NON-LINEAR AND SPARSE DISCRIMINANT ANALYSIS WITH DATA COMPRESSION

A Thesis

by

ALEXANDER FRANK LAPANOWSKI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Irina Gaynanova
Committee Members,	Anirban Bhattacharya
	Debdeep Pati
	Simon Foucart
Head of Department,	Daren Cline

August 2020

Major Subject: Statistics

ABSTRACT

Large-sample data became prevalent as data acquisition became cheaper and easier. While a large sample size has theoretical advantages for many statistical methods, it presents computational challenges either in the form of a large number of features or a large number of training samples. We consider the two-group classification problem and adapt Linear Discriminant Analysis to the problems above. Linear Discriminant Analysis is a linear classifier and will under-fit when the true decision boundary is non-linear.

To address non-linearity and sparse feature selection, we propose a kernel classifier based on the optimal scoring framework which trains a non-linear classifier. 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. We also propose automatic methods for ridge parameter selection and guassian kernel parameter selection.

To address the computational challenges of a large sample size, we adapt compression to the classification setting. Sketching, or compression, is a well-studied approach to address sample reduction in regression settings, but considerably less is known about its performance in classification settings. Here we consider the computational issues due to large sample size within the discriminant analysis framework. We propose a new compression approach for reducing the number of training samples for linear and quadratic discriminant analysis, in contrast to existing compression methods which focus on reducing the number of features. We support our approach with a theoretical bound on the misclassification error rate compared to the Bayes classifier. Empirical studies confirm the significant computational gains of the proposed method and its superior predictive ability compared to random sub-sampling.

DEDICATION

To Mom, whose love and support made this possible.

ACKNOWLEDGMENTS

I offer my love and thanks to my Mom, whose love and care are everything to me.

To Taylor, Shahina, and all my friends at Texas A&M who have offered friendship and endless shared memories.

To my teachers for believing in me and encouraging me along the way. In particular, I want to thank Mr. David Kirck, Brother Xavier, Mr. Brian Bacon, Mr. Thaier Mukhtar, Dr. Mitya Boyarchenko, Dr. Roman Vershynin, Dr. Yaniv Plan, Dr. Mark Rudelson, and Dr. Alexei Poltoratski.

Finally, to Dr. Irina Gaynanova, whose commitment, thoughtfulness, and statistical insight made for a wonderful advisor. I have grown in every facet as a researcher under her mentoring.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Irina Gaynanova, Professor Anirban Bhattacharya, and Professor Debdeep Pati all from the Department of Statistics, and Professor Simon Foucart of the Department of Mathematics.

All work conducted for the thesis (or) dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a graduate fellowship from Texas A&M University.

NOMENCLATURE

$\mathbb{R}^p$	p-dimensional Euclidean Space
$\|\cdot\|_2$	Euclidean norm on $\mathbb{R}^p$
$\|\cdot\|_\infty$	Supremum norm on $\mathbb{R}^p$
$\|\cdot\|_F$	Frobenius norm on the set of matrices with fixed dimensions
$\mathbf{x}_i$	p-dimensional feature vector
y_i	Class label corresponding to $\mathbf{x}_i$
$(\mathbf{x}_i, y_i)$	Training sample-label pair
X	$n \times p$ matrix of training samples
Y	Vector of training labels corresponding to X
$\mathcal{H}$	Reproducing Kernel Hilbert Space
k	Reproducing kernel associated with $\mathcal{H}$
Φ	Map from p-dimensional Euclidean Space to $\mathcal{H}$
$\langle \cdot\,,\,\cdot \rangle_{\mathcal{H}}$	Inner product in $\mathcal{H}$
$\|\cdot\|_{\mathcal{H}}$	Norm in $\mathcal{H}$ induced by $\langle \cdot\,,\,\cdot \rangle_{\mathcal{H}}$
Q^g	$m_g \times n_g$ matrix corresponding to group $g = 1, 2$
$\overline{X}_g$	g-th group sample mean $n_g^{-1} \sum \mathbf{x}_i^g$

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Problem Statement

Linear Discriminant Analysis (LDA) [2, Chapter 11] is a popular classification technique which seeks to separate classes of training data with hyper-planes. However, it has several drawbacks: (i) it will under-fit the data when the true decision boundaries between classes are non-linear; (ii) it uses all p features in the decision rule, and consequently over-fits in the high-dimensional setting; and (iii) it is computationally expensive when the training data has a large number of samples and medium-sized number of features.

This dissertation addresses (i)-(iii) by proposing several variants of LDA for the two-class setting. In particular, we propose a kernel discriminant classifier based on the optimal scoring framework which has simultaneous sparse feature selection. We also propose a novel sample-reduction technique based on compression within LDA and provide the theoretical framework for adapting compression to kernel discriminant analysis. Lastly, we include an R package vignette which instructs researchers on using the package biClassify. This package implements all of the proposed methods.

1.2 Review of Linear Discriminant Analysis

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be independent pairs of feature vectors $\mathbf{x}_i \in \mathbb{R}^p$ and labels $y_i \in \{1, 2\}$. Let $X = \begin{pmatrix} X^{1\top} & X^{2\top} \end{pmatrix}^\top$ be the corresponding $n \times p$ matrix of training samples, where $X^g \in \mathbb{R}^{n_g \times p}$ is the sub-matrix consisting of n_g samples $\mathbf{x}_i^g$ belonging to class $g = 1, 2$. Let $Y = (\{1\}^{n_1}, \{2\}^{n_2})^\top$ be the corresponding vector of class labels. We let $\overline{X} := n^{-1} \sum_{i=1}^n \mathbf{x}_i$ be the overall training sample mean, and let $\overline{X}_g$ be the gth class sample mean $n_g^{-1} \sum_{i=1}^{n_g} \mathbf{x}_i^g$.

Assumption 1. *Conditional on group membership, the training samples $\mathbf{x}_i$ are i.i.d. normal random vectors $N(\mu_g, \Sigma_w)$ with group mean $\mu_g \in \mathbb{R}^p$ and covariance matrix $\Sigma_w \in \mathbb{R}^{p \times p}$ such that $\mu_1 \neq \mu_2$.*

Assumption 1 states that the group distribution means are different but that the group covariances are equal.

There are several equivalent variants of the two-class LDA problem which are presented below.

1.2.1 Fisher Discriminant Analysis

Fisher Discriminant Analysis (FDA) [2, Section 11.5] seeks a vector $\beta \in \mathbb{R}^p$ such that the values $\beta^\top \mathbf{x}_i^g$ are well-separated between classes.

Given the within-class covariance matrix and between-class covariance matrices

$$\widehat{\Sigma}_w := \frac{1}{n}\sum_{g=1}^{2}\sum_{i=1}^{n_g}(\mathbf{x}_i^g - \overline{X}_g)(\mathbf{x}_i^g - \overline{X}_g)^\top \text{ and } \widehat{\Sigma}_b = \sum_{g=1}^{2}\frac{n_g}{n}(\overline{X}_g - \overline{X})(\overline{X}_g - \overline{X})^\top, \tag{1.1}$$

the *Fisher Discriminant Ratio* is defined as

$$\frac{\beta^\top \widehat{\Sigma}_b \beta}{\beta^\top \widehat{\Sigma}_w \beta}. \tag{1.2}$$

FDA seeks that vector $\widehat{\beta} \in \mathbb{R}^p$ which maximizes (1.2). One can solve for the discriminant vector by solving

$$\begin{gathered}\underset{\beta \in \mathbb{R}^p}{\text{maximize}}\, \beta^\top \widehat{\Sigma}_b \beta \\ \text{subject to } \beta^\top \widehat{\Sigma}_w \beta = 1.\end{gathered}$$

Let $d \in \mathbb{R}^p$ be the vector of the class mean differences

$$d := \frac{\sqrt{n_1 n_2}}{n}(\overline{X}_1 - \overline{X}_2), \tag{1.3}$$

then FDA estimates β as $\widehat{\beta} := \widehat{\Sigma}_w^{-1} d$.

Theorem 1 (Theorem 11.5.1 of [2])**.** *The vector $\widehat{\beta}$ in Fisher's linear discriminant function is the eigenvector of $\widehat{\Sigma}_w^{-1}\widehat{\Sigma}_b$ corresponding to the largest eigenvalue.*

Given the estimated discriminant vector $\widehat{\beta} \in \mathbb{R}^p$, the FDA classification rule labels a new

$\mathbf{x} \in \mathbb{R}^p$ by minimizing the Mahalanobis distance to the group centers

$$\operatorname*{argmin}_{g=1,2} \big\{(\mathbf{x} - \overline{X}_g)^\top \widehat{\beta}\, (\widehat{\beta}^\top \widehat{\Sigma}_w \widehat{\beta})^{-1} \widehat{\beta}^\top (\mathbf{x} - \overline{X}_g) - 2\log(n_g/n)\big\}. \tag{1.4}$$

1.2.2 Linear Discriminant Analysis Using Discriminant Functions

An equivalent formulation of LDA maximizes the likelihood ratio of the group distributions with plug-in estimates for the parameters. Let π_g be the prior group probabilities of sampling from class $g = 1, 2$. The likelihood function for group g is

$$\mathcal{L}(\mu_g, \Sigma_w|\mathbf{x}) = \frac{1}{(2\pi|\Sigma_w|)^{p/2}} \exp\Bigg(- \frac{(\mathbf{x} - \mu_g)^\top \Sigma_w^{-1} (\mathbf{x} - \mu_g)}{2}\Bigg)\, \pi_g.$$

Classify a sample $\mathbf{x}$ as belonging to group 1 if and only if $L(\mu_1, \Sigma_w|\mathbf{x})\pi_1 \geq L(\mu_2, \Sigma_w|\mathbf{x})\pi_2$. Equivalently, consider the ratio of likelihood functions

$$\begin{aligned}
\frac{\mathcal{L}(\mathbf{x}; \mu_1, \Sigma_w)}{\mathcal{L}(\mathbf{x}; \mu_2, \Sigma_w)} &= \exp\Bigg(- \frac{1}{2}(\mathbf{x} - \mu_1)^\top \Sigma_w^{-1}(\mathbf{x} - \mu_1) + \frac{1}{2}(\mathbf{x} - \mu_2)^\top \Sigma_w^{-1}(\mathbf{x} - \mu_2)\Bigg) \frac{\pi_1}{\pi_2} \\
&= \exp\Bigg(- \frac{1}{2}\Big[\{-2\mathbf{x}^\top \Sigma_w^{-1}\mu_1 + \mu_1^\top \Sigma_w^{-1}\mu_1\} - \{-2\mathbf{x}^\top \Sigma_w^{-1}\mu_2 + \mu_2^\top \Sigma_w^{-1}\mu_2\}\Big]\Bigg) \frac{\pi_1}{\pi_2} \\
&= \exp\Bigg(- \frac{1}{2}[\mathbf{x}^\top \Sigma_2^{-1}(\mu_2 - \mu_1) + \mu_1^\top \Sigma_w^{-1}\mu_1 - \mu_2^\top \Sigma_w^{-1}\mu_2]\Bigg) \frac{\pi_1}{\pi_2}.
\end{aligned}$$

The *Bayes Rule* classifies $\mathbf{x}$ to class 1 if and only if the likelihood ration is greater than or equal to 1. Taking the logarithm of the likelihood ratio gives the equivalent rule of labelling $\mathbf{x}$ as belonging to class 1 if and only if

$$\frac{1}{2}\mathbf{x}^\top \Sigma_w^{-1}(\mu_1 - \mu_2) \geq \mu_1^\top \Sigma_w^{-1}\mu_1 - \mu_2^\top \Sigma_w^{-1}\mu_2 - \log(\pi_1/\pi_2).$$

In practice, the population parameters μ_g, Σ_w, and π_g are replaced by their sample estimates

$\overline{X}_g, \widehat{\Sigma}_w$, and n_g/n, yielding

$$\frac{1}{2}\mathbf{x}^\top \widehat{\Sigma}_w^{-1}(\overline{X}_1 - \overline{X}_2) \geq \overline{X}_1^\top \widehat{\Sigma}_w^{-1}\overline{X}_1 - \overline{X}_2^\top \widehat{\Sigma}_w^{-1}\overline{X}_2 - \log(n_1/n_2). \tag{1.5}$$

Equation (1.5) can be expressed in terms of discriminant functions

$$\delta_g(\mathbf{x}) = \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\overline{X}_g - \frac{1}{2}\overline{X}_g \widehat{\Sigma}_w^{-1}\overline{X}_g + \log(n_g).$$

That is, (1.5) is equivalent to the rule which maximizes the discriminant functions

$$\underset{g=1,2}{\text{maximize}}\, \delta_g(\mathbf{x}). \tag{1.6}$$

1.2.3 Equivalence of the Fisher Discriminant Analysis and Discriminant Function Decision Rules

This section proves that the Fisher Discriminant Rule (1.4) and the discriminant function decision rule (1.6) are equivalent.

We first show that maximizing the discriminiant functions δ_g is equivalent to minimizing the Mahalanobis Distance.

Since $\mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x}$ is not class-dependent, we may add $\mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x}$ to both $\delta_g(\mathbf{x})$ $(g = 1, 2)$ and preserve the classification rule. This gives

$$\delta_2(\mathbf{x}) + \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x} \geq \delta_1(\mathbf{x}) + \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x},$$

where

$$\begin{aligned} \delta_g(\mathbf{x}) + \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x} &= \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\overline{X}_g - \frac{1}{2}\overline{X}_g^\top \widehat{\Sigma}_w^{-1}\overline{X}_g + \log(\pi_g) + \mathbf{x}^\top \widehat{\Sigma}_w^{-1}\mathbf{x} \\ &= -\frac{1}{2}(\mathbf{x} - \overline{X}_g)^\top \widehat{\Sigma}_w^{-1}(\mathbf{x} - \overline{X}_g) + \log(n_g/n). \end{aligned} \tag{1.7}$$

Thus, maximizing the discriminant functions is equivalent to minimizing the Mahalanobis distance penalized according to class proportion.

The Mahalanobis distance is the squared Euclidean distance of the data normalized by the sample within-group covariance

$$(\mathbf{x}-\overline{X}_g)^\top\widehat{\Sigma}_w^{-1}(\mathbf{x}-\overline{X}_g)=\|\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\|^2.$$

In computing the distance of a sample $\mathbf{x}$ to the class means, we may consider only the distance of the projection onto subspace spanned by the difference of group means. Let $P_{\widehat{\Sigma}_w^{-1/2}d}$ be the orthogonal projection onto the span of $\widehat{\Sigma}_w^{-1/2}d$. This projection is

$$P_{\widehat{\Sigma}_w^{-1/2}d}=\widehat{\Sigma}_w^{-1/2}d\,[(\widehat{\Sigma}_w^{-1/2}d)^\top(\widehat{\Sigma}_w^{-1/2}d)]^{-1}\,(\widehat{\Sigma}_w^{-1/2}d)^\top=\frac{\widehat{\Sigma}_w^{-1/2}(d\,d^\top)\widehat{\Sigma}_w^{-1/2}}{d^\top\widehat{\Sigma}_w^{-1}d}$$

The Pythagorean decomposition gives

$$\|\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\|^2=\|P_{\widehat{\Sigma}_w^{-1/2}d}\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\|^2+\|(I-P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\|^2.$$

For $\beta:=\widehat{\Sigma}_w^{-1}d$,

$$\begin{aligned}\|P_{\widehat{\Sigma}_w^{-1/2}d}\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\|^2&=\left\|\frac{\widehat{\Sigma}_w^{-1/2}(d\,d^\top)\widehat{\Sigma}_w^{-1/2}}{d^\top\widehat{\Sigma}_w^{-1}d}\widehat{\Sigma}_w^{-1/2}(\mathbf{x}-\overline{X}_g)\right\|^2\\&=(d^\top\widehat{\Sigma}_w^{-1}d)^{-2}\|\widehat{\Sigma}_w^{-1/2}(d\,d^\top)\widehat{\Sigma}_w^{-1}(\mathbf{x}-\overline{X}_g)\|^2\\&=(d^\top\widehat{\Sigma}_w^{-1}d)^{-2}\|\widehat{\Sigma}_w^{-1/2}d\,\beta^\top(\mathbf{x}-\overline{X}_g)\|^2\\&=\frac{(\mathbf{x}-\overline{X}_g)^\top\beta d^\top\widehat{\Sigma}_w^{-1/2}\widehat{\Sigma}_w^{-1/2}d\,\beta^\top(\mathbf{x}-\overline{X}_g)}{(d^\top\widehat{\Sigma}_w^{-1}d)^2}\\&=(\mathbf{x}-\overline{X}_g)^\top\beta(d^\top\widehat{\Sigma}_w^{-1}d)^{-1}\,\beta^\top(\mathbf{x}-\overline{X}_g)\\&=(\mathbf{x}-\overline{X}_g)^\top\beta(\beta^\top\widehat{\Sigma}_w\beta)^{-1}\,\beta^\top(\mathbf{x}-\overline{X}_g).\end{aligned}$$

Thus, the projected distance of the sample $\mathbf{x}$ to the class sample mean projected onto the difference of group means vector equals the Mahalanobis distance of the sample projected onto the discriminant vector β.

We now prove that $\|(I - P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}(\mathbf{x} - \overline{X}_g)\|^2$ does not depend on the class g. The projection $(I - P_{\widehat{\Sigma}_w^{-1/2}d})$ collapses the sphered difference in group means $\widehat{\Sigma}_w^{-1/2}d$ into the zero vector. Hence, consider the difference

$$\begin{aligned}&(I - P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}(\mathbf{x} - \overline{X}_1) - (I - P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}(\mathbf{x} - \overline{X}_2)\\ &= (I - P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}(\overline{X}_2 - \overline{X}_1)\\ &= (I - P_{\widehat{\Sigma}_w^{-1/2}d})\widehat{\Sigma}_w^{-1/2}d = 0,\end{aligned}$$

proving the claim.

Thus, the Fisher Discriminant classification rule (1.4) is equivalent to the discriminant function rule (1.6).

1.3 Optimal Scoring

Optimal scoring [3] is an equivalent formulation of LDA, but it is solved as a least-squares regression problem. It proceeds by transforming the categorical response into a numeric response and then performing least-squares regression to produce a discriminant vector $\widehat{\beta}_{OS} \in \mathbb{R}^p$.

Assumption 2. *The training data $X \in \mathbb{R}^{n \times p}$ is column-centered. That is, $\mathbf{1}_n^\top X = \sum_{i=1}^n \mathbf{x}_i = 0$.*

Under Assumption 2, the Optimal Scoring problem for binary classification finds the discriminant vector $\beta \in \mathbb{R}^p$ and the scores vector $\theta \in \mathbb{R}^2$ which minimize

$$\begin{aligned}&\underset{\theta\in\mathbb{R}^2,\,\beta\in\mathbb{R}^p}{\text{minimize}}\ \frac{1}{n}\|Y\theta - X\beta\|_2^2\\ &\text{subject to } n^{-1}\theta^T Y^\top Y\theta = 1,\ \theta^T Y^\top Y\mathbf{1} = 0.\end{aligned} \tag{1.8}$$

To better understand the constraints imposed in (1.8), note that $n^{-1}\theta^\top Y^\top Y\theta = n^{-1}\|Y\theta\|_2^2$. When paired with the constraint $\theta^\top Y^\top Y\mathbf{1} = (Y\theta)^\top\mathbf{1} = 0$, the transformed response $Y\theta$ is constrained to be mean-zero and unit-variance. Geometrically, the equation $n^{-1}\|Y\theta\|_2^2 = 1$ defines an ellipse in $\mathbb{R}^2$, while the constraint $(Y\theta)^\top\mathbf{1}$ defines a line running through the origin. They intersect at the pair of antipodal points $\pm \left(\sqrt{\frac{n_2}{n_1}} \quad -\sqrt{\frac{n_1}{n_2}} \right)^\top$. We define the score vector to be

$$\widehat{\theta} := \left(\sqrt{\frac{n_2}{n_1}} \quad -\sqrt{\frac{n_1}{n_2}} \right)^\top .$$

Substituting $\widehat{\theta}$ into (1.8) gives the least-squares regression problem

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \; \frac{1}{n} \| Y\widehat{\theta} - X\beta \|_2^2 \tag{1.9}$$

which, in the $n > p$ setting, has the closed-form solution

$$\widehat{\beta}_{OS} = (X^\top X)^{-1} X^\top Y \widehat{\theta}. \tag{1.10}$$

Moreover, the solution $\widehat{\beta}$ corresponds to the discriminant vector in LDA up to scaling, see e.g. [4, Section 3.4] or Section 1.3.1 below. Thus, LDA can be reduced to finding the solution to computing (1.10).

1.3.1 Equivalence of Linear Discriminant Analysis and Optimal Scoring

This section proves the equivalence of Linear Discriminant Analysis and Optimal Scoring in the two class setting.

Theorem 2 (Equality of Classification Rules). *Let $\widehat{\beta} := \widehat{\Sigma}_w^{-1} d$ be the Fisher Discriminant Vector, and let $\widehat{\beta}_{OS}$ the optimal scoring solution* (1.10). *Then using $\widehat{\beta}$ and $\widehat{\beta}_{OS}$ in classification rule* (1.4) *gives equivalent classifiers.*

In order to prove Theorem 2, we first need a series of preliminary Lemmas.

Lemma 1. *We have $n^{-1} X^\top Y \hat{\theta} = \sqrt{n_1 n_2}\, n^{-1} \left(\overline{X}_1 - \overline{X}_2 \right) = d.$*

Proof. We have

$$\begin{aligned}\frac{1}{n}X^\top Y\widehat{\theta} = \frac{1}{n}\begin{pmatrix}X_1^\top & X_2^\top\end{pmatrix}\begin{pmatrix}Y_1\widehat{\theta}\\ Y_2\widehat{\theta}\end{pmatrix} &= \frac{1}{n}\left(\sqrt{\frac{n_2}{n_1}}\sum X_1 - \sqrt{\frac{n_1}{n_2}}\sum X_2\right)\\ &= \frac{1}{n}\left(\sqrt{\frac{n_2}{n_1}}n_1\overline{X_1} - \sqrt{\frac{n_1}{n_2}}n_2\overline{X_2}\right) = \frac{\sqrt{n_1n_2}}{n}(\overline{X_1}-\overline{X_2}).\end{aligned}$$

□

Lemma 2. *Let $\widehat{\Sigma}_w$ and $\widehat{\Sigma}_b$ be the within-group and between-group covariance matrices* (1.1)*. Then $n^{-1}X^\top X = \widehat{\Sigma}_w + \widehat{\Sigma}_b$.*

Proof. We start with

$$\begin{aligned}\widehat{\Sigma}_W &= \frac{1}{n}\sum_{g=1}^{2}\sum_{j=1}^{n_g}(\mathbf{x}_j\mathbf{x}_j^\top - \mathbf{x}_j\overline{X}_g^\top - \overline{X}_g\mathbf{x}_j^\top + \overline{X}_g\overline{X}_g^\top)\\ &= \frac{1}{n}\sum_{g=1}^{2}\sum_{j=1}^{n_g}\mathbf{x}_j\mathbf{x}_j^\top - \frac{1}{n}\sum_{i=1}^{2}\left(\sum_{j=1}^{n_g}\mathbf{x}_j\right)\overline{X}_g^\top - \frac{1}{n}\sum_{i=1}^{2}\overline{X}_g\left(\sum_{j=1}^{n_g}\mathbf{x}_j\right)^\top + \frac{1}{n}\sum_{g=1}^{2}\sum_{j=1}^{n_g}\overline{X}_g\overline{X}_g^\top\\ &= \frac{1}{n}\sum_{g=1}^{2}\sum_{j=1}^{n_g}\mathbf{x}_j\mathbf{x}_j^\top - \frac{1}{n}\sum_{g=1}^{2}n_g\overline{X}_g\overline{X}_g^\top - \frac{1}{n}\sum_{g=1}^{2}n_g\overline{X}_g\overline{X}_g + \frac{1}{n}\sum_{i=1}^{2}\sum_{j=1}^{n_g}\overline{X}_g\overline{X}_g\\ &= \frac{1}{n}\sum_{i=1}^{2}\sum_{j=1}^{n_g}\mathbf{x}_j\mathbf{x}_j^\top - \frac{1}{n}\sum_{g=1}^{2}n_g\overline{X}_g\overline{X}_g^\top\\ &= \frac{1}{n}X^TX - \widehat{\Sigma}_b\end{aligned}$$

which proves the Lemma. □

We now prove a result which states that the optimal scoring solution $\widehat{\beta}_{OS}$ of (1.10) is a scale multiple of the Fisher Discriminant vector $\widehat{\beta}$.

Theorem 3 (Scale Multiple)**.** *Let $\widehat{\beta}_{OS} \in \mathbb{R}^p$ be the optimal scoring solution* (1.10)*, and let $\widehat{\beta} = \widehat{\Sigma}_w^{-1}d$ be the linear discriminant vector. Then $\widehat{\beta}_{OS} = \alpha\widehat{\beta}$ with $\alpha = (1 + d^\top\widehat{\Sigma}_w^{-1}d)^{-1}$.*

Proof. The Woodbury Matrix Identity for adding a rank-one matrix gives

$$(\widehat{\Sigma}_w + \widehat{\Sigma}_b)^{-1} = \widehat{\Sigma}_w^{-1} - \frac{\widehat{\Sigma}_w^{-1}dd^\top\widehat{\Sigma}_w^{-1}}{1 + d^\top\widehat{\Sigma}_w^{-1}d}.$$

Then

$$\widehat{\beta}_{OS} = (\widehat{\Sigma}_w + \widehat{\Sigma}_b)^{-1} d = \Sigma_w^{-1} d - \left(\frac{\Sigma_w^{-1} d d^\top \widehat{\Sigma}_w^{-1}}{1 + d^\top \widehat{\Sigma}_w^{-1} d} \right) d$$
$$= \left(1 - \frac{d^\top \widehat{\Sigma}_w^{-1} d}{1 + d^\top \widehat{\Sigma}_w^{-1} d} \right) \widehat{\Sigma}_w^{-1} d = \left(\frac{1}{1 + d^\top \widehat{\Sigma}_W d} \right) \widehat{\Sigma}_w d.$$

This proves the theorem. □

We now present the proof of Theorem 2.

Proof of Theorem 2. Applying Theorem 3 gives $\widehat{\beta}_{OS} = \alpha \widehat{\beta}$ for some non-zero constant $\alpha \in \mathbb{R}$. The Mahalanobis distances between a test sample $\mathbf{x}$ and the class means $\overline{X}_g$ is equal to

$$\underset{g=1,2}{\text{minimize}} \left\{ (\mathbf{x} - \overline{X}_g)^\top \widehat{\beta}_{OS} (\widehat{\beta^\top}_{OS} \widehat{\Sigma}_W \widehat{\beta}_{OS})^{-1} \widehat{\beta^\top}_{OS} (\mathbf{x} - \overline{X}_g) - 2 \log(n_g/n) \right\}$$
$$= \underset{g=1,2}{\text{minimize}} \left\{ (\mathbf{x} - \overline{X}_g)^\top \alpha \widehat{\beta} (\alpha \widehat{\beta^\top} \widehat{\Sigma}_W \widehat{\beta} \alpha)^{-1} \alpha \widehat{\beta^\top} (\mathbf{x} - \overline{X}_g) - 2 \log(n_g/n) \right\}$$
$$= \underset{g=1,2}{\text{minimize}} \left\{ (\mathbf{x} - \overline{X}_g)^\top \widehat{\beta} (\widehat{\beta^\top} \widehat{\Sigma}_W \widehat{\beta})^{-1} \widehat{\beta^\top} (\mathbf{x} - \overline{X}_g) - 2 \log(n_g/n) \right\}$$

where the last equality comes from the α^2 inside the inverse term $(\alpha \widehat{\beta^\top} \widehat{\Sigma}_W \widehat{\beta} \alpha)^{-1}$ canceling with the α^2 coming from the pair of differences $(\mathbf{x} - \overline{X}_g)^\top \alpha \widehat{\beta}$. This last term is the Fisher Discriminant classification rule (1.4). □

1.4 Review of Reproducing Kernel Hilbert Spaces

Reproducing Kernel Hilbert Spaces (RKHS) generalize linear regression and classification models into flexible non-linear ones. 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 $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$. We let $\| \cdot \|_{\mathcal{H}}$ be the norm induced by the inner product $\langle \cdot , \cdot \rangle_{\mathcal{H}}$. By the *reproducing property* of $\mathcal{H}$: $\langle \Phi(\mathbf{x}), f \rangle_{\mathcal{H}} = f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^p$ 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 $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma^{-2}\|\mathbf{x} - \mathbf{x}'\|_2^2)$ with parameter $\sigma > 0$, the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$ with d being a positive integer, and the sigmoid kernel $k(\mathbf{x}, \mathbf{x}') = \tanh(c \langle \mathbf{x}, \mathbf{x}' \rangle + t)$ with $c > 0$, $t \geq 0$. We refer the reader to [5, Chapter 13] for a review on kernel construction and selection. We let $\mathbf{K} \in \mathbb{R}^{n \times n}$ denote the kernel matrix $\mathbf{K}_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ based on observed feature vectors $\{\mathbf{x}_i\}_{i=1}^n$.

1.4.1 Constructing Reproducing Kernel Hilbert Spaces

This sub-section summarizes the process for constructing Reproducing Kernel Hilbert Spaces. This treatment is merely a summary, and the reader is referred to [5] for additional details.

Constructing $\mathcal{H}$ and Φ proceeds in two broad steps: (i) build a pre-Hilbert space $\mathcal{H}^0$ satisfying all of the desired properties (ii) construct $\mathcal{H}$ by taking the completion of $\mathcal{H}^0$ and checking that all the desired properties continuously extend from $\mathcal{H}^0$ to $\mathcal{H}$.

Fix some kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, and let

$$\mathcal{H}^0 = \left\{ \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) \;\middle|\; \text{for } n \in \mathbb{N} \text{ and with } \alpha_i \in \mathbb{R} \text{ and } \mathbf{x}_i \in \mathbb{R}^p \text{ for } i = 1, \ldots, n \right\} \tag{1.11}$$

be the vector space of all finite linear combinations of kernels which are centered on a finite subset of points in $\mathbb{R}^p$. Additionally, let $\Phi^0 : \mathbb{R}^p \to \mathcal{H}$ be the map defined by $\Phi^0(\mathbf{x}) := k(\mathbf{x}, \cdot)$.

Let

$$f = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) \quad \text{and} \quad g = \sum_{j=1}^{m} \beta_j k(\mathbf{x}'_j, \cdot)$$

be any two elements in $\mathcal{H}^0$, where $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_m\}$ are arbitrary finite subsets of $\mathbb{R}^p$. The pre-inner product between f and g is defined to be

$$\langle f, g \rangle_{\mathcal{H}^0} = \left\langle \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) \, , \, \sum_{j=1}^{m} \beta_j k(\mathbf{x}'_j, \cdot) \right\rangle_{\mathcal{H}^0} := \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}'_j). \tag{1.12}$$

An important property of Reproducing Kernel Hilbert Spaces is that they have the *reproducing property*, which means that function evaluation $f \mapsto f(\mathbf{x})$ is a continuous linear functional for any fixed $\mathbf{x} \in \mathbb{R}^p$. From (1.12), we have

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) = \left\langle \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot)\,,\, k(\mathbf{x}, \cdot) \right\rangle_{\mathcal{H}^0} = \left\langle f, \Phi^0(\mathbf{x}) \right\rangle_{\mathcal{H}^0}.$$

A function $f \in \mathcal{H}^0$ could have multiple expressions of the form (1.11). One can check, using the reproducing property, that the pre-inner product (1.12) is invariant under the particular expression of f and g used. That is, if $f = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) = \sum_{\ell=1}^{\tilde{n}} \tilde{\alpha}_\ell k(\tilde{\mathbf{x}}_\ell, \cdot)$ and $g = \sum_{j=1}^{m} \beta_j k(\mathbf{x}'_j, \cdot) = \sum_{t=1}^{\tilde{m}} \tilde{\beta}_t k(\tilde{\mathbf{x}}'_t, \cdot)$, then

$$\left\langle \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot)\,,\, \sum_{j=1}^{m} \beta_j k(\mathbf{x}'_j, \cdot) \right\rangle_{\mathcal{H}^0} = \left\langle \sum_{\ell=1}^{\tilde{n}} \tilde{\alpha}_\ell k(\tilde{\mathbf{x}}_\ell, \cdot)\,,\, \sum_{t=1}^{\tilde{m}} \tilde{\beta}_t k(\tilde{\mathbf{x}}'_t, \cdot) \right\rangle_{\mathcal{H}^0}.$$

The pre-inner product induces a semi-norm $\|\cdot\|_{\mathcal{H}^0}$ on $\mathcal{H}^0$ defined by

$$\|f\|_{\mathcal{H}^0} := \sqrt{\langle f, f \rangle_{\mathcal{H}^0}} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)}.$$

Since $\langle \cdot, \cdot \rangle_{\mathcal{H}^0}$ is invaraint under the particular representation of f, so too is the semi-norm.

