Sparse Feature Selection in Kernel Discriminant Analysis via Optimal Scoring

Alexander F. Lapanowski **Irina Gaynanova**
Texas A&M University
{alapanow, irinag}@stat.tamu.edu

Abstract

We consider the two-group classification problem and propose a kernel classifier based on the optimal scoring framework. Unlike previous approaches, we provide theoretical guarantees on the expected risk consistency of the method. We also allow for feature selection by imposing structured sparsity using weighted kernels. We propose fully-automated methods for selection of all tuning parameters, and in particular adapt kernel shrinkage ideas for ridge parameter selection. Numerical studies demonstrate the superior classification performance of the proposed approach compared to existing nonparametric classifiers.

1 Introduction

Linear Discriminant Analysis (LDA) is a popular linear classification rule [13, Section 4.3], but it has two limitations. First, it will underfit the data when the best decision boundary is nonlinear. Secondly, LDA uses all p features even though not all may contribute to class separation. Including such "noise" features into the classification rule can harm classification performance.

To account for non-linearity, several authors consider kernel discriminant analysis [4, 29, 31, 32]. While the methods have good empirical performance, to our knowledge there is a lack of theoretical guarantees on the risk of the learned classifiers. Recently, [12, 22, 25] provided such guarantees, however under modified classification criterion with respect to worst-case training data realization. At the same time, none of the above methods perform feature selection, and as such will overfit in the presence of "noise" features.

On the other hand, several sparse generalizations of LDA have been proposed [6, 10, 15], however the methods still result in linear classification boundaries.

This paper addresses the gap between kernel and sparse LDA methods by using an optimal scoring framework [19] to construct a kernel-based classifier. Unlike previous approaches, we provide theoretical guarantees on the risk consistency of the proposed kernel optimal scoring. We also allow the method to perform feature selection by adapting the weighted kernel idea from [1]. To avoid computational costs associated with selecting multiple tuning parameters, we develop a new Stabilization method for ridge parameter selection. The method is based on the shrinkage ideas from [24] for stabilization of kernel matrices. Our empirical results indicate that the Stabilization method leads to better error rates than generalized cross-validation (GCV) [11, 16, 36], and we believe this method of parameter selection could be of independent interest.

In summary, this work makes the following contributions: (i) we develop a kernel LDA method based on optimal scoring framework; (ii) we provide theoretical results on the risk consistency of the proposed classifier; (iii) we use weighted kernels to implement feature selection within kernel LDA; and (iv) we propose a new stabilization method for ridge parameter selection.

1.1 Related Work

In this section we draw connections between our work and existing literature on kernelized optimal scoring as well as sparse feature selection within kernels.

To our knowledge, the kernelized version of the optimal scoring problem has not been considered in the literature except for [31]. Unlike [31], we fix the scores and provide theoretical guarantees for the method. Another major distinction of our method is the feature selection which is achieved by weighting the kernel and

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

adding a sparsity penalty to the weights.

Weighted kernels with sparse weights have been considered in [1, 8] in the context of kernel regression and kernel support vector machines. The framework can not be applied to the original kernel LDA method [29], however it could be adapted to the proposed kernel optimal scoring problem due to its least squares formulation.

Learning the optimal weight vector can be viewed as a kernel learning problem. While most of the kernel learning literature focuses on finding linear or quadratic combination of predetermined kernels [3, 33], learning the weights corresponds to adjusting the feature support of the kernel matrix. This is also distinctive from the sparse kernel learning literature, where the kernel is assumed to be additive with respect to the features [2, 34]. Our framework does not impose additivity, thus enabling interactions between the features.

1.2 Notation

For a vector $v \in \mathbb{R}^p$, let $\|v\|_2 := \sqrt{\sum_{i=1}^p |v_i|^2}$ be the Euclidean norm, $\|v\|_1 := \sum_{i=1}^p |v_i|$ be the ℓ^1 norm, and $\|v\|_\infty := \max |v_i|$ be the ℓ^∞ norm. Let $\langle x, x' \rangle := \sum_{i=1}^p x_i x'_i$ be the Euclidean inner product in $\mathbb{R}^p$. For a matrix $M \in \mathbb{R}^{n\times k}$, let $M_{i,j}$ denote the (i,j) element of M. Let $\|M\|_{\text{op}} := \sup_{\|x\|_2=1} \|Mx\|_2$ be the operator norm, and let $\|M\|_F := \sqrt{\sum_{i=1}^n \sum_{j=1}^k |M_{i,j}|^2}$ be the Frobenius norm. Let I be the $n \times n$ identity matrix. Let $\mathbf{1} \in \mathbb{R}^n$ be the vector of all 1s, and let $C = I - n^{-1}\mathbf{1}\mathbf{1}^\top$ be the centering matrix.

2 Kernel Optimal Scoring

2.1 Linear Discriminant Analysis and Optimal Scoring

Let $\{(x_i, y_i)\}_{i=1}^n$ be independent pairs, where $x_i \in \mathbb{R}^p$ is the vector of features, and $y_i \in \mathbb{R}^2$ is the indicator of class membership such that $y_{ik} = 1$ if ith sample belongs to class k, $i \in C_k$, and $y_{ik} = 0$ otherwise. Let n_1 and n_2 denote the number of samples in each respective class so that $n = n_1 + n_2$. Let $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times 2}$ denote the corresponding feature and indicator matrices, and without loss of generality let X be column-centered.

The optimal scoring problem [19] finds the discriminant vector $\beta \in \mathbb{R}^p$ and the scores vector $\theta \in \mathbb{R}^2$ by solving

$$\begin{aligned} &\underset{\theta,\beta}{\text{minimize}}\, \|Y\theta - X\beta\|_2^2 \\ &\text{subject to } n^{-1}\theta^\top Y^\top Y\theta = 1,\, \theta^\top Y^\top Y\mathbf{1} = 0. \end{aligned} \tag{1}$$

Since the solution vector of scores has explicit form up to a sign, $\widehat{\theta} = (\sqrt{n_2/n_1}\ -\sqrt{n_1/n_2})^\top$, (1) is equivalent to the linear regression problem

$$\underset{\beta}{\text{minimize}}\, \|Y\widehat{\theta} - X\beta\|_2^2. \tag{2}$$

The solution $\widehat{\beta}$ corresponds to the discriminant vector in LDA up to scaling [17, Section 3.4]. Thus, linear discriminant analysis can be reduced to finding the solution to problem (2).

