Big(y_i \big(y_j + \vec{f}(x_i)\cdot(x_i - x_j) \big) \Big) + \lambda \|\vec{f}\|_K^2 \right\}.$$

This optimization problem does not follow from the Taylor expansion since in general y_j need not be close to $f_\phi(x_j)$, only the signs of the two functions need agree. This formulation does have an interesting interpretation for variable selection in that variables that are relevant in the classification problem will have large gradient norms and those not relevant will have norms near zero. In practice, for large values of λ the gradient estimates of the above formulation will be similar to those given by the optimization in (3).

3. Fully Bayesian model: The Tikhonov regularization framework coupled with the use of an RKHS allows us to implement a fully Bayesian version of the procedure in the context of Bayesian radial basis (RB) models (Liang et al., 2006). The Bayesian RB framework can be extended to develop a proper probability model for the gradient learning problem. The optimization procedure (3) would be replaced by Markov Chain Monte-carlo methods and the full posterior rather than the maximum a posteriori estimate would be computed. A very

useful result of this is that in addition to the point estimates for the gradient we would also be able to compute credible (confidence) intervals.

4. Intrinsic dimension: In Theorem 9 the rate of convergence of the gradient has the form of $O(m^{-1/n})$ which can be extremely slow if n is large. However, in most data sets and when variable selection is meaningful the data are concentrated on a much lower dimensional manifold embedded in the high dimensional space. In this setting an analysis that replaces the ambient dimension n with the intrinsic dimension of the manifold $n_{\mathcal{M}}$ would be of great interest.

5. Semi-supervised setting: Intrinsic properties of the manifold X can be further studied by unlabeled data. This is one of the motivations of semi-supervised learning. In many applications, it is much easier to obtain unlabeled data with a larger sample size $u \gg m$. For our purpose, unlabeled data $\mathbf{x} = (x_i)_{i=m+1}^{m+u}$ can be used to reduce the dimension or correlation. Since we learn the gradient by $\vec{f}$, it is natural to use the unlabeled data to control the approximate norm of $\vec{f}$ in some Sobolev spaces and introduce a semi-supervised learning algorithm as minimizing over $(g, \vec{f}) \in \mathcal{H}_K^{n+1}$

$$\left\{ \mathcal{E}_{\mathbf{z}}(g, \vec{f}) + \frac{\mu}{(m+u)^2} \sum_{i,j=1}^{m+u} W_{i,j} |\vec{f}(x_i) - \vec{f}(x_j)|^2_{\ell^2(\mathbb{R}^n)} + \lambda \|\vec{f}\|_K^2 \right\},$$

where $\{W_{i,j}\}$ are edge weights in the data adjacency graph, μ is another regularization parameter and often satisfies $\lambda = o(\mu)$.

6. Conclusion

The practical motivation for this work came from a problem in computational biology: pathway extraction. The basic problem is given model systems with known genetic or molecular perturbations infer gene expression "signatures of pathways" (sets of genes that characterize the perturbation in the model system). The term pathway has both a biological and statistical connotation. A statistical definition of a pathway is a set of genes that given a perturbation coordinately differentially co-express with respect to the perturbation. This statistical definition allows us to formulate the biological problem in the mathematical and computational framework of variable (gene) selection. A variety of methods have been proposed for variable selection (Tibshirani, 1996; Chen et al., 1999; Golub et al., 1999; Tusher et al., 2001; Chapelle et al., 2002; Guyon et al., 2002). However, all of these methods have the shortcoming that they cannot determine which variables covary in addition to being salient. This is the primary motivation for the method we propose.

Our proposal is that by studying the gradient of the classification function we can determine which variables are salient with respect to the classification problem and how these variables covary. The conceptual key is that an estimate of the gradient allows us to measure coordinate covariation since the inner product between partial derivatives $\left\langle \frac{\partial f_\phi}{\partial x^j}, \frac{\partial f_\phi}{\partial x^\ell} \right\rangle$ indicates the covariance of the j-th and ℓ-th coordinates with respect to variation in the classification function f_ϕ. This information is of central importance when an understanding is required of the effect of perturbing a salient explanatory variables (input features) on the other explanatory variables in addition to the response variable (the output). The method proposed in this paper gives an estimate of this covariation quantity. We implemented the method and tested it on a variety of simulated and real data sets,

further testing is provided in Mukherjee et al. (2006). These simulations suggest that the method does work for variable selection and some degree of covariation can be estimated. The efficacy of the method was clearly demonstrated on the simulated data and applying the method to gene expression data as well as images of digits (Mukherjee et al., 2006) gave an indication of its utility in understanding models of real data.

The method as currently implemented is designed for the setting of few samples and many dimensions. In this context it is more computationally intensive than methods that consider dimensions separately (Golub et al., 1999; Tusher et al., 2001) and of a similar complexity as methods based upon penalized loss (Tibshirani, 1996; Chen et al., 1999; Chapelle et al., 2002; Guyon et al., 2002). The method as is will scale very poorly as the number of examples increases. This can be addressed by using a basis set different than the difference between data points, for example the bases proposed in Lin and Zhang (2006).

To realize the objective of providing methodology and software to be used by biologists and clinicians for pathway extraction a system that works "right out of the box" is required. This means that the setting of the parameters of our algorithm (see Section 2.1) as well as decisions as to which variables are salient and which covary need to be automated. In addition finding blocks in the covariance matrix is a problem that needs to be addressed. We provide matlab code for the method—`http://www.stat.duke.edu/~sayan/covar1.html`.

Acknowledgments

We would like to thank D.X. Zhou for very useful discussions. We would like to thank the reviewers for many useful suggestions and comments. We would like to acknowledge support for this project from the Institute for Genome Sciences & Policy at Duke as well as the NIH grant for the Center for Public Genomics.

Appendix A. Proof of Theorem 10

The idea behind the proof is to first bound the $L^2_{\rho_X}$ differences by the excess error in Section A.1 and then bound the excess error in Section A.2. The proof is finished in Section A.3.