Let

$$\text{Null}(\|\cdot\|_{\mathcal{H}^0}) := \{g \in \mathcal{H}^0 \mid \|g\|_{\mathcal{H}^0} = 0\}$$

be the null-space of the semi-norm $\|\cdot\|_{\mathcal{H}^0}$. The equivalence class $[f]$ of all representations of the function $f \in \mathcal{H}^0$ is equal to the coset $f + \text{Null}(\|\cdot\|_{\mathcal{H}^0})$. Instead of $\mathcal{H}^0$, consider the quotient space $\mathcal{H}/\text{Null}(\|\cdot\|_{\mathcal{H}^0})$, where quotienting by the null-space of $\|\cdot\|_{\mathcal{H}^0}$ collapses all equivalent representations of the same function f into one equivalence class $[f]$.

The pre-inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}^0}$ and semi-norm $\|\cdot\|_{\mathcal{H}^0}$ induce a proper inner product and norm on the quotient space $\mathcal{H}/\text{Null}(\|\cdot\|_{\mathcal{H}^0})$. The quotient space has the reproducing property as well.

The final step to defining $\mathcal{H}$ and Φ is to take the completion of the quotient space $\mathcal{H}/\text{Null}(\|\cdot\|_{\mathcal{H}^0})$ with respect to the induced norm $\|\cdot\|_{\mathcal{H}^0}$. Denote this completion by $\mathcal{H}$. The quotient space isomorphically embeds within its completion $\mathcal{H}/\text{Null}(\|\cdot\|_{\mathcal{H}^0}) \to \mathcal{H}$, and so each equivalence class

$[\Phi^0(\mathbf{x})]$ has a unique representation in $\mathcal{H}$, denoted by $\Phi(x)$.

One can check that any continuous function defined on $\mathcal{H}/$Null$(\|\cdot\|_{\mathcal{H}^0})$ extends continuously to $\mathcal{H}$- including the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}^0}$ and its induced norm $\|\cdot\|_{\mathcal{H}^0}$.[1] Denote by $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and $\|\cdot\|_{\mathcal{H}}$ the extended inner product and norm.

For more on completing normed spaces and continuously extending functions, see [6, Chapter 2].

[1] This fact tacitly uses the completeness of $\mathbb{R}$ or $\mathbb{C}$ (or whichever field the inner product maps to).

2. SPARSE FEATURE SELECTION IN KERNEL DISCRIMINANT ANALYSIS VIA OPTIMAL SCORING[*]

2.1 Introduction

Linear Discriminant Analysis (LDA) is a popular linear classification rule [2, Section 11], however 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 [7, 8, 9, 5]. While the methods have good empirical performance, to our knowledge there is a lack of theoretical guarantees on the risk of the learned classifiers. At the same time, the methods do not perform feature selection, and as such will overfit in the presence of “noise" features.

The majority of kernel theory assumes a convex loss function. An additional challenge with kernel FDA is incorporating sparse feature selection, as the method developed in [10] assumes a convex loss function as well.

On the other hand, several sparse generalizations of LDA have been proposed [11, 12, 13], however the methods still result in linear classification boundaries.

This Chapter addresses the gap between kernel and sparse LDA methods by using an optimal scoring framework [3] 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 [10]. 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 [14] for stabilization of kernel matrices. Our empirical results indicate that the Stabilization method leads to better error rates than generalized cross-validation (GCV) [15, 16, 17], and we

[*] Reprinted with permission from “Sparse feature selection in kernel discriminant analysis via optimal scoring” by Alexander F. Lapanowski and Irina Gaynanova, 2019. *Proceedings of Machine Learning Research*, 1704-1713, Copyright 2019 by Alexander F. Lapanowski and Irina Gaynanova.

believe this method of parameter selection could be of independent interest.

Kernel classifiers often require the selection of several parameters, but [10] does not provide guidance for doing so. This Chapter provides fully-automatic selection methods for gaussian kernel, ridge, and sparsity parameters which avoids cross-validation over all three parameters. This is done by a new automatic ridge parameter selection technique based on [14], which could be of independent interest.

In summary, this Chapter makes the following contributions:

- we develop a kernel LDA method based on optimal scoring framework
- we provide theoretical results on the risk consistency of the proposed classifier
- we use weighted kernels to implement feature selection within kernel LDA
- we propose a new stabilization method for ridge parameter selection.

2.1.1 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 the work of [9]. Unlike [9], 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 adding a sparsity penalty to the weights.

Weighted kernels with sparse weights have been considered in [10, 18] in the context of kernel regression and kernel support vector machines. The framework can not be applied to the original kernel LDA method [8], 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 the linear or quadratic combination of predetermined kernels [19, 20], 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 [21, 22]. Our framework does not impose additivity, thus enabling interactions between the features.

Kernel methods often require selection of multiple tuning parameters. In particular, sparse KOS has the kernel parameters, ridge penalty, and sparsity penalty which all must be selected.

Several methods for automatic kernel parameter selection have been proposed. [23] selects the parameters which minimizes an upper bound on the number of errors made by a leave-one-out cross-validation procedure. [9] minimizes an estimate of the VC-dimension of the set of learnable functions. [24] minimizes a trade off between kernel values of points in the same class with kernel values of points in separate classes. Our approach is inspired by [25] in implementing cross-validation over quantiles of distances between points. Unlike [25], we restrict our consideration to distances between points in separate classes.

Likewise, there is no consensus on a method for selecting a ridge parameter. One of the most commonly used approaches is k-fold cross-validation, as in [26, 27, 28, 29]. [30, Corollaries 3 and 4.] selects the ridge parameter based the rate of eigenvalue decay for the kernel. [31] minimizes a validation mean-squared error over a uniform grid of values. The most popular method for automatic ridge parameter selection is generalized cross-validation (GCV) [15, 17, 16]. However, we have found that GCV can under-select the ridge penalty when the data contains noise features. The method developed here is based on a covariance stabilization technique presented in [14]. Empirical results show the superior performance of our stabilization method compared to GCV.

2.1.2 Notation

We use the following notation throughout the Chapter. 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 $\langle \mathbf{x}, \mathbf{x}' \rangle := \sum_{i=1}^p \mathbf{x}_i \mathbf{x}'_i$ 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 $\|M\|_{\text{op}} := \sup_{\|\mathbf{x}\|_2 = 1} \|M\mathbf{x}\|_2$ 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 $C = I - n^{-1}\mathbf{1}\mathbf{1}^T$ be the centering matrix.

2.2 Kernel Optimal Scoring

2.2.1 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 $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$. We let $\| \cdot \|_{\mathcal{H}}$ be the norm induced by the inner product $\langle \cdot \, , \, \cdot \rangle_{\mathcal{H}}$. By the *reproducing property* of $\mathcal{H}$: $\langle \Phi(\mathbf{x}), f \rangle_{\mathcal{H}} = f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^p$ 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 $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma^{-2} \|\mathbf{x} - \mathbf{x}'\|_2^2)$ with parameter $\sigma > 0$, the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$ with d being a positive integer, and the sigmoid kernel $k(\mathbf{x}, \mathbf{x}') = \tanh(c \langle \mathbf{x}, \mathbf{x}' \rangle + t)$ with $c > 0$, $t \geq 0$. We refer the reader to [5, Chapter 13] for a review on kernel construction and selection. We let $\mathbf{K} \in \mathbb{R}^{n \times n}$ denote the kernel matrix $\mathbf{K}_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ based on observed feature vectors $\{\mathbf{x}_i\}_{i=1}^n$.

2.2.2 Kernel Optimal Scoring

In this section we derive the kernelized formulation of the optimal scoring problem (1.9). Let f be the discriminant function in $\mathcal{H}$ with corresponding map Φ and kernel k. We substitute each inner product in the original space $\mathbf{x}_i^\top \beta = \langle x_i, \beta \rangle$ with inner product in $\mathcal{H}$, $\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \rangle_{\mathcal{H}}$, where we apply centering to $\Phi(\mathbf{x}_i)$ via $\overline{\Phi} := n^{-1} \sum_{i=1}^n \Phi(\mathbf{x}_i)$ to take into account column-centering of X. The corresponding optimal scoring problem in $\mathcal{H}$ takes the form

$$\underset{f \in \mathcal{H}}{\text{minimize}} \left\| Y\widehat{\theta} - \begin{pmatrix} \left\langle \Phi(\mathbf{x}_1) - \overline{\Phi} \, , \, f \right\rangle_{\mathcal{H}} \\ \vdots \\ \left\langle \Phi(\mathbf{x}_n) - \overline{\Phi} \, , \, f \right\rangle_{\mathcal{H}} \end{pmatrix} \right\|_2^2 .$$

By the Representer Theorem [32], the minimizing $\widehat{f}$ lies in the finite-dimensional span of the centered data, that is it is sufficient to consider minimization over $f = \sum_{i=1}^{n} \alpha_i[\Phi(\mathbf{x}_i) - \overline{\Phi}]$ 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:

$$\underset{\alpha \in \mathbb{R}^n}{\text{minimize}} \; \|Y\widehat{\theta} - C\mathbf{K}C\alpha\|_2^2. \tag{2.1}$$

Kernel methods may over-fit the training data without further restriction on the set of functions $f \in \mathcal{H}$, [33, 5, 34]. A common approach is to restrict the norm $\|f\|_{\mathcal{H}}^2 = \alpha^\top C\mathbf{K}C\alpha$, and we add a ridge penalty to the objective function (2.1)

$$\underset{\alpha \in \mathbb{R}^n}{\text{minimize}} \left\{ \frac{1}{n}\|Y\widehat{\theta} - C\mathbf{K}C\alpha\|_2^2 + \gamma\alpha^T C\mathbf{K}C\alpha \right\}, \tag{2.2}$$

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 [8, 9]. We fix $\varepsilon = 10^{-5}$ throughout the Chapter. The problem has a closed-form solution leading to

$$\widehat{\alpha} = \{(C\mathbf{K}C)^2 + n\gamma(C\mathbf{K}C + \varepsilon I)\}^{-1} C\mathbf{K}CY\widehat{\theta}. \tag{2.3}$$

We call (2.2) the kernel optimal scoring problem or KOS.

2.2.3 Classification of a New Data Point

In this section we describe how to use KOS for classification. Let $\widehat{\alpha}$ be as in (2.3), and let $\widehat{f} = \sum_{i=1}^{n} \widehat{\alpha}_i[\Phi(\mathbf{x}_i) - \overline{\Phi}]$. Given a new data point $\mathbf{x} \in \mathbb{R}^p$, let

$$K(X, \mathbf{x}) = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}) & \cdots & k(\mathbf{x}_n, \mathbf{x}) \end{pmatrix}^\top.$$

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

$$\left\langle \Phi(\mathbf{x}) - \overline{\Phi}, \, \widehat{f} \right\rangle_{\mathcal{H}} = (K(X, \mathbf{x})^\top - n^{-1}\mathbf{1}^\top \mathbf{K}) C \widehat{\alpha}. \tag{2.4}$$

The derivation of (2.4) is in Section 2.8.

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

$$\operatorname*{argmin}_{k=1,2} |P(\mathbf{x}) - \mu_k|.$$

2.3 Error Bounds for Kernel Optimal Scoring

Problem (2.2) can be viewed as a regularized empirical risk minimization problem

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ R_{\text{emp}}(f) + \gamma \|f\|_{\mathcal{H}}^2 \right\}, \tag{2.5}$$

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

$$R_{\text{emp}}(f) := \frac{1}{n} \sum_{i=1}^{n} |y_i^\top \widehat{\theta} - \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle|^2. \tag{2.6}$$

By duality, for every $\gamma \geq 0$ there exists a $\tau \geq 0$ such that

$$\widehat{f} = \operatorname*{argmin}_{\|f\|_{\mathcal{H}} \leq \tau} \left\{ R_{\text{emp}}(f) \right\}. \tag{2.7}$$

While the relationship between γ and τ is data-dependent, Lemma 5 in Section 2.10 shows that $\tau \leq C \min(\gamma^{-1}, \gamma^{-1/2})$ for some constant $C > 0$. For technical clarity, we analyze (2.7) throughout.

There are two complications in analyzing the empirical risk in (2.6): $\widehat{\theta}$ is dependent on all y_i

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

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

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

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

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

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

Let $\widehat{f}$ be as in (2.7) 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 3. *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 4. *There exists a constant $\kappa > 0$ such that $\|\Phi(\mathbf{x})\|_{\mathcal{H}} \leq \kappa$ for all $\mathbf{x} \in \mathbb{R}^p$. Equivalently, $\sup_{\mathbf{x} \in \mathbb{R}^p} k(\mathbf{x}, \mathbf{x}) \leq \kappa^2$.*

Assumption 5. *The RKHS $\mathcal{H}$ is separable.*

Remark 1. *The gaussian kernel satisfies Assumption 4 with $\kappa = 1$ and satisfies Assumption 5 by Theorem 7 in [35].*

Using (2.7), 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 4. The extra term $\|\theta^*\|_\infty$ comes from minimizing the modified empirical risk.*

Let

$$(\widetilde{f}, \widetilde{\beta}) := \underset{f \in \mathcal{H}_\tau, \beta \in I_\tau}{\operatorname{argmin}} \widetilde{R}_{\text{emp}}(f, \beta). \tag{2.8}$$

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

$$(f^*, \beta^*) = \underset{f \in \mathcal{H}_\tau, \beta \in I_\tau}{\operatorname{argmin}} R(f, \beta) \tag{2.9}$$

be the minimizers of the expected risk over the set of admissible functions and intercepts. Our proofs rely on characterizing (i) the difference between (2.7) and (2.8), and (ii) the difference between (2.8) and (2.9). The detailed proofs are in Section 2.9, and below we state the main results.

Theorem 4. *Under Assumptions 3–5, 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 5. *Under Assumptions 3–5, 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 $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\} \exp(C_2 \tau^2 \varepsilon^{-2})$.

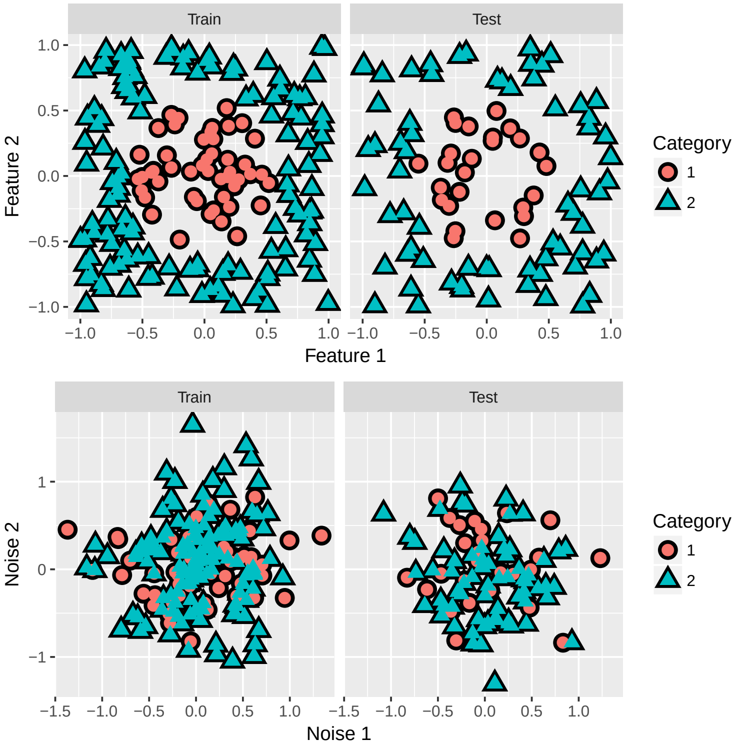

Figure 2.1: Simulated training and test data with four features, only features 1 and 2 contribute to class separation. Reproduced from [1].

Theorem 4 bounds the expected risk of $\widehat{f}$ compared to the best in-class expected risk, whereas Theorem 5 bounds it in terms of the empirical risk of $\widehat{f}$.

2.4 Sparse Kernel Optimal Scoring

The regularized KOS problem (2.2) performs no feature selection, that is 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 the features contribute to class separation. Including such noisy features in the discriminant rule can lead to poor classification performance. Figure 2.1 shows an example of this phenomenon 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 2.2 shows the projected data values (2.4) 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

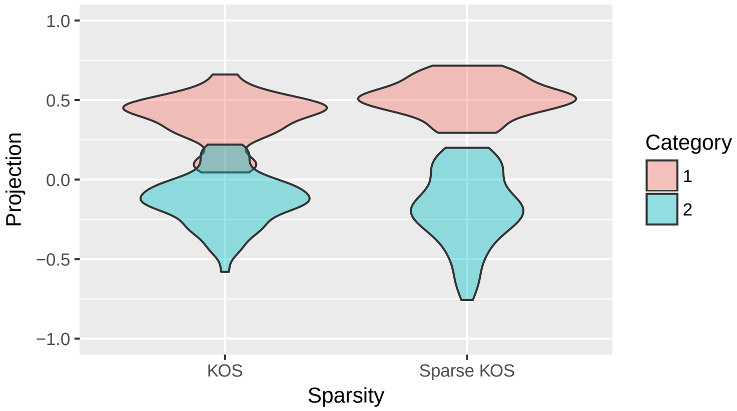

Figure 2.2: Comparing the projection values (2.4) of the test data in Figure 2.1 with and without sparsity. Reproduced from [1].

feature selection within KOS.

To incorporate feature selection, we borrow the ideas from [10] 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 $(\mathbf{K}_w)_{i,j} = k(w\mathbf{x}_i, w\mathbf{x}_j)$, where $wx = (w_1\mathbf{x}_1, \dots, w_p\mathbf{x}_p)^T$ is the Hadamard product between the weight vector w and observed feature vector $\mathbf{x}$. If $w = \mathbf{1}$, $\mathbf{K}_w = \mathbf{K}$ from Section 2.2.2. 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, of course, is that the optimal weight vector w is unknown, and therefore has to be learned in addition to learning the discriminant function f.

Guided by these considerations, we adjust (2.2) 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 [10] leading to the following minimization problem:

$$\begin{aligned} &\underset{\alpha\in\mathbb{R}^n,\, w\in\mathbb{R}^p}{\text{minimize}} \left\{ \frac{1}{n}\|Y\widehat{\theta} - C\mathbf{K}_w C\alpha\|_2^2 + \lambda\|w\|_1 + \gamma\alpha^T(C\mathbf{K}_w C + \varepsilon I)\alpha \right\} \\ &\qquad \text{subject to} \quad -1 \le w_i \le 1 \text{ for } i = 1, \dots, p. \end{aligned} \tag{2.10}$$

Here $\lambda \geq 0$ is the tuning parameter that controls the sparsity of the weight vector w, with larger values leading to sparser solutions. We call (2.10) sparse kernel optimal scoring. Given the solution pair $(\widehat{w}, \widehat{\alpha})$, we perform classification as in Section 2.2.3 with $\mathbf{K}_{\widehat{w}}$ being substituted for $\mathbf{K}$ and $\widehat{w}x$ substituted for $\mathbf{x}$ in forming the projected values $P(x)$ in (2.4).

Remark 3. *Unlike our restriction $w_k \in [-1, 1]$, [10] 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.*

2.4.1 Optimization Algorithm

In this section we describe the optimization algorithm for problem (2.10) given the fixed values of $\gamma, \lambda \geq 0$. Methods for parameter selection are presented in Section 2.5. We define the objective function in (2.10) as

$$\text{Obj}(w, \alpha) = \frac{1}{n}\|Y\widehat{\theta} - C\mathbf{K}_w C\alpha\|_2^2 + \lambda\|w\|_1 + \gamma\alpha^T(C\mathbf{K}_w C + \varepsilon I)\alpha. \tag{2.11}$$

There are two main challenges in solving (2.10): (i) non-convexity of the objective function (2.11) in (α, w) and (ii) non-convex mapping $w \mapsto \mathbf{K}_w$. [10] 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 weight vector. We adapt the algorithm from [10] to problem (2.10).

Given the current value of the weight vector w, we form the corresponding weighted kernel matrix $\mathbf{K}_w$ and update α according to (2.3) with $\mathbf{K}$ substituted with $\mathbf{K}_w$. Given the current value of the coefficient vector α, we update w using linearization of kernel matrix.

Consider the first-order Taylor approximation of $\mathbf{K}_w$ with respect to w centered at previous value $w^{(t-1)}$:

$$\widetilde{\mathbf{K}}_w = \mathbf{K}_{w^{(t-1)}} + \nabla_w \mathbf{K}_{w^{(t-1)}}^T (w^{(t)} - w^{(t-1)}).$$

```
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 $\widehat{w}$.
$t \leftarrow 0$
$w^{(0)} \leftarrow \mathbf{1}$
$(\mathbf{K}_{w^{(0)}})_{i,j} \leftarrow k(w^0\mathbf{x}_i, w^0\mathbf{x}_j)$, $\mathbf{K}_{w^{(0)}} \leftarrow \{(\mathbf{K}_{w_0})_{i,j}\}$
repeat
    $t \leftarrow t+1$
    Update $\alpha^{(t)}$ according to (2.3) with $\mathbf{K} = \mathbf{K}_{w^{(t-1)}}$
    Update $w^{(t)}$ using coordinate descent with updates according to (2.14)
    $(\mathbf{K}_{w^{(t)}})_{i,j} \leftarrow k(w^{(t)}\mathbf{x}_i, w^{(t)}\mathbf{x}_j)$
until $\textit{Obj}(\alpha^{(t)}, w^{(t)}) - \textit{Obj}(\alpha^{(t-1)}, w^{(t-1)}) < \varepsilon_{con}$
return $\widehat{\alpha} = \alpha^{(t)}$, $\widehat{w} = w^{(t)}$
```

Algorithm 1: Sparse Kernel Optimal Scoring

We substitute $\widetilde{\mathbf{K}}_w$ in place of $\mathbf{K}_w$ within (2.10). Let $T \in \mathbb{R}^{n\times p}$ be

$$T := \begin{pmatrix} \sum_{\ell=1}^{n}(C\alpha)_\ell \nabla_w \, \mathbf{K}_{w^{(t-1)}}(\mathbf{x}_1, \mathbf{x}_\ell)^T \\ \vdots \\ \sum_{\ell=1}^{n}(C\alpha)_\ell \nabla_w \, \mathbf{K}_{w^{(t-1)}}(\mathbf{x}_n, \mathbf{x}_\ell)^T \end{pmatrix}.$$

For fixed α, the minimization problem (2.10) with respect to w can be written as

$$\begin{aligned} &\underset{w}{\text{minimize}} \left\{ \frac{1}{2} w^\top Q w - \beta^\top w + \frac{\lambda}{2}\|w\|_1 \right\} \\ &\text{subject to } -1 \leq w_i \leq 1 \text{ for } i = 1, \ldots, p; \end{aligned} \tag{2.12}$$

where

$$\begin{aligned} Q &= \frac{1}{n}(CT)^\top CT \in \mathbb{R}^{p\times p}, \\ \beta &= \frac{1}{n} T^\top C[Y\widehat{\theta} - C\mathbf{K}_{w^{(t-1)}}C\alpha + CTw^{(t-1)}] - 2^{-1}\gamma T^\top C\alpha \in \mathbb{R}^p. \end{aligned} \tag{2.13}$$

Section 2.4.2 provides details on kernel linearization and weight vector update, while the full algorithm is presented in Algorithm 1.

2.4.2 Update of Weights

In this section, we describe the update of weight vector using the linearization of kernel matrix as proposed in [10].

Problem (2.12) is of the same form as the penalized lasso problem [36, Chapter 5] with extra convex constraints on w. Therefore, we can use coordinate-descent algorithm to solve (2.12).

Consider optimizing (2.12) with respect to w_k. From the KKT conditions [37], the solution must satisfy

$$\widehat{w}_k = \operatorname{sign}(\widetilde{w}_k)\min(|\widetilde{w}_k|, 1), \tag{2.14}$$

where

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

and $S_{\lambda/2}(x) := \operatorname{sign}(x)\max\{|x| - \lambda/2,\, 0\}$ is the soft-thresholding function. The coordinate-descent algorithm proceeds by applying update (2.14) on each feature k until convergence.

The full algorithm for (2.10) is summarized as Algorithm 1. While the update of w is based on approximation of objective function (2.11), 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 [10]. We refer to [10] for further discussion of algorithmic convergence.

2.5 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.

2.5.1 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

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

This approach is similar to the one used in the R package kernlab [25], which takes values between $.1$ and $.9$ quantiles of the distance statistic $\|\mathbf{x}-\mathbf{x}'\|_2$ between distinct data points taken from a random subset of the full data. [38] and [25] 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 $\|\mathbf{x}_{i_1}-\mathbf{x}_{i_2}\|_2$. 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.

2.5.2 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. [14] proposes to stabilize the kernel matrix via shrinkage towards a target matrix, and derives an optimal value for the shrinkage parameter. Following [14], in KOS we want to stabilize $(C\mathbf{K}_wC)^2$ with the target matrix $C\mathbf{K}_wC+\varepsilon I$, and therefore consider

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

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) [15, 17, 16] is another common method for selection of ridge parameter, however we found that it performs poorly compared to proposed Stabilization method. Figure 2.3 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 2.6.1. Each time we consider five possible kernel parameters σ^2 based on the distance quantiles as in Section 2.5.1. We then select ridge parameters by either GCV or proposed stabilization method,

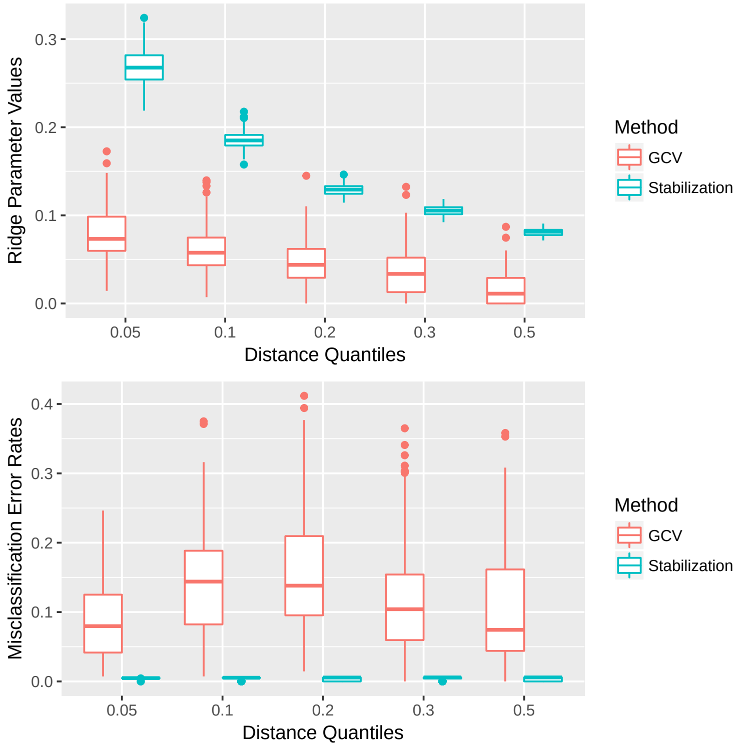

Figure 2.3: Comparison between generalized cross-validation (GCV) and proposed Stabilization method for selection of ridge parameter γ over 100 replications. **Top:** Selected values of γ; **Bottom:** Misclassification error rates. Reproduced from [1].

and choose the best sparsity parameter for each as in Section 2.5.3. 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.

2.5.3 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 (2.13), since the solution $\widehat{w}$ to (2.12) is zero if $\lambda \geq \lambda_{\max}$ (see Lemma 3 of Section 2.10).

2.6 Empirical studies

We compare the performance of the following methods: (i) sparse kernel optimal scoring (Sparse KOS); (ii) 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 2.5, KOS is implemented by setting $\lambda = 0$ and $w = \mathbf{1}$. We use the R package randomForest [39] to create a classifier with 50 decision trees. We use the R package kernlab [25] for kernel SVM using the gaussian kernel with parameter selected as in Section 2.5.1. We use keras [40] to implement a neural network with the ReLU activation function, 50 units, 100 epochs, and the default batch size. We use class [41] for KNN with $K = 5$. We use the R package MGSDA [42] for sparse LDA.

2.6.1 Simulated model 1

We generate data as in Figure 2.1 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, $\mathbf{x}_{ij} \sim \mathcal{N}(0, 2^{-1})$ 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 data sets are presented in Figure 2.4.

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. It gives nonzero weight to the first two features in all 100 splits, and it gives $\widehat{w}_j = 1$ 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.

The results show that sparse kernel optimal scoring out performs the six other non-parameteric classifiers. The median misclassification error rate for sparse KOS is 0.00%, and the upper quartile error rate is 1.11%. By comparison, the lower quartile error rate for random forest classification is 1.08% and the median is 2.15%. Sparse linear discriminant analysis has a median error rate of

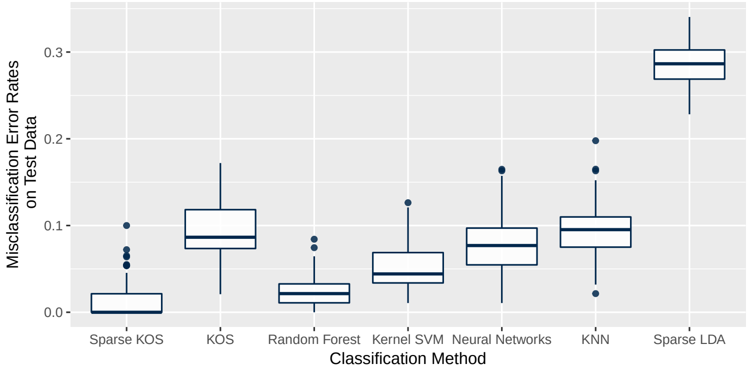

Figure 2.4: Misclassification error rates based on 100 replications of simulated model 1. Reproduced from [1].

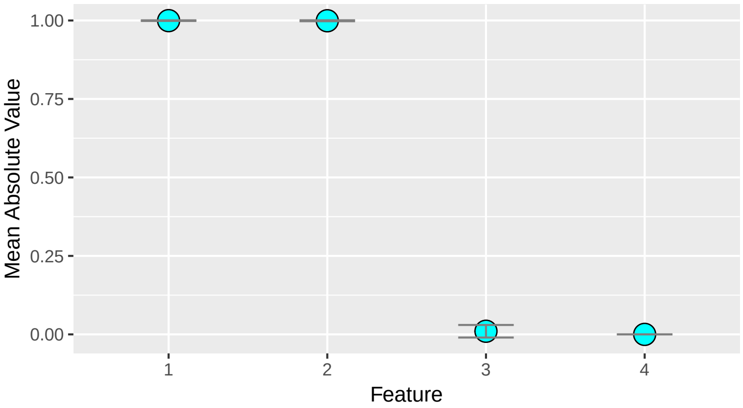

Figure 2.5: Average of the absolute values of the weight values for each feature across the 100 independent simulations of model 1. Bars represent plus or minus twice the standard error. Reproduced from [1].

28.65%. Kernel SVM has a median error rate of 5.38%, while KOS has a median error rate of 8.60%. Neural Networks have a median error rate of 7.53%.

2.6.2 Simulated model 2

We generate data with $p = 10$ features and $n = 400$ samples such that $\mathbf{x}_{i3} + \sin(\mathbf{x}_{i4} + \mathbf{x}_{i1}) < (\mathbf{x}_{i2})^2$ if sample i belongs to class 1, and $\mathbf{x}_{i3} + \sin(\mathbf{x}_{i4} + \mathbf{x}_{i1}) \geq (\mathbf{x}_{i2})^2$ if sample i belongs to class 2. We use the uniform distribution on $[-1, 1]$ for each $\mathbf{x}_{ij}$, so that the last 6 features are uniform noise.

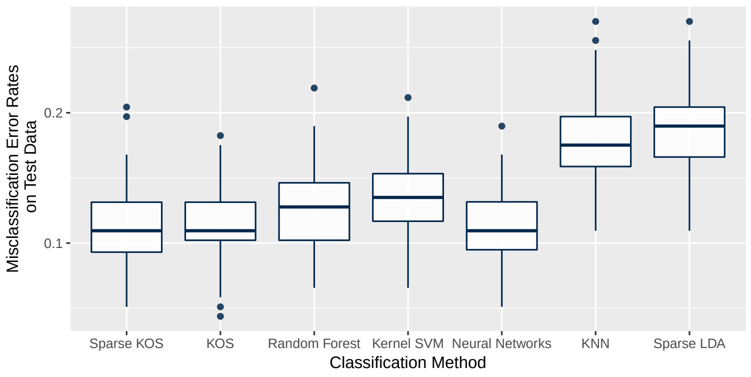

Figure 2.6: Misclassification error rates based on 100 replications of simulated model 2. Reproduced from [1].

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 2.6.

The lowest misclassification error rates are achieved 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 2.7. As with the previous example, sparse LDA performs the worst due to optimal classification boundary being non-linear.

2.6.3 Benchmark datasets

We consider three datasets summarized in Table 2.1, 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 2.2.

In the blood donation study [43], the goal is to determine whether 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

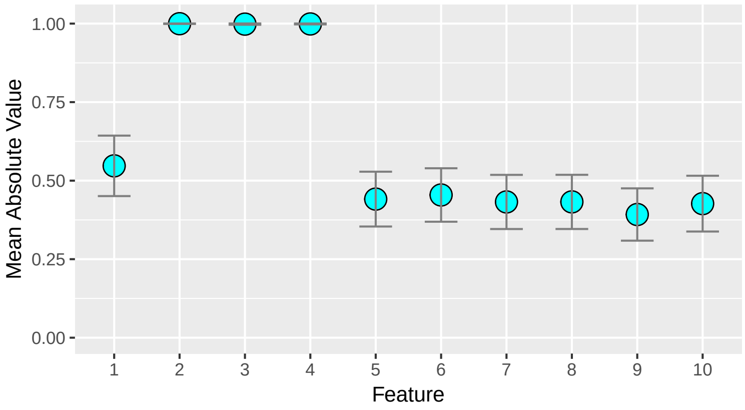

Figure 2.7: The mean absolute values of weights $|w_j|$ for each feature across 100 replications of simulated model 2. The bars represent ±2 standard errors. Reproduced from [1].