2.2 Reproducing Kernel Hilbert Spaces

Reproducing Kernel Hilbert Spaces (RKHS) are commonly used in creating non-linear classifiers. The data is mapped into a RKHS $\mathcal{H}$ via $\Phi : \mathbb{R}^p \to \mathcal{H}$ with an accompanying kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ such that $\langle \Phi(x), \Phi(x')\rangle_{\mathcal{H}} = k(x, x')$ for any $x, x' \in \mathbb{R}^p$. We let $\|\cdot\|_{\mathcal{H}}$ be the norm induced by the inner product $\langle\cdot\,,\,\cdot\rangle_{\mathcal{H}}$. By the *reproducing property* of $\mathcal{H}$: $\langle \Phi(x), f\rangle_{\mathcal{H}} = f(x)$ for all $x \in \mathbb{R}^p$ and $f \in \mathcal{H}$. Thus, any classifier that relies on the training data only through the inner products can be *kernelized* by substituting kernel evaluations in place of inner products. This effectively creates a classifier in $\mathcal{H}$ rather than in $\mathbb{R}^p$.

Some commonly-used kernels are the gaussian kernel $k(x, x') = \exp(-\sigma^{-2}\|x - x'\|_2^2)$ with $\sigma > 0$, the polynomial kernel $k(x, x') = (1 + \langle x, x'\rangle)^d$ with d a positive integer, and the sigmoid kernel $k(x, x') = \tanh(c\langle x, x'\rangle + t)$ with $c > 0$, $t \geq 0$. We refer the reader to [32, Chapter 13] for a review on kernel construction and selection. We let $\mathbf{K} \in \mathbb{R}^{n\times n}$ denote the kernel matrix $\mathbf{K}_{i,j} := k(x_i, x_j)$ based on observed feature vectors $\{x_i\}_{i=1}^n$.

2.3 Kernel Optimal Scoring

We derive the kernelized formulation of the optimal scoring problem (2). Let f be the discriminant function in $\mathcal{H}$ with corresponding map Φ and kernel k. We substitute each inner product $x_i^\top\beta = \langle x_i, \beta\rangle$ with inner product in $\mathcal{H}$, $\langle \Phi(x_i) - \overline{\Phi}, f\rangle_{\mathcal{H}}$, where we apply centering to $\Phi(x_i)$ via $\overline{\Phi} := n^{-1}\sum_{i=1}^n \Phi(x_i)$ to take into account column-centering of X. The corresponding optimal scoring problem in $\mathcal{H}$ takes the form

$$\underset{f\in\mathcal{H}}{\text{minimize}} \left\| Y\widehat{\theta} - \begin{pmatrix} \left\langle \Phi(x_1) - \overline{\Phi}\,,\, f\right\rangle_{\mathcal{H}} \\ \vdots \\ \left\langle \Phi(x_n) - \overline{\Phi}\,,\, f\right\rangle_{\mathcal{H}} \end{pmatrix} \right\|_2^2 .$$

By the Representer Theorem [23], the minimizing $\widehat{f}$ lies in the finite-dimensional span of the centered data, that is it is sufficient to consider minimization over $f = \sum_{i=1}^n \alpha_i[\Phi(x_i) - \overline{\Phi}]$ for some $\alpha_i \in \mathbb{R}$. Combining the Representer Theorem with kernel representation of inner-products in $\mathcal{H}$ leads to the equivalent coefficient space formulation of the kernel optimal scoring problem:

$$\underset{\alpha\in\mathbb{R}^n}{\text{minimize}}\ \|Y\widehat{\theta}-C\mathbf{K}C\alpha\|_2^2. \tag{3}$$

Kernel methods may over-fit the training data without further restriction on the set of functions $f\in\mathcal{H}$, [13, 32, 30]. A common approach is to restrict the norm $\|f\|_{\mathcal{H}}^2=\alpha^\top CKC\alpha$, and we add a ridge penalty to the objective function (3)

$$\underset{\alpha\in\mathbb{R}^n}{\text{minimize}}\left\{\frac{1}{n}\|Y\widehat{\theta}-C\mathbf{K}C\alpha\|_2^2+\gamma\alpha^\top C\mathbf{K}C\alpha\right\}, \tag{4}$$

where $\gamma>0$ controls the level of regularization. For numerical stability, we also add εI with small $\varepsilon>0$ to the ridge penalty so that CKC is replaced with $CKC+\varepsilon I$. A similar adjustment is used in [29, 31]. We fix $\varepsilon=10^{-5}$ throughout the manuscript. The problem has a closed-form solution leading to

$$\widehat{\alpha}=\{(C\mathbf{K}C)^2+n\gamma(C\mathbf{K}C+\varepsilon I)\}^{-1}C\mathbf{K}CY\widehat{\theta}. \tag{5}$$

We call (4) the kernel optimal scoring problem or KOS.

2.4 Classification of a New Data Point

In this section we describe how to use KOS for classification. Let $\widehat{\alpha}$ be as in (5), and let $\widehat{f}=\sum_{i=1}^n\widehat{\alpha}_i[\Phi(x_i)-\overline{\Phi}]$. Given a new data point $x\in\mathbb{R}^p$, let

$$K(X,x)=\begin{pmatrix}k(x_1,x) & \cdots & k(x_n,x)\end{pmatrix}^\top.$$

We define the projected value $P(x)$ as the inner-product between x mapped and centered in $\mathcal{H}$ and $\widehat{f}$ so that $P(x)$ is equal to

$$\left\langle\Phi(x)-\overline{\Phi},\,\widehat{f}\right\rangle_{\mathcal{H}}=(K(X,x)^\top-n^{-1}\mathbf{1}^\top\mathbf{K})C\widehat{\alpha}. \tag{6}$$

The derivation of (6) is in the Supplement.