A.1 Bounding $L^2_{\rho_X}$ Differences by the Excess Error

Recall the empirical error (Definition 1) for $(g, \vec{f}) : X \to \mathbb{R}^{n+1}$

$$\mathcal{E}_{\mathbf{z}}(g, \vec{f}) = \frac{1}{m^2} \sum_{i,j=1}^{m} w_{i,j}^{(s)} \phi\big(y_i(g(x_j) + \vec{f}(x_i) \cdot (x_i - x_j))\big).$$

One can similarly define the expected error

$$\mathcal{E}(g, \vec{f}) = \int_Z \int_X w(x-u)\phi(y(g(u) + \vec{f}(x) \cdot (x-u)))\mathrm{d}\rho_X(u)\mathrm{d}\rho(x,y).$$

Unlike the standard setting of classification and regression $\mathcal{E}(g, \vec{f})$ and $\mathcal{E}_{\mathbf{z}}(g, \vec{f})$ are not respectively the expected and empirical mean of a random variable. This is due to the extra $\mathrm{d}\rho_X$ in the expected

error term. However, since

$$\mathbb{E}_{\mathbf{z}}[\mathcal{E}_{\mathbf{z}}(g,\vec{f})] = \frac{1}{ms^{n+2}}\mathcal{R}(g) + \frac{m-1}{m}\mathcal{E}(g,\vec{f}),$$

the empirical and expected errors should be close to each other if the empirical error concentrates with m increasing.

Define

$$\mathcal{R}_s = \int_X \int_Z w(x-u)\phi(yf_\phi(x))\mathrm{d}\rho(x,y)\mathrm{d}\rho_X(u).$$

We will use the *excess error,* $\mathcal{E}(g,\vec{f}) - \mathcal{R}_s$, to bound the $L^2_{\rho_X}$ differences.

For $r > 0$, denote

$$\mathcal{F}_r = \left\{(g,\vec{f}) \in \mathcal{H}_K^{n+1} : \|g\|_K^2 + \|\vec{f}\|_K^2 \le r^2\right\}.$$

Theorem 15 *Assume ρ_X satisfies the conditions* (7) *and* (8) *and $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$. For $(g,\vec{f}) \in \mathcal{F}_r$ with some $r > 1$, there exist constants $C_0, C_1 > 0$ such that*

$$\|g - f_\phi\|^2_{L^2_{\rho_X}} \le C_0\left(s^\theta r^2 + s^{2-\theta}B_r(\mathcal{E}(g,\vec{f}) - \mathcal{R}_s)\right)$$

and

$$\|f - \nabla f_\phi\|^2_{L^2_{\rho_X}} \le C_1\left(s^\theta r^2 + s^{-\theta}B_r(\mathcal{E}(g,\vec{f}) - \mathcal{R}_s)\right),$$

where $B_r = \min\left\{\frac{1}{q_1(c_0 r)}, r\right\}$ with some $c_0 > 0$.

To prove Theorem 15 we will need the following several lemmas which require the definition of the following quantities.

Definition 16 *Define for $(g,\vec{f}) : X \to \mathbb{R}^{n+1}$ the square error functional*

$$Q(g,\vec{f}) = \int_X \int_X w(x-u)\left(g(x) - f_\phi(x) + (\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u)\right)^2 \mathrm{d}\rho_X(u)\mathrm{d}\rho_X(x),$$

the border set

$$X_s = \left\{x \in X : d(x,\partial X) > s \text{ and } p(x) \ge (1+c_\rho)s^\theta\right\},$$

and the moments for $0 \le p < \infty$,

$$N_p = \int_{\{t\in\mathbb{R}^n : |t|\le 1\}} e^{-\frac{|t|^2}{2}}|t|^p \mathrm{d}t, \quad \text{and} \quad \tilde{N}_p = \int_{\mathbb{R}^n} e^{-\frac{|t|^2}{2}}|t|^p \mathrm{d}t.$$

Note that X_s is nonempty when s is small enough.

Lemma 17 *Under assumptions of Theorem 15*

$$\frac{N_0}{s^{2-\theta}}\int_{X_s}(g(x) - f_\phi(x))^2 \mathrm{d}\rho_X(x) + \frac{N_2 s^\theta}{n}\int_{X_s}|\vec{f}(x) - \nabla f_\phi(x)|^2 \mathrm{d}\rho_X(x) \le Q(g,\vec{f}).$$

Proof For $x \in X_s$, $\{u \in X : |u-x| \le s\} \subset X$ since $d(x,\partial X) > s$. For $u \in X$ such that $|u-x| \le s$

$$p(u) = p(x) - (p(x) - p(u)) \ge (1+c_\rho)s^\theta - c_\rho |u-x|^\theta \ge s^\theta.$$

Therefore,

$$\begin{aligned}
Q(g,\vec{f}) &\ge \int_{X_s}\int_{|u-x|\le s} w(x-u)\Big(g(x) - f_\phi(x) + (\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u)\Big)^2 p(u)\mathrm{d}u\mathrm{d}\rho_X(x) \\
&\ge s^\theta \int_{X_s}\int_{|u-x|\le s} w(x-u)\Big(g(x) - f_\phi(x) + (\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u)\Big)^2 \mathrm{d}u\mathrm{d}\rho_X(x) \\
&= s^\theta \int_{X_s}\int_{|u-x|\le s} w(x-u)(g(x)-f_\phi(x))^2 \mathrm{d}u\mathrm{d}\rho_X(x) \\
&\quad + 2s^\theta \int_{X_s}\int_{|u-x|\le s} w(x-u)(g(x)-f_\phi(x))((\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u))\mathrm{d}u\mathrm{d}\rho_X(x) \\
&\quad + s^\theta \int_{X_s}\int_{|u-x|\le s} w(x-u)((\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u))^2 \mathrm{d}u\mathrm{d}\rho_X(x) \\
&:= J_1 + J_2 + J_3.
\end{aligned}$$