Dataset	Features size	Sample size
Blood donation [43]	$p = 4$	$n = 748$
Climate model failure [44]	$p = 18$	$n = 540$
Credit card default [45]	$p = 24$	$n = 3,000$

Table 2.1: Description of benchmark datasets. Reproduced from [1].

	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 2.2: Mean misclassification errors (%) over 100 random splits, standard errors are in brackets. Reproduced from [1].

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 four features are important for classification.

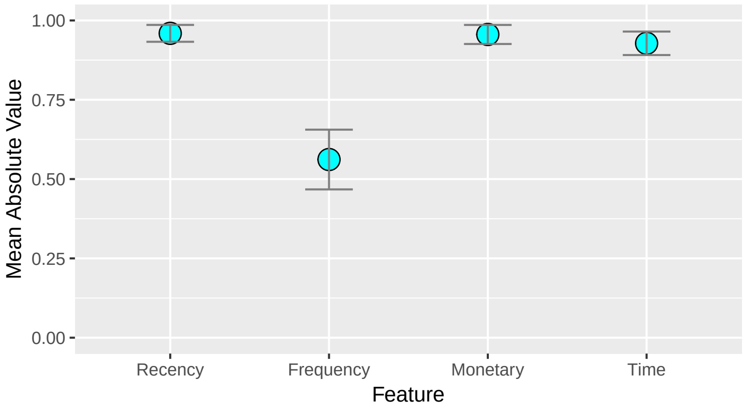

Figure 2.8: Average of the absolute values of the weight values based on 100 replications of the Blood Donation simulation. Error bars indicate plus or minus two standard errors of the mean. Reproduced from [1].

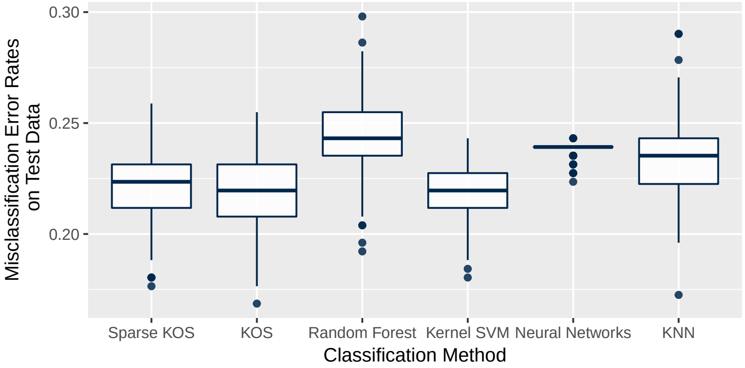

Figure 2.9: Misclassification error rates based on 100 replications for the blood donation data set. Reproduced from [1].

In the climate model study [44], the goal is to predict whether 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).The error rates for 100 iterations are shown in Figure 2.10. Figure 2.13 shows a boxplot of the model sizes over those 100 iterations. The median number of nonzero coefficients used in sparse KOS is 7. Sparse KOS has the best classification performance on these data, which is likely due to its feature selection.

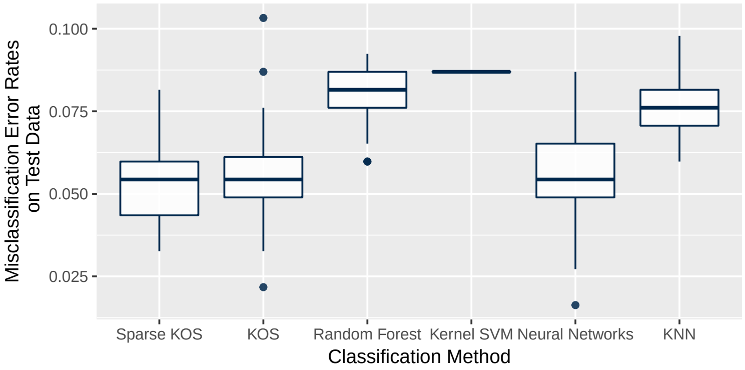

Figure 2.10: Misclassification error rates based on 100 replications of the climate model failure simulation data. Reproduced from [1].

The credit card data [45] have 30,000 data points, but we restrict to $n = 3,000$ for computational simplicity. The goal is to predict the default of a customer for the credit payment based on 24 available features. Sparse KOS has the best classification performance, followed by KOS and random forests. We found that sparse KOS always selects feature 6 (the repayment status in September, 2005, the latest monthly payment recorded), and almost never selects other features. This indicates that the most recent payment history is strongly indicative of credit default.

The misclassification error rates for 100 iterations are depicted in Figure 2.11. Sparse KOS clearly performs better than the other six classifiers. The median misclassification error rate for sparse KOS is 18.0$\%$, and the upper quartile rate is 18.5$\%$. By comparison, the next best classifier, random forests, has a lower quartile error rate of 18.6$\%$ and a median rate of 19.3$\%$. KOS has a lower quartile rate of 18.7$\%$ and a median rate of 19.2$\%$. Sparse LDA has a constant error rate across all iterations because it classifies all test data points as belonging to the same class.

2.7 Discussion

In this Chapter, we propose a kernel discriminant classifier with sparse feature selection called sparse kernel optimal scoring. An advantage of sparsity is that it often improves the classification performance (see Section 2.6), and leads to more interpretable classification rules. The nonzero weights produced by sparse KOS can be used to judge the importance of features. While we have

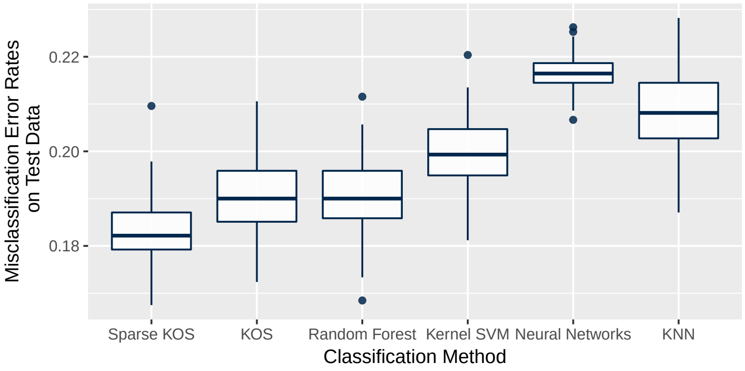

Figure 2.11: Credit card Default simulation. Misclassification error rates based on 100 replications for sparse kernel optimal scoring (sparse KOS), kernel optimal scoring (KOS), random forests, kernel support vector machines (Kernel SVM), neural networks, K-nearest neighbors (KNN), and sparse linear discriminant analysis (Sparse LDA). Reproduced from [1].

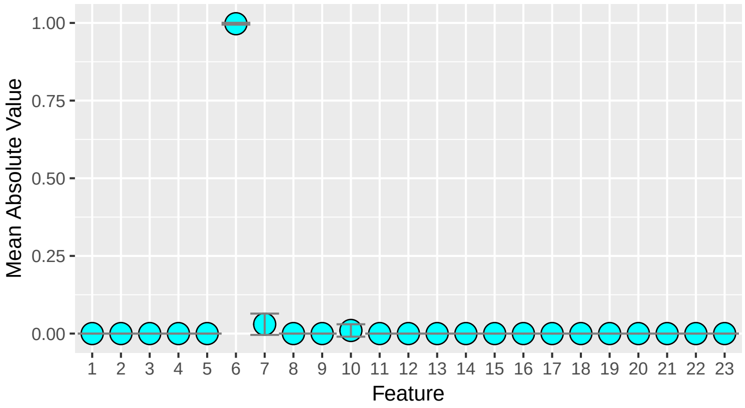

Figure 2.12: Average absolute values for the feature weight values across the 100 simulations of the Credit Card Default simulation. Bars represent plus or minus twice the standard error of the mean. Reproduced from [1].

focused the discussion on the case of two classes, the method can be generalized to multiple classes using optimal scoring formulation in [46].

One limitation of sparse KOS is that it requires the construction of a $n \times n$ kernel matrix $\mathbf{K}$, and therefore is computationally prohibitive for large n cases. An interesting direction for future research is to investigate the appropriate low-dimensional approximations of $\mathbf{K}$ within kernel optimal scoring framework.

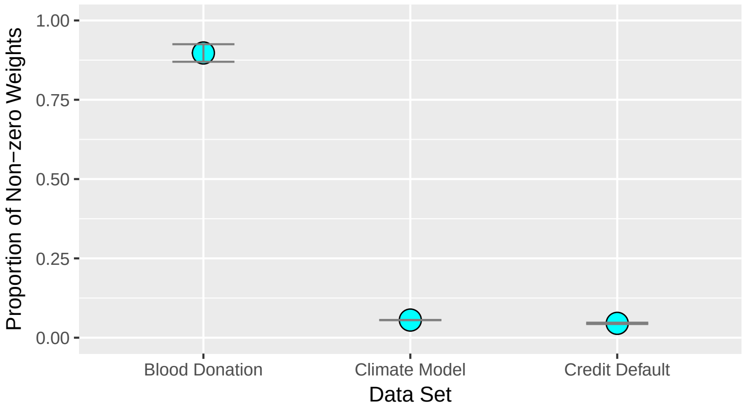

Figure 2.13: Average ratio of number of nonzero weights across the 100 splits in each simulation study using the benchmark data sets. Reproduced from [1].

2.8 Derivation of Projection Formula (2.4)

Proof. Since $\widehat{f} = \sum_{i=1}^{n} \widehat{\alpha}_i[\Phi(\mathbf{x}_i) - \overline{\Phi}]$,

$$\begin{aligned}
&\left\langle \Phi(\mathbf{x}) - \overline{\Phi}, \widehat{f} \right\rangle_{\mathcal{H}} \\
&= \left\langle \Phi(\mathbf{x}) - \overline{\Phi}\,, \sum_{i=1}^{n} \widehat{\alpha}_i[\Phi(\mathbf{x}_i) - \overline{\Phi}] \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \widehat{\alpha}_i \left\langle \Phi(\mathbf{x}) - \overline{\Phi}, \Phi(\mathbf{x}_i) - \overline{\Phi} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \widehat{\alpha}_i \left\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \right\rangle_{\mathcal{H}} - \sum_{i=1}^{n} \widehat{\alpha}_i \left\langle \Phi(\mathbf{x}), \overline{\Phi} \right\rangle_{\mathcal{H}} - \sum_{i=1}^{n} \widehat{\alpha}_i \left\langle \overline{\Phi}, \Phi(\mathbf{x}_i) \right\rangle_{\mathcal{H}} + \sum_{i=1}^{n} \widehat{\alpha}_i \left\langle \overline{\Phi}, \overline{\Phi} \right\rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{n} \widehat{\alpha}_i k(\mathbf{x}, \mathbf{x}_i) - (\mathbf{1}^\top \widehat{\alpha}) \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i) - \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \widehat{\alpha}_i k(\mathbf{x}_j, \mathbf{x}_i) + (\mathbf{1}^\top \widehat{\alpha}) \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{x}_i, \mathbf{x}_j).
\end{aligned}$$

Let $K(X, \mathbf{x}) := \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}) & \cdots & k(\mathbf{x}_n, \mathbf{x}) \end{pmatrix}^\top$. Then from the above display

$$\begin{aligned}
\left\langle \Phi(\mathbf{x}) - \overline{\Phi}, \widehat{f} \right\rangle_{\mathcal{H}} &= K(X, \mathbf{x})^\top \widehat{\alpha} - n^{-1} K(X, \mathbf{x})^\top \mathbf{1}\mathbf{1}^\top \widehat{\alpha} - n^{-1} \mathbf{1}^\top \mathbf{K} \widehat{\alpha} + \frac{1}{n^2} \mathbf{1}^\top \mathbf{K} \mathbf{1} (\mathbf{1}^\top \widehat{\alpha}) \\
&= K(X, \mathbf{x})^\top C \widehat{\alpha} - \frac{1}{n} \mathbf{1}^\top \mathbf{K} C \widehat{\alpha} \\
&= (K(X, \mathbf{x})^\top - \frac{1}{n} \mathbf{1}^\top K) C \widehat{\alpha},
\end{aligned}$$

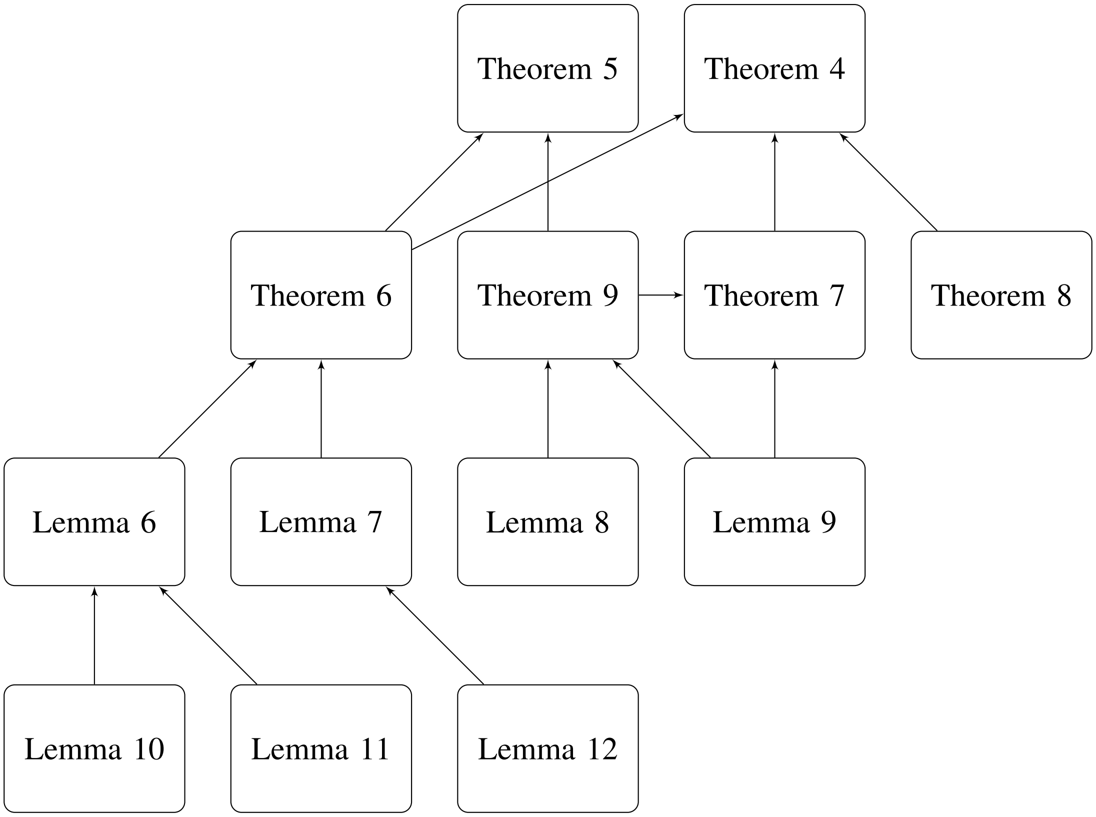

Figure 2.14: Proof charts for Theorems 4 and 5. Reproduced from [1].

where $C = I - n^{-1}\mathbf{1}\mathbf{1}^\top$ is the centering matrix. $\square$

2.9 Technical Proofs

In this section we prove the results stated within the main text. We use C, C_1, C_2, $\ldots$ to denote absolute positive constants that don't depend on the sample size n but which may depend on $\|\theta^*\|_\infty$, κ, or τ. Their values may change from line to line. The dependence between the main Theorems and supplementary results is depicted below.

2.9.1 Proofs of Theorems 1 and 2

Proof of Theorem 4. Consider

$$\begin{aligned}
&R(\widehat{f},\widehat{\beta}) - R(f^*,\beta^*)\\
&= \underbrace{R(\widehat{f},\widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widehat{f},\widehat{\beta})}_{I_1} + \underbrace{\widetilde{R}_{\text{emp}}(\widehat{f},\widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta})}_{I_2} + \underbrace{\widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta}) - R(f^*,\beta^*)}_{I_3}.
\end{aligned}$$

By the union bound and de Morgan's law,

$$\mathbb{P}\Big(R(\widehat{f},\widehat{\beta}) - R(f^*,\beta^*) > \varepsilon\Big) \le \mathbb{P}\Big(I_1 > \frac{\varepsilon}{3}\Big) + \mathbb{P}\Big(I_2 > \frac{\varepsilon}{3}\Big) + \mathbb{P}\Big(I_3 > \frac{\varepsilon}{3}\Big).$$

Applying Theorems 6, 7 and 8 to I_1, I_2 and I_3 correspondingly, there exist constants $C, C_i > 0$ such that

$$\begin{aligned}
&\mathbb{P}\Big(R(\widehat{f},\widehat{\beta}) - R(f^*,\beta^*) > \varepsilon\Big)\\
&\le 2\mathcal{N}_\varepsilon \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big) + C_2 \exp\Big(-\frac{C_3 n\varepsilon^2}{1+(\kappa\tau)^2}\Big) + 2\exp\Big(-\frac{n\varepsilon^2}{16(\|\theta^*\|_\infty + \kappa\tau)^4}\Big)\\
&\le C_4 \mathcal{N}_\varepsilon \exp\Big(-\frac{C_5 n\varepsilon^2}{(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),
\end{aligned}$$

where $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\}\exp(C\tau^2\varepsilon^{-2})$. This concludes the proof of Theorem 4.

□

Proof of Theorem 5. Consider

$$R(\widehat{f},\widehat{\beta}) - R_{\text{emp}}(\widehat{f}) = \underbrace{R(\widehat{f},\widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widehat{f},\widehat{\beta})}_{I_1} + \underbrace{\widetilde{R}_{\text{emp}}(\widehat{f},\widehat{\beta}) - R_{\text{emp}}(\widehat{f})}_{I_2}.$$

By the union bound and de Morgan's law,

$$\mathbb{P}\Big(R(\widehat{f},\widehat{\beta}) - R_{\text{emp}}(\widehat{f}) > \varepsilon\Big) \le \mathbb{P}\Big(I_1 > \frac{\varepsilon}{2}\Big) + \mathbb{P}\Big(I_2 > \frac{\varepsilon}{2}\Big).$$

Applying Theorem 6 for I_1 and Theorem 9 for I_2, the exist constants $C_i > 0$ such that

$$\mathbb{P}\Big(R(\widehat{f},\widehat{\beta}) - R_{\text{emp}}(\widehat{f}) > \varepsilon\Big) \leq 2\mathcal{N}_\varepsilon \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big) + C_3\exp\Big(-\frac{C_4 n\varepsilon^2}{1+(\kappa\tau)^2}\Big)$$
$$\leq C_5\mathcal{N}_\varepsilon \exp\Big(-\frac{C_6 n\varepsilon^2}{(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),$$

where $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\}\exp(C_1\tau^2\varepsilon^{-2})$. This concludes the proof of Theorem 5. □

2.9.2 Supplementary Theorems

Theorem 6. *Under Assumptions 3-5, there exists a constant $C_2 > 0$ such that for all $\varepsilon > 0$,*

$$\mathbb{P}\Big(\sup_{f\in\mathcal{H}_\tau,\beta\in I_\tau}\{R(f,\beta) - \widetilde{R}_{emp}(f,\beta)\} > \varepsilon\Big) \leq 2\mathcal{N}_\varepsilon \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),$$

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

Theorem 7. *Let $\widehat{\beta} = -\Big\langle \overline{\Phi}, \widehat{f}\Big\rangle_{\mathcal{H}}$. Under Assumptions 3 and 4, there exist constants $C_1, C_2 > 0$ such that for all $\varepsilon > 0$,*

$$\mathbb{P}\Big(\Big|\widetilde{R}_{emp}(\widehat{f},\widehat{\beta}) - \widetilde{R}_{emp}(\widetilde{f},\widetilde{\beta})\Big| > \varepsilon\Big) \leq C_1\exp\Big(-\frac{C_2 n\varepsilon^2}{1+(\kappa\tau)^2}\Big).$$

Theorem 8. *Under Assumptions 3 and 4, for all $\varepsilon > 0$*

$$\mathbb{P}\Big(\widetilde{R}_{emp}(\widetilde{f},\widetilde{\beta}) - R(f^*,\beta^*) > \varepsilon\Big) \leq 2\exp\Big(-\frac{n\varepsilon^2}{16(\|\theta^*\|_\infty + \kappa\tau)^4}\Big).$$

Theorem 9. *Let Assumptions 3 and 4 be true, and let $\beta(f) := n^{-1}\sum_{i=1}^n y_i^\top\theta^* - \langle\overline{\Phi}, f\rangle_{\mathcal{H}} = \overline{Y\theta^*} - \langle\overline{\Phi}, f\rangle_{\mathcal{H}}$ be the minimizing $\beta \in I_\tau$ for fixed $f \in \mathcal{H}_\tau$ in the modified empirical risk. There exists constants $C_1, C_2 > 0$ such that for all $\varepsilon > 0$*

$$\mathbb{P}\Big(\sup_{f\in\mathcal{H}_\tau}|R_{emp}(f) - \widetilde{R}_{emp}(f,\beta(f))| > \varepsilon\Big) \leq C_1\exp\Big(-\frac{C_2 n\varepsilon^2}{1+(\kappa\tau)^2}\Big).$$

Definition 2. *The* empirical measure *T_x with respect to $\{x_i\}_{i=1}^n$ is defined as $T_x := n^{-1}\sum_{i=1}^n \delta(x_i)$,* *where $\delta(x_i)$ is the point mass at $\mathbf{x}_i$. The space $L^2(T_x)$ is the set $\mathcal{H}_\tau$ equipped with the semi-norm*

$$\|f\|_{L^2(T_x)} := \sqrt{\frac{1}{n}\sum_{i=1}^{n} |f(x_i)|^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |\left\langle \Phi(\mathbf{x}_i), f\right\rangle_{\mathcal{H}}|^2}.$$

Definition 3. *Let (X, d) be a pseudometric space. An ε-net is any subset $\widetilde{X} \subset X$ such that for any $\mathbf{x} \in X$, there exists a $\widetilde{x} \in \widetilde{X}$ satisfying $d(x, \widetilde{x}) < \varepsilon$. The ε-*covering number *of (X, d) is the minimum size of an ε-net for X.*

Remark 4. *Distances in $\mathcal{H}_\tau$ are given by the semi-norm generated by $L^2(T_x)$. Distances in I_τ are given by the Euclidean distance $d(\beta_1, \beta_2) = |\beta_1 - \beta_2|$.*

2.9.3 Proofs of Supplementary Theorems

Proof of Theorem 6. Let $\{(x_j, y_j)\}_{j=n+1}^{2n}$ be independent from $\{(x_i, y_i)\}_{i=1}^{n}$ and identically distributed set of n pairs, and let T_x be the empirical measure on $\{(x_i, y_i)\}_{i=1}^{2n}$. Let $\widetilde{R}_{\text{emp}}(f, \beta)$ be the modified empirical risk on $\{(x_i, y_i)\}_{i=1}^{n}$, and $\widetilde{R}'_{\text{emp}}(f, \beta)$ on $\{(x_j, y_j)\}_{i=n+1}^{2n}$. By symmetrization lemma (see, for example, Lemma 2 in [47]), for $n\varepsilon^2 \geq 2$

$$\mathbb{P}\bigg(\sup_{f\in\mathcal{H}_\tau,\beta\in I_\tau} \{R(f, \beta) - \widetilde{R}_{\text{emp}}(f, \beta)\} > \varepsilon\bigg) \leq 2\mathbb{P}\bigg(\sup_{f\in\mathcal{H}_\tau,\beta\in I_\tau} \{\widetilde{R}'_{\text{emp}}(f, \beta) - \widetilde{R}_{\text{emp}}(f, \beta)\} > \frac{\varepsilon}{2}\bigg).$$

Let $c = 64(\|\theta^*\|_\infty + \kappa\tau)$, and let $\{f_1, \ldots, f_M\}$ be the smallest $L^2(T_x)$ $\varepsilon/\sqrt{2}c$-net of $\mathcal{H}_\tau$ and $\{\beta_1, \ldots, \beta_K\}$ an ε/c-net of I_τ. Applying Lemma 6 to the above display

$$\mathbb{P}\bigg(\sup_{f\in\mathcal{H}_\tau,\beta\in I_\tau} \{R(f, \beta) - \widetilde{R}_{\text{emp}}(f, \beta)\} > \varepsilon\bigg) \leq 2\mathbb{P}\bigg(\underset{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}{\text{maximize}} \{\widetilde{R}'_{\text{emp}}(f, \beta) - \widetilde{R}_{\text{emp}}(f, \beta)\} > \frac{\varepsilon}{4}\bigg).$$

Applying Lemma 7 to the right-hand expression gives the final inequality

$$\begin{aligned}&\mathbb{P}\bigg(\sup_{f\in\mathcal{H}_\tau,\beta\in I_\tau} \{R(f, \beta) - \widetilde{R}_{\text{emp}}(f, \beta)\} > \varepsilon\bigg)\\ &\quad\leq 2\{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\} \exp\Big(\frac{C_1\tau^2}{\varepsilon^2}\Big) \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big).\end{aligned}$$

This completes the proof of Theorem 6. □

Proof of Theorem 7. Let $\beta(f) = \overline{Y\theta^*} - \langle \overline{\Phi}, f \rangle_{\mathcal{H}}$. By definition of $\widetilde{f}$, $\widetilde{\beta} = \beta(\widetilde{f})$, $\widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) \geq \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta})$. On the other hand, since $R_{\text{emp}}(\widehat{f}) \leq R_{\text{emp}}(\widetilde{f})$,

$$\begin{aligned}
&\widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta}) \\
&= \widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - R_{\text{emp}}(\widehat{f}) + R_{\text{emp}}(\widehat{f}) - R_{\text{emp}}(\widetilde{f}) + R_{\text{emp}}(\widetilde{f}) - \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta}) \\
&\leq \widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - R_{\text{emp}}(\widehat{f}) + R_{\text{emp}}(\widetilde{f}) - \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta}) \\
&\leq \widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widehat{f}, \beta(\widehat{f})) + \widetilde{R}_{\text{emp}}(\widehat{f}, \beta(\widehat{f})) - R_{\text{emp}}(\widehat{f}) + R_{\text{emp}}(\widetilde{f}) - \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta}) \\
&\leq \underbrace{\left| \widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widehat{f}, \beta(\widehat{f})) \right|}_{I_1} + \underbrace{2 \sup_{f \in \mathcal{H}_\tau} \left| R_{\text{emp}}(f) - \widetilde{R}_{\text{emp}}(f, \beta(f)) \right|}_{I_2}.
\end{aligned}$$

The union bound and de Morgan's law proves

$$\mathbb{P}\Big(\widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widetilde{f}, \widetilde{\beta}) > \varepsilon \Big) \leq \mathbb{P}\Big(I_1 > \frac{\varepsilon}{2} \Big) + \mathbb{P}\Big(I_2 > \frac{\varepsilon}{2} \Big).$$

Consider I_1

$$\begin{aligned}
&\left| \widetilde{R}_{\text{emp}}(\widehat{f}, \widehat{\beta}) - \widetilde{R}_{\text{emp}}(\widehat{f}, \beta(\widehat{f})) \right| \\
&= \Big| \frac{1}{n} \sum_{i=1}^{n} \Big(y_i^\top \theta^* - \Big\langle \Phi(\mathbf{x}_i) - \overline{\Phi}\,,\, \widehat{f} \Big\rangle_{\mathcal{H}} \Big)^2 - \frac{1}{n} \sum_{i=1}^{n} \Big(y_i^\top \theta^* - \overline{Y\theta^*} - \Big\langle \Phi(\mathbf{x}_i) - \overline{\Phi}\,,\, \widehat{f} \Big\rangle_{\mathcal{H}} \Big)^2 \Big| \\
&= \Big| 2 \frac{1}{n} \sum_{i=1}^{n} \overline{Y\theta^*} \Big(y_i^\top \theta^* - \Big\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, \widehat{f} \Big\rangle_{\mathcal{H}} \Big) - \frac{1}{n} \sum_{i=1}^{n} (\overline{Y\theta^*})^2 \Big| \\
&= \Big| (\overline{Y\theta^*})^2 - 2(\overline{Y\theta^*}) \frac{1}{n} \sum_{i=1}^{n} \Big\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, \widehat{f} \Big\rangle_{\mathcal{H}} \Big| \\
&= |\overline{Y\theta^*}|^2.
\end{aligned}$$

By Lemma 9, there exists $C_1 > 0$ such that $\mathbb{P}(I_1 > \varepsilon/2) \leq 2\exp(-C_1 n \varepsilon)$ for all $\varepsilon > 0$. By Theorem 9, there exists constants $C_2, C_3 > 0$ such that $\mathbb{P}(I_2 > \varepsilon/2) \leq C_2 \exp[-C_3(n\varepsilon^2)/\{1 +$

$(\kappa\tau)^2\}]$. Combining the bounds for I_1 and I_2 gives

$$\begin{aligned}\mathbb{P}\Big(\widetilde{R}_{\text{emp}}(\widehat{f},\widehat{\beta})-\widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta})>\varepsilon\Big)&\le 2\exp(-C_1n\varepsilon)+C_2\exp\Big(-\frac{C_3n\varepsilon^2}{1+(\kappa\tau)^2}\Big)\\&\le C_4\exp\Big(-\frac{C_5n\varepsilon^2}{1+(\kappa\tau)^2}\Big)\end{aligned}$$

for some constants $C_i>0$. This completes the proof of Theorem 7. □

Proof of Theorem 8. Consider

$$\begin{aligned}\widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta})-R(f^*,\beta^*)&=\widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta})-\widetilde{R}_{emp}(f^*,\beta^*)+\widetilde{R}_{emp}(f^*,\beta^*)-R(f^*,\beta^*)\\&\le\widetilde{R}_{emp}(f^*,\beta^*)-R(f^*,\beta^*),\end{aligned}$$

where the last inequality follows since $\widetilde{R}_{\text{emp}}(\widetilde{f},\widetilde{\beta})\le\widetilde{R}_{emp}(f^*,\beta^*)$ by the definition of $\widetilde{f}$, $\widetilde{\beta}$.

Let $z_i:=|y_i^\top\theta^*-\beta^*-\langle\Phi(\mathbf{x}_i),f^*\rangle_{\mathcal{H}}|^2$, then $\widetilde{R}_{\text{emp}}(f^*,\beta^*)=n^{-1}\sum_{i=1}^n z_i$ is the average of i.i.d. random variables with $\mathbb{E}z_i=R(f^*,\beta^*)$ by definition of expected risk. Since $|z_i|\le 4(\|\theta^*\|_\infty+\kappa\tau)^2$, by Hoeffding's inequality

$$\mathbb{P}(|\widetilde{R}_{\text{emp}}(f^*,\beta^*)-R(f^*,\beta^*)|>\varepsilon)=\mathbb{P}\Big(\Big|n^{-1}\sum_{i=1}^n(z_i-\mathbb{E}z_i)\Big|>\varepsilon\Big)\le 2\exp\Big(-\frac{n\varepsilon^2}{16(\|\theta^*\|_\infty+\kappa\tau)^4}\Big).$$

□

Proof of Theorem 9. By definition of $R_{\text{emp}}(f)$ and $\widetilde{R}_{\text{emp}}(f,\beta(f))$,

$$\begin{aligned}&R_{\text{emp}}(f)-\widetilde{R}_{\text{emp}}(f,\beta(f))\\&=\frac{1}{n}\sum_{i=1}^n|y_i^\top\widehat{\theta}-\big\langle\Phi(\mathbf{x}_i)-\overline{\Phi},f\big\rangle_{\mathcal{H}}|^2-\frac{1}{n}\sum_{i=1}^n|y_i^\top\theta^*-\beta(f)-\langle\Phi(\mathbf{x}_i),f\rangle_{\mathcal{H}}|^2\\&=\frac{1}{n}\sum_{i=1}^n|y_i^\top\widehat{\theta}-\big\langle\Phi(\mathbf{x}_i)-\overline{\Phi},f\big\rangle_{\mathcal{H}}|^2-\frac{1}{n}\sum_{i=1}^n|y_i^\top\theta^*-\overline{Y\theta^*}-\big\langle\Phi(\mathbf{x}_i)-\overline{\Phi},f\big\rangle_{\mathcal{H}}|^2.\end{aligned}$$

Expanding the squares and cancelling equal terms yields

$$
\begin{aligned}
&R_{\text{emp}}(f) - \widetilde{R}_{\text{emp}}(f, \beta(f)) \\
&= \frac{1}{n}\sum_{i=1}^{n} \Big\{ (y_i^\top \widehat{\theta})^2 - (y_i^\top \theta^*)^2 - 2y_i^\top (\widehat{\theta} - \theta^*) \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} \\
&\qquad\qquad - 2\overline{Y\theta^*} \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} + 2y_i^\top \theta^* \overline{Y\theta^*} - (\overline{Y\theta^*})^2 \Big\} \\
&= \frac{1}{n}\sum_{i=1}^{n} \Big\{ (y_i^\top \widehat{\theta})^2 - (y_i^\top \theta^*)^2 \Big\} - \frac{1}{n}\sum_{i=1}^{n} \Big\{ 2y_i^\top (\widehat{\theta} - \theta^*) \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} \Big\} + (\overline{Y\theta^*})^2 \\
&= I_1 + I_2(f) + I_3,
\end{aligned}
$$

where I_1 and I_3 are independent of f. By the union bound and de Morgan's law,

$$
\mathbb{P}\Big(\sup_{f\in\mathcal{H}_\tau} |R_{\text{emp}}(f) - \widetilde{R}_{\text{emp}}(f, \beta(f))| > \varepsilon \Big) \le \mathbb{P}\Big(|I_1| > \frac{\varepsilon}{3} \Big) + \mathbb{P}\Big(\sup_{f\in\mathcal{H}_\tau} |I_2(f)| > \frac{\varepsilon}{3} \Big) + \mathbb{P}\Big(|I_3| > \frac{\varepsilon}{3} \Big).
$$

We bound each probability separately. Since $y_i \in \mathbb{R}^2$ is an indicator vector of class membership for sample i, using the definition of $\widehat{\theta}$ and θ^*

$$
\begin{aligned}
|I_1| &= \left| \frac{1}{n} \sum \Big\{ (y_i^\top \widehat{\theta})^2 - (y_i^\top \theta^*)^2 \Big\} \right| \le \max_i |(y_i^\top \widehat{\theta})^2 - (y_i^\top \theta^*)^2| \\
&= \max \Big(|n_1/n_2 - \pi_1/\pi_2|, |n_2/n_1 - \pi_2/\pi_1| \Big).
\end{aligned}
$$

By Lemma 8, there exist $C_1, C_2 > 0$ such that $\mathbb{P}(|I_1| > \varepsilon/3) \le C_1 \exp(-C_2 n\varepsilon^2)$.