KOS classifies $x\in\mathbb{R}^p$ using nearest centroids classification on the projected values. Specifically, let $\mu_k=\frac{1}{n_k}\sum_{i\in G_k}P(x_x)$ be the mean projected values of group k (projected centroid). We classify $x\in\mathbb{R}^p$ according to the minimal distance to projected centroids

$$\underset{k=1,2}{\text{argmin}}\,|P(x)-\mu_k|.$$

3 Error Bounds for Kernel Optimal Scoring

Problem (4) can be viewed as a regularized empirical risk minimization problem

$$\widehat{f}=\underset{f\in\mathcal{H}}{\text{argmin}}\left\{R_{\text{emp}}(f)+\gamma\|f\|_{\mathcal{H}}^2\right\}, \tag{7}$$

where for a fixed $f\in\mathcal{H}$

$$R_{\text{emp}}(f):=\frac{1}{n}\sum_{i=1}^n|y_i^\top\widehat{\theta}-\left\langle\Phi(x_i)-\overline{\Phi},f\right\rangle|^2. \tag{8}$$

By duality, for every $\gamma\geq0$ there exists a $\tau\geq0$ such that

$$\widehat{f}=\underset{\|f\|_{\mathcal{H}}\leq\tau}{\text{argmin}}\left\{R_{\text{emp}}(f)\right\}. \tag{9}$$

While the relationship between γ and τ is data-dependent, Lemma 3 in the Supplement shows that $\tau\leq C\min(\gamma^{-1},\gamma^{-1/2})$ for some constant $C>0$. For technical clarity, we analyze (9) throughout.

There are two complications in analyzing the empirical risk in (8): $\widehat{\theta}$ is dependent on all y_i through n_1, n_2, and $\overline{\Phi}$ is dependent on all x_i. Hence, the error terms $|y_i^\top\widehat{\theta}-\left\langle\Phi(x_i)-\overline{\Phi},f\right\rangle|^2$ are dependent. The empirical risk can be equivalently written as

$$R_{\text{emp}}(f,\beta)=\frac{1}{n}\sum_{i=1}^n|y_i^\top\widehat{\theta}-\beta-\langle\Phi(x_i),f\rangle|^2,$$

with the minimizing $\widehat{\beta}=-\langle\overline{\Phi},f\rangle$ since $\mathbf{1}^\top Y\widehat{\theta}=0$. We therefore introduce a modified empirical risk using population scores θ^* and an extra intercept parameter $\beta\in\mathbb{R}$. The population scores θ^* result from substituting π_k instead of n_k/n in $\widehat{\theta}$.

Definition 1. *Let $\pi_k=P(i\in C_k)$ be the prior class probabilities, $k=1,2$. The* population scores *are defined as $\theta^*=(\sqrt{\pi_2/\pi_1}\ \ -\sqrt{\pi_1/\pi_2})^\top$.*

For a fixed $f\in\mathcal{H}$ and $\beta\in\mathbb{R}$, the modified empirical risk is

$$\widetilde{R}_{\text{emp}}(f,\beta)=\frac{1}{n}\sum_{i=1}^n|y_i^\top\theta^*-\beta-\langle\Phi(x_i),f\rangle|^2.$$

Unlike the empirical risk, the modified empirical risk is the average of iid terms. For a fixed $f\in\mathcal{H}$ and $\beta\in\mathbb{R}$, the corresponding expected risk is

$$R(f,\beta):=\mathbb{E}_{(x,y)}|y^\top\theta^*-\beta-\langle\Phi(x),f\rangle|^2.$$

Let $\widehat{f}$ be as in (9) and let $\widehat{\beta}=-\langle\overline{\Phi},\widehat{f}\rangle$. We next derive probabilistic bounds on the expected risk of $\widehat{f}$. Throughout, we use the following assumptions.

Assumption 1. *Let $\pi_{\max}=\max(\pi_1,\pi_2)$, $\pi_{\min}=\min(\pi_1,\pi_2)$. There exists a constant $C>0$ such that $\|\theta^*\|_\infty=\sqrt{\pi_{\max}/\pi_{\min}}\leq C$.*

This assumption implies that the prior group probabilities are not degenerate, that is $\pi_1\asymp\pi_2$.

Assumption 2. *There exists a constant $\kappa>0$ such that $\|\Phi(x)\|_{\mathcal{H}}\leq\kappa$ for all $x\in\mathbb{R}^p$. Equivalently, $\sup_{x\in\mathbb{R}^p}k(x,x)\leq\kappa^2$.*

Assumption 3. *The RKHS $\mathcal{H}$ is separable.*

Remark 1. *The gaussian kernel satisfies Assumption 2 with $\kappa = 1$ and satisfies Assumption 3 by Theorem 7 in [20].*

Using (9), we define the set of admissible functions f as $\mathcal{H}_\tau := \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq \tau\}$, and the set of admissible intercepts β as $I_\tau := \{\beta \in \mathbb{R} : |\beta| \leq \|\theta^*\|_\infty + \kappa\tau\}$.

Remark 2. *The intercept $\widehat{\beta} \in I_\tau$ by Assumption 2. The extra term $\|\theta^*\|_\infty$ comes from minimizing the modified empirical risk.*

Let

$$(\widetilde{f}, \widetilde{\beta}) := \operatorname*{argmin}_{f \in \mathcal{H}_\tau\,,\, \beta \in I_\tau} \widetilde{R}_{\text{emp}}(f, \beta). \tag{10}$$

be the minimizers of the modified empirical risk over the set of admissible functions and intercepts, and let

$$(f^*, \beta^*) = \operatorname*{argmin}_{f \in \mathcal{H}_\tau\,,\, \beta \in I_\tau} R(f, \beta) \tag{11}$$

be the minimizers of the expected risk over the set of admissible functions and intercepts. Our proofs rely on characterizing (i) the difference between (9) and (10), and (ii) the difference between (10) and (11). The detailed proofs are in the Supplement, and below we state the main results.