It can be verified that

$$J_1 = \frac{1}{s^{2-\theta}}\int_{X_s}(g(x)-f_\phi(x))^2 \int_{|t|\le 1} e^{-\frac{|t|^2}{2}}\mathrm{d}t\mathrm{d}\rho_X(x) = \frac{N_0}{s^{2-\theta}}\int_{X_s}|g(x)-f_\phi(x)|^2\mathrm{d}\rho_X(x) \ .$$

In the following, denote by the superscripts of $x,u,t \in \mathbb{R}^n$ the corresponding coordinate indices. For every $i \in \{1,\ldots,n\}$

$$\int_{|u-x|\le s} w(x-u)(x^i-u^i)\mathrm{d}u = \frac{1}{s}\int_{|t|\le 1} e^{-\frac{|t|^2}{2}} t^i \mathrm{d}t = 0.$$

It follows that $J_2 = 0$.

Note that $((\vec{f}(x) - \nabla f_\phi(x))\cdot(x-u))^2$ equals

$$\sum_{i=1}^n\sum_{j=1}^n \left(f^i(x) - \frac{\partial f_\phi}{\partial x^i}(x)\right)\left(f^j(x) - \frac{\partial f_\phi}{\partial x^j}(x)\right)(x^i-u^i)(x^j-u^j).$$

But when $j \ne i$,

$$\int_{|u-x|\le s} w(x-u)(x^i-u^i)(x^j-u^j)\mathrm{d}u = \int_{|t|\le 1} e^{-\frac{|t|^2}{2}} t^i t^j \mathrm{d}t = 0.$$

Therefore

$$J_3 = s^\theta \sum_{i=1}^n \int_{X_s}(f^i(x) - \frac{\partial f_\phi}{\partial x^i}(x))^2 \int_{|t|\le 1} e^{-\frac{|t|^2}{2}}(t^i)^2 \mathrm{d}t\mathrm{d}\rho_X(x) = \frac{N_2 s^\theta}{n}\int_{X_s}|\vec{f}(x) - \nabla f_\phi(x)|^2\mathrm{d}\rho_X(x).$$

Plugging J_1 and J_3 into the inequality completes the proof. ■

In Lemma 20 below we will bound $Q(g,\vec{f})$ by the excess error $\mathcal{E}(g,\vec{f}) - \mathcal{R}_s$. For this purpose, we prove two facts which we state in Lemmas 18 and 19 and define the local error function of $t \in \mathbb{R}$ at $x \in X$ as

$$\mathrm{err}_x(t) = \mathbb{E}_{y\sim Y}[\phi(yt)] = \phi(t)P(1|x) + \phi(-t)P(-1|x),$$

which is a twice differentiable, univariate convex function for every $x \in X$.

Lemma 18 *For almost every $x \in X$, the following hold*

(i) *$f_\phi(x)$ is a minimizer of the function* $\mathrm{err}_x(t)$, i.e., $f_\phi(x) = \arg\min_{t\in\mathbb{R}} \mathrm{err}_x(t)$.

(ii) *If $T > \max\{|t|, \|f_\phi\|_\infty\}$, then*

$$\tfrac{1}{2}q_1(T)(t - f_\phi(x))^2 \le \mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \le \tfrac{1}{2}q_2(T)(t - f_\phi(x))^2.$$

(iii) *If $T \ge \{|t|, 3\|f_\phi\|_\infty\}$ there exists a constant $c_1 > 0$ such that*

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \ge c_1 \max\left\{q_1(T), \frac{1}{T}\right\}(t - f_\phi(x))^2.$$

Proof The first conclusion is a direct consequence of the fact

$$\mathcal{R}(f) = \int_X \mathrm{err}_x(f(x))d\rho_x(x).$$

Note that $(\mathrm{err}_x)'(f_\phi(x)) = 0$ since $f_\phi(x)$ is a minimizer of $\mathrm{err}_x(t)$. By a Taylor series expansion, there exists t_0 between t and $f_\phi(x)$ such that

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) = \frac{1}{2}(\mathrm{err}_x)''(t_0)(t - f_\phi(x))^2.$$

Since $(\mathrm{err}_x)''(t_0) = \phi''(t_0)P(1|x) + \phi''(-t_0)P(-1|x)$ and $|t_0| \le T$ the following holds

$$q_1(T) \le \phi''(t_0), \phi''(-t_0) \le q_2(T).$$

It follows $q_1(T) \le (\mathrm{err}_x)''(t_0) \le q_2(T)$ which proves (ii).

To show (iii), write

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) = \int_{f_\phi(x)}^{t}\int_{f_\phi(x)}^{r} (\mathrm{err}_x)''(a)\mathrm{d}a\mathrm{d}r.$$

Since $(\mathrm{err}_x)''(a)$ is positive, if $t \ge 3\|f_\phi\|_\infty := 3M_\phi$, then

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \ge \int_{2M_\phi}^{t}\int_{|f_\phi(x)|}^{2M_\phi} (\mathrm{err}_x)''(a)\mathrm{d}a\mathrm{d}r \ge q_1(2M_\phi)M_\phi(|t| - 2M_\phi)$$

and, if $t \le -3M_\phi$, then

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \ge \int_{-2M_\phi}^{t}\int_{-|f_\phi(x)|}^{-2M_\phi} (\mathrm{err}_x)''(a)\mathrm{d}a\mathrm{d}r \ge q_1(2M_\phi)M_\phi(|t| - 2M_\phi).$$

So, if $|t| > 3M_\phi$,

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \ge q_1(2M_\phi)M_\phi(|t| - 2M_\phi) \ge \frac{3q_1(2M_\phi)M_\phi}{16T}(t - f_\phi(x))^2,$$

where we have used the facts $|t|-2M_\phi \geq \frac{1}{4}|t-f_\phi(x)|$ and $|t-f_\phi(x)| \leq T+M_\phi \leq \frac{4}{3}T$. On the other hand, by (ii), if $|t| \leq 3M_\phi$