By Hölder's and Cauchy-Schwarz inequalities

$$
\begin{aligned}
|I_2(f)| &= \left| \frac{1}{n} \sum_{i=1}^{n} 2y_i^\top (\widehat{\theta} - \theta^*) \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} \right| \\
&\le \frac{1}{n} \sum_{i=1}^{n} 2|y_i^\top (\widehat{\theta} - \theta^*)| \cdot | \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} | \\
&\le 2\|\widehat{\theta} - \theta^*\|_\infty \max_i | \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, f \right\rangle_{\mathcal{H}} | \\
&\le 2 \max \Big(|\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2}|, |\sqrt{n_2/n_1} - \sqrt{\pi_2/\pi_1}| \Big) \max_i \|\Phi(\mathbf{x}_i) - \overline{\Phi}\|_{\mathcal{H}} \|f\|_{\mathcal{H}} \\
&\le 4 \max \Big(|\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2}|, |\sqrt{n_2/n_1} - \sqrt{\pi_2/\pi_1}| \Big) \kappa\tau,
\end{aligned}
$$

where we used Assumption 4 in the last inequality. Since the upper bound does not depend on f, the same bound holds for $\sup_{f \in \mathcal{H}_\tau} |I_2(f)|$. Combining the bound with Lemma 8 gives for some $C_3, C_4 > 0$

$$
\begin{aligned}
\mathbb{P}\Big(\sup_{f \in \mathcal{H}_\tau} |I_2(f)| > \varepsilon \Big) &\le \mathbb{P}\Bigg(\max \Big(|\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2}|, |\sqrt{n_2/n_1} - \sqrt{\pi_2/\pi_1}| > \frac{\varepsilon}{4\kappa\tau} \Big) \\
&\le C_3 \exp(-C_4 \frac{n\varepsilon^2}{(\kappa\tau)^2}).
\end{aligned}
$$

By Lemma 9, there exists $C_5 > 0$ such that $\mathbb{P}(|I_3| > \varepsilon/3) \le 2\exp(-C_5 n\varepsilon)$.

Combining the bounds for I_1, I_2 and I_3 gives

$$
\begin{aligned}
&\mathbb{P}\Big(\sup_{f \in \mathcal{H}_\tau} |R_{\text{emp}}(f) - \widetilde{R}_{\text{emp}}(f, \beta(f))| > \varepsilon \Big) \\
&\le C_1 \exp(-C_2 n\varepsilon^2) + C_3 \exp(-C_4 \frac{n\varepsilon^2}{(\kappa\tau)^2}) + 2\exp(-C_5 n\varepsilon) \\
&\le C_6 \exp\Big(-C_7 \frac{n\varepsilon^2}{1 + (\kappa\tau)^2} \Big)
\end{aligned}
$$

for some $C_6, C_7 > 0$. This completes the proof of Theorem 9. □

2.10 Supplementary Lemmas

Lemma 3. *Consider minimizing $f(w) = 2^{-1}w^TQw - \beta^Tw + 2^{-1}\lambda\|w\|_1$ with respect to $w \in \mathbb{R}^p$ with $w_i \in [-1, 1]$, where Q is positive semi-definite and $\lambda \geq 0$. If $\lambda \geq 2\|\beta\|_\infty$, then the minimizing w is the zero vector.*

Proof. Consider $2^{-1}\lambda\|w\|_1 - \beta^Tw = \sum_{i=1}^p(\lambda/2|w_i| - \beta_i w_i)$. If $\lambda \geq 2\|\beta\|_\infty$, this expression is non-negative for all $w \in \mathbb{R}^p$ and a minimum occurs at $w = 0$. Since Q is positive semi-definite, $w^T\frac{1}{2}Qw$ is always non-negative with a minimum at $w = 0$. It follows that for $\lambda \geq 2\|\beta\|_\infty$ the sum of these terms attains minimum at $w = 0$. □

For a matrix $A \in \mathbb{R}^{n\times n}$, let A^- denote a generalized inverse.

Lemma 4. *Let $M = [(C\mathbf{K}C)^2 + n\gamma(C\mathbf{K}C)]^- C\mathbf{K}C$, then $\|M\|_{op} \leq (n\gamma)^{-1}$.*

Proof of Lemma 4. The kernel matrix $\mathbf{K}$ is positive semi-definite since by the reproducing property for any $\alpha \in \mathbb{R}^n$

$$\alpha^\top\mathbf{K}\alpha = \left\langle \sum_{i=1}^n \alpha_i\Phi(\mathbf{x}_i)\,, \sum_{i=1}^n \alpha_i\Phi(\mathbf{x}_i)\right\rangle_{\mathcal{H}} = \Big\| \sum_{i=1}^n \alpha_i\Phi(\mathbf{x}_i)\Big\|_{\mathcal{H}}^2 \geq 0.$$

It follows that $C\mathbf{K}C$ is also positive semi-definite. Let $\{\lambda_i\}_{i=1}^k$ be the set of non-zero eigenvalues of $C\mathbf{K}C$, then $\{\lambda_i/(\lambda_i^2 + n\gamma\lambda_i)\}_{i=1}^k$ are the non-zero eigenvalues of $M = [(C\mathbf{K}C)^2 + n\gamma(C\mathbf{K}C)]^- C\mathbf{K}C$. The function $t \mapsto t/(t^2 + n\gamma t)$ is bounded above by $(n\gamma)^{-1}$ for $t > 0$, hence $\|M\|_{\text{op}} \leq (n\gamma)^{-1}$. □

Lemma 5. *Let $\gamma > 0$. The minimizer $\widehat{f}$ in* (2.2) *satisfies $\|\widehat{f}\|_{\mathcal{H}} \leq 1/\sqrt{\gamma}$. Additionally, if Assumption 4 holds for $\kappa > 0$, then $\|\widehat{f}\|_{\mathcal{H}} \leq 2\kappa/\gamma$.*

Proof of Lemma 5. Comparing the value of objective function in (2.2) at $f = \widehat{f}$ with the value at $f = 0$ gives

$$\gamma\|\widehat{f}\|_{\mathcal{H}}^2 \leq \frac{1}{n}\sum_{i=1}^n \left| y_i^\top\widehat{\theta} - \left\langle \Phi(\mathbf{x}_i) - \overline{\Phi}, \widehat{f}\,\right\rangle_{\mathcal{H}}\right|^2 + \gamma\|\widehat{f}\|_{\mathcal{H}}^2 \leq \frac{1}{n}\sum_{i=1}^n |y_i^\top\widehat{\theta}|^2 = 1.,$$

where the last equality follows since $n^{-1}\widehat{\theta}Y^\top Y\widehat{\theta} = 1$. It follows that $\|\widehat{f}\|_{\mathcal{H}} \le 1/\sqrt{\gamma}$.

On the other hand, since $\widehat{f} = \sum_{i=1}^n \alpha_i(\Phi(\mathbf{x}_i) - \overline{\Phi})$, by the triangle inequality and Assumption 4

$$\|\widehat{f}\|_{\mathcal{H}} = \Big\| \sum_{i=1}^n \alpha_i(\Phi(\mathbf{x}_i) - \overline{\Phi}) \Big\|_{\mathcal{H}} \le \sum_{i=1}^n |\alpha_i| \|\Phi(\mathbf{x}_i) - \overline{\Phi}\|_{\mathcal{H}}$$
$$\le \max_i \|\Phi(\mathbf{x}_i) - \overline{\Phi}\|_{\mathcal{H}} \|\alpha\|_1 \le 2\kappa\|\alpha\|_1 \le 2\kappa\sqrt{n}\|\alpha\|_2.$$

Since $\alpha = \{(C\mathbf{K}C)^2 + \gamma n C\mathbf{K}C\}^{-}C\mathbf{K}CY\widehat{\theta}$, applying Lemma 4 and using $\|Y\widehat{\theta}\|_2 = \sqrt{\widehat{\theta}Y^\top Y\widehat{\theta}} = \sqrt{n}$ gives

$$\|\alpha\|_2 \le \|\{(C\mathbf{K}C)^2 + \gamma n C\mathbf{K}C\}^{-}C\mathbf{K}C\|_{\text{op}}\|Y\widehat{\theta}\|_2 \le \frac{\|Y\widehat{\theta}\|_2}{n\gamma} \le \frac{1}{\sqrt{n}\gamma}.$$

Combining the above two displays gives $\|\widehat{f}\|_{\mathcal{H}} \le 2\kappa/\gamma$. □

Lemma 6. *Under Assumptions 3 and 4, let $\{(x_i, y_i)\}_{i=1}^n$ and $\{(x_j, y_j)\}_{j=n+1}^{2n}$ be two independent copies of i.i.d. data, and let T_x be the empirical measure on their union. Let $\widetilde{R}_{emp}(f, \beta)$ be the modified empirical risk on $\{(x_i, y_i)\}_{i=1}^n$, and $\widetilde{R}'_{emp}(f, \beta)$ on $\{(x_j, y_j)\}_{i=n+1}^{2n}$. Let $c = 64(\|\theta^*\|_\infty + \tau\kappa)$, and let $\{f_1, \ldots, f_M\}$ be the smallest $L^2(T_x)$ $\varepsilon/\sqrt{2}c$-net of $\mathcal{H}_\tau$, and let $\{\beta_1, \ldots, \beta_K\}$ be an ε/c-net of I_τ. Then*

$$\mathbb{P}\Bigg(\sup_{\substack{f\in H_\tau \\ \beta\in I_\tau}} \{\widetilde{R}_{emp}(f,\beta) - \widetilde{R}'_{emp}(f,\beta)\} > \frac{\varepsilon}{2} \Bigg) \le \mathbb{P}\Bigg(\operatorname*{maximize}_{\substack{f\in\{f_1,\ldots,f_M\} \\ \beta\in\{\beta_1,\ldots,\beta_K\}}} \{\widetilde{R}_{emp}(f,\beta) - \widetilde{R}'_{emp}(f,\beta)\} > \frac{\varepsilon}{4} \Bigg).$$

Proof of Lemma 6. Let $f \in \mathcal{H}_\tau$, $\beta \in I_\tau$ be such that $\widetilde{R}_{\text{emp}}(f, \beta) - \widetilde{R}'_{\text{emp}}(f, \beta) > \varepsilon/2$. There exists $f_j \in \{f_1, \ldots, f_M\}$ and $\beta_\ell \in \{\beta_1, \ldots, \beta_K\}$ such that $\|f_j - f\|_{L^2(T_x)} < \varepsilon/\sqrt{2}c$ and $|\beta - \beta_\ell| < \varepsilon/c$. Applying Lemma 11 gives

$$\sqrt{\frac{1}{n}\sum_{i=1}^n |f(x_i) - f_j(x_i)|^2} < \frac{\varepsilon}{c} \quad \text{and} \quad \sqrt{\frac{1}{n}\sum_{i=n+1}^{2n} |f(x_i) - f_j(x_i)|^2} < \frac{\varepsilon}{c}.$$

Applying Lemma 10 yields

$$|\widetilde{R}_{\text{emp}}(f,\beta) - \widetilde{R}_{\text{emp}}(f_j,\beta_\ell)| < 8\frac{\varepsilon}{c}(\|\theta^*\|_\infty + \kappa\tau) = \frac{\varepsilon}{8},$$

and similarly $|\widetilde{R}'_{\text{emp}}(f,\beta) - \widetilde{R}'_{\text{emp}}(f_j,\beta_\ell)| < \varepsilon/8$. Therefore, $\widetilde{R}'_{\text{emp}}(f,\beta) - \widetilde{R}_{\text{emp}}(f,\beta) > \varepsilon/2$ for some $f \in \mathcal{H}_\tau$, $\beta \in I_\tau$ implies $\widetilde{R}'_{\text{emp}}(f_j,\beta_\ell) - \widetilde{R}_{\text{emp}}(f_j,\beta_\ell) > \varepsilon/4$ for some f_j and β_ℓ. Therefore,

$$\begin{aligned}&\mathbb{P}\Big(\sup_{f\in\mathcal{H}_\tau,\,\beta\in I_\tau}\{\widetilde{R}'_{\text{emp}}(f,\beta) - \widetilde{R}_{\text{emp}}(f,\beta)\} > \frac{\varepsilon}{2}\Big)\\ &\leq \mathbb{P}\Big(\underset{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}{\text{maximize}}\{\widetilde{R}'_{\text{emp}}(f_j,\beta_\ell) - \widetilde{R}_{\text{emp}}(f_j,\beta_\ell)\} > \frac{\varepsilon}{4}\Big).\end{aligned}$$

□

Lemma 7. *Under Assumptions 3-5, let $\{f_1,\ldots,f_M\}$ and $\{\beta_1,\ldots,\beta_K\}$ be as in Lemma 6. There exist a constant $C_1 > 0$ such that for all $\varepsilon > 0$,*

$$\mathbb{P}\Big(\underset{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}{\text{maximize}}\{\widetilde{R}_{emp}(f,\beta) - \widetilde{R}'_{emp}(f,\beta)\} > \frac{\varepsilon}{4}\Big) \leq \mathcal{N}_\varepsilon \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big),$$

where $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\}\exp(C_1\tau^2\varepsilon^{-2})$.

Proof of Lemma 7. Let $\sigma = \{\sigma_i\}_{i=1}^n$ be $i.i.d.$ Radamacher random variables, $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$. Let

$$\widetilde{R}^\sigma_{\text{emp}} = \frac{1}{n}\sum_{i=1}^n \sigma_i|y_i^\top\theta^* - \beta - \langle\Phi(\mathbf{x}_i), f\rangle_{\mathcal{H}}|^2, \quad \widetilde{R}'^\sigma_{\text{emp}} = \frac{1}{n}\sum_{i=n+1}^{2n} \sigma_i|y_i^\top\theta^* - \beta - \langle\Phi(\mathbf{x}_i), f\rangle_{\mathcal{H}}|^2.$$

Since (y_i, x_i) and (y_{n+i}, x_{n+i}) are independent, and have the same distribution, the distribution of $\xi_i := (|y_i^\top\theta^* - \beta - \langle\Phi(x_i), f\rangle_{\mathcal{H}}|^2 - |y_{n+i}^\top\theta^* - \beta - \langle\Phi(x_{n+i}), f\rangle_{\mathcal{H}}|^2)$ is the same as distribution of

$\sigma_i\xi_i$. Let $Z = \{(x_i, y_i)\}_{i=1}^{2n}$, then

$$\begin{aligned}
&\mathbb{P}_Z\Big(\max_{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}\{\widetilde{R}_{\text{emp}}(f,\beta) - \widetilde{R}'_{\text{emp}}(f,\beta)\} > \frac{\varepsilon}{4}\Big)\\
&\quad= \mathbb{P}_{Z,\sigma}\Big(\max_{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}\{\widetilde{R}^{\sigma}_{\text{emp}}(f,\beta) - \widetilde{R}'^{\sigma}_{\text{emp}}(f,\beta)\} > \frac{\varepsilon}{4}\Big).
\end{aligned}$$

Let $\mathcal{A}_{m,k}$ be the event $\mathcal{A}_{m,k} = \{\widetilde{R}^{\sigma}_{\text{emp}}(f_m,\beta_k) - \widetilde{R}'^{\sigma}_{\text{emp}}(f_m,\beta_k) > \varepsilon/4\}$ for $m = 1,\ldots,M(Z)$; $k = 1,\ldots,K$; where $M(Z)$ emphasizes the dependence of M on Z. Using properties of conditional expectation and union bound

$$\begin{aligned}
\mathbb{P}_{Z,\sigma}\Big(\max_{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}\{\widetilde{R}^{\sigma}_{\text{emp}}(f,\beta) - \widetilde{R}'^{\sigma}_{\text{emp}}(f,\beta)\} > \frac{\varepsilon}{4}\Big) &= \mathbb{P}_{Z,\sigma}(\cup_{m=1}^{M(Z)}\cup_{k=1}^{K}\mathcal{A}_{m,k})\\
&= \mathbb{E}_Z\Big\{\mathbb{P}_\sigma(\cup_{m=1}^{M(Z)}\cup_{k=1}^{K}\mathcal{A}_{m,k}|Z)\Big\}\\
&\le \mathbb{E}_Z\left\{M(Z)K\mathbb{P}_\sigma(\mathcal{A}_{m,k}|Z)\right\}.
\end{aligned}$$

For fixed f_m, β_k and conditionally on Z, the terms $\psi_i := \sigma_i(|y_i^\top\theta^* - \beta_k - \langle\Phi(x_i), f_m\rangle_{\mathcal{H}}|^2 - |y_{n+i}^\top\theta^* - \beta_k - \langle\Phi(x_{n+i}), f_m\rangle_{\mathcal{H}}|^2)$, $i = 1,\ldots,n$, are independent, mean-zero random variables with $|\psi_i| \le 4(\|\theta^*\|_\infty + \kappa\tau)^2$. Applying Hoeffding's inequality gives

$$\mathbb{P}_\sigma(\mathcal{A}_{m,k}|Z) = \mathbb{P}_\sigma\Big(\frac{1}{n}\sum_{i=1}^{n}\psi_i > \varepsilon/4\,\Big|\,Z\Big) \le \exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big).$$

On the other hand, since I_τ is a one-dimensional sphere of radius $\|\theta^*\| + \kappa\tau$, K is independent of the data and $K \le 1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon$. Combining this with the above two displays gives

$$\begin{aligned}
&\mathbb{P}_{Z,\sigma}\Big(\max_{\substack{f\in\{f_1,\ldots,f_M\}\\ \beta\in\{\beta_1,\ldots,\beta_K\}}}\{\widetilde{R}^{\sigma}_{\text{emp}}(f,\beta) - \widetilde{R}'^{\sigma}_{\text{emp}}(f,\beta)\} > \frac{\varepsilon}{4}\Big)\\
&\quad\le \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\}\,\mathbb{E}_Z\{M(Z)\}\exp\Big(-\frac{n\varepsilon^2}{128(\|\theta^*\|_\infty + \kappa\tau)^4}\Big).
\end{aligned}$$

Recall that $\{f_1,\ldots,f_M\}$ is the smallest $L^2(T_x)$ $\varepsilon/\sqrt{2}c$-net of $\mathcal{H}_\tau$, with $c = 64(\|\theta^*\|_\infty + \tau\kappa)$.

By Lemma 12

$$\mathbb{E}_Z\{M(Z)\} \leq \sup_{Z=\{(x_i,y_i)\}_{i=1}^{2n}} M(Z) \leq \exp\left(\frac{C_1\tau^2}{\varepsilon^2}\right) \tag{2.15}$$

for some constant $C_1 > 0$. Setting $\mathcal{N}_\varepsilon = \{1 + 2(\|\theta^*\|_\infty + \kappa\tau)/\varepsilon\}\exp(C_1\tau^2\varepsilon^{-2})$ completes the proof of Lemma 7. □

Lemma 8. *Under Assumption 3 there exist constants $C_1, C_2 > 0$ such that for all $\varepsilon > 0$,*

$$\mathbb{P}\biggl(\max\Big(|n_1/n_2 - \pi_1/\pi_2|, |n_2/n_1 - \pi_2/\pi_1|\Big) > \varepsilon\biggr) \leq C_1\exp\Big(-C_2n\varepsilon^2\Big),$$

$$\mathbb{P}\biggl(\max\Big(|\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2}|, |\sqrt{n_2/n_1} - \sqrt{\pi_2/\pi_1}|\Big) > \varepsilon\biggr) \leq C_1\exp\Big(-C_2n\varepsilon^2\Big).$$

Proof of Lemma 8. We provide the proof for n_1/n_2, the proof for n_2/n_1 is analogous. The first inequality is equivalent to Lemma 1 in [48]. For the second inequality, by Taylor expansion of the square root function centered at π_1/π_2

$$\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2} = 2^{-1}\sqrt{\pi_2/\pi_1}(n_1/n_2 - \pi_1/\pi_2) + o(n_1/n_2 - \pi_1/\pi_2).$$

Since $|n_1/n_2 - \pi_1/\pi_2| = O_p(n^{-1/2})$ by the first inequality, it follows that there exist a constant $C_3 > 0$ such that $|\sqrt{n_1/n_2} - \sqrt{\pi_1/\pi_2}| \leq C_2\{\log(\eta^{-1})/n\}^{1/2}$ with probability at least $1-\eta$. Setting $\varepsilon = C_3\{\log(\eta^{-1})/n\}^{1/2}$ and solving for η completes the proof. □

Lemma 9. *Let Assumption 3 be true. For all $\varepsilon > 0$, we have*

$$\mathbb{P}\big((\overline{Y\theta^*})^2 > \varepsilon\big) \leq 2\exp(-n\varepsilon/\|\theta^*\|_\infty).$$

Proof of Lemma 9. Let $z_i = y_i^\top\theta^*$, then z_i are independent,

$$\mathbb{E}(z_i) = \mathbb{E}(y_i)^\top\theta^* = \pi_1\sqrt{\frac{\pi_2}{\pi_1}} - \pi_2\sqrt{\frac{\pi_1}{\pi_2}} = \sqrt{\pi_1\pi_2} - \sqrt{\pi_1\pi_2} = 0$$

and

$$(\overline{Y\theta^*})^2 = (n^{-1}\sum_{i=1}^n y_i^\top \theta^*)^2 = (n^{-1}\sum_{i=1}^n z_i)^2.$$

Since $|z_i| \leq \|\theta^*\|_\infty = \sqrt{\pi_{\max}/\pi_{\min}}$, by Hoeffding's inequality for $\varepsilon > 0$

$$\mathbb{P}\Big(\Big|n^{-1}\sum_{i=1}^n z_i\Big|^2 > \varepsilon\Big) = \mathbb{P}\Big(\Big|n^{-1}\sum_{i=1}^n z_i\Big| > \sqrt{\varepsilon}\Big) \leq 2\exp(-n\varepsilon/\|\theta^*\|_\infty).$$

□

Lemma 10. *Let Assumptions 3 and 4 be true, and suppose that $\{f_1, \dots, f_M\}$ is an $L^2(T_x)$ ε-net of $\mathcal{H}_\tau$ and that $\{\beta_1, \dots, \beta_K\}$ be an ε-net of I_τ. Then for any admissible f and β, let f_j and β_ℓ be members of the ε-nets so that $\|f - f_j\|_{L^2(T_x)} < \varepsilon$ and $|\beta - \beta_\ell| < \varepsilon$. Then*

$$\Big|\widetilde{R}_{emp}(f,\beta) - \widetilde{R}_{emp}(f_j,\beta_l)\Big| \leq 8\varepsilon\Big(\|\theta^*\|_\infty + \kappa\tau\Big). \tag{2.16}$$

Proof of Lemma 10. By the reproducing property of $\mathcal{H}$, $\langle \Phi(\mathbf{x}_i), f\rangle_{\mathcal{H}} = f(x_i)$, and

$$\begin{aligned}
&\Big|\widetilde{R}_{emp}(f,\beta) - \widetilde{R}_{emp}(f_j,\beta_l)\Big| \\
&= \Big|\frac{1}{n}\sum_{i=1}^n |y_i^\top\theta^* - \beta - \langle\Phi(\mathbf{x}_i), f\rangle_{\mathcal{H}}|^2 - \frac{1}{n}\sum_{i=1}^n |y_i^\top\theta^* - \beta_\ell - \langle\Phi(\mathbf{x}_i), f_j\rangle_{\mathcal{H}}|^2\Big| \\
&= \Big|\frac{1}{n}\sum_{i=1}^n |y_i^\top\theta^* - \beta - f(x_i)|^2 - \frac{1}{n}\sum_{i=1}^n |y_i^\top\theta^* - \beta_\ell - f_j(x_l)|^2\Big| \\
&= \Big| - 2\frac{1}{n}\sum_{i=1}^n y_i^\top\theta^*\{\beta + f(x_i) - \beta_\ell - f_j(x_i)\} + \frac{1}{n}\sum_{i=1}^n[\{\beta + f(x_i)\}^2 - \{\beta_\ell + f_j(x_i)\}^2]\Big| \\
&\leq \underbrace{2\|\theta^*\|_\infty\Big|\beta - \beta_l + \frac{1}{n}\sum_{i=1}^n\{f(x_i) - f_j(x_i)\}\Big|}_{I_1} + \underbrace{\Big|\frac{1}{n}\sum_{i=1}^n[\{\beta + f(x_i)\}^2 - \{\beta_\ell + f_j(x_i)\}^2]\Big|}_{I_2}.
\end{aligned}$$

Consider

$$I_1 = 2\|\theta^*\|_\infty \left|\beta - \beta_l + \frac{1}{n}\sum_{i=1}^{n}\{f(x_i) - f_j(x_i)\}\right| \leq 2\|\theta^*\|_\infty \left\{|\beta - \beta_l| + \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - f_j(x_i)|\right\}$$
$$\leq 2\|\theta^*\|_\infty \left\{\varepsilon + \left[\frac{1}{n}\sum_{i=1}^{n}|f(x_i) - f_j(x_i)|^2\right]^{1/2}\right\}$$
$$\leq 4\|\theta^*\|_\infty \varepsilon,$$

where we used $n^{-1}\sum_{i=1}^{n}[|f(x_i) - f_j(x_i)|^2]^{1/2} \leq [n^{-1}\sum_{i=1}^{n}|f(x_i) - f_j(x_i)|^2]^{1/2}$ due to Jensen's inequality, and that $\|f - f_j\|_{L^2(T_x)} < \varepsilon$ and $|\beta - \beta_\ell| < \varepsilon$.

Consider I_2. Using $a^2 - b^2 = (a+b)(a-b)$, the Cauchy-Schwarz inequalty, and Jensen's inequality,

$$\begin{aligned}
I_2 &= \frac{1}{n}\left|\sum_{i=1}^{n}\{\beta + f(x_i) + \beta_\ell + f_j(x_i)\}\{\beta - \beta_\ell + f(x_i) - f_j(x_i)\}\right| \\
&\leq 2(\sup_{\beta \in I_\tau}|\beta| + \sup_{x, f \in \mathcal{H}_\tau}|f(x)|)\frac{1}{n}\sum_{i=1}^{n}(|\beta - \beta_j| + |f(x_i) - f_j(x_i)|) \\
&\leq 2(\|\theta^*\|_\infty + \kappa\tau + \sup_{x, f \in \mathcal{H}_\tau}|\langle \Phi(\mathbf{x}), f\rangle_{\mathcal{H}}|)(\varepsilon + \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - f_j(x_i)|) \\
&\leq 2\Big(\|\theta^*\|_\infty + \kappa\tau + \kappa\tau\Big)\Big(\varepsilon + \sqrt{\frac{1}{n}\sum_{i=1}^{n}|f(x_i) - f_j(x_i)|^2}\Big) \\
&= 4\varepsilon\Big(\|\theta^*\|_\infty + 2\kappa\tau\Big).
\end{aligned}$$

Combining the bounds for I_1 and I_2 completes the proof of Lemma 10. □

Lemma 11. *Let $\{(x_i, y_i)\}_{i=1}^{2n}$ be the data, and consider an $L^2(T_x)$ ε-net $\{f_1, \ldots, f_M\}$ of $\mathcal{H}_\tau$. Then $\{f_1, \ldots, f_M\}$ is an $\sqrt{2}\varepsilon$-net with respect to the empirical measure on half of the data $\{(x_i, y_i)\}_{i=1}^{n}$.*

Proof of Lemma 11. Since $\{f_1, \ldots, f_M\}$ is ε-net with respect to $\{(x_i, y_i)\}_{i=1}^{2n}$, for any $f \in \mathcal{H}_\tau$, there exists f_j such that

$$\sqrt{\frac{1}{2n}\sum_{i=1}^{2n}|f(x_i) - f_j(x_i)|^2} < \varepsilon.$$

If $\frac{1}{2n}\sum_{i=1}^{2n}|f(x_i)-f_j(x_i)|^2=0$, then $\frac{1}{n}\sum_{i=1}^{n}|f(x_i)-f_j(x_i)|^2=0$. Otherwise

$$\begin{aligned}\sqrt{\frac{1}{n}\sum_{i=1}^{n}|f(x_i)-f_j(x_i)|^2} &= \sqrt{\frac{2n}{2n}\frac{1}{n}\sum_{i=1}^{n}|f(x_i)-f_j(x_i)|^2\frac{\sum_{i=1}^{2n}|f(x_i)-f_j(x_i)|^2}{\sum_{i=1}^{2n}|f(x_i)-f_j(x_i)|^2}} \\ &= \sqrt{\frac{2n}{n}\frac{\sum_{i=1}^{n}|f(x_i)-f_j(x_i)|^2}{\sum_{i=1}^{2n}|f(x_i)-f_j(x_i)|^2}}\sqrt{\frac{1}{2n}\sum_{i=1}^{2n}|f(x_i)-f_j(x_i)|^2} < \sqrt{2}\varepsilon,\end{aligned}$$

hence $\{f_1,\ldots,f_M\}$ is $\sqrt{2}\varepsilon$-net with respect to $\{(x_i,y_i)\}_{i=1}^n$. □

Lemma 12 (Theorem 2.1 of [49])**.** *Let Assumption 5 be true, and Let $M(Z)$ be the size of an $L^2(T_x)$ ε-covering number of $\mathcal{H}_\tau$ with data $Z=\{(x_i,y_i)\}_{i=1}^n$. There exists a $C>0$ independent of n, such that*

$$\sup_{Z=\{(x_i,y_i)\}_{i=1}^n} M(Z) \le \exp\left(\frac{C\tau^2}{\varepsilon^2}\right). \tag{2.17}$$

Remark 5. *[50] notes that “Theorem 2.1 of [49] considered only the Gaussian RKHS, however the proof of the entropy bound for p = 2 in their notation only requires that the RKHS is separable.” It is this case which is presented in Lemma 12.*

3. COMPRESSING LARGE SAMPLE DATA FOR DISCRIMINANT ANALYSIS

3.1 Introduction

Linear Discriminant Analysis (LDA) [33] is a linear classification rule which separates the classes by maximizing between-class variability compared to within-class variability. Applying LDA requires constructing the within-class covariance matrix, which has complexity $O(n\,p^2)$ in the number of training samples n and number of features p. As large-sample data acquisition became prevalent, it became computationally expensive to apply LDA to such data even for moderately-sized p.

Compression [51, 52, 53, 54, 55], or sketching, is a popular approach for scaling algorithms to large data. Given the training data $X \in \mathbb{R}^{n\times p}$, compression uses a random matrix Q to either reduce the number of rows (samples) or columns (features) in X. The corresponding reduced-size QX or XQ is called a sketch of the original X. The sketch is used in place of X to approximate the solution of the full algorithm. For example, compression is used in least-squares regression [56, 55]; non-negative least-squares regression [51]; ridge regression [57, 58] and ℓ^1-penalized regression [59]. Compression for a broader class of convex minimization problems is considered in [53].

Despite the widespread use of compression in regression contexts, and considerable progress in theoretical understanding of its performance in regression, compression has not been widely used in discriminant analysis. Specifically, existing works on compression in LDA [60, 61] focus on reducing the number of features p, and thus do not consider the case where the computational bottleneck is due to the large number of samples n. On the other hand, existing results on compression due to large n in the regression literature [57, 58] can not be applied to discriminant analysis. In regression, the training data $X \in \mathbb{R}^{n\times p}$ is treated as fixed, with continuous response $Y \in \mathbb{R}^n$ modelled conditionally on X. In contrast, in discriminant analysis the observations in $X \in \mathbb{R}^{n\times p}$ are treated as random, and are modelled conditionally on the discrete class membership

$Y \in \{1, 2\}^n$. Thus, the theoretical analysis of compression in LDA requires different techniques than for regression.