Theorem 1. *Under Assumptions 1–3, there exist constants $C_1, C_2, C_3 > 0$ such that*

$$\mathbb{P}\Big(R(\widehat{f}, \widehat{\beta}) > R(f^*, \beta^*) + \varepsilon\Big) \leq C_1 \mathcal{N}_\varepsilon \exp\Big(-\frac{C_3 n \varepsilon^2}{(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),$$

where $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\} \exp(C_2 \tau^2 \varepsilon^{-2})$.

Theorem 2. *Under Assumptions 1–3, there exist constants $C_1, C_2, C_3 > 0$ such that*

$$\mathbb{P}\Big(R(\widehat{f}, \widehat{\beta}) > R_{emp}(\widehat{f}) + \varepsilon\Big) \leq C_1 \mathcal{N}_\varepsilon \exp\Big(-\frac{C_3 n \varepsilon^2}{(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),$$

where $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\} \exp(C_2 \tau^2 \varepsilon^{-2})$.

Theorem 1 bounds the expected risk of $\widehat{f}$ compared to the best in-class expected risk, whereas Theorem 2 bounds it in terms of the empirical risk of $\widehat{f}$.

4 Sparse Kernel Optimal Scoring

The regularized KOS problem (4) performs no feature selection. All p features are used in construction of $\widehat{f}$ and the subsequent classification rule. In many applications, however, it is reasonable to expect that not all features contribute to class separation. Including such noisy features in the discriminant rule can lead to poor classification performance. Figure 1 shows an example based on simulated data with four features. Only the first two features contribute to class separation, while the third and fourth features are noise.

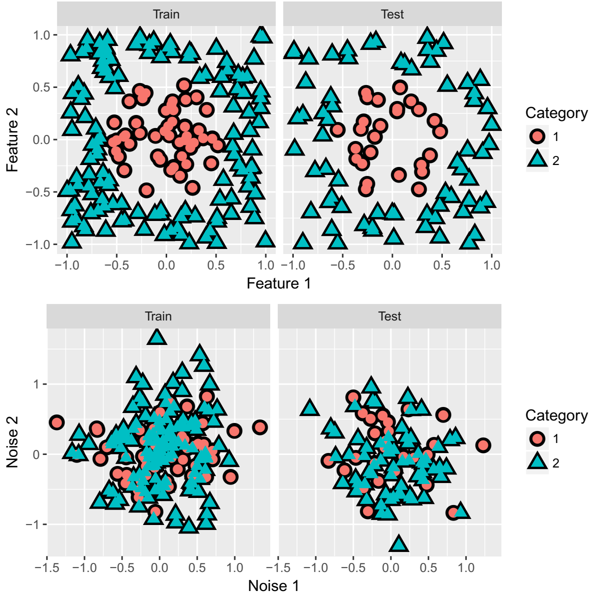

Figure 1: Simulated training and test data with four features, only features 1 and 2 contribute to class separation.

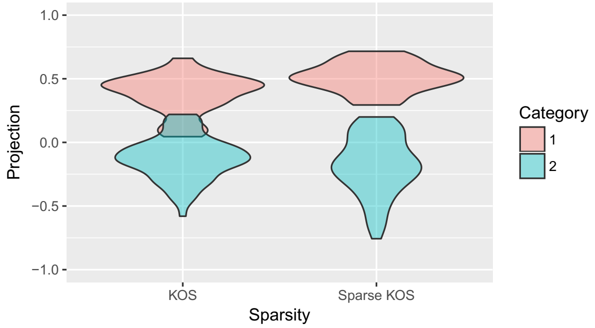

Figure 2: Comparing the projection values (6) of the test data in Figure 1 with and without sparsity.

Figure 2 shows the projected data values (6) formed by applying KOS to (i) all four features and (ii) only the first two features. The class separation is perfect based on the two “true” features, but the projected values overlap with the addition of noisy features, thus illustrating the need for feature selection within KOS.

To incorporate feature selection, we borrow the ideas from [1] and introduce a weight vector $w \in \mathbb{R}^p$, where we restrict each feature as $w_j \in [-1, 1]$. The weight vector is used to form the weighted kernel matrix $(\mathbf{K}_w)_{i,j} = k(wx_i, wx_j)$, where $wx = (w_1x_1, \dots, w_px_p)^\top$ is the Hadamard product between the weight vector w and observed feature vector x. If $w = \mathbf{1}$, $\mathbf{K}_w = \mathbf{K}$ from Section 2.3. Otherwise, w can be used to rescale features with respect to each other, and more importantly perform feature selection. If $w_j = 0$ for some feature j, then the kernel matrix $\mathbf{K}_w$ is formed without the jth feature, successfully eliminating that feature from the classification rule. The main difficulty is that the optimal weight vector w is unknown, and therefore has to be learned in addition to learning the discriminant function f.

We adjust (4) to perform joint minimization over the coefficient vector $\alpha \in \mathbb{R}^n$ and the weight vector $w \in \mathbb{R}^p$. To encourage feature selection, we add an ℓ_1-penalty on w as in [1] leading to the following minimization problem:

$$\begin{aligned}\underset{\alpha\in\mathbb{R}^n,\, w\in\mathbb{R}^p}{\text{minimize}} \; & \left\{\frac{1}{n}\|Y\widehat{\theta} - C\mathbf{K}_w C\alpha\|_2^2 + \lambda\|w\|_1 \right.\\ & \qquad \left. + \gamma\alpha^\top(C\mathbf{K}_w C + \varepsilon I)\alpha\right\} \\ \text{subject to} \quad & -1 \le w_i \le 1 \text{ for } i = 1, \dots, p.\end{aligned} \tag{12}$$

Here $\lambda \ge 0$ is the tuning parameter that controls the sparsity of the weight vector w, with larger values leading to sparser solutions. We call (12) sparse kernel optimal scoring. Given the solution pair $(\widehat{w}, \widehat{\alpha})$, we perform classification as in Section 2.4 with $\mathbf{K}_{\widehat{w}}$ being substituted for $\mathbf{K}$ and $\widehat{w}x$ substituted for x in forming the projected values $P(x)$ in (6).