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \geq \frac{1}{2}q_1(3M_\phi)(t-f_\phi(x))^2 \geq \frac{3q_1(3M_\phi)M_\phi}{2T}(t-f_\phi(x))^2.$$

Hence for all $|t| \leq T$,

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \geq \frac{3q_1(3M_\phi)M_\phi}{16T}(t-f_\phi(x))^2.$$

Together with (ii), we obtain

$$\mathrm{err}_x(t) - \mathrm{err}_x(f_\phi(x)) \geq c_1 \max\left\{q_1(T), \frac{1}{T}\right\}(t-f_\phi(x))^2$$

with $c_1 = \min\left\{\frac{1}{2}, \frac{3q_1(3M_\phi)M_\phi}{16}\right\}$. ■

Lemma 19 *If $K \in C^2$, then there exists a constant $c_K > 0$ depending only on K such that*

$$|f(x)-f(u)| \leq c_K \|f\|_K |x-u|, \quad \forall f \in \mathcal{H}_K,\ x,u \in X.$$

Proof It follows from the reproducing property that

$$|f(x)-f(u)| = |\langle f, K(x,\cdot)-K(u,\cdot)\rangle| \leq \|f\|_K \sqrt{K(x,x)-2K(x,u)+K(u,u)}.$$

Denote $\nabla_1 K(x,u)$ as the gradient of $K(x,u)$ with respect to the first variable x. Since $K \in C^2$, we have

$$\begin{aligned} & K(x,x)-2K(x,u)+K(u,u) \\ = & \int_0^1 (\nabla_1(K(u+t(x-u),x) - \nabla_1 K(u+t(x-u),y)) \cdot (x-u)\mathrm{d}t \\ \leq & \int_0^1 |\nabla_1(K(u+t(x-u),x) - \nabla_1 K(u+t(x-u),y)|\,|x-u|\mathrm{d}t \\ \leq & (c_K)^2|x-u|^2 \end{aligned}$$

with

$$(c_K)^2 = \max\left\{\left\|\frac{\partial^2 K}{\partial x^i \partial u^j}\right\|_\infty, i,j = 1,\ldots,n\right\}.$$

Hence the conclusion is true. ■

Lemma 20 *Under the assumptions of Theorem 15, there exists a constant $c_2 > 0$ such that*

$$Q(g,\vec{f}) \leq c_2\left(r^2 s^2 + B_r(\mathcal{E}(g,\vec{f}) - \mathcal{R}_s)\right),$$

were B_r is defined as in Theorem 15 with $c_0 = \kappa \max\left\{3\|f_\phi\|_K, (1+D)\right\}$.

Proof For $(g,\vec{f})\in\mathcal{F}_r$ and $u,x\in X$, we have

$$|g(u)+\vec{f}(x)(x-u)|\le\kappa\|g\|_K+\kappa D\|\vec{f}\|_K\le c_0 r.$$

Since $c_0 r\ge 3\kappa\|f_\phi\|_K\ge 3\|f_\phi\|_\infty$, by Lemma 18 (iii),

$$\begin{aligned}\mathcal{E}(g,\vec{f})-R_s &= \int_X\int_X w(x-u)\Big(\mathrm{err}_x\big(g(u)+\vec{f}(x)\cdot(x-u)\big)-\mathrm{err}_x\big(f_\phi(x)\big)\Big)\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u)\\ &\ge \frac{c_1}{c_0}\frac{1}{B_r}\int_X\int_X w(x-u)\Big(g(u)+\vec{f}(x)\cdot(x-u)-f_\phi(x)\Big)^2\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u),\end{aligned}$$

Denote

$$\begin{aligned}t_1 &= g(u)-f_\phi(u)+(\vec{f}(u)-\nabla f_\phi(u))\cdot(x-u),\\ t_2 &= \big(f_\phi(u)-f_\phi(x)+\nabla f_\phi(u)\cdot(x-u)\big)+(\vec{f}(x)-\vec{f}(u))\cdot(x-u).\end{aligned}$$

We have

$$\mathcal{Q}(g,\vec{f})=\int_X\int_X w(x-u)(t_1)^2\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u).$$

Note that

$$\Big(g(u)+\vec{f}(x)\cdot(x-u)-f_\phi(x)\Big)^2=(t_1+t_2)^2\ge(t_1)^2+2t_1t_2\ge(t_1)^2-2|t_1||t_2|.$$

There holds

$$\frac{c_0}{c_1}B_r(\mathcal{E}(g,\vec{f})-R_s)\ge\mathcal{Q}(g,\vec{f})-2\int_X\int_X w(x-u)|t_1||t_2|\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u).$$

By the fact $\nabla f_\phi\in\mathcal{H}_K^n$ and Lemma 19, we have

$$|t_2|\le c_K(\|\nabla f_\phi\|_K+\|\vec{f}\|_K)|x-u|^2\le c_K(\|\nabla f_\phi\|_K+r)|x-u|^2.$$

Together with the assumption $p(x)\le c_\rho$ we obtain

$$\begin{aligned}&\int_X\int_X w(x-u)|t_1||t_2|\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u)\le\sqrt{\mathcal{Q}(g,\vec{f})}\left(\int_X\int_X w(x-u)|t_2|^2\mathrm{d}\rho_x(x)\mathrm{d}\rho_x(u)\right)^{1/2}\\ &\le c_K\left(\|\nabla f_\phi\|_K+r\right)\sqrt{\mathcal{Q}(g,\vec{f})}\left(c_\rho\int_X\int_{\mathbb{R}^n}w(x-u)|x-u|^4\mathrm{d}x\mathrm{d}\rho_x(u)\right)^{1/2}\\ &\le c_K\left(\|\nabla f_\phi\|_K+r\right)\sqrt{c_\rho\tilde{N}_4 s}\sqrt{\mathcal{Q}(g,\vec{f})}.\end{aligned}$$