In this Chapter, we address these challenges and bridge the existing gap between compression with large n in regression and compression with large n in discriminant analysis. This Chapter makes the following contributions:

- We develop a new method, Compressed LDA, for large sample data that is based on *separate* compression within each class in contrast to joint compression of existing approaches [62];
- We derive a finite-sample bound on misclassification error rate of Compressed LDA compared to the optimal error rate of the Bayes classifier;
- We extend Compressed LDA to the setting with unequal class covariance matrices leading to Compressed Quadratic Discriminant Analysis (QDA) [33], to our knowledge this it the first method that considers compression within the QDA context;
- We demonstrate significant computational advantages of our methods compared to discriminant analysis on the full data and their superior classification performance compared to methods based on random sub-sampling or joint compression [62].

3.1.1 Related Works

Existing works on compression in LDA [60, 61] focus on reducing the number of features p, and thus do not consider the case where the computational bottleneck is due to the large number of samples n. To our knowledge, the only exception is the Fast Random Fisher Discriminant Analysis (FRF) of [62].

In [62], the authors use joint compression of classes to form a sketch $QX \in \mathbb{R}^{m \times p}$, $m \ll n$, via a random matrix $Q \in \mathbb{R}^{m \times n}$, and then use the sketch within the generalized eigenvalue formulation of LDA to form the approximate discriminant vector $\beta_{\mathbf{c}} \in \mathbb{R}^p$. The discriminant vector is applied to form the projected training data $\beta_{\mathbf{c}}^\top \mathbf{x}_i \in \mathbb{R}$, which is used to train LDA instead of original $\mathbf{x}_i \in \mathbb{R}^p$. The m compressed samples in $QX \in \mathbb{R}^{m \times p}$ are thus only used to form β_c. This is because these m samples can not be assigned class labels, as multiplication by Q allows mixing of both classes. Furthermore, due to this mixing, it is not possible to form class-specific covariance matrices based on compressed samples in QX, and thus the method of [62] cannot be extended to QDA. In contrast, our method applies separate class compression, not only allowing an extension to QDA, but also leading to significantly better empirical performance (in terms of both lower error rate and lower variance).

Another difference between this Chapter and the work of [62] is the corresponding theoretical analysis. In [62], the authors compare the compressed discriminant vector $\beta_{\mathbf{c}}$ to the discriminant vector $\widehat{\beta}$ based on the full data by deriving the bound on the difference of projection values $|(\mathbf{x}-\overline{X})^\top(\beta_{\mathbf{c}}-\widehat{\beta})|$, where $\overline{X}=n^{-1}\sum_{i=1}^{n}\mathbf{x}_i$ is the training sample mean and $\mathbf{x}\in\mathbb{R}^p$ is a random test sample. It is unclear, however, whether this bound directly translates into a similar difference in misclassficiation error rates, which is a more natural loss within a classification context. Furthermore, since the bound is provided with respect to $\widehat{\beta}$ rather than the true population β^*, it is unclear how the performance of the method of [62] compares to the performance of the Bayes classifier. In contrast, we directly analyze the misclassification error rate of the proposed Compressed LDA method, and derive a finite-sample bound on its rate compared to the Bayes classifier.

In the regression literature on compression, the quality of the compressed solution β_c is typically evaluated either by bounding mean-squared error compared to the underlying true parameter vector β^* [58], or by considering the ε-optimality. Let f be the objective function that is minimized within the given algorithm (e.g. standard least-squares, ℓ_1-penalized least-squares, etc.) over some subset S of $\mathbb{R}^p$, where the function f is based on the full training data. The compressed solution $\beta_{\mathbf{c}}$ is said to be ε-optimal [63, 55] if

$$\min_{\beta\in S} f(\beta)\leq f(\beta_{\mathbf{c}})\leq(1+\varepsilon)^2\min_{\beta\in S} f(\beta).$$

While ε-optimality is natural in a regression context, where the loss in the objective function represents the sample average of targeted population loss, LDA solves a generalized eigenvalue problem rather than directly minimizing the misclassification error rate. Thus, bounding the misclassification error rate of Compressed LDA directly in terms of the Bayes error rate provides a more direct answer regarding its theoretical performance, and it is consistent with results in the LDA literature without compression [64, 65, 66].

Another sample size reduction method outside of compression is squashing [67, 68, 69], which partitions the n training samples into d distinct segments, calculates a fixed number of moments k for each segment, and then generates a smaller number of new samples within each segment preserving the corresponding original moments. Each new sample comes with a weight that accounts for a possible discrepancy between the distribution of samples across segments in the original data and the distribution of samples across segments in the new data. Because of the weights, one can not simply apply LDA to the new "squashed" data, as the weights will need to be included to modify the estimation algorithm. Furthermore, while squashing reduces the number of training samples, its computational complexity depends on the number of partitions d, number of calculated moments k, and the number of newly-generated samples. Since partitioning the data may lead to an exponential number of segments d in the number of features p, applying squashing in LDA context may be more computationally expensive than training LDA on the full data, and thus we do not pursue this approach here.

3.1.2 Notation

For a vector $v \in \mathbb{R}^p$, we let $\|v\|_2$ be the Euclidean norm $\sqrt{\sum_{i=1}^{p} |v_i|^2}$. For a matrix $M \in \mathbb{R}^{k\times p}$, we let $M_{i,j}$ be its (i,j)-th element, $\|M\|_{\text{op}} = \sup_{\|v\|_2\leq 1} \|Mv\|_2$ be its operator norm, and $\|M\|_F = \sqrt{\sum_{i,j} |M_{i,j}|^2}$ be the Frobenius norm. For a random variable Z, we let $\|Z\|_{\Psi_2} = \inf\{t > 0 : \mathbb{E}\exp(Z^2/t^2) \leq 2\}$ be its sub-Gaussian norm and $\|Z\|_{\Psi_1} = \inf\{t > 0 : \mathbb{E}\exp(|Z|/t) \leq 2\}$ its sub-Exponential norm. We use $\Phi(\cdot)$ and $\phi(\cdot)$ to denote the cdf and the pdf of the standard normal distribution, respectively.

3.2 Compressed LDA

Our goal is to reduce the computational complexity of LDA while maintaining its classification performance. To achieve this, we propose to separately compress each class of training data $X^g \in \mathbb{R}^{n_g \times p}$ via a sparse Rademacher matrix $Q^g \in \mathbb{R}^{m_g \times n_g}$ as defined below.

Definition 4. *A matrix $Q^g \in \mathbb{R}^{m_g \times n_g}$ is a* sparse Rademacher *matrix with parameter $s \in (0,1)$ if the elements $Q^g_{j,k}$ are i.i.d. with distribution*

$$\mathbb{P}(Q^g_{j,k} = 1) = \mathbb{P}(Q^g_{j,k} = -1) = \frac{s}{2}, \, \mathbb{P}(Q^g_{j,k} = 0) = 1 - s.$$

Definition 5. *The j-th* compressed data sample *in class g is*

$$\mathbf{x}^g_{j,\mathbf{c}} = \frac{1}{\sqrt{n_g\, s}} \sum_{i=1}^{n_g} Q^g_{j,i}(\mathbf{x}^g_i - \overline{X}_g) + \overline{X}_g, \tag{3.1}$$

where $Q^g_{j,i}$ are entries of the sparse Rademacher matrix $Q^g \in \mathbb{R}^{m_g \times n_g}$ of Definition 4.

Definition 6. *The compressed within-class sample covariance matrix $\widehat{\Sigma}_{w,\mathbf{c}} \in \mathbb{R}^{p \times p}$ is defined as the within-class sample covariance matrix of the compressed $\mathbf{x}^g_{j,\mathbf{c}}$*

$$\widehat{\Sigma}_{w,\mathbf{c}} := \frac{1}{m} \sum_{g=1}^{2} \sum_{j=1}^{m_g} (\mathbf{x}^g_{j,\mathbf{c}} - \overline{X}_g)(\mathbf{x}^g_{j,\mathbf{c}} - \overline{X}_g)^\top. \tag{3.2}$$

The compressed discriminant vector *is $\beta_\mathbf{c} := \widehat{\Sigma}^{-1}_{w,\mathbf{c}} d$, where d is defined as in* (1.3).

The proposed Compressed LDA classifies a new $\mathbf{x} \in \mathbb{R}^p$ as in (1.4), with $\widehat{\beta}$ and $\widehat{\Sigma}_w$ replaced by $\beta_\mathbf{c}$, and $\widehat{\Sigma}_{w,\mathbf{c}}$. Algorithm 1 summarizes the full workflow for Compressed LDA.

Our proposed compression scheme is analogous to partial compression within the compressed regression literature, see e.g. Section 2.1 of [58]. Given the matrix of covariates $X \in \mathbb{R}^{n \times p}$ and response $Y \in \mathbb{R}^n$, partial compression calculates the inner-product $X^\top Y$ on the full data and only uses compression to approximate $X^\top X$. The rationale is that calculating $X^\top Y$ only has complexity $O(n\, p)$ compared to complexity $O(n\, p^2)$ for calculating $X^\top X$. Similarly in discriminant

```
Input  : $X \in \mathbb{R}^{n\times p}$, $Y \in \{1,2\}^n$, $s \in (0,1)$, $m \ll n$.
Output: Compressed discriminant vector $\beta_{\mathbf{c}}$.
Compute $\overline{X}_g$, $g = 1,2$, and $d$ as in (1.3).
Set $m_g = \lfloor n_g m/n \rfloor$, $g = 1,2$.
Form compressed samples $\mathbf{x}^g_{j,\mathbf{c}}$ in (3.1), $j = 1,\dots,m_g$, $g = 1,2$.
Form $\widehat{\Sigma}_{w,\mathbf{c}} \in \mathbb{R}^{p\times p}$ as in (3.2).
Set $\beta_{\mathbf{c}} = \widehat{\Sigma}^{-1}_{w,\mathbf{c}} d$.
Use $\beta_{\mathbf{c}}$, $\widehat{\Sigma}_{w,\mathbf{c}}$ in rule (1.4) instead of $\widehat{\beta}$, $\widehat{\Sigma}_w$.
return $\beta_{\mathbf{c}}$
```

Algorithm 2: Compressed LDA

analysis, calculating d on the full data only has complexity $O(n\,p)$, whereas calculating $\widehat{\Sigma}_w$ has complexity $O(n\,p^2)$, and thus we only use compression to approximate the latter term.

The proposed compression scheme has several advantages. First, by compressing the classes individually, we are able to unambiguously assign labels to the compressed samples, thus allowing us to form the compressed within-class covariance matrix. This is not possible with the method of [62], which allows mixing samples from both classes in one compressed sample. Secondly, using sparse Radamacher compression matrices leads to both memory and computational advantages compared to e.g. random Gaussian compression matrices. Due to sparsity, the average complexity of data compression (3.1) is $O(nmps)$ rather than $O(nmp)$ for dense matrices. Thus, the overall average complexity of data compression and construction of $\widehat{\Sigma}_{w,\mathbf{c}}$ is $O(nmps + mp^2)$ compared to the complexity $O(np^2)$ of LDA on the full data. Choosing m and s so that $ms << p$ ensures that Compressed LDA is faster than full LDA. The computational costs of compression (3.1) can be further reduced by parallelizing the construction of $Q^g X^g$.

3.3 Error bound of Compressed LDA

In this section we derive a bound on the misclassification error rate of Compressed LDA compared to the optimal rate of the Bayes classifier. To our knowledge, this is the first such result for a sample compression method within the discriminant analysis framework. We use Assumption 1, which is standard for LDA (see e.g. [2, Section 11]).

We next define the Bayes classifier, which gives the optimal (minimal) error rate under Assumption 1.

Definition 7. *Under Assumption 1, and for equal prior class probabilities* $\pi_1 = \pi_2$*, the* Bayes decision rule *classifies* $\mathbf{x} \in \mathbb{R}^p$ *to class* 1 *if and only if* $\delta^\top \Sigma_w^{-1}(\mathbf{x} - \mu) \geq 0$*, where* $\delta = (\mu_1 - \mu_2)/2$*, and* $\mu = (\mu_1 + \mu_2)/2$*.*

The corresponding optimal misclassification error rate is given by [2, Chapter 11.6]

$$R_{\text{opt}} := \Phi(-\sqrt{\delta^\top \Sigma_w^{-1} \delta}). \tag{3.3}$$

We consider the case of equal prior class probabilities for clarity of technical derivations, which focus on the effects of compression. For the same reason, we assume equality of class sizes and their corresponding compression dimensions.

Assumption 6. $n_1 = n_2 = n/2$ *and* $m_1 = m_2 = m/2$*.*

These assumptions can be relaxed at the expense of more technical proofs without affecting the resulting rates, e.g. Hoeffding inequality bounds n_g/n in terms of π_g with rate $O(n^{-1/2})$. As our main focus is the effect of compression, we do not pursue this relaxation here.

We next bound the misclassification error rate of the proposed Compressed LDA in Section 3.2 in terms of the optimal rate R_{opt} in (3.3). Under Assumption 6, the Compressed LDA rule assigns new $\mathbf{x}$ to class 1 if and only if $d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \geq 0$. Under Assumptions 1-6, by [65, Section 2], the corresponding error rate of Compressed LDA is given by

$$R_{\mathbf{c}} = \frac{1}{2}\sum_{g=1}^{2} \Phi\bigg(\frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\{(-1)^g(\mu_g - \overline{X}_g) - d\}}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\, \Sigma_w\, \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\bigg). \tag{3.4}$$

We now state our main result, which compares the misclassification error rate R_c of Compressed LDA to the optimal rate R_{opt}. Since our focus is on the large sample data, we treat p as fixed which implies that $R_{\text{opt}} > 0$ is bounded from below, and hence $R_{\mathbf{c}} - R_{\text{opt}} \to 0$ implies $R_{\mathbf{c}}/R_{\text{opt}} \to 1$. We state our result in the latter form below.

Theorem 10. *Under Assumptions 1 and 6, and for $\pi_1 = \pi_2$, there exists an absolute constant $C > 0$ such that with probability at least $1 - \eta$,*

$$|R_{\mathbf{c}} - R_{opt}| \leq C\, P\, K_s^2 \sqrt{\frac{\log(\eta^{-1}) + p}{m}},$$

where $P = \phi(\sqrt{\delta^\top \Sigma_w^{-1} \delta})\,(\sqrt{\delta^\top \Sigma_w^{-1} \delta} + 1)$, and $K_s^2 = [s \log\left(1 + s^{-1}\right)]^{-1}$.

The upper bound depends on the sparsity level s through K_s, which appears in the proofs as the sub-Gaussian norm of the elements of $Q^g/\sqrt{s}$ (see Lemma 18 in Section 3.10). As $s \to 0$, fewer training samples are used when forming each compressed sample, and the upper bound of Theorem 10 increases. As $s \to 1$, more training samples are included, and the upper bound decreases. However, as s increases so does the run time for Compressed LDA. Thus, there is a trade-off between accuracy and speed determined by s.

Existing results in the LDA literature (i.e. [65]) have error rates $\mathcal{O}_p(n^{-1/2})$. Since Compressed LDA reduces the sample size to m, the rate $\mathcal{O}_p(m^{-1/2})$ in Theorem 10 is expected. While the decay rate is typical, our theoretical approach is not. The main difficulty in analyzing Compressed LDA is dependency across m compressed samples as (i) they share the sample class mean $\overline{X}_g$, and (ii) different rows of the compression matrix Q^g can share the location of non-zero entries, and thus the same $\mathbf{x}_i^g$ may appear in (3.1) for different values of j. To overcome these difficulties, we use independence between the compression matrices Q^g and original data matrices X^g when bounding the difference between $\widehat{\Sigma}_{w,\mathbf{c}}$ and Σ_w. The detailed proof of Theorem 10, as well as supplementary Theorems and Lemmas, are presented in Sections 3.9 and 3.10.

Finally, while the scaling $\mathcal{O}_p(m^{-1/2})$ in Theorem 10 is the same as what would be expected under sub-sampling (randomly selecting $m/2$ samples from each class and discarding the rest), we found that empirically compression offers two advantages: (i) it has the smaller misclassification error rate variance (see e.g. Figures 3.1-3.3), which is likely due to using multiple $\mathbf{x}_i^g$ in forming each compressed sample; (ii) it is more robust to violations of normality assumption in the original data as the summation within (3.1) induces normality of compressed samples (see Figure 3.6).

3.4 Extensions

3.4.1 Projected LDA

The Compressed LDA proposed in Section 3.2 proceeds by (i) forming a discriminant vector $\beta_{\mathbf{c}}$ based on compressed samples in (3.1); (ii) using $\beta_{\mathbf{c}}$ and compressed within-class sample covariance matrix $\widehat{\Sigma}_{w,\mathbf{c}}$ in classification rule (1.4). An alternative approach is to use step (i) only, project the original training data using $\beta_{\mathbf{c}}$ to form $\mathbf{z}_i^g = \beta_{\mathbf{c}}^\top \mathbf{x}_i^g \in \mathbb{R}$, and then apply LDA on the pairs $\{\mathbf{z}_i, y_i\}$, where now the samples $\mathbf{z}_i$ are one-dimensional scalars rather than p-dimensional vectors. Thus, the within-class variance of the projected data $\beta_{\mathbf{c}}^\top \widehat{\Sigma}_w \beta_{\mathbf{c}}$ is used in decision rule (1.4) rather than $\beta_{\mathbf{c}}^\top \widehat{\Sigma}_{w,\mathbf{c}} \beta_{\mathbf{c}}$. We call this alternative approach Projected LDA. If the two classes have equal sample sizes, that is Assumption (6) holds, Compressed LDA and Projected LDA rules coincide as both will classify a new $\mathbf{x}$ according to

$$\underset{g=1,2}{\operatorname{argmin}}\{(\mathbf{x} - \overline{X}_g)^\top \beta_{\mathbf{c}}\}^2.$$

However, if $n_1 \neq n_2$, the two methods will in general differ due to discrepancy between $\beta_{\mathbf{c}}^\top \widehat{\Sigma}_w \beta_{\mathbf{c}}$ and $\beta_{\mathbf{c}}^\top \widehat{\Sigma}_{w,\mathbf{c}} \beta_{\mathbf{c}}$.

The Projected LDA is analogous to the Fast Random Fisher Discriminant Analysis proposed in [62]: both use compression to form the discriminant vector $\beta_{\mathbf{c}}$, and then apply LDA on the projected values. The key difference between the two approaches is the compression scheme: [62] jointly compress both classes when forming $\beta_{\mathbf{c}}$, whereas we propose separate class compression. We found that the latter is preferable, and Section 3.5 shows that Projected LDA has consistently better classification performance than the method of [62].

In terms of computational efficiency, Projected LDA described here and Compressed LDA of Section 3.2 are comparable - the main computational bottleneck of both is calculation of compressed $\widehat{\Sigma}_{w,\mathbf{c}}$. In terms of theoretical guarantees, since the methods coincide under Assumption 6, the results of Theorem 10 apply to Projected LDA as well. In practice, the sample sizes are often not exactly equal, and thus in Section 3.5 we observe some difference in the empirical performance

```
**Input**  : $X \in \mathbb{R}^{n\times p}$, $Y \in \{1,2\}^n$, $s \in (0,1)$, $m \ll n$.
**Output:** Compressed discriminant vector $\beta_{\mathbf{c}}$.
Compute $\overline{X}_g$ for $g = 1,2$.
Form compression matrices $\frac{1}{\sqrt{n_g s}} Q^g \in \mathbb{R}^{m_g \times n_g}$ as in Definition 4.
Form compressed samples $\frac{1}{\sqrt{n_g s}} Q^g (X^g - \overline{X}_g) + \overline{X}_g$ as in Definition 5.
Form $d \in \mathbb{R}^p$ and $\widehat{\Sigma}_{w,\mathbf{c}} \in \mathbb{R}^{p\times p}$ as in Definition 6. Set $\beta_{\mathbf{c}} = \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d$.
Project Training data $\beta_{\mathbf{c}}^\top \mathbf{x}_i^g$ and form projected covariance $\beta_{\mathbf{c}}^\top \widehat{\Sigma}_w \beta_{\mathbf{c}}$.
Use $\beta_{\mathbf{c}}$ and $\beta_{\mathbf{c}}^\top \widehat{\Sigma} \beta_{\mathbf{c}}$ in classification rule (1.4).
**return** $\beta_{\mathbf{c}}$
```

Algorithm 3: Projected LDA

of Compressed LDA and Projected LDA. We found, however, that neither method has uniformly better classification performance over the other.

3.4.2 Compressed QDA

The proposed compression scheme (3.1) is applied separately to each class, and thus allowing us to assign classes to the compressed samples. This, in turn, allows us to compute class-specific compressed covariance matrices, which motivates us to consider an extension of Compressed LDA to the case of unequal class covariance structures.

Quadratic Discriminant Analaysis (QDA) [33] is a generalization of LDA to the case of unequal class covariance matrices, which weakens Assumption 1.

Assumption 7. *Conditional on class membership $g = 1,2$, the samples $\mathbf{x}_i^g \in \mathbb{R}^p$ are i.i.d. $N(\mu_g\,,\,\Sigma_w^g)$.*

Under Assumption 7, the Bayes decision rule classifies a new sample $\mathbf{x} \in \mathbb{R}^p$ by minimizing

$$\operatorname*{argmin}_{g=1,2} \big\{ (\mathbf{x} - \mu_g)^\top (\Sigma_w^g)^{-1} (\mathbf{x} - \mu_g) + \log |\Sigma_w^g| - 2\log(\pi_g) \big\}, \tag{3.5}$$

where $|\Sigma_w^g|$ is the determinant of Σ_w^g. The QDA classification rule is the sample plug-in rule, where the population parameters μ_g, Σ_w^g, and π_g are replaced by their sample estimates $\overline{X}_g$, $\widehat{\Sigma}_w^g$, and n_g/n.

As our compression scheme proposed in (3.1) is applied separately to each class, it can be used

```
**Input :** $X \in \mathbb{R}^{n\times p}$, $Y \in \{1,2\}^n$, $s \in (0,1)$, $m \ll n$.
**Output:** Compressed discriminant vector $\beta_{\mathbf{c}}$.
Compute $\overline{X}_g$ for $g = 1, 2$.
Form compression matrices $\frac{1}{\sqrt{n_g s}} Q^g \in \mathbb{R}^{m_g \times n_g}$ as in Definition 4.
Form compressed samples $\frac{1}{\sqrt{n_g s}} Q^g(X^g - \overline{X}_g) + \overline{X}_g$ as in Definition 5.
Form $\widehat{\Sigma}^g_{w,\mathbf{c}} \in \mathbb{R}^{p\times p}$ for $g = 1, 2$ as in Definition 6.
Use $\widehat{\Sigma}^1_{w,\mathbf{c}}$ and $\widehat{\Sigma}^2_{w,\mathbf{c}}$ in decision rule (3.5).
**return** $\widehat{\Sigma}^g_{w,\mathbf{c}} \in \mathbb{R}^{p\times p}$ *for* $g = 1, 2$
```

Algorithm 4: Compressed QDA

to form class-specific compressed covariance matrices.

Definition 8. *The compressed sample covariance matrix for class* $g = 1, 2$ *is defined as*

$$\widehat{\Sigma}^g_{w,\mathbf{c}} := \frac{1}{m_g} \sum_{j=1}^{m_g} (\mathbf{x}^g_{j,\mathbf{c}} - \overline{X}_g)(\mathbf{x}^g_{j,\mathbf{c}} - \overline{X}_g)^\top.$$

We define the Compressed QDA decision rule by substituting $\widehat{\Sigma}^g_{w,\mathbf{c}}$ instead of Σ^g_w in (3.5), and $\overline{X}_g$, n_g/n instead of μ_g, π_g, respectively. Thus, Compressed QDA classifies a new sample $\mathbf{x} \in \mathbb{R}^p$ according to

$$\operatorname*{argmin}_{g=1,2} \big\{(\mathbf{x} - \overline{X}_g)^\top (\widehat{\Sigma}^g_{w,\mathbf{c}})^{-1} (\mathbf{x} - \overline{X}_g) + \log|\widehat{\Sigma}^g_{w,\mathbf{c}}| - 2\log(n_g/n)\big\}.$$

Algorithm 4 summarizes Compressed QDA.

3.5 Simulation Studies

In this section we empirically evaluate the performance of the proposed compression methods on three publicly available datasets: Zip Code [33], MNIST [70] and Skin Segmentation [71]. For each dataset, we compare five linear classifiers: (L1) Compressed LDA of Section 3.2; (L2) Projected LDA of Section 3.4.1; (L3) Fast Random Fisher Discriminant Analysis (FRF) of [62]; (L4) LDA trained on sub-sampled data drawn uniformly from both classes; and (L5) LDA trained on the full data (Full LDA). We also separately compare three quadratic classifiers: (Q1) Compressed

QDA of Section 3.4.2; (Q2) QDA trained on sub-sampled data drawn uniformly from both classes; and (Q3) QDA trained on the full data (Full QDA).

For each method, we evaluate the out-of-sample misclassification error rate as a function of reduced number of training samples $m = m_1 + m_2$ (with $m = n$ for full methods L5 and Q3). To assess variability due to compression or sub-sampling, we use 100 replications for each value of m. Within each classifier, a small multiple of the identity matrix γI_p is added to the corresponding estimate of the within-class covariance matrix Σ_w for numerical stability. We use $\gamma = 10^{-4}$ for Zip Code and Skin Segmentation data, and $\gamma = 10^{-3}$ for the MNIST data as it has a much larger number of features p compared to other datasets, and thus requires stronger regularization. We use $s = 0.01$ for Zip Code and MNIST datasets, and $s = 10^{-3}$ for the Skin Segmentation dataset as the latter has considerably larger sample size n; thus for all datasets $s = O(n^{-1/2})$.

We also compare the execution times of forming the compressed within-class covariance matrix $\widehat{\Sigma}_{w,\mathbf{c}}$ and full within-class covariance matrix $\widehat{\Sigma}_w$. For compression, we consider the time required to both compress the data via Q^g and to form $\widehat{\Sigma}_{w,\mathbf{c}}$. The timing results are reported using a Linux Machine with Intel Xeon E5-2690 with 2.90 GHz.

3.5.1 ZIP Code Data

The Zip Code Data [33] has $n = 7,291$ training samples with $p = 256$ features. The samples are images of handwritten digits for zip codes, and each feature corresponds to a normalized gray-scale pixel of an image. The original data has ten classes, each corresponding to a digit from 0 to 9, which we merge into two classes of even and odd digits. The classes are well-balanced, with 48% to 52% split between the class 1 odd digits and class 2 even digits. The corresponding test data has $n = 2,007$ samples.

The top of Figure 3.1 displays the misclassification error rates of (L1)-(L5) across 100 independent trials for each value of m. As expected, the performance of all methods improves with the increase in compression dimension m. Both Compressed LDA and Projected LDA have better classification performance compared to FRF and sub-sampled LDA. For example, when $m = 500$, Compressed LDA has a mean misclassification error rate of 12.60% (se 0.08%), and Projected

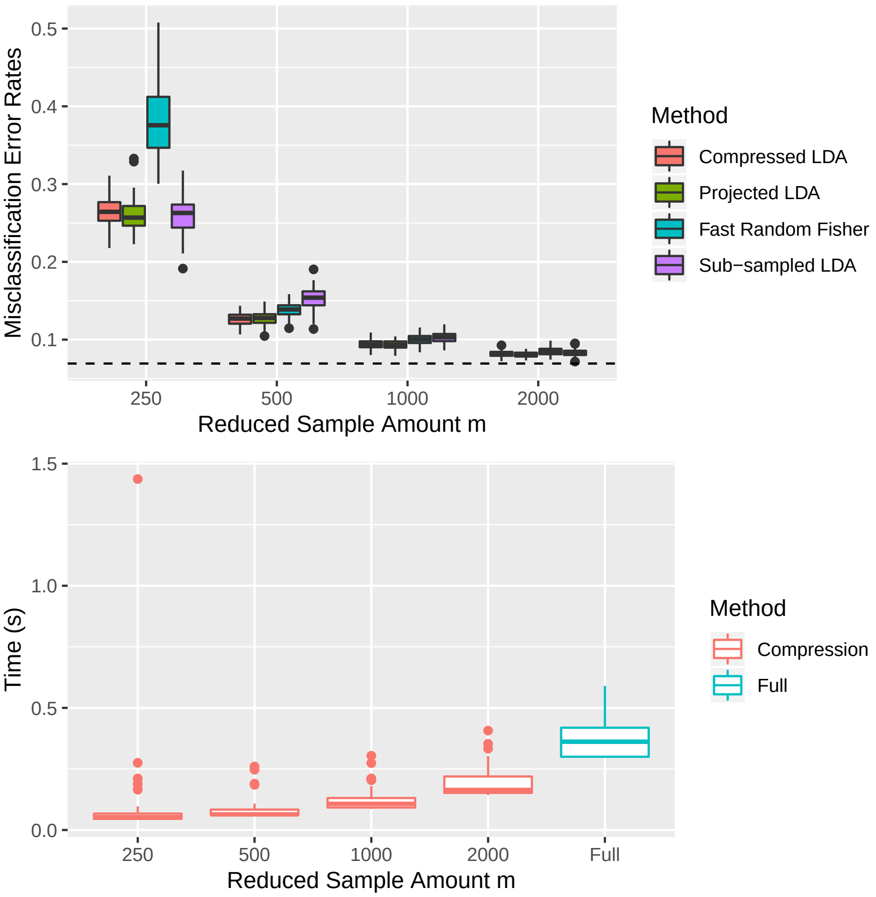

Figure 3.1: Zip Code Data. **Top:** Misclassification error rates across 100 replications for each value of m with $s = 0.01$ and $\gamma = 10^{-4}$. The dashed line represents the 6.88% error rate of Full LDA. **Bottom:** The execution times for 100 independent compressed and full covariance formations.

LDA has mean error rate 12.73% (se 0.08%). In contrast, FRF has a mean rate of 13.84% (se 0.08%), and sub-sampling has mean rate 15.31% (se 0.13%). Compressed and Projected LDA have similar error rates due to the balanced class sizes in this dataset, see Section 3.4.1.

Compressed and Projected LDA have the lowest mean error rates and standard errors across all values of m. Sub-sampling has the highest mean error rates for $m \geq 500$, which is likely because pixel values for images of handwritten digits are not normally distributed. Unexpected to us, FRF has the highest error rates for $m = 250$ despite using compression. We suspect this is due to its joint compression of both classes (rather than separate class compression used by our methods), which likely leads to higher variance in the estimated discriminant vector when m is relatively

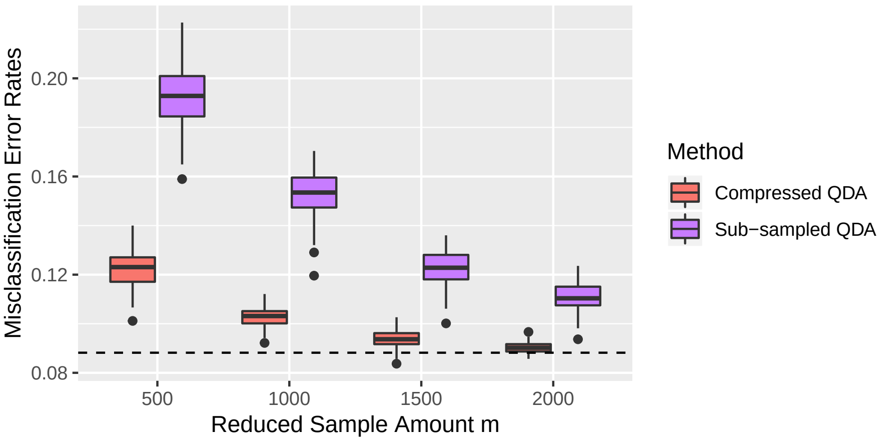

Figure 3.2: Zip Code Data. Misclassification error rates of compressed and sub-sampled QDA across 100 replications for each value of m with $s = 0.01$ and $\gamma = 10^{-3}$. The dashed line represents the 8.82% error rate of Full QDA.

small. When $m \geq 500$, the error rates of FRF are better than sub-sampling, but still worse than the proposed approaches.

The bottom of Figure 3.1 compares the execution times of forming compressed and full within-class covariance matrices, where the execution time for compression includes both formation of compressed samples in (3.1) and calculation of $\widehat{\Sigma}_{w,\mathbf{c}}$. As expected, compression is significantly faster. For instance, when $m = 2,000$, the compression takes on average 0.19 seconds (se 0.01 s), while the construction of full covariance matrixtakes on average 0.36 seconds (se 0.01 s).

Figure 3.2 displays the misclassification error rates of (Q1)-(Q3). Compressed QDA has uniformly lower mean error rates and lower variance than QDA on sub-sampled data for the same values of m. For instance, when $m = 500$, Compressed QDA has a mean error rate of 12.22% (se 0.08%) while sub-sampled QDA has the mean error rate of 19.27% (se 0.14%). For $m \geq 2,000$, the misclassification error rate of Compressed QDA matches that of Full QDA.