Remark 3. *Unlike our restriction $w_k \in [-1, 1]$, [1] considers $w_k \in [0, 1]$. Both lead to $w_k^2 \in [0, 1]$, but we found that the latter may force all the weights to zero even when $\lambda = 0$. This behavior is avoided when the weights are allowed to be negative.*

4.1 Optimization Algorithm

In this section we describe the optimization algorithm for problem (12) given the fixed values of $\gamma, \lambda \ge 0$. Methods for parameter selection are presented in Section 5. We define the objective function in (12) as

$$\begin{aligned}Obj(w, \alpha) = \frac{1}{n}\|Y\widehat{\theta} - C\mathbf{K}_w C\alpha\|_2^2 + \lambda\|w\|_1 \\ + \gamma\alpha^\top(C\mathbf{K}_w C + \varepsilon I)\alpha.\end{aligned} \tag{13}$$

There are two challenges in solving (12): (i) non-convexity of the objective function (13) in (α, w) and (ii) non-convex mapping $w \mapsto \mathbf{K}_w$. [1] propose to overcome these challenges by (i) iterative minimization over α and w and (ii) linearization of the weighted kernel matrix $\mathbf{K}_w$ with respect to the current value of the weight vector. We adapt the algorithm from [1] to problem (12).

Given the current value of the weight vector w, we form the corresponding weighted kernel matrix $\mathbf{K}_w$ and update α according to (5) with $\mathbf{K}$ substituted with $\mathbf{K}_w$. Given the current value of the coefficient vector α, we update w by linearizing the kernel matrix. Consider the first-order Taylor approximation of $\mathbf{K}_w$ with respect to w centered at the previous value $w^{(t-1)}$ elementwise:

$$\begin{aligned}&\widetilde{\mathbf{K}}_w(x_i, x_j) := \\ &\mathbf{K}_{w^{(t-1)}}(x_i, x_j) + \{\nabla_w \mathbf{K}_{w^{(t-1)}}(x_i, x_j)\}^\top (w - w^{(t-1)}),\end{aligned}$$

where $\nabla_w \mathbf{K}_{w^{(t-1)}}(x_i, x_j) \in \mathbb{R}^p$ is the gradient of $k(wx_i, wx_j)$ with respect to w evaluated at $w^{(t-1)}$. We substitute $\widetilde{\mathbf{K}}_w$ in place of $\mathbf{K}_w$ within (12). Let $T \in \mathbb{R}^{n\times p}$ be

$$T := \begin{pmatrix} \sum_{\ell=1}^n (C\alpha)_\ell \nabla_w \mathbf{K}_{w^{(t-1)}}(x_1, x_\ell)^\top \\ \vdots \\ \sum_{\ell=1}^n (C\alpha)_\ell \nabla_w \mathbf{K}_{w^{(t-1)}}(x_n, x_\ell)^\top \end{pmatrix}.$$

For fixed α, the minimization problem (12) with respect to w can be written as

$$\begin{aligned}\underset{w}{\text{minimize}} & \left\{\frac{1}{2}w^\top Q w - \beta^\top w + \frac{\lambda}{2}\|w\|_1\right\} \\ \text{subject to} & \; -1 \le w_i \le 1 \text{ for } i = 1, \dots, p;\end{aligned} \tag{14}$$

where

$$\begin{aligned}Q &= \frac{1}{n}(CT)^\top CT \in \mathbb{R}^{p\times p}, \\ \beta &= \frac{1}{n}T^\top C[Y\widehat{\theta} - C\mathbf{K}_{w^{(t-1)}} C\alpha + CTw^{(t-1)}] \\ &\qquad - 2^{-1}\gamma T^\top C\alpha \in \mathbb{R}^p.\end{aligned} \tag{15}$$

Problem (14) is of the same form as the penalized lasso problem [18, Chapter 5] with extra convex constraints on w. Therefore, we can use the coordinate-descent algorithm to solve (14).

Consider optimizing (14) with respect to w_k. From the KKT conditions [5], the solution must satisfy

$$\widehat{w}_k = \text{sign}(\widetilde{w}_k)\min(|\widetilde{w}_k|, 1), \tag{16}$$

where

$$\widetilde{w}_k := \frac{1}{Q_{kk}} S_{\lambda/2}\Big(\beta_k - \sum_{i\neq k} Q_{ki} w_i\Big),$$

and $S_{\lambda/2}(x) := \text{sign}(x)\max\{|x| - \lambda/2,\, 0\}$ is the soft-thresholding function. The coordinate-descent algorithm proceeds by applying the update (16) on each feature k until convergence.

Algorithm 1: Sparse Kernel Optimal Scoring

```
Input  : $X \in \mathbb{R}^{n \times p}$, $Y \in \mathbb{R}^{n \times 2}$, $\widehat{\theta}$, $\sigma > 0$, $\gamma > 0$,
         $\lambda \geq 0$ , convergence threshold $\varepsilon_{\text{con}}$
Output: Discriminant coefficients $\widehat{\alpha}$ and feature
         weights $\widehat{w}$.
$t \leftarrow 0$
$w^{(0)} \leftarrow \mathbf{1}$
$(\mathbf{K}_{w^{(0)}})_{i,j} \leftarrow k(w^0 x_i, w^0 x_j)$, $\mathbf{K}_{w^{(0)}} \leftarrow \{(\mathbf{K}_{w_0})_{i,j}\}$
repeat
    $t \leftarrow t + 1$
    Update $\alpha^{(t)}$ according to (5) with $\mathbf{K} = \mathbf{K}_{w^{(t-1)}}$
    Update $w^{(t)}$ using coordinate descent with
     updates according to (16)
    $(\mathbf{K}_{w^{(t)}})_{i,j} \leftarrow k(w^{(t)} x_i, w^{(t)} x_j)$
until $Obj(\alpha^{(t)}, w^{(t)}) - Obj(\alpha^{(t-1)}, w^{(t-1)}) < \varepsilon_{con}$
return $\widehat{\alpha} = \alpha^{(t)}$, $\widehat{w} = w^{(t)}$
```

The full algorithm for (12) is summarized as Algorithm 1. While the update of w is based on approximation of objective function (13), in our experience the objective function is always decreasing at each iteration. In case of convergence issues, one can use a line search along a descent direction of w [1]. We refer to [1] for further discussion of algorithmic convergence.