Combining the above arguments we obtain

$$\mathcal{Q}(g,\vec{f})-2c_K\left(\|\nabla f_\phi\|_K+r\right)\sqrt{c_\rho\tilde{N}_4 s}\sqrt{\mathcal{Q}(g,\vec{f})}\le\frac{1}{c_1}\min\left\{\frac{1}{q_1(c_0r)},c_0r\right\}(\mathcal{E}(g,\vec{f})-\mathcal{R}_s).$$

Solving this inequality gives

$$\sqrt{\mathcal{Q}(g,\vec{f})}\le 2c_K\left(\|\nabla f_\phi\|_K+r\right)\sqrt{c_\rho\tilde{N}_4 s}+\sqrt{\frac{1}{c_1}\min\left\{\frac{1}{q_1(c_0r)},c_0r\right\}(\mathcal{E}(g,\vec{f})-\mathcal{R}_s)}.$$

This implies the conclusion with $c_2 = 2\max\left\{2(c_K)^2\left(\|\nabla f_\phi\|_K+1\right)^2 c_\rho \tilde{N}_4, \frac{c_0}{c_1}\right\}$. ■

Proof of Theorem 15. Write

$$\|g-f_\phi\|^2_{L^2_{\rho_X}} = \int_{X\setminus X_s}(g(x)-f_\phi(x))^2 d\rho_x(x) + \int_{X_s}(g(x)-f_\phi(x))^2 d\rho_x(x). \tag{10}$$

We have

$$\rho_x(X\setminus X_s) \le c_\rho s + (1+c_\rho)c_\rho|X|s^\theta \le (c_\rho+(1+c_\rho)c_\rho|X|)s^\theta,$$

where $|X|$ is the Lebesgue measure of X. So the first term on the right of (10) is bounded by

$$\kappa^2(r+\|f_\phi\|_K)^2(c_\rho+(1+c_\rho)c_\rho|X|)s^\theta.$$

By Lemmas 17 and 20, the second term on the right of (10) is bounded by

$$\frac{s^{2-\theta}}{N_0}c_2\left(r^2s^2+B_r\left(\mathcal{E}(f,\vec{f})-\mathcal{R}_s\right)\right)$$

Combing these two estimates finishes the proof of the first claim with

$$C_0 = \kappa^2(1+\|f_\phi\|_K)^2(c_\rho+(1+c_\rho)c_\rho|X|)+\frac{c_2}{N_0}.$$

Similarly, we can show the second claim with

$$C_1 = \kappa^2(1+\|\nabla f_\phi\|_K)^2(c_\rho+(1+c_\rho)c_\rho|X|)+\frac{nc_2}{N_2}.$$ ■

In order to apply Theorem 15 to $(g_{\mathbf{z}},\vec{f}_{\mathbf{z}})$, we need a bound on $\|g_{\mathbf{z}}\|^2_K+\|\vec{f}_{\mathbf{z}}\|^2_K$. We first state a rough bound.

Lemma 21 *For every $s>0$ and $\lambda>0$, $\|g_{\mathbf{z}}\|^2_K+\|\vec{f}_{\mathbf{z}}\|^2_K \le \frac{2\phi(0)}{\lambda s^{n+2}}$.*

Proof The conclusion follows from the fact

$$\frac{\lambda}{2}\left(\|g_{\mathbf{z}}\|^2_K+\|\vec{f}_{\mathbf{z}}\|^2_K\right) \le \mathcal{E}_{\mathbf{z}}(g_{\mathbf{z}},\vec{f}_{\mathbf{z}})+\frac{\lambda}{2}\left(\|g_{\mathbf{z}}\|^2_K+\|\vec{f}\|^2_K\right) \le \mathcal{E}_{\mathbf{z}}(0,\vec{0})+0 = \frac{\phi(0)}{s^{n+2}}.$$

■

Remark 22 *Using this quantity the bound in Theorem 15 is at least of order $O(\frac{1}{\lambda s^{n+2-\theta}})$ which tends to ∞ as $s\to 0$ and $\lambda\to 0$. So a sharper bound is needed. We will obtain such a bound in Section A.3.*

A.2 Bounding the Excess Error

In this section, we bound the quantity $\mathcal{E}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) - \mathcal{R}_s$. Let

$$(g_\lambda, \vec{f}_\lambda) = \operatorname*{arg\,min}_{(g,\vec{f})\in\mathcal{H}_K^{n+1}} \left\{ \mathcal{E}(g,\vec{f}) + \frac{\lambda}{2}(\|g\|_K^2 + \|\vec{f}\|_K^2) \right\}.$$

Theorem 23 *If $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$, $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_r$ for some $r \geq 1$, then with confidence $1-\delta$*

$$\mathcal{E}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) - \mathcal{R}_s \leq C_2 \left(\frac{L_r r + M_r \log\frac{2}{\delta}}{\sqrt{m} s^{n+2}} + s^2 + \lambda \right),$$

where $C_2 > 0$ is a constant depending on ϕ and ρ but not on r, s and λ.

By a standard decomposition procedure, we have the following result.