3.5.2 MNIST Data

The MNIST Data [70] has $n = 60,000$ training samples with $p = 784$ features. The samples are pictures of handwritten digits, and each feature corresponds to a normalized grayscale pixel for an image. The original data has ten classes, each corresponding to a digit from 0 to 9, which we merge into two classes of even and odd digits. The classes are well-balanced with a 51% to 49% split between the class 1 odd digits and class 2 even digits. The test data has $n = 10,000$ samples.

The top of Figure 3.3 shows the misclassification error rates of the linear methods across 100 independent trials for each value of m. As with the Zip Code data, both Compressed LDA and Projected LDA have the lowest misclassification error rates compared to FRF and sub-sampled LDA. For instance, when $m = 2,000$, the mean error rate for Compressed LDA is 13.93% (se 0.04%), and the mean error rate for Projected LDA is 13.98 (se 0.04%). In contrast, FRF has mean rate 15.71% (se 0.05%), and sub-sampled LDA has mean rate 16.05% (se 0.05%). As with the Zip Code data, Compressed and Projected LDA have similar rates due to the balanced class sizes, see Section 3.4.1. Unlike the Zip Code data, FRF performs comparable to sub-sampling even for larger values of m. This suggests that joint class compression leads to sub-optimal classification performance compared to proposed separate class compression, and the difference is particularly striking when the number of features p is large.

The bottom of Figure 3.3 compares the execution times of forming compressed and full within-class covariance matrices. As expected, compression is considerably faster. Even when $m = 10,000$, the mean time for compression (9.31 seconds, se 1.29) is significantly smaller than the time of forming $\widehat{\Sigma}_w$ on the full data (23.53 seconds, se 2.29).

Figure 3.4 shows the misclassification error rates of the quadratic methods. Compressed QDA has uniformly better performance than sub-sampling, it has both lower mean error rates and lower variances. For example, when $m = 1,000$, Compressed QDA has mean error rate 19.24% (se 0.06%) while sub-sampled QDA has mean error 29.42% (se 0.21%).

3.5.3 Skin Segmentation Data

The Skin Segmentation Data [71] has $n = 245,057$ samples with $p = 3$ features. The features are Red, Blue, and Green pixel values for randomly sampled image pixels. The goal is to learn which colors represent skin, and subsequently classify those pixels as corresponding to skin or not. Unlike the Zip Code and MNIST datasets, here the classes are unbalanced, with 21% (skin) to 79% (not skin) split. We select 90% of the data from each class for training, and use the remaining 10%

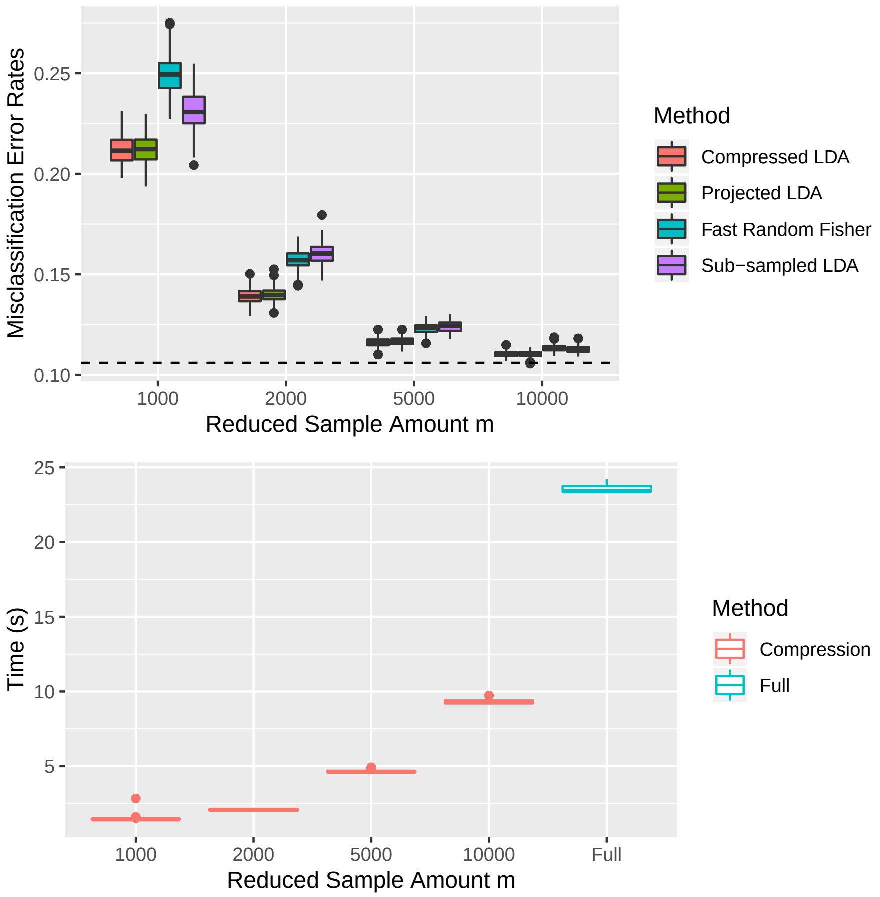

Figure 3.3: MNIST Data. **Top:** Misclassification error rates across 100 replications for each value of m with $s = 0.01$ and $\gamma = 10^{-3}$. The dashed line represents the 10.60% misclassification error rate of Full LDA. **Bottom:** The execution times for 100 independent compressed and full covariance formations.

for testing.

The top of Figure 3.5 displays the misclassification error rates of the linear methods across 100 independent trials for each value of m. Compressed LDA, Projected LDA, and FRF all have superior classification performance over sub-sampled LDA, especially in terms of variance for the same value of m. For instance, when $m = 25$, Compressed LDA has an average error rate of 7.42% (se 0.09%), with 7.57% (se 0.09%) for Projected LDA, and 7.38% (se 0.09%) for FRF. In contrast, sub-sampled LDA has error 8.78% (se 0.40%). Unlike the Zip Code and MNIST datasets, FRF performs comparably to the proposed approaches, which supports our previous conjecture

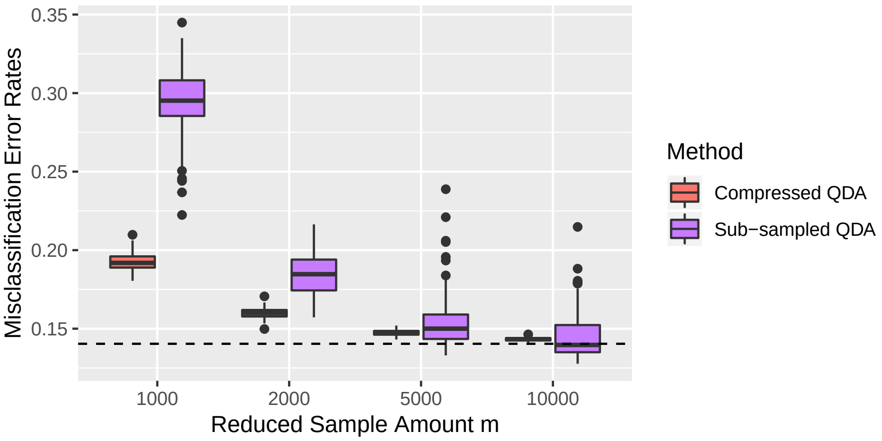

Figure 3.4: MNIST Data. Misclassification error rates of compressed and sub-sampled QDA across 100 replications for each value of m with $s = 0.01$ and $\gamma = 10^{-3}$. The dashed line represents the 14.04% error rate of Full QDA.

that the difference between joint compression and separate class compression is more pronounced for larger values of p. The bottom of Figure 3.5 displays the corresponding error rates for the quadratic methods. While the mean error rates between Compressed QDA and sub-sampled QDA are similar, Compressed QDA has much smaller variance, which is consistent with results we observed for other datasets.

The Skin Segmentation Data only has $p = 3$ features, and thus one may ask whether the compression is really necessary since it doesn't offer significant computational advantages for small values of p. We found, however, that compression still allows to use much smaller number of samples to obtain good predictive accuracy, as Compressed LDA reaches the Full LDA error rate of 6.93% at only $m = 100$. Furthermore, our main reason for including this dataset as an example is to illustrate how compression can induce normality in the compressed samples when the normality for original samples does not hold. The top of Figure 3.6 shows the first two principal components of $5,000$ original training samples, whereas the bottom of Figure 3.6 shows the first two principal components of $5,000$ compressed samples. The original training samples clearly are not normally distributed as the main directions of variation display non-linear class separation. In contrast, each class of compressed data has an elliptical shape suggesting the normal distribution and a linear

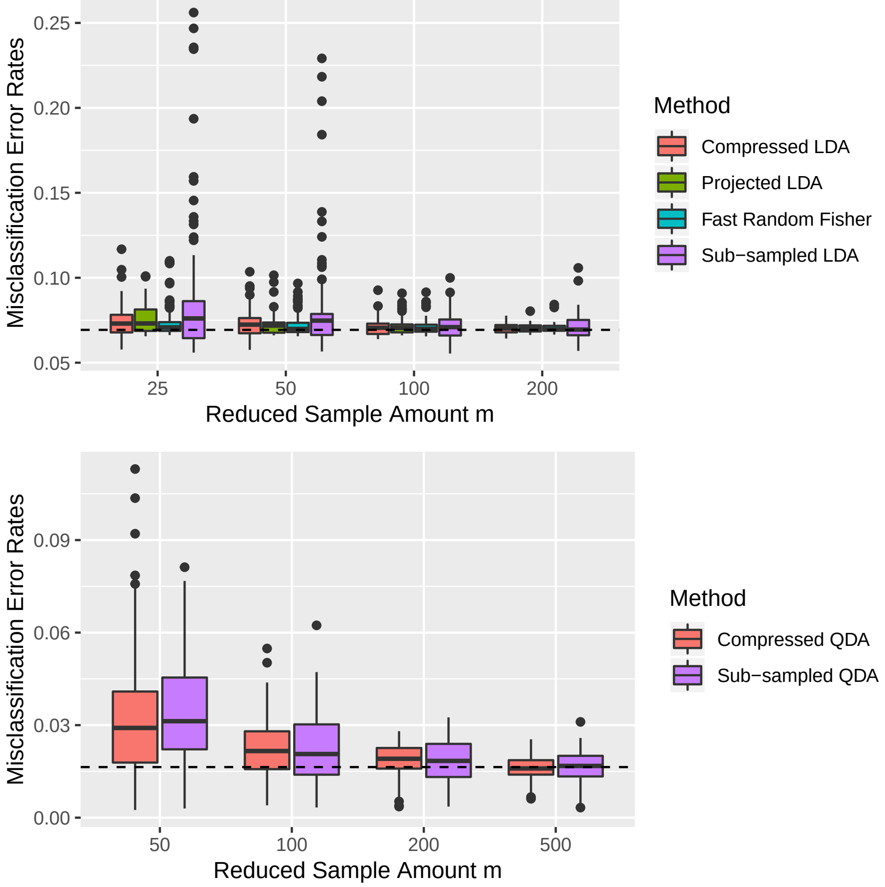

Figure 3.5: Skin Segmentation Data, misclassification error rates across 100 replications for each vale of m. **Top:** Linear classification methods with $s = 10^{-3}$ and $\gamma = 10^{-4}$. The dashed line represents the 6.93% error rate of Full LDA. **Bottom:** Qadratic classification methods with $s = 10^{-3}$ and $\gamma = 10^{-4}$. The dashed line represents the 1.64% error rate of Full QDA.

classification boundary. Thus, Compressed LDA is more robust to the assumption of normality than sub-sampling. For the Skin Segmentation Data, this leads to Compressed LDA having slightly lower mean misclassification error rates compared to sub-sampling, and significantly smaller error variances across the replications.

3.6 Discussion

We propose a sample reduction scheme for discriminant analysis through compression. The advantage of compression over sub-sampling is illustrated in Section 3.5, where the proposed Compressed LDA consistently has better classification performance than LDA trained on sub-

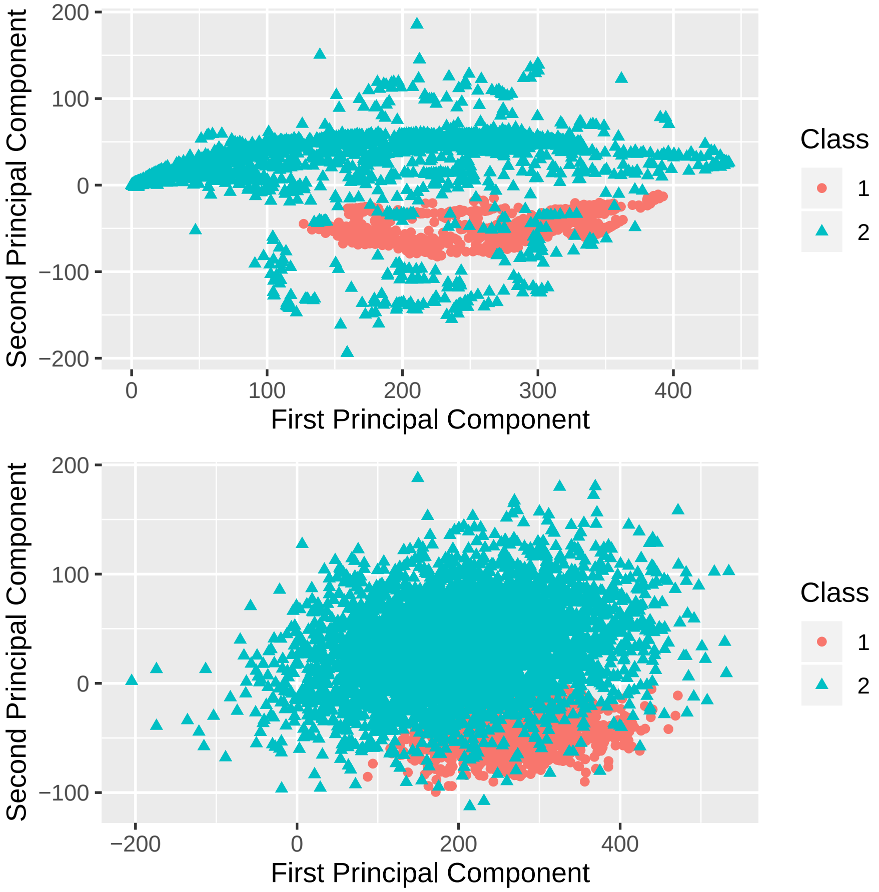

Figure 3.6: Skin Segmentation Data, the two classes are separated by both shape and color. **Top:** First two principal components based on the $5,000$ training samples. **Bottom:** First two principal components based on $5,000$ compressed samples with $s = 0.001$.

sampled data. The compression scheme is further extended to Projected LDA and Compressed QDA, which again show superior predictive accuracy compared to the same classifiers trained on sub-sampled data.

There are several directions of future research that could be pursued. First, while we only considered binary classification, our approach can be extended to the multi-class setting by applying compression (3.1) to all G classes. Secondly, given our results on compressing in the number of samples, and existing results on compressing in the number of features [60, 61], it would be of interest to simultaneously consider both compression schemes within discriminant analysis. Finally, here we focused on linear and quadratic classification rules which may be too restrictive. Exploring

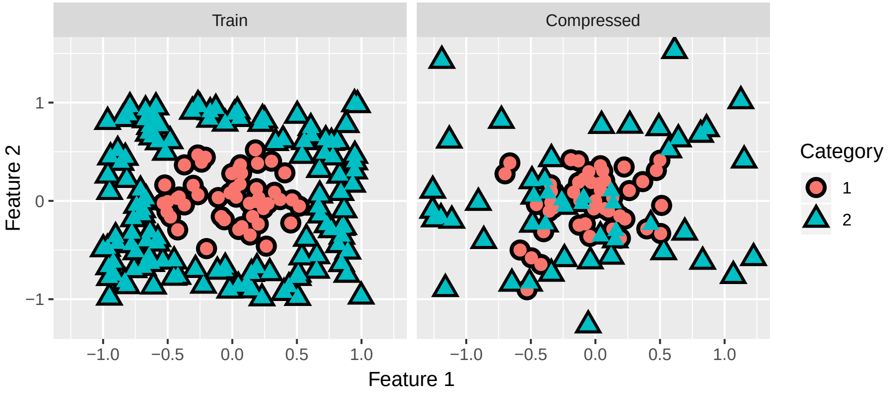

Figure 3.7: **Left**: 500 simulated data set where the classes are separated by shape and color. Right: 100 compressed samples. While the original data set is separable with respect to the classes, the compressed samples are not.

compression within the kernel discriminant analysis framework [72] will allow for more flexible non-linear classification boundaries.

3.7 Extension to Kernel Discriminant Analysis

Much work has been done in scaling kernel methods to large n data. Two of the most common approaches are the Nyström method and Random Fourier Features (RFF). However, both of these methods require $O(nm)$ memory. The compression method presented here requires only $O(m^2)$ or $O(n)$ bits of memory while also having competitive classification performance.

The above compression scheme extends to the setting of Reproducing Kernel Hilbert Spaces to perform non-linear classification on large data. However, naively compressing the original samples can ruin non-linear separation, as shown in Figure 3.7.

As in Chapter 2, let $\Phi : \mathbb{R}^p \to \mathcal{H}$ be a mapping so that the kernel trick $\langle \Phi(x) \, , \, \Phi(x') \rangle_{\mathcal{H}} = k(x, x')$ hold for some kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. Let

$$\Phi_g = \Phi(X^g) = \begin{pmatrix} \Phi(x_1^g) \\ \vdots \\ \Phi(x_{n_g}^g) \end{pmatrix}$$

be the mapped class-g data, $g = 1, 2$. Let

$$\begin{pmatrix} \Phi_1 \\ \\ \Phi_2 \end{pmatrix} = \begin{pmatrix} \Phi(X^1) \\ \\ \Phi(X^2) \end{pmatrix} = \Phi(X) = \begin{pmatrix} \Phi(x_1) \\ \vdots \\ \Phi(x_n) \end{pmatrix}$$

be the block representation of the entire mapped data set. Let $\overline{\Phi_g}$ represent the g-th class mean of the mapped data $\frac{1}{n_g}\sum_{j=1}^{n_g} \Phi(x_j^g)$. Finally, let

$$Q = \begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix} \in \mathbb{R}^{m \times n}$$

be the block-diagonal compression matrix with $Q^g \in \mathbb{R}^{m_g \times n_g}$, $g = 1, 2$.

3.7.1 Compressed Kernel Matrices

We first present the compression scheme in $\mathcal{H}$ and then translate it to operations on the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ in the coefficient space.

The compressed mapped data in $\mathcal{H}$ has the block form

$$\begin{pmatrix} Q^1(\Phi_1 - \overline{\Phi_1}) + \overline{\Phi_1} \\ Q^2(\Phi_2 - \overline{\Phi_2}) + \overline{\Phi_2} \end{pmatrix}.$$

By the Representer Theorem, the discriminant function $f \in \mathcal{H}$ lies in the span of the compressed data

$$f = \sum_{g=1}^{2} \sum_{\ell=1}^{m_g} \alpha_\ell^g \Big\{ \sum_{j=1}^{n_g} Q_{\ell,s}^g (\Phi(\mathbf{x}_i^g) - \overline{\Phi_g}) + \overline{\Phi_g} \Big\}.$$

Definition 9. *Let $M \in \mathbb{R}^{m\times n}$ be the matrix with block diagonal structure*

$$M = \begin{pmatrix} Q^1 + \mathbf{1}_{m_1}\frac{1}{n_1}\mathbf{1}_{n_1}^\top & \\ & \\ & Q^2 + \mathbf{1}_{m_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix},$$

and let

$$C_d = \begin{pmatrix} C_{n_1} & \\ & C_{n_2} \end{pmatrix} \in \mathbb{R}^{n\times n}$$

be a block diagonal centering matrix for

$$C_{n_g} = I - n_g^{-1}\mathbf{1}\mathbf{1}^\top, g = 1, 2.$$

Then $\mathbf{K_c} = MC_d\mathbf{K}C_dM^\top \in \mathbb{R}^{m\times m}$ is the compressed kernel matrix *corresponding to the kernel matrix evaluated on the compressed samples in $\mathcal{H}$.*

Lemma 13. *The Matrix $MC_d \in \mathbb{R}^{m\times n}$ reduces to $QC_d \in \mathbb{R}^{m\times n}$, and hence $\mathbf{K_c} = QC_d\mathbf{K}(QC_d)^\top$.*

Proof. We have

$$\begin{aligned} &MC_d = \\ &\begin{pmatrix} Q^1 + \mathbf{1}_{m_1}\frac{1}{n_1}\mathbf{1}_{n_1}^\top & \\ & \\ & Q^2 + \mathbf{1}_{m_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix} \begin{pmatrix} I_{n_1} - \frac{1}{n_1}\mathbf{1}_{n_1}\mathbf{1}_{n_1}^\top & \\ & I_{n_2} - \frac{1}{n_2}\mathbf{1}_{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix} = \\ &= \begin{pmatrix} Q^1C_1 + \frac{1}{n_1}\mathbf{1}_{m_1}\mathbf{1}_{n_1}^\top - \frac{1}{n_1}\mathbf{1}_{m_1}\mathbf{1}_{n_1}^\top & \\ & Q^2C_2 + \frac{1}{n_2}\mathbf{1}_{m_2}\mathbf{1}_{n_2}^\top - \frac{1}{n_2}\mathbf{1}_{m_2}\mathbf{1}_{n_2}^\top \end{pmatrix} \\ &= \begin{pmatrix} Q^1C_1 & \\ & Q^2C_2 \end{pmatrix}. \end{aligned}$$

□

This, in turn, is equal to

$$\left[\begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix} - \begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix}\right] \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix}$$
$$\left[\begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix} - \begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix}\right]^\top$$

This will expand into four separate terms:

$$\underbrace{\begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix} \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix} \begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix}^\top}_{(I)}$$
$$-\underbrace{\begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix} \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix} \begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix}^\top}_{(II)}$$
$$-\underbrace{\begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix} \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix} \begin{pmatrix} Q^1 & \\ & Q^2 \end{pmatrix}^\top}_{(III)}$$
$$+\underbrace{\begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix} \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix} \begin{pmatrix} Q^1\mathbf{1}_{n_1}\frac{1}{n_1}\mathbf{1}^\top & \\ & Q^2\mathbf{1}_{n_2}\frac{1}{n_2}\mathbf{1}_{n_2}^\top \end{pmatrix}^\top}_{(IV)}$$

Our strategy for computing these quantities are to go row-by-row for each $\mathbf{K}_{i,j}$ and compute the row mean and the vectors $Q^i\mathbf{K}_{i,j}$. We can then store the results in an array

Derivation of Compressed Kernel Matrix. Compute

$$
\begin{pmatrix}
\left\langle \sum_{s=1}^{n_1} Q^1_{1,s}(\Phi(x^1_s) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\ f \right\rangle \\
\vdots \\
\left\langle \sum_{s=1}^{n_i} Q^1_{m_1,s}(\Phi(x^1_s) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\ f \right\rangle \\
\left\langle \sum_{s=1}^{n_1} Q^2_{1,s}(\Phi(x^2_s) - \overline{\Phi_2}) + \overline{\Phi_2}\,,\ f \right\rangle \\
\vdots \\
\left\langle \sum_{s=1}^{n_2} Q^1_{m_2,s}(\Phi(x^2_s) - \overline{\Phi_2}) + \overline{\Phi_2}\,,\ f \right\rangle
\end{pmatrix}
=
$$

$$
\begin{pmatrix}
\left\langle \sum_{s=1}^{n_1} Q^1_{1,s}(\Phi(x^1_s) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\ \sum_{i=1}^{2}\sum_{\ell=1}^{m_i} \alpha^i_\ell \left\{ \sum_{s=1}^{n_i} Q^i_{\ell,s}(\Phi(x^i_s) - \overline{\Phi_i}) + \overline{\Phi_i} \right\} \right\rangle \\
\vdots \\
\left\langle \sum_{s=1}^{n_i} Q^1_{m_1,s}(\Phi(x^1_s) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\ \sum_{i=1}^{2}\sum_{\ell=1}^{m_i} \alpha^i_\ell \left\{ \sum_{s=1}^{n_i} Q^i_{\ell,s}(\Phi(x^i_s) - \overline{\Phi_i}) + \overline{\Phi_i} \right\} \right\rangle \\
\left\langle \sum_{s=1}^{n_1} Q^2_{1,s}(\Phi(x^2_s) - \overline{\Phi_2}) + \overline{\Phi_2}\,,\ \sum_{i=1}^{2}\sum_{\ell=1}^{m_i} \alpha^i_\ell \left\{ \sum_{s=1}^{n_i} Q^i_{\ell,s}(\Phi(x^i_s) - \overline{\Phi_i}) + \overline{\Phi_i} \right\} \right\rangle \\
\vdots \\
\left\langle \sum_{s=1}^{n_2} Q^1_{m_2,s}(\Phi(x^2_s) - \overline{\Phi_2}) + \overline{\Phi_2}\,,\ \sum_{i=1}^{2}\sum_{\ell=1}^{m_i} \alpha^i_\ell \left\{ \sum_{s=1}^{n_i} Q^i_{\ell,s}(\Phi(x^i_s) - \overline{\Phi_i}) + \overline{\Phi_i} \right\} \right\rangle
\end{pmatrix}
$$

$$
= \mathbf{K_c}\alpha
$$

for some $m \times m$ compressed kernel matrix $\mathbf{K_c}$.

The (i, j) coordinate of the $(1, 1)$ upper left block of M is equal to

$$
\left\langle \sum_{s=1}^{n_1} Q^1_{i,s}(\Phi(x^1_s) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\ \sum_{t=1}^{n_1} Q^1_{j,t}(\Phi(x^1_t) - \overline{\Phi_1}) + \overline{\Phi_1} \right\rangle =
$$

$$
\left\langle \frac{1}{n_1}\sum_{s=1}^{n_1} Q^1_{i,s}(\Phi(x^1_s) - \overline{\Phi_1})\,,\ \frac{1}{n_1}\sum_{t=1}^{n_1} Q^1_{j,s}(\Phi(x^1_t) - \overline{\Phi_1}) \right\rangle
$$

$$
+ \left\langle \sum_{s=1}^{n_1} Q^1_{i,s}(\Phi(x^1_s) - \overline{\Phi_1})\,,\ \overline{\Phi_1} \right\rangle + \left\langle \sum_{t=1}^{n_1} Q^1_{j,t}(\Phi(x^1_t) - \overline{\Phi_1})\,,\ \overline{\Phi_1} \right\rangle + \frac{1}{n}\mathbf{1}^\top \mathbf{K}^{1,1}\frac{1}{n_1}\mathbf{1}
$$

The intermediary equation

$$\left\langle \frac{1}{n_1}\sum_{s=1}^{n_1} Q^1_{i,s}(\Phi(x^1_s) - \overline{\Phi_1})\,,\,\overline{\Phi_1}\right\rangle = \frac{1}{n_1}\sum_{s=1}^{n_1} Q_{i,s}\left\langle \Phi(x_s),\overline{\Phi_1}\right\rangle - (\frac{1}{n}Q_i\mathbf{1})\frac{1}{n}\mathbf{1}^\top\mathbf{K}^{1,1}\frac{1}{n}\mathbf{1}$$
$$= \frac{1}{n_1}Q_i\mathbf{K}^{1,1}\frac{1}{n_1}\mathbf{1} - (\frac{1}{n_1}Q_i\mathbf{1})\frac{1}{n_1}\mathbf{1}^\top\mathbf{K}^{1,1}\frac{1}{n}\mathbf{1} = \frac{1}{n_1}Q_i(I-\frac{1}{n_1}\mathbf{1}\mathbf{1}^\top)\mathbf{K}^{1,1}\frac{1}{n_1}\mathbf{1} = \frac{1}{n}Q_iC_1\mathbf{K}^{1,1}\frac{1}{n_1}\mathbf{1},$$

and applying this over both i and j gives

$$\frac{1}{n}Q_iC_1\mathbf{K}^{1,1}C_1Q_j^\top\frac{1}{n} + \frac{1}{n_1}Q_iC_1\mathbf{K}^{1,1}\frac{1}{n_1}\mathbf{1} + \frac{1}{n_1}\mathbf{1}^\top\mathbf{K}^{1,1}C_1\frac{1}{n_1}Q_j^\top + \frac{1}{n_1}\mathbf{1}^\top\mathbf{K}^{1,1}\mathbf{1}\frac{1}{n_1}.$$

Combining like terms gives

$$Q_iC_1\mathbf{K}^{1,1}\left(C_1Q_j^\top + \frac{1}{n}\mathbf{1}\right) + \frac{1}{n_1}\mathbf{1}_{n_1}^\top\mathbf{K}^{1,1}\left(C_1Q_j^\top + \mathbf{1}\frac{1}{n_1}\right)$$
$$= \left(\frac{1}{n_1}\mathbf{1}^\top + Q_iC_1\right)\mathbf{K}^{1,1}\left(C_1Q_j^\top + \frac{1}{n_1}\mathbf{1}_{n_1}\right).$$

This is the (i,j)-component, which implies that the $(1,1)$ upper block of the matrix M is equal to

$$M^{1,1} = \left(\frac{1}{n_1}\mathbf{1}_{m_1}\mathbf{1}_{n_1}^\top + \frac{1}{n_1}Q^1C_1\right)\mathbf{K}^{1,1}\left(C_1\frac{1}{n_1}Q^{1^\top} + \frac{1}{n_1}\mathbf{1}_{n_1}\mathbf{1}_{m_1}^\top\right)$$

Likewise, by changing the index i, the lower right $(2,2)$ is equal to

$$M^{2,2} = \left(\frac{1}{n_2}\mathbf{1}_{m_2}\mathbf{1}_{n_2}^\top + Q^2C_2\right)\mathbf{K}^{2,2}\left(C_2Q^{2^\top} + \frac{1}{n_2}\mathbf{1}_{n_2}\mathbf{1}_{m_2}^\top\right)$$

Let us now compute the off diagonal matrices of M. Here, we focus on the $(1,2)$ diagonal, for the $(2,1)$ diagonal will merely be its transpose. We have

$$\begin{aligned}
&\left\langle \frac{1}{n_1}\sum_{s=1}^{n_1} Q_{i,s}^1(\Phi(x_s^1) - \overline{\Phi_1}) + \overline{\Phi_1}\,,\, \frac{1}{n_2}\sum_{s=1}^{n_2} Q_{i,s}^2(\Phi(x_s^2) - \overline{\Phi_2}) + \overline{\Phi_2} \right\rangle \\
&= \left\langle \frac{1}{n_1}\sum_{s=1}^{n_1} Q_{i,s}^1(\Phi(x_s^1) - \overline{\Phi_1})\,,\, \frac{1}{n_2}\sum_{s=1}^{n_2} Q_{i,s}^2(\Phi(x_s^2) - \overline{\Phi_2}) \right\rangle + \left\langle \frac{1}{n_1}\sum_{s=1}^{n_1} Q_{i,s}^1(\Phi(x_s^1) - \overline{\Phi_1})\,,\, \overline{\Phi_2} \right\rangle \\
&\qquad\qquad + \left\langle \overline{\Phi_1}\,,\, \frac{1}{n_2}\sum_{s=1}^{n_2} Q_{i,s}^2(\Phi(x_s^2) - \overline{\Phi_2}) \right\rangle + \langle \overline{\Phi_1}\,,\, \overline{\Phi_2} \rangle \\
&= \frac{1}{n_1}Q_i^1 C_1 \mathbf{K}^{1,2} C_2 \frac{1}{n_2} Q_j^{2^\top} + \frac{1}{n_1}Q_i^1 C_1 \mathbf{K}^{1,2}\frac{1}{n_2}\mathbf{1}_{n_2} + \frac{1}{n_1}\mathbf{1}_{n_1}\mathbf{K}^{1,2} C_2 \frac{1}{n_2} Q_j^{2^\top} + \frac{1}{n_1}\mathbf{1}_{n_1}^\top \mathbf{K}^{1,2}\frac{1}{n_2}\mathbf{1}_{n_2} \\
&= \left(\frac{1}{n_1}\mathbf{1}_{n_1}^\top + \frac{1}{n_1}Q_i^1 C_1\right)\mathbf{K}^{1,2}\left(C_2\frac{1}{n_2}Q_j^{2^\top} + \frac{1}{n_2}\mathbf{1}_{n_2}\right).
\end{aligned}$$

It follows that the upper $(1,2)$ coordinate block matrix is

$$\left(\mathbf{1}_{m_1}\frac{1}{n_1}\mathbf{1}_{n_1}^\top + Q^1 C_1\right)\mathbf{K}^{1,2}\left(C_2 Q^{2^\top} + \frac{1}{n_2}\mathbf{1}_{n_2}\mathbf{1}_{m_2}^\top\right).$$

□

3.8 Discussion

We propose a sample reduction scheme for discriminant analysis through compression. The advantage of compression over sub-sampling is illustrated in Section 3.5, where the proposed Compressed LDA consistently has better classification performance than LDA trained on sub-sampled data. Also, we derive a non-asymptotic bound on the misclassification error rate of Compressed LDA compared to the Bayes classifier. The compression scheme is further extended to Projected LDA and Compressed QDA, which again show superior predictive accuracy compared to the same classifiers trained on sub-sampled data.