5 Parameter Selection

This section describes the selection of the kernel parameter (tailored to the gaussian kernel parameter σ^2), ridge parameter γ, and sparsity parameter λ.

5.1 Gaussian Kernel Parameter Selection

We propose to use 5-fold cross-validation to minimize the error rate. To reduce computational cost, we only consider five tuning parameters based on the $\{.05, .1, .2, .3, .5\}$ quantiles of the set of squared distances between the classes

$$\{\|x_{i_1} - x_{i_2}\|_2^2 \, : x_{i_1} \in C_1, \, x_{i_2} \in C_2\}.$$

This approach is similar to the one used in the R package kernlab [21], which takes values between .1 and .9 quantiles of the distance statistic $\|x - x'\|_2$ between distinct data points taken from a random subset of the full data. [7] and [21] state that good performance can be achieved with any value of σ in this range. Our approach is different in that (i) we select one value based on CV, (ii) only look at the distances between classes, and (iii) only consider lower quantiles. We find that this yields good predictive accuracy, and we conjecture that the reason is the presence of noise features, which inflate the distance values $\|x_{i_1} - x_{i_2}\|_2$. This is supported by empirical observation that the quantiles based on the full set of features will exceed the corresponding quantiles based on the reduced set of informative features.

5.2 Ridge parameter selection

Due to the computational expense of cross-validation, we propose an alternative approach for ridge parameter selection based on the shrinkage of kernel matrix. [24] proposes to stabilize the kernel matrix via shrinkage towards a target matrix and derives an optimal value for the shrinkage parameter. Following [24], in KOS we want to stabilize $(C\mathbf{K}_w C)^2$ with the target matrix $C\mathbf{K}_w C + \varepsilon I$, and therefore consider

$$(C\mathbf{K}_w C)^2 + \gamma(C\mathbf{K}_w C + \varepsilon I)$$

for $\gamma > 0$. Let $t = \gamma/(1+\gamma)$, then the optimal value of t is $\widehat{t} = \min(\max(0, \widetilde{t}), 1)$, where

$$\widetilde{t} := \frac{n}{(n-2)}\left(\frac{\|\text{diag}(C\mathbf{K}C)\|_F^2 - \frac{1}{n}\|C\mathbf{K}C\|_F^2}{\|C\mathbf{K}C\|_F^2}\right).$$

Solving back for γ gives the ridge penalty $\widehat{\gamma} = \widehat{t}/(1-\widehat{t})$. We call this approach Stabilization.

Generalized cross-validation (GCV) [11, 36, 16] is another common method for selection of ridge parameter, however we found that it performs poorly compared to proposed Stabilization method. Figure 3 compares the selected ridge parameters as well as corresponding error rates for two methods. We generate 100 training and testing datasets following the model in Section 6.1. Each time we consider five possible kernel parameters σ^2 based on the distance quantiles as in Section 5.1. We then select ridge parameters by either GCV or proposed stabilization method, and choose the best sparsity parameter for each as in Section 5.3. We find that GCV consistently selects smaller value for the ridge parameter than our approach leading to higher error rates. We conjecture that surprisingly poor performance of GCV is due to the presence of noise variables, although we do not have the formal justification.

5.3 Sparsity parameter selection

We select λ using 5-fold cross-validation (CV) to minimize the error rate over a grid of 20 equally-spaced values in $[10^{-10}\lambda_{\max}, \lambda_{\max}]$. We set $\lambda_{\max} = 2\|\beta\|_\infty$, where β is as in (15), since the solution $\widehat{w}$ to (14) is zero if $\lambda \geq \lambda_{\max}$ (see Lemma 1 in the Supplement).

6 Empirical studies

We compare the performance of the following methods: (i) sparse kernel optimal scoring (Sparse KOS); (ii)

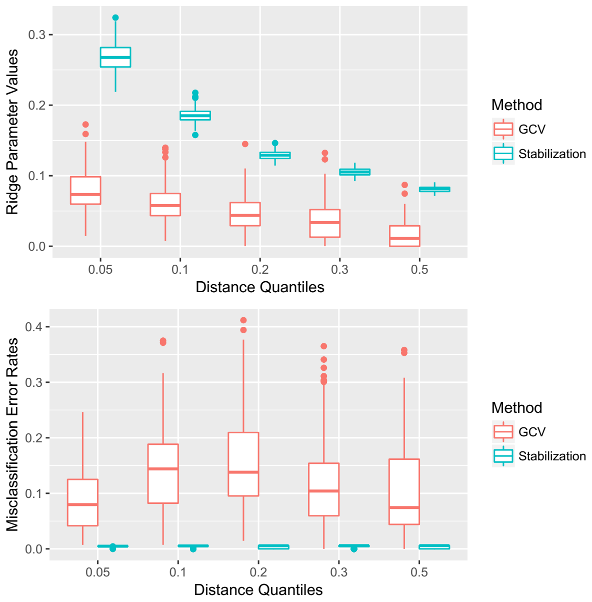

Figure 3: Comparison between generalized cross-validation (GCV) and proposed Stabilization method for selection of ridge parameter γ over 100 replications. **Top:** Selected values of γ; **Bottom:** Misclassification error rates.

kernel optimal scoring (KOS); (iii) random forests; (iv) kernel support vector machines (kernel SVM); (v) neural networks; (vi) K-nearest neighbors (KNN); and (vii) sparse linear discriminant analysis (sparse LDA).