Proposition 24 *The following hold*

$$\mathcal{E}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) - \mathcal{R}_s + \frac{\lambda}{2}\left(\|g_{\mathbf{z}}\|_K^2 + \|\vec{f}_{\mathbf{z}}\|_K^2 \right) \leq \mathscr{S}(\mathbf{z}) + \mathscr{A}(\lambda)$$

where

$$\mathscr{S}(\mathbf{z}) = \left(\mathcal{E}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) - \mathcal{E}_{\mathbf{z}}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) \right) + \left(\mathcal{E}_{\mathbf{z}}(g_\lambda, \vec{f}_\lambda) - \mathcal{E}(g_\lambda, \vec{f}_\lambda) \right)$$

and

$$\mathscr{A}(\lambda) = \inf_{(g,\vec{f})\in\mathcal{H}_K^{n+1}} \left\{ \mathcal{E}(g,\vec{f}) - \mathcal{R}_s + \frac{\lambda}{2}\left(\|g\|_K^2 + \|\vec{f}\|_K^2 \right) \right\}.$$

The quantity $\mathscr{S}(\mathbf{z})$ is called the sample error and can be bound by controlling

$$S(\mathbf{z}, r) := \sup_{(g,\vec{f})\in\mathcal{F}_r} |\mathcal{E}_{\mathbf{z}}(g,\vec{f}) - \mathcal{E}(g,\vec{f})|.$$

In fact, if both $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_r$ for some $r > 0$, then

$$\mathscr{S}(\mathbf{z}) \leq 2S(\mathbf{z}, r). \tag{11}$$

Again $\mathcal{E}_{\mathbf{z}}(g,\vec{f})$ and $\mathcal{E}(g,\vec{f})$ are not the empirical and expected means of a random variable. We will use McDiarmid's inequality (McDiarmid, 1989) to bound $S(\mathbf{z}, r)$.

Lemma 25 *For every $r > 0$*

$$\mathbf{P}\mathrm{rob}\left\{ |S(\mathbf{z}, r) - \mathbb{E}S(\mathbf{z}, r)| > \varepsilon \right\} \leq 2\exp\left(-\frac{m\varepsilon^2 s^{2(n+2)}}{2M_r^2} \right).$$

Proof Denote by $\mathbf{z}_i'$ the sample which coincides with $\mathbf{z}$ except for the i-th pair (x_i, y_i) replaced by (x_i', y_i'). It is easy to verify that

$$\begin{aligned} S(\mathbf{z}, r) - S(\mathbf{z}_i', r) &= \sup_{(g,\vec{f})\in\mathcal{F}_r} \left(\mathcal{E}_{\mathbf{z}}(g,\vec{f}) - \mathcal{E}(g,\vec{f}) \right) - \sup_{(g,\vec{f})\in\mathcal{F}_r} \left(\mathcal{E}_{\mathbf{z}_i'}(g,\vec{f}) - \mathcal{E}(g,\vec{f}) \right) \\ &\leq \sup_{(g,\vec{f})\in\mathcal{F}_r} \left(\mathcal{E}_{\mathbf{z}}(g,\vec{f}) - \mathcal{E}_{\mathbf{z}_i'}(g,\vec{f}) \right) \leq \frac{2m-1}{m^2} \frac{M_r}{s^{n+2}}. \end{aligned}$$

Interchanging the roles of $\mathbf{z}$ and $\mathbf{z}_i'$ gives $|S(\mathbf{z},r)-S(\mathbf{z}_i',r)| \le \frac{2M_r}{ms^{n+2}}$. By McDiarmid's inequality we obtain the desired estimate. ■

Lemma 26 *For every $r>0$*

$$\mathbb{E}S(\mathbf{z},r) \le \frac{8L_r(\kappa(1+2D)r+\phi(0))}{s^{n+2}\sqrt{m}} + \frac{2M_r}{ms^{n+2}}.$$

In order to prove this lemma, we need Rademacher complexities. We refer to Koltchinskii and Panchenko (2000) and van der Vaart and Wellner (1996) for definitions and properties.

Proof Denote $\xi(x,y,u) = w(x-u)\phi(y(g(u)+\vec{f}(x)\cdot(x-u)))$ for simplicity. Then $\mathcal{E}(g,\vec{f}) = \mathbb{E}_u\mathbb{E}_{(x,y)}\xi(x,y,u)$ and $\mathcal{E}_{\mathbf{z}}(g,\vec{f}) = \sum_{i,j=1}^m \xi(x_i,y_i,x_j)$. One can easily check that

$$\begin{aligned}
S(\mathbf{z},r) &\le \sup_{(g,\vec{f})\in\mathcal{F}_r}\left|\mathcal{E}(g,\vec{f}) - \frac{1}{m}\sum_{j=1}^m \mathbb{E}_{(x,y)}\xi(x,y,x_j)\right| + \sup_{(g,\vec{f})\in\mathcal{F}_r}\left|\frac{1}{m}\sum_{j=1}^m \mathbb{E}_{(x,y)}\xi(x,y,x_j) - \mathcal{E}_{\mathbf{z}}(g,\vec{f})\right| \\
&\le \mathbb{E}_{(x,y)}\sup_{(g,\vec{f})\in\mathcal{F}_r}\left|\mathbb{E}_u\xi(x,y,u) - \frac{1}{m}\sum_{i=1}^m \xi(x,y,x_j)\right| \\
&\quad + \frac{1}{m}\sum_{j=1}^m \sup_{(g,\vec{f})\in\mathcal{F}_r}\sup_{u\in X}\left|\mathbb{E}_{(x,y)}\xi(x,y,u) - \frac{1}{m-1}\sum_{\substack{i=1\\ i\ne j}}^m \xi(x_i,y_i,u)\right| \\
&\quad + \frac{1}{m}\sum_{j=1}^m\left(\frac{1}{m}\xi(x_j,y_j,x_j) + \frac{1}{m(m-1)}\sum_{\substack{i=1\\ i\ne j}}^m \xi(x_i,y_i,x_j)\right) \\
&:= S_1+S_2+S_3.
\end{aligned}$$