There are several directions of future research that could be pursued. First, while we only considered binary classification, our approach can be extended to the multi-class setting by applying compression (3.1) to all G classes. Secondly, given our results on compressing in the number of samples, and existing results on compressing in the number of features [60, 61], it would be of interest to simultaneously consider both compression schemes within discriminant analysis. Finally, here we focused on linear and quadratic classification rules which may be too restrictive. Exploring compression within the kernel discriminant analysis framework [72] will allow for more flexible non-linear classification boundaries.

3.9 Proof of Miscalculation Error Rate

Proof. If $\mathbf{x} \sim N(\mu_1, \Sigma_w)$ belongs to class 1, it is misclassified if and only if $d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \geq 0$. The probability of misclassifying $\mathbf{x}$ conditioned on the class y being 1 is

$$\mathbb{P}\big(d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) < 0 \,\big|\, y = 1\big).$$

Note that

$$d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \sim N(m_1\,,\, \widetilde{\sigma}_w).$$

where

$$m_1 = d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_1 - \overline{X}), \quad \widetilde{\sigma}_w = d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d.$$

Thus, the probability of misclassifying $\mathbf{x}$ is

$$\Phi\left(\frac{-m_1}{\sqrt{\widetilde{\sigma}_w}}\right) = \Phi\left(- \frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_1 - \overline{X})}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\right) = \Phi\left(- \frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_1 - \overline{X}_1) + d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\right).$$

where the last equality results from the identity $\mu_1 - \overline{X} = (\mu_1 - \overline{X}_1) + d$.

Likewise, let us assume that $\mathbf{x} \sim N(\mu_2, \Sigma_w)$ belongs to class 2. Just as before,

$$d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \sim N(m_2\,,\, \widetilde{\sigma}_w).$$

with

$$m_2 = d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_2 - \overline{X}), \quad \widetilde{\sigma}_w = d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d.$$

The conditional probability of misclassifying $\mathbf{x}$ is

$$\mathbb{P}\big(d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \geq 0 \,\big|\, y = 2\big).$$

Thus, the probability of misclassifying a data point which belongs to group 2 is

$$\begin{aligned} 1 - \Phi\bigg(\frac{-m_2^{\text{comp}}}{\sqrt{\widetilde{\sigma}_w}}\bigg) &= 1 - \Phi\bigg(- \frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_2 - \overline{X})}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\bigg) \\ &= \Phi\bigg(\frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_2 - \overline{X})}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\bigg) = \Phi\bigg(\frac{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_2 - \overline{X}_2) - d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\bigg), \end{aligned}$$

where the second equality is a result of $1 - \Phi(-t) = \Phi(t)$ and the last equality is a result of $\mu_2 - \overline{X} = (\mu_2 - \overline{X}_2) - d$.

Thus, the total misclassification error rate for compressed Linear discriminant analysis is

$$\begin{aligned} &\mathbb{P}\big(d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) < 0 \,\big|\, y = 1\big)\pi_1 + \mathbb{P}\big(d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mathbf{x} - \overline{X}) \geq 0 \,\big|\, y = 2\big)\pi_2 \\ &= \frac{1}{2}\sum_{g=1}^{2} \Phi\bigg(\frac{(-1)^g d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_g - \overline{X}_g) - d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}{\sqrt{d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d}}\bigg). \end{aligned}$$

□

3.10 Technical Proofs

This supplement contains a proof of Theorem 10 along with supplemental Theorems and Lemmas. In the following C denotes an absolute constant which may change from line to line. If multiple constants appear in the same expression, C_1, C_2, etc. will be used to differentiate them.

We make the following assumption which is useful for simplifying expressions in the theory.

Assumption 8. *The number of compressed samples m is large enough so that $\log(\eta^{-1})/m \leq 1$. Additionally, the number of original training samples n is large enough so that $\log(\eta^{-1}) \leq \sqrt{n}$.*

Remark 6. *Assumption 8 is mild. For instance, if $\eta = 10^{-10}$, then m must be at least 24, and n must be at least 531. If $\eta = 10^{-2}$, then m muust be at least 5, and n must be at least 22*

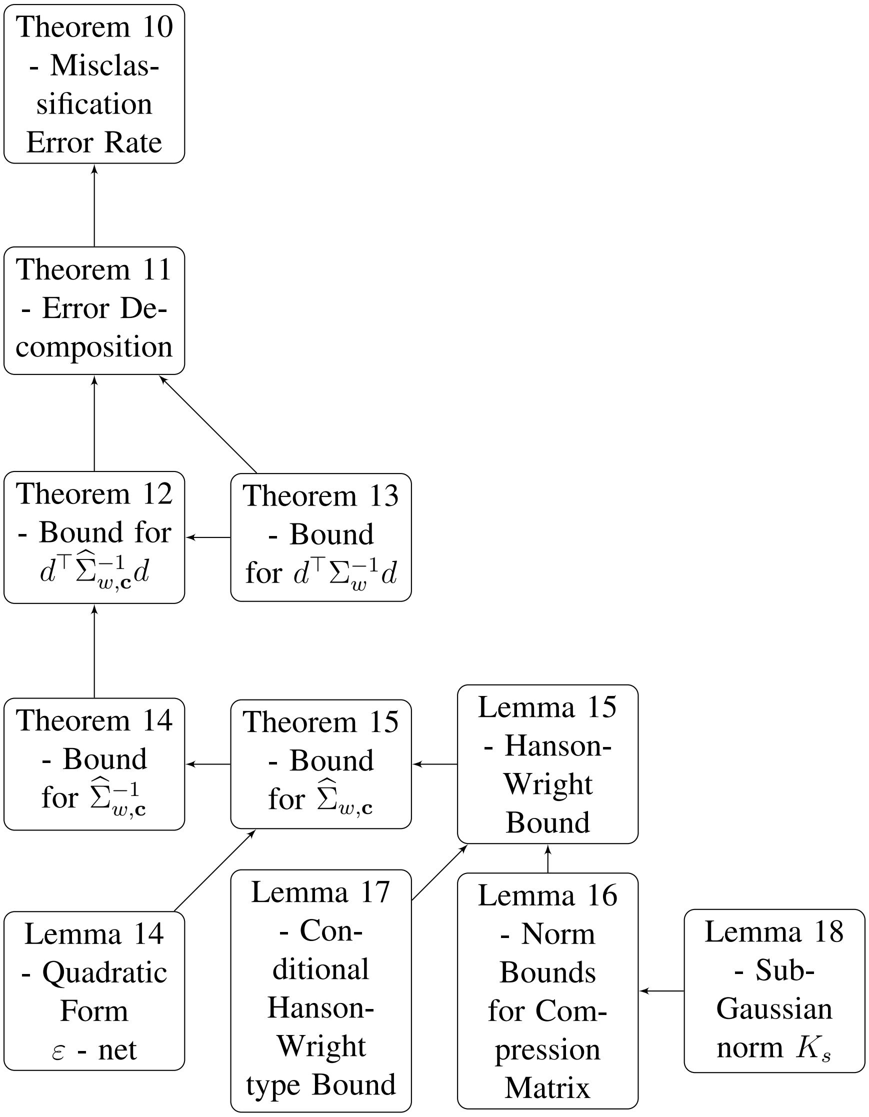
Theorem 10 - Misclassification Error Rate
Theorem 11 - Error Decomposition
Theorem 12 - Bound for $d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d$
Theorem 13 - Bound for $d^\top \Sigma_w^{-1} d$
Theorem 14 - Bound for $\widehat{\Sigma}_{w,\mathbf{c}}^{-1}$
Theorem 15 - Bound for $\widehat{\Sigma}_{w,\mathbf{c}}$
Lemma 15 - Hanson-Wright Bound
Lemma 14 - Quadratic Form ε - net
Lemma 17 - Conditional Hanson-Wright type Bound
Lemma 16 - Norm Bounds for Compression Matrix
Lemma 18 - Sub-Gaussian norm K_s

Figure 3.8: Proof chart for Theorem 10.

Proof of Theorem 10. By Theorem 11, the compressed LDA misclassification error rate $R_\mathbf{c}$ has the form

$$R_\mathbf{c} = f(\varepsilon_1^1, \varepsilon_1^2, \varepsilon_2) = \frac{1}{2}\sum_{g=1}^{2} \Phi\left(\frac{\varepsilon_1^g - \delta^\top \Sigma_w^{-1}\delta}{\sqrt{\varepsilon_2 + \delta^\top \Sigma_w^{-1}\delta}}\right),$$

where ε_1^g and ε_2 are defined in Theorem 11. Let $\varepsilon = (\varepsilon_1^1, \varepsilon_1^2, \varepsilon_2)$. Taking the first-order Taylor expansion of f centered at 0 gives

$$R_{\mathbf{c}} = f(\varepsilon) = \Phi(-\sqrt{\delta^\top \Sigma_w^{-1} \delta}) + \nabla f(0)^\top \varepsilon + o_p(\|\varepsilon\|_2) = R_{\text{opt}} + \nabla f(0)^\top \varepsilon + o_p(\|\varepsilon\|_2).$$

Plugging this expansion into $|R_{\mathbf{c}} - R_{\text{opt}}|$ gives

$$\begin{aligned} |R_{\mathbf{c}} - R_{\text{opt}}| &= \left| \Phi(-\sqrt{\delta^\top \Sigma_w^{-1} \delta}) + \nabla f(0)^\top \varepsilon \ + o_p(\|\varepsilon\|_2) - R_{\text{opt}} \right| \\ &\leq \left| R_{\text{opt}} + \nabla f(0)^\top \varepsilon - R_{\text{opt}} \right| + o_p(\|\varepsilon\|_2) \\ &= \left| \nabla f(0)^\top \varepsilon \right| + o_p(\|\varepsilon\|_2) \\ &\leq C \|\nabla f(0)\|_2 \, \|\varepsilon\|_2, \end{aligned}$$

where we absorbed the lower-order $o_p(\|\varepsilon\|_2)$ into the absolute constant $C > 0$.

We now compute $\|\nabla f(0)\|_2$. The partial derivatives are

$$\frac{\partial f}{\partial \varepsilon_1^g}(0) = \frac{1}{2} \phi\left(\frac{-\delta^\top \Sigma_w^{-1} \delta}{\sqrt{\delta^\top \Sigma_w^{-1} \delta}} \right) \left[\frac{1}{\sqrt{\delta^\top \Sigma_w^{-1} \delta}} \right] = \frac{\phi(\sqrt{\delta^\top \Sigma_w^{-1} \delta})}{2\sqrt{\delta^\top \Sigma_w^{-1} \delta}}$$

and

$$\frac{\partial f}{\partial \varepsilon_2}(0) = -\frac{1}{4} \phi\left(\frac{-\delta^\top \Sigma_w^{-1} \delta}{\sqrt{\delta^\top \Sigma_w^{-1} \delta}} \right) \left[\frac{-\delta^\top \Sigma_w^{-1} \delta}{(\delta^\top \Sigma_w^{-1} \delta)^{3/2}} \right] = \frac{\phi(\sqrt{\delta^\top \Sigma_w^{-1} \delta})}{4\sqrt{\delta^\top \Sigma_w^{-1} \delta}},$$

where ϕ denotes the standard normal density. It follows that

$$\|\nabla f(0)\|_2 = \frac{\phi(-\sqrt{\delta^\top \Sigma_w^{-1} \delta})}{2\sqrt{\delta^\top \Sigma_w^{-1} \delta}} \| \begin{pmatrix} 1 & 1 & 1/2 \end{pmatrix} \|_2 = \frac{3\, \phi(\sqrt{\delta^\top \Sigma_w^{-1} \delta})}{4\sqrt{\delta^\top \Sigma_w^{-1} \delta}}.$$

We now focus on bounding the error term $\|\varepsilon\|_2$. We have

$$\|\varepsilon\|_2 \leq \|\varepsilon\|_1 = |\varepsilon_1^1| + |\varepsilon_1^2| + |\varepsilon_2|.$$

Applying Theorem 11 proves that with probability at least $1 - \eta$:

$$\begin{aligned}|\varepsilon_1^g| &\leq C\, K_s^2\, (\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1}) + p}{m}}\\ |\varepsilon_2| &\leq C\, K_s^2\, (\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1}) + p}{m}}.\end{aligned}$$

It follows that with probability at least $1 - \eta$:

$$\begin{aligned}|R_{\mathbf{c}} - R_{\text{opt}}| &\leq C\frac{\phi(\sqrt{\delta^\top \Sigma_w^{-1}\delta})}{\sqrt{\delta^\top \Sigma_w^{-1}\delta}}\, K_s^2\, (\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1}) + p}{m}}\\ &\leq C\, \phi(\sqrt{\delta^\top \Sigma_w^{-1}\delta})\, K_s^2\, (\sqrt{\delta^\top \Sigma_w^{-1}\delta} + 1)\sqrt{\frac{\log(\eta^{-1}) + p}{m}}.\end{aligned}$$

This proves the Theorem. □

Theorem 11. *Let $R_{\mathbf{c}}$ be the misclassification error rate* (3.4) *of the compressed LDA decision rule. Then $R_{\mathbf{c}}$ has the form*

$$R_{\mathbf{c}} = \frac{1}{2}\sum_{g=1}^{2}\Phi\left(\frac{\varepsilon_1^g - \delta^\top \Sigma_w^{-1}\delta}{\sqrt{\varepsilon_2 + \delta^\top \Sigma_w^{-1}\delta}}\right),$$

where

$$\begin{aligned}\varepsilon_1^g &= (-1)^g d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_g - \overline{X}_g) - d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d + \delta^\top \Sigma_w^{-1}\delta\\ \varepsilon_2 &= d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\, \Sigma_w\, \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - \delta^\top \Sigma_w^{-1}\delta.\end{aligned}$$

Then the error terms ε_1 and ε_2 have the following upper bounds with probability at least $1 - \eta$:

$$|\varepsilon_1^g| \leq C\, K_s^2(\|\Sigma_w^{-1/2}\delta\|_2 + \|\Sigma_w^{-1/2}\delta\|_2^2)\sqrt{\frac{\log(\eta^{-1}) + p}{m}},$$

and

$$|\varepsilon_2| \leq C\, K_s^2 (\|\Sigma_w^{-1/2}\delta\|_2 + \|\Sigma_w^{-1/2}\delta\|_2^2)\sqrt{\frac{\log(\eta^{-1}) + p}{m}}.$$

Here, $C > 0$ is an absolute constant, and $K_s = \{s\log(1+s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q^g_{i,j}/\sqrt{s}$- the entries of the compression matrices.

Proof of Theorem 11. We have

$$|\varepsilon_1^g| \leq \underbrace{|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_g - \overline{X}_g)|}_{(I)} + \underbrace{|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - \delta^\top \Sigma_w^{-1}\delta|}_{(II)}$$

We first bound (I). Consider

$$\begin{aligned} |(I)| = |d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_g - \overline{X}_g)| &= |d^\top \Sigma_w^{-1/2}(\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2})\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)| \\ &\leq \underbrace{\|d^\top \Sigma_w^{-1/2}\|_2}_{A_1} \underbrace{\|\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2}\|_{\text{op}}}_{A_2} \underbrace{\|\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\|_2}_{A_3}. \end{aligned}$$

We bound $A_1 - A_3$ separately.

For A_1, by Assumptions 1 and 6, $\Sigma_w^{-1/2} d \sim N(\Sigma_w^{-1/2}\delta, n^{-1}I_p)$. By the triangle inequality and Proposition 1.1 of [73], the following holds with probability at least $1-\eta$ for any $\eta \in (0, e^{-1})$:

$$\begin{aligned} \|\Sigma_w^{-1/2} d\|_2 &\leq \|\Sigma_w^{-1/2}\delta\|_2 + \|\Sigma_w^{-1/2}(d-\delta)\|_2 \\ &\leq \|\Sigma_w^{-1/2}\delta\|_2 + \left(\frac{p}{n} + \frac{2\sqrt{p\log(\eta^{-1})}}{n} + \frac{2\log(\eta^{-1})}{n}\right)^{1/2} \\ &\leq \|\Sigma_w^{-1/2}\delta\|_2 + \left(\frac{p\log(\eta^{-1})}{n} + \frac{2\sqrt{p\log(\eta^{-1})}}{n} + \frac{2p\log(\eta^{-1})}{n}\right)^{1/2} \\ &\leq \|\Sigma_w^{-1/2}\delta\|_2 + C\sqrt{\frac{p\log(\eta^{-1})}{n}} \end{aligned}$$

We now bound A_2. By Theorem 5, the following inequality holds with probability at least

$1-\eta/3:$

$$\|\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2}\|_{\text{op}} \le \|I_p\|_{\text{op}} + \|\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2} - I_p\|_{\text{op}} \le 1 + C_2\, K_s^2 \sqrt{\frac{\log(\eta^{-1})+p}{m}}.$$

We now bound A_3. By Assumptions 1 and 6, $\Sigma_w^{-1/2}(\mu_g - \overline{X}_g) \sim N(0, n_g^{-1} I_p)$. By Proposition 1.1 of [73], the following holds with probability at least $1-\eta$:

$$\|\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\|_2 \le C\sqrt{\frac{p\log(\eta^{-1})}{n}}.$$

Combining the bounds for A_1-A_3, with probability at least $1-\eta$:

$$\begin{aligned}
&|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}(\mu_g - \overline{X}_g)| \\
&\le C\left(\|\Sigma_w^{-1/2}\delta\|_2 + C\sqrt{\frac{p\log(\eta^{-1})}{n}}\right)\left(1 + C_2\, K_s^2 \sqrt{\frac{\log(\eta^{-1})+p}{m}}\right)\sqrt{\frac{p\log(\eta^{-1})}{n}} \\
&\le CK_s^2\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}},
\end{aligned} \tag{3.6}$$

where the last inequality came from absorbing lower-order terms into the absolute constant C.

We now bound (II). By the triangle inequality and Theorems 12–13, with probability at least $1-\eta$:

$$\begin{aligned}
|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - \delta^\top \Sigma_w^{-1}\delta| &\le |d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w^{-1} d| + |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta| \\
&\le C_1\, K_s^2\, \|\Sigma_w^{-1/2}\delta\|_2^2 \sqrt{\frac{\log(\eta^{-1})+p}{m}} + C_2\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}} \\
&\le C\,(K_s^2\, \|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1})+p}{m}}.
\end{aligned}$$

For $s \le 0.8$, we have $K_s \ge 1$. Thus,

$$\begin{aligned}
|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - \delta^\top \Sigma_w^{-1}\delta| &\le C\,(K_s^2\, \|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1})+p}{m}} \\
&\le C\, K_s^2\,(\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1})+p}{m}}.
\end{aligned} \tag{3.7}$$

Combining (3.6) and (3.7) gives with probability at least $1-\eta$:

$$\begin{aligned}|\varepsilon_1^g| &\leq C_1\, K_s^2\, (\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1})+p}{m}} + C_2 K_s^2 \|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}}\\ &\leq C\, K_s^2\, (\|\Sigma_w^{-1/2}\delta\|_2^2 + \|\Sigma_w^{-1/2}\delta\|_2)\sqrt{\frac{\log(\eta^{-1})+p}{m}},\end{aligned}$$

where the lower-order term has been absorbed into the absolute constant C_1.

We now focus on bounding ε_2. The triangle inequality gives

$$|\varepsilon_2| = |d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\,\Sigma_w\,\widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - \delta^\top \Sigma_w^{-1}\delta| \leq \underbrace{|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\,\Sigma_w\,\widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w^{-1} d|}_{A_1} + \underbrace{|d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta|}_{A_2}.$$

We bound A_1-A_2 separately.

First consider A_1. Using identity $I_p = \Sigma_w^{-1/2}\Sigma_w^{1/2}$ gives

$$\begin{aligned}|A_1| &= |d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w\widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w d|\\ &= |d^\top \Sigma_w^{-1/2}(\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2})(\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2})\Sigma_w^{-1/2} d - d^\top \Sigma_w^{-1} d|\\ &\leq \|\Sigma_w^{-1/2} d\|_2^2\, \|(\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2})^2 - I_p\|_{\text{op}}.\end{aligned}$$

Let $A = \Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2}$. Then $\|(\Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2})^2 - I_p\|_{\text{op}}$ is bounded above by

$$\begin{aligned}\|I_p - A^2\|_{\text{op}} &= \|(I_p + A)(I_p - A)\|_{\text{op}}\\ &\leq \|2I_p + (A - I_p)\|_{\text{op}}\|I_p - A\|_{\text{op}}\\ &\leq [2 + \|I_p - A\|_{\text{op}}]\, \|I_p - A\|_{\text{op}}.\end{aligned}$$

Using the assumption that $\|I_p - A\|_{\text{op}} < 1$ and Theorem 14, we have with probability at least $1-\eta$:

$$\|I_p - A^2\|_{\text{op}} < 3\|I_p - A\|_{\text{op}} \leq C\, K_s^2 \sqrt{\frac{\log(\eta^{-1})+p}{m}} \tag{3.8}$$

for some absolute constant $C > 0$.

By Theorem 13, the following holds with probability at least $1-\eta/2$:

$$\|\Sigma_w^{-1/2}d\|_2^2 \leq \|\Sigma_w^{-1/2}\delta\|_2^2 + |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta| \leq \|\Sigma_w^{-1/2}\delta\|_2^2 + C\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\,\log(\eta^{-1})}{n}}. \tag{3.9}$$

Combining (3.8) and (3.9) proves that the following bound on A_1 holds with probability at least $1-\eta$:

$$\begin{aligned}
&|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w d| \\
&\leq \left(\|\Sigma_w^{-1/2}\delta\|_2^2 + C\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\,\log(\eta^{-1})}{n}}\right) C\,K_s^2\sqrt{\frac{\log(\eta^{-1})+p}{m}} \\
&\leq C\,K_s^2(\|\Sigma_w^{-1/2}\delta\|_2 + \|\Sigma_w^{-1/2}\delta\|_2^2)\sqrt{\frac{\log(\eta^{-1})+p}{m}}.
\end{aligned} \tag{3.10}$$

To bound A_2, Theorem 13 proves that with probability at least $1-\eta/2$,

$$|A_2| = |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta| \leq C\|\Sigma_w^{-1}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}}.$$

Since A_2 is a smaller-order term compared to (3.10), we absorb it into the absolute constant C. Thus, with probability at least $1-\eta$:

$$|\varepsilon_2| \leq C\,K_s^2(\|\Sigma_w^{-1/2}\delta\|_2 + \|\Sigma_w^{-1/2}\delta\|_2^2)\sqrt{\frac{\log(\eta^{-1})+p}{m}}.$$

This completes the proof. □

Theorem 12. *Let the samples $X \in \mathbb{R}^{n\times p}$ be distributed according to Assumption 1. Let d and $\delta \in \mathbb{R}^p$ be as in Definition 6, and let $\widehat{\Sigma}_{w,\mathbf{c}}$ be the compressed within-group covariance matrix. Then with probability at least $1-\eta$,*

$$|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w^{-1} d| \leq C\,K_s^2\,\|\Sigma_w^{-1/2}\delta\|_2^2\,\sqrt{\frac{\log(\eta^{-1})+p}{m}},$$

where $C > 0$ is an absolute constant, and $K_s = \{s \log(1 + s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q^g_{i,j}/\sqrt{s}$.

Proof of Theorem 12. We have

$$\begin{aligned}|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w^{-1} d| &= |d^\top \Sigma_w^{-1/2} \Sigma_w (\widehat{\Sigma}_{w,\mathbf{c}}^{-1} - \Sigma_w^{-1}) \Sigma_w^{1/2} \Sigma_w^{-1/2} d| \\ &\leq \|\Sigma_w^{-1/2} d\|_2^2 \, \|\Sigma_w^{1/2} \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w^{1/2} - I_p\|_{\text{op}}.\end{aligned}$$

By Theorem 14, with probability at least $1 - \eta/2$,

$$\|\Sigma_w^{1/2} \widehat{\Sigma}_{w,\mathbf{c}}^{-1} \Sigma_w^{1/2} - I_p\|_{\text{op}} \leq C\, K_s^2 \sqrt{\frac{\log(\eta^{-1}) + p}{m}}.$$

for some absolute constant $C > 0$, and where $K_s = \{s \log(1+s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q^g_{i,j}/\sqrt{s}$ by Lemma 18.

By the triangle inequality and Theorem 13, with probability at least $1 - \eta$:

$$\begin{aligned}\|\Sigma_w^{-1/2} d\|_2^2 &= |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1} \delta + \delta^\top \Sigma_w^{-1} \delta| \\ &\leq \|\Sigma_w^{-1/2} \delta\|_2^2 + |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1} \delta| \leq \|\Sigma_w^{-1/2} \delta\|_2^2 + C \|\Sigma_w^{-1/2} \delta\|_2 \sqrt{\frac{p \log(\eta^{-1})}{n}}.\end{aligned}$$

Combining the two displays above and absorbing the lower order term into the absolute constant C, we have that with probability at least $1 - \eta$

$$\begin{aligned}|d^\top \widehat{\Sigma}_{w,\mathbf{c}}^{-1} d - d^\top \Sigma_w^{-1} d| &\leq \left(\|\Sigma_w^{-1/2} \delta\|_2^2 + C \|\Sigma_w^{-1/2} \delta\|_2 \sqrt{\frac{p \log(\eta^{-1})}{n}} \right) C_1\, K_s^2 \sqrt{\frac{\log(\eta^{-1}) + p}{m}} \\ &\leq C\, K_s^2 \|\Sigma_w^{-1/2} \delta\|_2^2 \sqrt{\frac{\log(\eta^{-1}) + p}{m}}.\end{aligned}$$

□

Theorem 13. *Let the samples in $X \in \mathbb{R}^{n \times p}$ be distributed according to Assumption 1, and let d and δ be as in Definition 6. Then for $\eta \in (0, e^{-1})$, the following upper bound holds with probability* *at least* $1-\eta$,

$$|d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta| \le C\|\Sigma_w^{-1/2}\delta\|_2 \sqrt{\frac{p\log(\eta^{-1})}{n}}$$

for some absolute constant $C > 0$.

Proof of Theorem 13. Completing the square gives

$$\begin{aligned} |d^\top \Sigma_w^{-1} d - \delta^\top \Sigma_w^{-1}\delta| &= |(d-\delta)^\top \Sigma_w^{-1}(d-\delta) + 2(d-\delta)^\top \Sigma_w^{-1}\delta| \\ &\le \|\Sigma_w^{-1/2}(d-\delta)\|_2^2 + 2\|\Sigma_w^{-1/2}\delta\|_2\, \|\Sigma_w^{-1/2}(d-\delta)\|_2. \end{aligned}$$

Assumptions 1 and 6 give $\Sigma_w^{-1/2}(d-\delta) \sim N(0, n^{-1}I_p)$. By Proposition 1.1 of [73], with probability at least $1-\eta$,

$$\|\Sigma_w^{-1/2}(d-\delta)\|_2^2 \le \frac{p}{n} + \frac{2\sqrt{p\log(\eta^{-1})}}{n} + \frac{2\log(\eta^{-1})}{n}.$$

For $\eta \in (0, e^{-1})$, we have $\log(\eta^{-1}) \ge 1$. It follows that

$$\begin{aligned} \|\Sigma_w^{-1/2}(d-\delta)\|_2^2 &\le \frac{p}{n} + \frac{2\sqrt{p\log(\eta^{-1})}}{n} + \frac{2\log(\eta^{-1})}{n} \\ &\le \frac{p\log(\eta^{-1})}{n} + \frac{2\sqrt{p\log(\eta^{-1})}}{n} + \frac{2p\log(\eta^{-1})}{n} \\ &\le C\frac{p\log(\eta^{-1})}{n}. \end{aligned}$$

Then

$$\begin{aligned} &\|\Sigma_w^{-1/2}(d-\delta)\|_2^2 + 2\|\Sigma_w^{-1/2}\delta\|_2\, \|\Sigma_w^{-1/2}(d-\delta)\|_2 \\ &\le C_1\frac{p\log(\eta^{-1})}{n} + C_2\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}} \\ &\le C\|\Sigma_w^{-1/2}\delta\|_2\sqrt{\frac{p\log(\eta^{-1})}{n}}. \end{aligned}$$

□

Theorem 14 (Inverse Covariance Bound). *Let the samples $X \in \mathbb{R}^{n\times p}$ be distributed according to Assumption 1 with shared covariance $\Sigma_w \in \mathbb{R}^{p\times p}$. Let $\widehat{\Sigma}_{w,\mathbf{c}}$ be the within-group sample covariance matrix of the compressed data with sparsity parameter $s > 0$. Then with probability at least $1-\eta$,*

$$\|I_p - \Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2}\|_{op} \leq C\, K_s^2\sqrt{\frac{\log(\eta^{-1})+p}{m}}$$

for some absolute constant $C > 0$, and where $K_s = \{s\log(1+s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q^g_{i,j}/\sqrt{s}$.

Proof. For $A := \Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2}$, the above is of the form $\|\Sigma_w^{-1}\|_{\text{op}}\|A^{-1}-I\|_{\text{op}}$. By Theorem 15, $\|I-A\|_{\text{op}} < 1$ with high probability. Then A has the geometric sum expansion of its inverse $A^{-1} = \sum_{k=0}^{\infty}(I-A)^k$. Thus,

$$\begin{aligned}
\|I_p - A^{-1}\|_{\text{op}} &= \left\|I_p - \sum_{k=0}^{\infty}(I_p-A)^k\right\|_{\text{op}} = \left\|\sum_{k=1}^{\infty}(I_p-A)^k\right\|_{\text{op}} \\
&\leq \sum_{k=1}^{\infty}\|I_p-A\|_{\text{op}}^k = \sum_{k=0}^{\infty}\|I_p-A\|_{\text{op}}^k - 1 \\
&= \frac{1}{1-\|I_p-A\|_{\text{op}}} - 1 = \frac{\|I_p-A\|_{\text{op}}}{1-\|I_p-A\|_{\text{op}}} = \|I_p-A\|_{\text{op}} + o_p(\|I_p-A\|_{\text{op}}),
\end{aligned}$$

where the last equality comes from the Taylor Expansion of the function $t/(1-t)$ centered at 0.

Applying Theorem 15 and absorbing the lower-order $o_p(\|I_p-A\|_{\text{op}})$ into the absolute constant C proves that with probability at least $1-\eta$,

$$\|I_p - \Sigma_w^{1/2}\widehat{\Sigma}_{w,\mathbf{c}}^{-1}\Sigma_w^{1/2}\|_{\text{op}} \leq C\, K_s^2\sqrt{\frac{\log(\eta^{-1})+p}{m}}$$

□

Theorem 15 (Covariance Bound). *Let the samples $X \in \mathbb{R}^{n\times p}$ be distributed according to Assumption 1 with shared covariance $\Sigma_w \in \mathbb{R}^{p\times p}$. Let $\widehat{\Sigma}_{w,\mathbf{c}} \in \mathbb{R}^{p\times p}$ be the within-group sample* *covariance matrix of the compressed data with sparsity parameter $s > 0$. Then with probability at least $1 - \eta$:*

$$\|\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p\|_{op} \leq C\, K_s^2 \sqrt{\frac{\log(\eta^{-1}) + p}{m}}, \tag{3.11}$$

for some absolute constant $C > 0$, and where $K_s = \{s \log(1 + s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q_{i,j}^g/\sqrt{s}$.

Proof of Theorem 15. By the definition of $\widehat{\Sigma}_{w,\mathbf{c}}$,

$$
\begin{aligned}
&\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} \\
&= \frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Sigma_w^{-1/2}(\mathbf{x}_{j,\mathbf{c}}^g - \overline{X}_g)(\mathbf{x}_{j,\mathbf{c}}^g - \overline{X}_g)^\top\Sigma_w^{-1/2} \\
&= \frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Sigma_w^{-1/2}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g(\mathbf{x}_i^g - \overline{X}_g) + \overline{X}_g - \overline{X}_g\Bigg) \\
&\qquad\qquad\qquad \Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g(\mathbf{x}_\ell^g - \overline{X}_g) + \overline{X}_g - \overline{X}_g\Bigg)^\top\Sigma_w^{-1/2} \\
&= \frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g\Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g + \mu_g - \overline{X}_g)\Bigg) \\
&\qquad\qquad\qquad \Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g\Sigma_w^{-1/2}(\mathbf{x}_\ell^g - \mu_g + \mu_g - \overline{X}_g)\Bigg)^\top \\
&= \underbrace{\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g\Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g)\Bigg)\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g\Sigma_w^{-1/2}(\mathbf{x}_\ell^g - \mu_g)\Bigg)^\top}_{A_1} \\
&\qquad -\underbrace{\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g\Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g)\Bigg)\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\Bigg)^\top}_{A_2} \\
&\qquad -\underbrace{\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\Bigg)\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g\Sigma_w^{-1/2}(\mathbf{x}_\ell^g - \mu_g)\Bigg)^\top}_{A_3} \\
&\qquad +\underbrace{\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q_{j,i}^g\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\Bigg)\Bigg(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q_{j,\ell}^g\Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\Bigg)^\top}_{A_4}.
\end{aligned}
$$

We bound $A_1 - A_4$ separately. We do this by considering a fixed $v \in \mathbb{R}^p$ with norm $\|v\|_2 = 1$. We first bound each $v^\top A_i v$ and then generalize to a norm bound using an ϵ-net argument.