We implement sparse KOS using the gaussian kernel with parameters selected as in Section 5, KOS is implemented by setting $\lambda = 0$ and $w = \mathbf{1}$. We use the R package randomForest [27] to create a classifier with 50 decision trees. We use the R package kernlab [21] for kernel SVM using the gaussian kernel with parameter selected as in Section 5.1. We use keras [9] to implement a neural network with the ReLU activation function, 50 units, 100 epochs, and the default batch size. We use class [35] for KNN with $K = 5$. We use the R package MGSDA [15] for sparse LDA.

6.1 Simulated model 1

We generate data as in Figure 1 with $p = 4$ features (x_1, x_2, x_3, x_4). The first two features satisfy $\sqrt{x_{i1}^2 + x_{i2}^2} \geq 2/3$ if the ith sample is in class 1, and $\sqrt{x_{i1}^2 + x_{i2}^2} \leq 2/3 - 1/10$ if the ith sample is in class 2. We generate 300 samples with each feature from the uniform distribution on $[-1, 1]$ and only leave samples that satisfy one of the class requirements ($n \approx 270$). The remaining two features are generated as independent gaussian noise variables, $x_{ij} \sim \mathcal{N}(0, 2^{-1})$ for $j = 3, 4$ and all samples i. We use 2/3 of the samples for training, and 1/3 for testing, maintaining the class proportions. We repeat the data generation process and the split 100 times, the misclassification error rates over test datasets are presented in Figure 4.

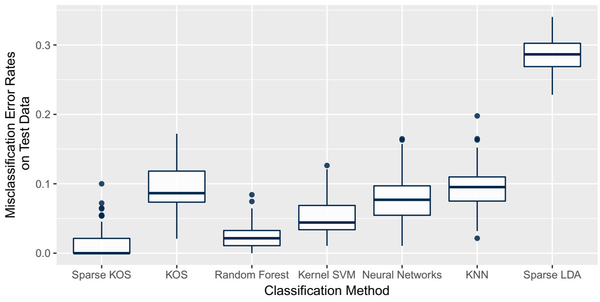

Figure 4: Misclassification error rates based on 100 replications of simulated model 1.

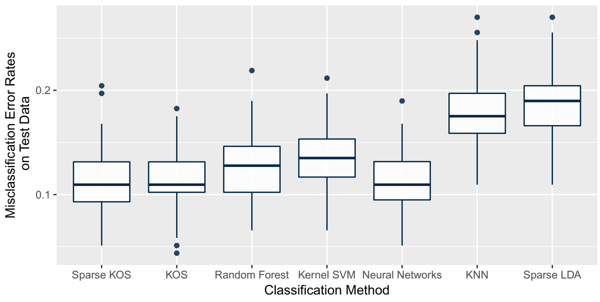

Figure 5: Misclassification error rates based on 100 replications of simulated model 2.

Sparse KOS performs the best out of all classifiers with random forest being second-best. Sparse LDA performs the worst, likely due to non-linear optimal classification boundary. Sparse KOS has excellent feature selection in this study- giving nonzero weight to the first two features in all 100 splits while giving $\widehat{w}_j = 1$ for $j = 1, 2$ in 98 out of 100 replications and $\widehat{w}_j = 0$ for $j = 3, 4$ in 99 out of 100 replications.

6.2 Simulated model 2

We generate data with $p = 10$ features and $n = 400$ samples such that $x_{i3} + \sin(x_{i4} + x_{i1}) < (x_{i2})^2$ if sample i belongs to class 1, and $x_{i3} + \sin(x_{i4} + x_{i1}) \geq (x_{i2})^2$ if sample i belongs to class 2. We use the uniform distribution on $[-1, 1]$ for each x_{ij}, so that the last 6 features are uniform noise. As with the previous example, we use 2/3 of the samples for training, and 1/3 for testing, where the split is performed to maintain the class proportions. We repeat the data generation process and the split 100 times. The misclassification error rates over test datasets are presented in Figure 5.

The lowest misclassification error rates are achieved

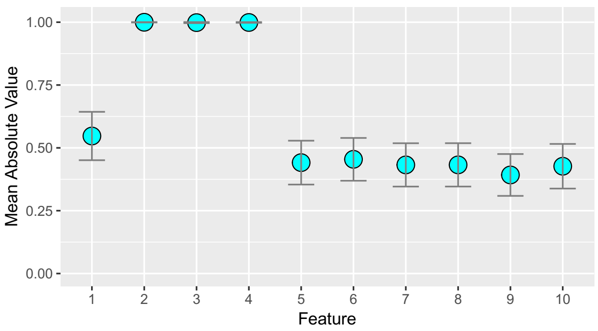

Figure 6: The mean absolute values of weights $|w_j|$ for each feature across 100 replications of simulated model 2. The bars represent ±2 standard errors.

Dataset	Features size	Sample size
Blood donation [38]	$p = 4$	$n = 748$
Climate model failure [28]	$p = 18$	$n = 540$
Credit card default [37]	$p = 24$	$n = 3,000$

Table 1: Description of benchmark datasets

by sparse KOS, KOS, and neural network classifiers. Sparse KOS behaves similarly to KOS because sparse KOS is unable to consistently select true features. Nevertheless, it gives higher weight values to true features as displayed in Figure 6. As with the previous example, sparse LDA performs the worst.

6.3 Benchmark datasets

We consider three datasets, summarized in Table 1, which are publicly available from the UCI Machine Learning Repository. We randomly split each dataset 100 times preserving the class proportions and use 2/3 for training and 1/3 for testing. We do not present the error rates for sparse LDA due to its poor performance on these datasets (it classifies every point to the largest of two groups), the misclassification error rates for all other methods are in Table 2.

In the blood donation study [38], the goal is to determine if a person will donate blood given four features: Recency (months since last donation), Frequency (total number of donations), Monetary (total blood donated in cubic centimetres), and Time since first donation. Sparse KOS consistently gives large weights ($|w_j| > 0.9$) to every feature but Frequency. The latter gets large weight in only 50% of splits. Sparse KOS performs similarly to KOS, and we conjecture this is because all features are important for classification.