Let $\varepsilon_i, i=1,\dots,m$ be independent Rademacher variables. Denote

$$G_{(x,y)} = \left\{h(u) = y(g(u)+\vec{f}(x)\cdot(x-u)) : (g,\vec{f})\in\mathcal{F}_r\right\}$$

for every $(x,y)\in Z$. For S_1, by using the properties of Rademacher complexities, we have

$$\begin{aligned}
\mathbb{E}S_1(\mathbf{z}) &= \mathbb{E}_{(x,y)}\mathbb{E}\sup_{h\in G_{x,y}}\left|\mathbb{E}_u[w(x-u)\phi(h(u))] - \frac{1}{m}\sum_{j=1}^m w(x-x_j)\phi(h(x_j))\right| \\
&\le 2\sup_{(x,y)\in Z}\mathbb{E}\sup_{h\in G_{(x,y)}}\left|\frac{1}{m}\sum_{j=1}^m \varepsilon_j w(x-x_j)\phi(h(x_j))\right| \\
&\le \frac{4}{s^{n+2}}\sup_{(x,y)\in Z}\mathbb{E}\sup_{h\in G_{(x,y)}}\left|\frac{1}{m}\sum_{j=1}^m \varepsilon_j\phi(h(x_j))\right| \\
&\le \frac{4L_r}{s^{n+2}}\left(\sup_{(x,y)\in Z}\mathbb{E}\sup_{h\in G_{(x,y)}}\left|\frac{1}{m}\sum_{j=1}^m \varepsilon_j h(x_j)\right| + \frac{\phi(0)}{\sqrt{m}}\right) \\
&\le \frac{4L_r}{s^{n+2}}\left(\mathbb{E}\sup_{\|g\|_K^2\le r^2}\left|\sum_{j=1}^m \varepsilon_j g(x_j)\right| + 2\kappa r\sup_{x\in X}\mathbb{E}\left|\sum_{j=1}^m \varepsilon_j\|x-x_j\|\right| + \frac{\phi(0)}{\sqrt{m}}\right) \\
&\le \frac{4L_r\big(\kappa(1+2D)r+\phi(0)\big)}{s^{n+2}\sqrt{m}}.
\end{aligned}$$

Similarly, we can verify

$$\mathbb{E}S_2(\mathbf{z}) \leq \frac{4L_r(\kappa(1+2D)r+\phi(0))}{s^{n+2}\sqrt{m-1}}.$$

Obviously $S_3 \leq \frac{2M_r}{ms^{n+2}}$. Combining the estimates for S_1, S_2, and S_3 completes the proof. ■

Proposition 27 *Assume $r > 1$. There exists a constant $c_2 > 0$ such that with confidence at least $1-\delta$*

$$\mathscr{S}(\mathbf{z}) \leq c_3 \frac{L_r r + M_r \log\frac{2}{\delta}}{\sqrt{m}s^{n+2}}.$$

Proof The result is a direct application of inequality (11) and Lemmas 25 and 26. ■

We now bound the approximation error $\mathscr{A}(\lambda)$.

Proposition 28 *If $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$, then $\mathscr{A}(\lambda) \leq c_4(s^2+\lambda)$ for some $c_4 > 0$.*

Proof By the definition of $\mathscr{A}(\lambda)$ and the fact that $(f_\phi, \nabla f_\phi) \in \mathcal{H}_K^{n+1}$

$$\mathscr{A}(\lambda) \leq \mathcal{E}(f_\phi, \nabla f_\phi) - \mathcal{R}_s + \frac{\lambda}{2}(\|f_\phi\|_K^2 + \|\nabla f_\phi\|_K^2).$$

By Lemma 18 (ii), we have

$$\begin{aligned}
\mathcal{E}(f_\phi, \nabla f_\phi) - \mathcal{R}_s &= \int_X \int_X w(x-u)\Big(\mathrm{err}_x(f_\phi(u) + \nabla f_\phi(x)\cdot(x-u)) - \mathrm{err}_x(f_\phi(x)\Big)\mathrm{d}\rho_x(u)\mathrm{d}\rho_x(x) \\
&\leq q_2(\tilde{M}_\phi)\int_X \int_X w(x-u)\Big(f_\phi(u) - f_\phi(x) + \nabla f_\phi(x)\cdot(x-u)\Big)^2 \mathrm{d}\rho_x(u)\mathrm{d}\rho_x(x) \\
&\leq q_2(\tilde{M}_\phi)(c_K)^2\|\nabla f_\phi\|_K^2 c_\rho \int_X \int_X w(x-u)|x-u|^4 \mathrm{d}u\mathrm{d}\rho_x(x) \leq q_2(\tilde{M}_\phi)(c_K)^2\|\nabla f_\phi\|_K^2 c_\rho \tilde{N}_4 s^2,
\end{aligned}$$

where $\tilde{M}_\phi = \kappa\|f_\phi\|_K + \kappa D\|\nabla f_\phi\|_K$. Taking

$$c_4 = \max\{q_2(\tilde{M}_\phi)(c_K)^2\|\nabla f_\phi\|_K^2 c_\rho \tilde{N}_4, \tfrac{1}{2}(\|f_\phi\|_K^2 + \|\nabla f_\phi\|_K^2)\},$$

the result follows. ■

Theorem 23 follows directly from Propositions 24, 27 and 28.

A.3 Proof of Theorem 10

We will use Theorems 15 and 23 to prove Theorem 10.

Notice that both theorems need a bound r so that $(g_\mathbf{z}, \vec{f}_\mathbf{z})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_r$. In Lemma 21 we have shown

$$\|g_\mathbf{z}\|_K^2 + \|\vec{f}_\mathbf{z}\|_K^2 \leq \frac{2\phi(0)}{\lambda s^{n+2}}.$$

Similarly we can show $\|g_\lambda\|_K^2 + \|\vec{f}_\lambda\|_K^2$ is also bounded by $\frac{2\phi(0)}{\lambda s^{n+2}}$. So $\sqrt{\frac{2\phi(0)}{\lambda s^{n+2}}}$ is a universal bound for r such that $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_r$. However, this bound is not sharp enough to be useful for Theorem 15 (see Remark 22).

A sharper bound will be given below. This bound also improves the sample error estimate and the estimate in Theorem 23.

Lemma 29 *Under the assumptions of Theorem 10*

$$\|g_\lambda\|_K^2 + \|\vec{f}_\lambda\|_K^2 \le 2c_4\left(\frac{s^2}{\lambda} + 1\right).$$

Proof Since $\mathcal{E}(g, \vec{f}) - \mathcal{R}_s$ is non-negative for all pairs $(g, \vec{f})$, we have

$$\tfrac{\lambda}{2}(\|g_\lambda\|_K^2 + \|\vec{f}_\lambda\|_K^2) \le \mathcal{E}(g_\lambda, \vec{f}_\lambda) - \mathcal{R}_s + \tfrac{\lambda}{2}(\|g_\lambda\|_K^2 + \|\vec{f}_\lambda\|_K^2) = \mathscr{A}(\lambda).$$

This in conjunction with Proposition 28 implies the conclusion. ■

Lemma 30 *Under the assumptions of Theorem 10 with confidence at least* $1-\delta$

$$\|g_{\mathbf{z}}\|_K^2 + \|\vec{f}_{\mathbf{z}}\|_K^2 \le c_5\left\{1 + \frac{s^2}{\lambda} + \left(\frac{L_{\lambda,s}}{\sqrt{\lambda s^{n+2}}} + M_{\lambda,s}\log\frac{2}{\delta}\right)\frac{1}{\sqrt{m\lambda s^{n+2}}}\right\}$$

for some $c_5 > 0$.

Proof By the fact $\mathcal{E}(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) - \mathcal{R}_s > 0$ and Proposition 24 we have

$$\tfrac{\lambda}{2}\left(\|g_{\mathbf{z}}\|_K^2 + \|\vec{f}_{\mathbf{z}}\|_K^2\right) \le \mathscr{S}(\mathbf{z}) + \mathscr{A}(\lambda).$$

Since both $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_{\sqrt{2\phi(0)/\lambda s^{n+2}}}$, we apply Proposition 27 to get with probability at least $1-\delta$

$$\mathscr{S}(\mathbf{z}) \le c_3\left(L_{\lambda,s}\sqrt{\frac{2\phi(0)}{\lambda s^{n+2}}} + M_{\lambda,s}\log\frac{2}{\delta}\right)\frac{1}{\sqrt{m}s^{n+2}}.$$

Together with Proposition 28, we obtain the desired estimate with $c_5 = 2\max\{c_3, c_4\}$. ■

We now prove Theorem 10.

Proof of Theorem 10. By Theorems 15 and 23 we have with probability at least $1-\delta$ both $\|g_{\mathbf{z}} - f_\phi\|_{L^2_{\rho_X}}$ and $\||\vec{f}_{\mathbf{z}} - \nabla f_\phi|\|_{L^2_{\rho_X}}^2$ are bounded by

$$\max\{C_0, C_1\}\left\{r^2 s^\theta + C_2 B_r\left(\frac{L_r r + M_r\log\frac{2}{\delta}}{\sqrt{m}s^{n+2}} + s^2 + \lambda\right)s^{-\theta}\right\}, \tag{12}$$

if both $(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}})$ and $(g_\lambda, \vec{f}_\lambda)$ are in $\mathcal{F}_r$ for some $r > 1$. By Lemmas 29 and 30 we can state that both $\{(g_{\mathbf{z}}, \vec{f}_{\mathbf{z}}) \in \mathcal{F}_r\}$ and $\{(g_\lambda, \vec{f}_\lambda) \in \mathcal{F}_r\}$ with probability at least $1-\delta$ if

$$r^2 = \max(c_4, c_5, 1)\left\{1 + \frac{s^2}{\lambda} + \left(\frac{L_{\lambda,s}}{\sqrt{\lambda s^{n+2}}} + M_{\lambda,s}\log\frac{2}{\delta}\right)\frac{1}{\sqrt{m\lambda s^{n+2}}}\right\}.$$

Substituting the above r into (12) gives us the desired bound with confidence at least $1-2\delta$. ■

References

N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68(6):337–404, 1950.

B. L. Bartlett, M. L. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2005.

O. Chapelle, V.N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

C. Cortes and V.N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations for object detection using kernel classifiers. In *Proceedings of Asian Conference on Computer Vision*, 2000a.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000b.

T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286: 531–537, 1999.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

L. Hermes and J.M. Buhmann. Feature selection for support vector machines. In *International Conference on Pattern Recognition*, 2000.

V.I. Koltchinskii and D. Panchenko. Rademacher processes and bounding the risk of function learning. In J. Wellner E. Giné, D. Mason, editor, *High Dimensional Probability II*, pages 443–459, 2000.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and applications to the classification of microarray data and satellite radiance data. *J. Amer. Stat. Soc.*, 99:67–81, 2004.

F. Liang, S. Mukherjee, and M. West. Understanding the use of unlabeled data in predictive modeling. *Statistical Science*, 2006. accepted.

Y. Lin and H.H. Zhang. Component selection and smoothing in smoothing spline analysis of variance models. *Annals of Statistics*, 2006. in press.

C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics*, volume 141, pages 148–188. LMS Lecture Notes Series, 1989.

C.A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

S. Mukherjee and D.X. Zhou. Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, 7:519–549, 2006.

S. Mukherjee, Q. Wu, and D. X. Zhou. Learning gradients and feature selection on manifolds. *Annals of Statistics*, 2006. submitted.

T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

B. Schoelkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

D.K. Slonim, P. Tamayo, J.P. Mesirov, T.R. Golub, and E.S. Lander. Class prediction and discovery using gene expression data. In *Proc. of the 4th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 263–272, 2000.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J Royal Stat Soc B*, 58(1):267–288, 1996.

V.G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, 98(9):5116–21, 2001.

A. van der Vaart and J. Wellner. *Weak convergence and empirical processes*. Springer-Verlag, New York, 1996.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

Q. Wu and D.X. Zhou. Support vector machine classifiers: Linear programming versus quadratic programming. *Neural Computation*, 17:1160–1187, 2005.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statis.*, 32:56–85, 2004.