In the climate model study [28], the goal is to predict if a climate simulation will crash based on 18 initial parameter values. Sparse KOS consistently selects 4 out of 18: features 1, 2 (variable viscosity parameters), feature 13 (tracer and momentum mixing coefficient), and feature 14 (base background vertical diffusivity). Sparse KOS has the best classification performance, which is likely due to feature selection.

	Blood Donation	Climate Model	Credit Default
Sparse KOS	**22.1** (0.18)	**4.9** (0.13)	**18.2** (0.06)
KOS	**22.2** (0.20)	5.4 (0.12)	19.1 (0.08)
Random Forest	24.3 (0.18)	8.2 (0.06)	19.1 (0.08)
Kernel SVM	22.4 (0.12)	8.7 (0.00)	20.0 (0.08)
Neural Network	23.9 (0.04)	5.4 (0.15)	21.7 (0.04)
KNN	23.5 (0.20)	7.6 (0.08)	20.8 (0.08)

Table 2: Mean misclassification errors (%) over 100 random splits, standard errors are in brackets.

The credit card data [37] has 30,000 data points, but we restrict to $n = 3,000$ for computational simplicity. The goal is to predict the default of a customer on credit payments based on 24 features. Sparse KOS has the best classification performance, followed by KOS and random forests. Sparse KOS always selects feature 6 (the repayment status in September, 2005, the latest monthly payment recorded) and rarely selects other features. The most recent payment history is strongly indicative of credit default.

7 Discussion

We propose a kernel discriminant classifier with sparse feature selection, called sparse kernel optimal scoring, which is implemented in the R package `sparseKOS` [26]. An advantage of sparsity is that it can improve classification performance (see Section 6) and lead to more interpretable classification rules. The nonzero weights produced by sparse KOS can be used to judge the importance of features. While we have focused the discussion on the case of two classes, the method can be generalized to multiple classes using optimal scoring formulation in [14].

Sparse KOS requires the construction of a $n \times n$ kernel matrix $\mathbf{K}$ and is therefore computationally prohibitive for large n cases. Future research could investigate the appropriate low-dimensional approximations of $\mathbf{K}$ within the kernel optimal scoring framework.

Acknowledgements: This work was supported in part by NSF-DMS 1712943.

References

[1] Allen, G. I. (2013). Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2):284–299.

[2] Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(Jun):1179–1225.

[3] Bach, F. R., Lanckriet, G. R., and Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM.

[4] Baudat, G. and Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404.

[5] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[6] Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106(496):1566–1577.

[7] Caputo, B., Sim, K., Furesjo, F., and Smola, A. (2002). Appearance-based object recognition using svms: which kernel should i use? In *Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, Whistler*, volume 2002.

[8] Chen, J., Zhang, C., Kosorok, M. R., and Liu, Y. (2017). Double sparsity kernel learning with automatic variable selection and data extraction. *arXiv preprint arXiv:1706.01426*.

[9] Chollet, F. et al. (2015). Keras. `https://keras.io`.

[10] Clemmensen, L., Witten, D. M., Hastie, T., and Ersbøll, B. (2011). Sparse Discriminant Analysis. *Technometrics*, 53(4):406–413.

[11] Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

[12] Diethe, T., Hussain, Z., Hardoon, D., and Shawe-Taylor, J. (2009). Matching pursuit kernel fisher discriminant analysis. In *Artificial Intelligence and Statistics*, pages 121–128.

[13] Friedman, J., Hastie, T., and Tibshirani, R. (2009). *The Elements of statistical learning*. Springer Series in Statistics New York, 2 edition.

[14] Gaynanova, I. (2018). Prediction and estimation consistency of sparse multi-class penalized optimal scoring. *arXiv preprint arXiv:1809.04669*.

[15] Gaynanova, I., Booth, J. G., and Wells, M. T. (2016). Simultaneous sparse estimation of canonical vectors in the $p >> N$ setting. *Journal of the American Statistical Association*, 111:696–706.

[16] Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.

[17] Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, pages 73–102.

[18] Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

[19] Hastie, T., Tibshirani, R. J., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270.

[20] Hein, M. and Bousquet, O. (2004). Kernels, associated structures and generalizations. *Max-Planck-Institut fuer biologische Kybernetik, Technical Report*.

[21] Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). Kernlab-an s4 package for kernel methods in r. *Journal of Statistical Software*, 11(9):1–20.

[22] Kim, S.-J., Magnani, A., and Boyd, S. (2006). Robust fisher discriminant analysis. In *Advances in Neural Information Processing Systems*, pages 659–666.

[23] Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502.

[24] Lancewicki, T. (2017). Regularization of the kernel matrix via covariance matrix shrinkage estimation. *arXiv preprint arXiv:1707.06156*.

[25] Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., and Jordan, M. I. (2002). A robust minimax approach to classification. *Journal of Machine Learning Research*, 3(Dec):555–582.

[26] Lapanowski, A. F. and Gaynanova, I. (2018). *sparseKOS: An R package for Sparse Kernel Optimal Scoring*. `https://github.com/aflapan/sparseKOS`.

[27] Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

[28] Lucas, D., Klein, R., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D., and Zhang, Y. (2013). Failure analysis of parameter-induced simulation

crashes in climate models. *Geoscientific Model Development*, 6(4):1157–1171.

[29] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE.

[30] Nosedal-Sanchez, A., Storlie, C. B., Lee, T. C., and Christensen, R. (2012). Reproducing kernel hilbert spaces for penalized regression: A tutorial. *The American Statistician*, 66(1):50–60.

[31] Roth, V. and Steinhage, V. (2000). Nonlinear discriminant analysis using kernel functions. In *Advances in Neural Information Processing Systems*, pages 568–574.

[32] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

[33] Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7(Jul):1531–1565.

[34] Sun, S., Kolar, M., and Xu, J. (2015). Learning structured densities via infinite dimensional exponential families. In *Advances in Neural Information Processing Systems*, pages 2287–2295.

[35] Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.

[36] Xiang, D. and Wahba, G. (1996). A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistica Sinica*, pages 675–692.

[37] Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

[38] Yeh, I.-C., Yang, K.-J., and Ting, T.-M. (2009). Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with Applications*, 36(3):5866–5871.