Consider

$$\begin{aligned}
v^\top A_1 v &= \frac{1}{2}\sum_{g=1}^{2} v^\top \left[\frac{1}{\sqrt{n_g}}\frac{1}{\sqrt{n_g}} \sum_{i,\ell=1}^{n_g} \left\{ \frac{1}{m_g}\sum_{j=1}^{m_g} \frac{1}{s} Q^g_{j,i} Q^g_{j,\ell} \right\} \Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g)(\mathbf{x}_j^g - \mu_g)^\top \Sigma_w^{-1/2} \right] v \\
&= \frac{1}{2}\sum_{g=1}^{2}\sum_{i,\ell=1}^{n_g} \left\{ \frac{1}{m_g}\sum_{j=1}^{m_g} \frac{1}{s} Q^g_{j,i} Q^g_{j,\ell} \right\} \frac{1}{\sqrt{n_g}} \left\langle \Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g)\,,\, v \right\rangle \frac{1}{\sqrt{n_g}} \left\langle \Sigma_w^{-1/2}(\mathbf{x}_j^g - \mu_g)\,,\, v \right\rangle \\
&= \frac{1}{2}\sum_{g=1}^{2} \frac{1}{n_g\, m_g} Z^{g\top} R_g Z^g,
\end{aligned}$$

where $Z^g \in \mathbb{R}^{n_g}$ is the vector with i-th coordinate $\left\langle \Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g), v \right\rangle$, and $R_g = \frac{1}{s} Q^{g\top} Q^g \in \mathbb{R}^{n_g \times n_g}$. By Assumption 1, $Z^g \sim N(0, I_{n_g})$. By Lemma 15, with probability at least $1-\eta$:

$$|v^\top (A_1 - I_p) v| = |v^\top A_1 v - 1| = \left| \frac{1}{2}\sum_{g=1}^{2} \frac{1}{n_g\, m_g} Z^{g\top} R_g Z^g - 1 \right| \le C\, K_s^2 \sqrt{\frac{\log(\eta^{-1})}{m}}. \tag{3.12}$$

The terms A_2 and A_3 are transposes of each other, and so we handle them simultaneously. Left and right multiplying by v gives

$$\begin{aligned}
&\frac{1}{2}\sum_{g=1}^{2} v^\top \left[\frac{1}{m_g}\sum_{j=1}^{m_g} \left(\frac{1}{\sqrt{n_g s}} \sum_{i=1}^{n_g} Q^g_{i,j} \Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g) \right) \left(\frac{1}{\sqrt{n_g s}} \sum_{\ell=1}^{n_g} Q^g_{\ell,j} \Sigma_w^{-1/2}(\mu_g - \overline{X}_g) \right)^\top \right] v \\
&= \frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g} \left(\frac{1}{\sqrt{n_g s}} \sum_{i=1}^{n_g} Q^g_{i,j} \left\langle \Sigma_w^{-1/2}(\mathbf{x}_i^g - \mu_g)\,,\, v \right\rangle \right) \left(\frac{1}{\sqrt{n_g s}} \sum_{\ell=1}^{n_g} Q^g_{\ell,j} \right) \left\langle \Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\,,\, v \right\rangle .
\end{aligned}$$

By Assumption 1, $\left\langle \Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\,,\, v \right\rangle \sim N(0, n_g^{-1})$. By the Gaussian concentration inequality, with probability at least $1 - \eta/3$:

$$\left| \left\langle \Sigma_w^{-1/2}(\mu_g - \overline{X}_g)\,,\, v \right\rangle \right| \le C \sqrt{\frac{\log(\eta^{-1})}{n_g}} = C' \sqrt{\frac{\log(\eta^{-1})}{n}} \tag{3.13}$$

for some absolute constants $C, C' > 0$. The last equality comes from Assumption 6.

By the general Hoeffding's Inequality, Theorem 2.6.3 of [74], with probability at least $1-\eta/3$:

$$\left|\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g} Q^g_{\ell,j}\right| \leq CK_s\sqrt{\log(\eta^{-1})}, \tag{3.14}$$

where $K_s = \{s\log(1+s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q^g_{i,j}/\sqrt{s}$ by Lemma 18.

Lastly,

$$\begin{aligned}
&\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\left(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g} Q^g_{i,j}\left\langle \Sigma_w^{-1/2}(\mathbf{x}^g_i-\mu_g)\,,\,v\right\rangle\right)\\
&=\frac{1}{2}\sum_{g=1}^{2}\frac{1}{\sqrt{n_g}}\sum_{i=1}^{n_g}\left(\frac{1}{m_g}\sum_{j=1}^{m_g}\frac{1}{\sqrt{s}}Q^g_{i,j}\right)Z^g_i,
\end{aligned} \tag{3.15}$$

where the Z^g_i are as above. Let $X_{ig} = m_g^{-1}\sum_{j=1}^{m_g} Q^g_{i,j}/\sqrt{s}$, then by Lemma 18 the sub-Gaussian norm of X_{ig} is $K_s/\sqrt{m}$. Conditioning on vectors $Z^g = (Z^g_1,\ldots,z^g_{n_g})$, and applying Hoeffding's Inequality to $Q^g_{i,j}$ gives that with probability at least $1-\eta/6$:

$$\begin{aligned}
\left|\frac{1}{2}\sum_{g=1}^{2}\frac{1}{\sqrt{n_g}}\sum_{i=1}^{n_g}\left(\frac{1}{m_g}\sum_{j=1}^{m_g}\frac{1}{\sqrt{s}}Q^g_{i,j}\right)Z^g_i\right| &= \left|\sum_{g=1}^{2}\sum_{i=1}^{n_g}\frac{1}{2\sqrt{n_g}}Z^g_i X_{ig}\right|\\
&\leq CK_s\sqrt{\frac{\log(\eta^{-1})}{m}}\left(\frac{\|Z^1\|_2^2+\|Z^2\|_2^2}{n}\right)^{1/2}.
\end{aligned}$$

Let $Z = \begin{pmatrix} Z^{1\top} & Z^{2\top}\end{pmatrix}^\top \in \mathbb{R}^n$. By Theorem 3.1.1 of [74],

$$\mathbb{P}\left(\left|\frac{1}{\sqrt{n}}\|Z\|_2 - 1\right| \geq t\right) = \mathbb{P}(|\|Z\|_2-\sqrt{n}| \geq \sqrt{n}\,t) \leq 2\exp(-c\,n\,t^2).$$

This is equivalent to the following upper bound holding with probability at least $1-\eta/6$:

$$\left(\frac{\|Z^1\|_2^2+\|Z^2\|_2^2}{n}\right)^{1/2} = \frac{1}{\sqrt{n}}\|Z\|_2 \leq 1 + C\sqrt{\frac{\log(\eta^{-1})}{n}},$$

where $C>0$ is an absolute constant. Combining the above two displays gives the following bound for (3.15), which holds with probability at least $1-\eta/3$:

$$\begin{aligned}\left|\frac{1}{2}\sum_{g=1}^{2}\frac{1}{\sqrt{n_g}}\sum_{i=1}^{n_g}\left(\frac{1}{m_g}\sum_{j=1}^{m_g}\frac{1}{\sqrt{s}}Q^g_{i,j}\right)Z^g_i\right| &\leq C_1K_s\sqrt{\frac{\log(\eta^{-1})}{m}}\left(1+C_2\sqrt{\frac{\log(\eta^{-1})}{n}}\right)\\ &\leq C\,K_s\sqrt{\frac{\log(\eta^{-1})}{m}}.\end{aligned} \tag{3.16}$$

Putting (3.13), (3.14) and (3.16) together shows that with probability at least $1-\eta$,

$$\begin{aligned}&|v^\top A_2 v| =\\ &\left|\frac{1}{m}\sum_{g=1}^{2}\sum_{j=1}^{m_g}\left(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g}Q^g_{i,j}\left\langle\Sigma_w^{-1/2}(\mathbf{x}^g_i-\mu_g)\,,\,v\right\rangle\right)\left(\frac{1}{\sqrt{n_g s}}\sum_{\ell=1}^{n_g}Q^g_{\ell,j}\right)\left\langle\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\,,\,v\right\rangle\right|\\ &\leq\left(C_1K_s\sqrt{\frac{\log(\eta^{-1})}{m}}\right)C_2K_s\sqrt{\log(\eta^{-1})}\left(C_3\sqrt{\frac{\log(\eta^{-1})}{n}}\right)\\ &\leq CK_s^2\,\frac{\log(\eta^{-1})}{\sqrt{n}}\sqrt{\frac{\log(\eta^{-1})}{m}}\leq CK_s^2\sqrt{\frac{\log(\eta^{-1})}{m}}.\end{aligned}$$

We have used Assumption 8 in the last inequality.

For A_4, left and right multiplying by v gives

$$\begin{aligned}&v^\top A_4 v\\ &=v^\top\left[\frac{1}{2}\sum_{g=1}^{2}\frac{1}{m_g}\sum_{j=1}^{m_g}\left(\frac{1}{\sqrt{n_g p}}\sum_{i=1}^{n_g}Q^g_{i,j}\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\right)\left(\frac{1}{\sqrt{n_g p}}\sum_{\ell=1}^{n_g}Q^g_{\ell,j}\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\right)^\top\right]v\\ &=\frac{1}{2}\sum_{g=1}^{2}\left\{\frac{1}{m_g}\sum_{j=1}^{m_g}\left(\frac{1}{\sqrt{n_g p}}\sum_{i=1}^{n_g}Q^g_{i,j}\right)^2\right\}\left\langle\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\,,\,v\right\rangle^2,\end{aligned}$$

where the last equality is true since $\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)$ is independent of i, j, and ℓ.

By Assumption 1, $\left\langle\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\,,\,v\right\rangle\sim N(0,n_g^{-1})$. The Gaussian concentration inequality proves that with probability at least $1-\eta/2$:

$$\left|\left\langle\Sigma_w^{-1/2}(\mu_g-\overline{X}_g)\,,\,v\right\rangle\right|^2\leq C\frac{\log(\eta^{-1})}{n_g}.$$

The squared terms

$$\left(\frac{1}{\sqrt{n_g p}}\sum_{i=1}^{n_g} Q_{i,j}^g\right)^2$$

are sub-Exponential because they are the squares of sub-Gaussian random variables. By Lemma 2.7.6 of [74], the sub-Exponential norm satisfies

$$\left\|\left(\frac{1}{\sqrt{n_g p}}\sum_{i=1}^{n_g} Q_{i,j}^g\right)^2\right\|_{\Psi_1} = \left\|\frac{1}{\sqrt{n_g p}}\sum_{i=1}^{n_g} Q_{i,j}^g\right\|_{\Psi_2}^2 = C\, K_s^2,$$

where $C > 0$ is an absolute constant and K_s is the sub-Gaussian norm of $Q_{i,j}^g/\sqrt{s}$ by Lemma 18. Thus, by Bernstein's Inequality, with probability at least $1-\eta$:

$$\left|\frac{1}{m_g}\sum_{j=1}^{m_g}\left(\frac{1}{\sqrt{n_g s}}\sum_{i=1}^{n_g} Q_{j,i}^g\right)^2\right| \le C\, K_s^2 \, \max\left\{\frac{\log(\eta^{-1})}{m_g}\, ,\, \sqrt{\frac{\log(\eta^{-1})}{m_g}}\right\} \le C\, K_s^2 \sqrt{\frac{\log(\eta^{-1})}{m_g}}.$$

Combining the above displays, with probability at least $1-\eta$, $|v^\top A_4 v|$ is bounded above by

$$|v^\top A_4 v| \le C K_s^2 \sqrt{\frac{\log(\eta)^{-1}}{m_g}}\,\frac{\log(\eta^{-1})}{n_g} \le C K_s^2\,\frac{\log(\eta^{-1})}{n},$$

where we have used Assumptions 6 and 8.

Combining the above bounds for $A_1 - A_4$ shows that with probability at least $1-\eta$:

$$\begin{aligned}|v^\top(\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p)v| &\le C_1\, K_s^2 \sqrt{\frac{\log(\eta^{-1})}{m}} + C_2\, K_s^2 \sqrt{\frac{\log(\eta^{-1})}{m}} + C_3 K_s^2\,\frac{\log(\eta^{-1})}{n} \\ &= C\, K_s^2 \sqrt{\frac{\log(\eta^{-1})}{m}}.\end{aligned}$$

We now generalize to a norm bound via an ε-net argument. Let $\mathcal{N}$ be a $1/3$-net on the unit sphere of $\mathbb{R}^p$. There exists a $1/3$-net such that $|\mathcal{N}| \leq 7^p$ (see Corollary 4.2.13 of [74]). Thus,

$$\begin{aligned}
\mathbb{P}\bigg(\sup_{v\in\mathcal{N}} |v^\top(\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p)v| \geq t\bigg) &= \mathbb{P}\bigg(\bigcup_{v\in\mathcal{N}} \{|v^\top(\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p)v| \geq t\}\bigg) \\
&\leq \sum_{v\in\mathcal{N}} \mathbb{P}(|v^\top(\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p)v| \geq t) \\
&\leq \sum_{v\in\mathcal{N}} \exp\bigg(-\frac{C\,m\,t^2}{K_s^4}\bigg) \\
&= |\mathcal{N}| \exp\bigg(-\frac{C\,m\,t^2}{K_s^4}\bigg) \\
&\leq \exp(p\log(7))\,\exp\bigg(-\frac{C\,m\,t^2}{K_s^4}\bigg) \\
&= \exp\bigg(C_1\,p - C_2\frac{m\,t^2}{K_s^4}\bigg).
\end{aligned}$$

This tail inequality is equivalent to the following upper bound holding with probability at least $1-\eta:$

$$\begin{aligned}
&\sup_{v\in\mathcal{N}} |v^\top(\Sigma_w^{-1/2}\widehat{\Sigma}_{w,\mathbf{c}}\Sigma_w^{-1/2} - I_p)v| \leq C_1\,K_s^2\sqrt{\frac{\log(\eta^{-1}) + C_2 p}{m}} \\
&\leq C\,K_s^2\,\max\{1, \sqrt{C_2}\}\,\sqrt{\frac{\log(\eta^{-1}) + p}{m}}.
\end{aligned}$$

Absorbing $\max\{1, \sqrt{C_2}\}$ into the absolute constant C_1 gives a uniform bound on the ε-net $\mathcal{N}$. Applying Lemma 14 proves the final reuslt. □

Lemma 14 (page 88 of [74])**.** *Let $\varepsilon \in [0, 1/2)$. Then for any ε-net $\mathcal{N}$ of the unit sphere of $\mathbb{R}^p$, we have*

$$\sup_{v\in\mathcal{N}} |v^\top(\widehat{\Sigma}_{w,\mathbf{c}} - \Sigma_w)v| \leq \|\widehat{\Sigma}_{w,\mathbf{c}} - \Sigma_w\|_{op} \leq \frac{1}{1-2\varepsilon}\sup_{v\in\mathcal{N}} |v^\top(\widehat{\Sigma}_{w,\mathbf{c}} - \Sigma_w)v|.$$

Lemma 15. *For $g = 1, 2$, let $Z^g \sim N(0, I_{n_g})$, let $Q^g \in \mathbb{R}^{m_g\times n_g}$ consist of i.i.d. sparse Rademacher random variables with sparsity parameter s, and let $R_g = Q^{g\top}Q^g/s$. Then with probability at least*

$1-\eta$:

$$\left|\frac{1}{2}\sum_{g=1}^{2}\frac{1}{n_g\,m_g}Z^{g\top}R_gZ^g-1\right|\le C\,K_s^2\sqrt{\frac{\log(\eta^{-1})}{m}},$$

where $C>0$ is an absolute constant, and $K_s=\{s\log(1+s^{-1})\}^{-1/2}$ is the sub-gaussian norm of $Q^g_{i,j}/\sqrt{s}$.

Proof of Lemma 15. By Lemma 17, with probability at least $1-\eta/2$:

$$\begin{aligned}\left|\frac{1}{2}\sum_{g=1}^{2}\frac{1}{n_g\,m_g}Z^{g\top}R_gZ^g-1\right| &\le \frac{1}{2}\sum_{g=1}^{2}\left|\frac{1}{n_g\,m_g}Z^{g\top}R_gZ^g-1\right|\\ &\le \frac{1}{2}\sum_{g=1}^{2}\left(\frac{C}{n_g}\|R_g\|_{\text{op}}\sqrt{\frac{\log(\eta^{-1})}{m_g}}+\left|\frac{1}{n_g\,m_g}\text{tr}(R_g)-1\right|\right)\end{aligned}\tag{3.17}$$

for some absolute constant $C>0$. We bound each term individually.

By Lemma 16, with probability at least $1-\eta/2$:

$$C\sum_{g=1}^{2}\frac{1}{n_g}\|R_g\|_{\text{op}}\sqrt{\frac{\log(\eta^{-1})}{m_g}}\le C\sum_{g=1}^{2}K_s^2\left[1+\sqrt{\frac{\log(\eta^{-1})}{n_g}}\right]\sqrt{\frac{\log(\eta^{-1})}{m_g}}\le C\,K_s^2\sqrt{\frac{\log(\eta^{-1})}{m}},$$

where we have absorbed the lower-order term into the absolute constant C and used Assumption 6.

Since $\text{tr}(R_g)=\|Q^g/\sqrt{s}\|_F^2$, by Hoeffding's Inequality, Theorem 2.6.3 of [74], the following inequalities hold with probability at least $1-\eta/2$:

$$\begin{aligned}&\left|\frac{1}{n_g\,m_g}\text{tr}(R_g)-1\right|\\ &=\frac{1}{2}\sum_{g=1}^{2}\left|\frac{1}{n_g\,m_g}\left\|\frac{1}{\sqrt{s}}Q^g\right\|_F^2-1\right|=\frac{1}{2}\sum_{g=1}^{2}\left|\frac{1}{n_g\,m_g}\sum_{i=1}^{n_g}\sum_{j=1}^{m_g}\left\{\left(\frac{1}{\sqrt{s}}Q^g_{i,j}\right)^2-1\right\}\right|\\ &\le\frac{1}{2}\sum_{g=1}^{2}K_s^2\sqrt{\frac{\log(\eta^{-1})}{n_g\,m_g}}=2\,K_s^2\sqrt{\frac{\log(\eta^{-1})}{n\,m}},\end{aligned}$$

where Assumption 6 was used in the last equality.

Combining the above two displays with (3.17), and absorbing the lower order terms gives

$$\left|\frac{1}{2}\sum_{g=1}^{2}\frac{1}{n_g\, m_g}Z^{g\top}R_gZ^g - 1\right| \leq C\, K_s^2\sqrt{\frac{\log(\eta^{-1})}{m}}$$

with probability at least $1-\eta$ for some absolute constant $C>0$. □

Lemma 16 (Norm Bound). *Let $Q \in \mathbb{R}^{m\times n}$ be a matrix consisting of i.i.d. sparse Rademacher random variables with sparsity parameter s, and let $R = Q^\top Q/s$. Then with probability at least $1-\eta$:*

$$\frac{\|R\|_{op}}{n\, m} \leq C\frac{K_s^2}{m}\left[1+\sqrt{\frac{\log(\eta^{-1})}{n}}\right],$$

where $C>0$ is an absolute constant, and $K_s = \{s\log(1+s^{-1})\}^{-1/2}$ is the sub-gaussian norm of $Q_{i,j}/\sqrt{s}$.

Proof of Lemma 16. By Lemma 18, $K_s = \{s\log(1+s^{-1})\}^{-1/2}$ is the sub-Gaussian norm of $Q_{i,j}/\sqrt{s}$. By Theorem 4.4.5 of [74], with probability at least $1-\eta$:

$$\|R\|_{\text{op}} = \left\|\frac{1}{\sqrt{s}}Q\right\|_{\text{op}}^2 \leq CK_s^2(\sqrt{m}+\sqrt{n}+\sqrt{\log(\eta^{-1})})^2$$

Including the scaling $(n\, m)^{-1}$ gives

$$\begin{aligned}\frac{\|R\|_{\text{op}}}{n\, m} &= \frac{CK_s^2}{n\, m}(\sqrt{m}+\sqrt{n}+\sqrt{\log(\eta^{-1})})^2 = \frac{C\, K_s^2}{m}\left[\sqrt{\frac{m}{n}}+1+\sqrt{\frac{\log(\eta^{-1})}{n}}\right]^2\\ &\leq \frac{CK_s^2}{m}\left[2+\sqrt{\frac{\log(\eta^{-1})}{n}}\right]^2 \leq \frac{CK_s^2}{m}\left[1+\sqrt{\frac{\log(\eta^{-1})}{n}}\right],\end{aligned}$$

where we have expanded the square and absorbed the lower-order terms into the absolute constant $C>0$. □

Lemma 17 (Conditional Hanson-Wright). *Let $Z \sim N(0, I_n)$, and let $R \in \mathbb{R}^{n\times n}$ be a matrix of rank m. Conditioning on R, and for $\eta \in (0, e^{-1})$, the following upper bound holds with probability at least $1-\eta$:*

$$\left|\frac{1}{n\,m} Z^\top R Z - 1\right| \le \frac{C}{n}\|R\|_{op}\sqrt{\frac{\log(\eta^{-1})}{m}} + \left|\frac{1}{n\,m} tr(R) - 1\right|, \tag{3.18}$$

where $C > 0$ is an absolute constant.

Proof of Lemma 17. Since $Z \sim N(0, I_n)$, the conditional expectation equals

$$\mathbb{E}[Z^\top R Z \mid R] = \text{tr}(R\, I_n) + 0^\top R\, 0 = \text{tr}(R).$$

The Hanson-Wright Inequality, Theorem 6.2.1 of [74], gives the conditional tail bound

$$\begin{aligned}\mathbb{P}(|Z^\top R Z - \text{tr}(R)\,| \ \ge t\, n\, m \mid R) &= \mathbb{P}(|Z^\top R Z - \mathbb{E}[Z^\top R\, Z \mid R]| \ge t\, n\, m \mid R) \\ &\le 2\exp\left(- C\min\left(\frac{t^2\, m^2\, n^2}{\|R\|_F^2}, \frac{t\, m\, n}{\|R\|_{\text{op}}}\right)\right)\end{aligned}$$

for some absolute $C > 0$. This is equivalent to the following upper bound holding with probability at least $1-\eta$:

$$\frac{1}{n\,m}|Z^\top R Z - \text{tr}(R)| \le \frac{C}{m\,n}\max\left\{\|R\|_F\sqrt{\log(\eta^{-1})}\,,\ \|R\|_{\text{op}}\log(\eta^{-1})\right\}.$$

Using the fact that $\|R\|_F \le \sqrt{m}\|R\|_{\text{op}}$ and $m \ge \log(\eta^{-1})$ for $\eta \le e^{-1}$, this is further bounded by

$$\begin{aligned}\frac{1}{n\,m}|Z^\top R Z - \text{tr}(R)| &\le \frac{C}{m\,n}\max\left\{\sqrt{m}\|R\|_{\text{op}}\sqrt{\log(\eta^{-1})}\,,\ \|R\|_{\text{op}}\log(\eta^{-1})\right\} \\ &\le \frac{C}{n}\|R\|_{\text{op}}\max\left\{\sqrt{\frac{\log(\eta^{-1})}{m}}\,,\ \frac{\log(\eta^{-1})}{m}\right\} = \frac{C}{n}\|R\|_{\text{op}}\sqrt{\frac{\log(\eta^{-1})}{m}}.\end{aligned} \tag{3.19}$$

Applying the triangle inequality and substituting (3.19) gives the final result:

$$\begin{aligned}\left|\frac{1}{n\,m}Z^\top RZ-1\right| &\le \left|\frac{1}{n\,m}Z^\top RZ-\frac{1}{n\,m}\operatorname{tr}(R)\right|+\left|\frac{\operatorname{tr}(R)}{n\,m}-1\right| \\ &\le \frac{C}{n}\|R\|_{\text{op}}\sqrt{\frac{\log(\eta^{-1})}{m}}+\left|\frac{\operatorname{tr}(R)}{n\,m}-1\right|. \qquad \square\end{aligned}$$

Lemma 18 (Sub-Gaussian Norm). *Let X be sparse Rademacher random variable satisfying for some $s \in (0,1)$*

$$P(X=0)=1-s, \quad P(X=1)=P(X=-1)=s/2.$$

Then the sub-Gaussian norm of X is $K=\{\log(1+s^{-1})\}^{-1/2}$, and the sub-Gaussian norm of $X/\sqrt{s}$ is $K_s=\{s\log(1+s^{-1})\}^{-1/2}$. Additionally, the sub-Gaussian norm of X^2/s is $s^{-1}\{\log(1+s^{-1})\}^{-1/2}$.

Proof. By definition of sub-Gaussian norm,

$$K=\inf\{t>0:\mathbb{E}\exp(X^2/t^2)\le 2\}.$$

Consider for some $t>0$,

$$\mathbb{E}\exp(X^2/t^2)=\exp(0/t^2)(1-s)+\exp(1/t^2)s=1-s+\exp(1/t^2)s.$$

Then $\mathbb{E}\exp(X^2/t^2)\le 2$ is equivalent to

$$\begin{aligned}1-s+\exp(1/t^2)s&\le 2\\ \exp(1/t^2)s&\le 1+s\\ \exp(1/t^2)&\le 1+s^{-1}\\ 1/t^2&\le\log(1+s^{-1})\\ t^2&\ge\{\log(1+s^{-1})\}^{-1}.\end{aligned}$$

The term $K_s = \{s \log(1+s^{-1})\}^{-1/2}$ follows from scaling X by $s^{-1/2}$.

Additionally, the sub-gaussian norm of the squared X^2/s is $\|X^2/s\|_{\Psi_2} = \|X^2\|_{\Psi_2}/s$. Because X has values 0 and ± 1, it follows that $X^4 = X^2$. Thus,

$$\begin{aligned}
\mathbb{E}\exp(X^4/t^2) &\leq 2 \\
\mathbb{E}\exp(X^2/t^2) &\leq 2 \\
\exp(1/t^2)s &\leq 1+s \\
\exp(1/t^2) &\leq 1+s^{-1} \\
t/t^2 &\leq \log(1+s^{-1}) \\
t^2 &\geq \{\log(1+s^{-1})\}^{-1}.
\end{aligned}$$

Hence, the sub-gaussian norm of X^2/s is $s^{-1}\{\log(1+s^{-1})\}^{-1/2}$. □

4. R PACKAGE FOR SPARSE KERNEL OPTIMAL SCORING AND COMPRESSED LINEAR DISCRIMINANT ANALYSIS

4.1 Introduction

biClassify is an R package for adapting Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Kernel Discriminant Analysis to a variety of situations where the conventional methods may not work. In particular, this package implements methodology for the following problems:

1. Linear and Quadratic classification in the large-sample case with small-to-medium sized number of features. The available compressed LDA and QDA methods of Sections 3.2 and 3.4.2 provide alternatives to random sub-sampling which are shown to produce lower mean misclassification error rates and lower standard error in the misclassification error rates.

2. Kernel classification where the data has a medium-to-large number of features. In this case, one would like to learn a non-linear decision boundary and have simultaneous sparse feature selection. The sparse kernel optimal scoring method is presented in Section 2.4.

The following is a vignette manual for instructing researchers how to use the the biClassify R package. Text appearing in the `verbatim font` denotes R code, commands, or function arguments.

4.2 Quick Start

The purpose of this Section is to give the user a quick overview of the package and the types of problems it can be used to solve. Accordingly, we implement only the basic versions of the available methods, and more detailed presentations are given in later sections.

We first load the package.

```
library(biClassify)
```

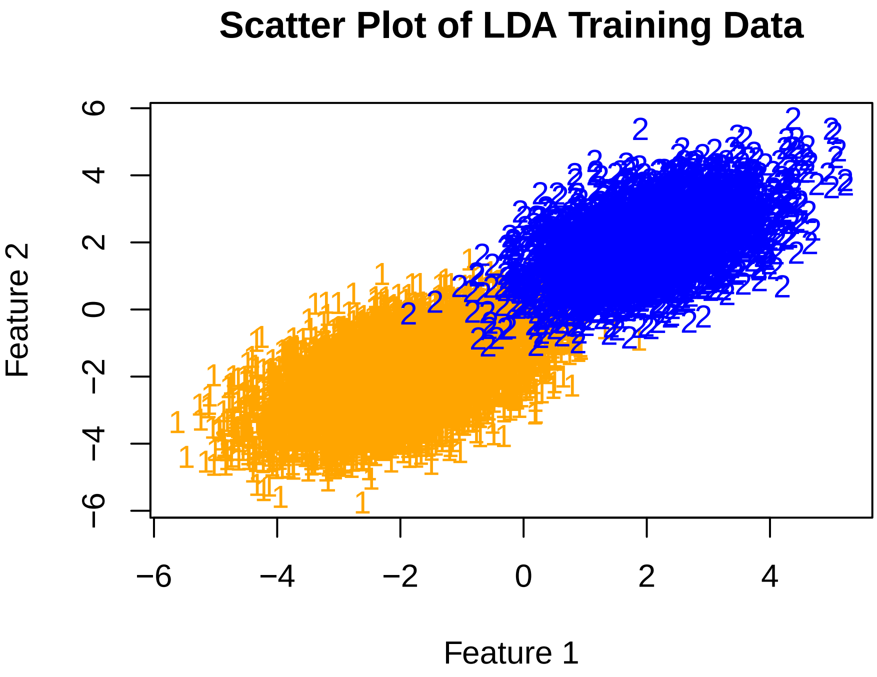

Figure 4.1: Scatter plot of first two features of LDA training data. Classes are distinguished by color and shape.

4.2.1 Quick LDA Example

Our first example illustrates the compressed LDA function on data well-suited for LDA. The first two features of the training data in `LDA_Data` are plotted below:

```
data(LDA_Data)

plot(LDA_Data$TrainData[,2]~LDA_Data$TrainData[,1],
     col = c("orange","blue")[LDA_Data$TrainCat],
     pch = c("1","2")[LDA_Data$TrainCat],
     xlab = "Feature 1",
     ylab = "Feature 2",
     main = "Scatter Plot of LDA Training Data")
```

Figure 4.1 displays the resulting scatter plot.

This data set has $n = 10,000$ training samples with $p = 10$ features. It is normally distributed with class means equal to $\pm\mathbf{1}$ and a shared covariance matrix with entries $\Sigma_{i,j} = (0.5)^{|i-j|}$. The test data was independently generated from the same class distributions and proportions, but it has only $n = 1,000$ samples.

Let us use compressed LDA to predict the test data labels.

```
test_pred <- LDA(TrainData = LDA_Data$TrainData,
                 TrainCat = LDA_Data$TrainCat,
                 TestData = LDA_Data$TestData,
                 Method = "Compressed")
mean(test_pred != LDA_Data$TestCat)
[1] 0
```

The automatic implementation of compressed LDA predicted the Test labels perfectly. However, this is due, in part, to the classes being well-separated and having the same covariance structure. Let us now consider an example of where LDA will not perform well.

4.2.2 Quick QDA Example

Our next example illustrates the compressed QDA function on data well-suited for QDA. The first two features of the training data in QDA Data are plotted below:

```
data(QDA_Data)

plot(QDA_Data$TrainData[,2]~QDA_Data$TrainData[,1],
     col = c("orange","blue")[QDA_Data$TrainCat],
     pch = c("1","2")[QDA_Data$TrainCat],
     xlab = "Feature 1",
     ylab = "Feature 2",
     main = "Scatter Plot of QDA Training Data")
```

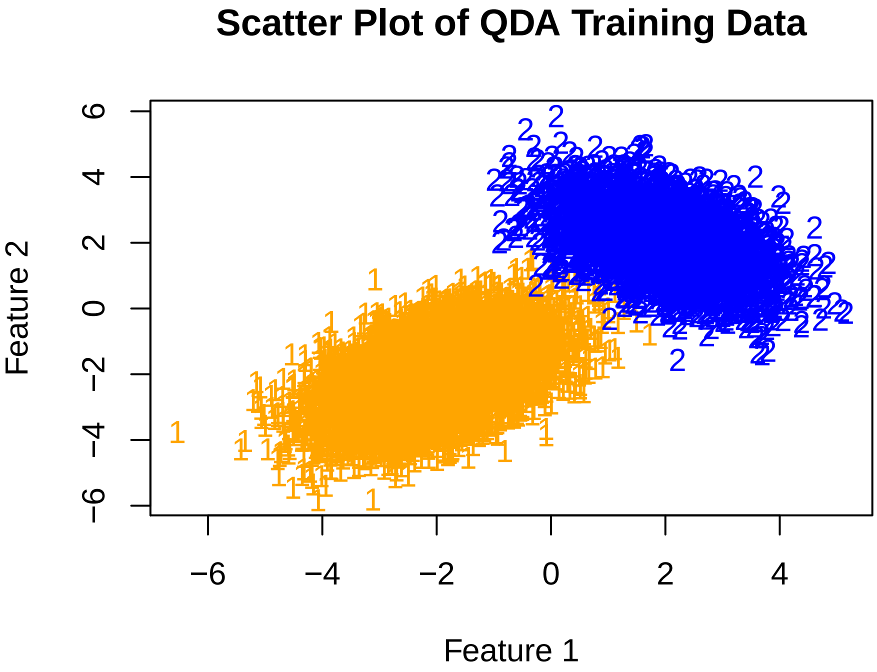

Figure 4.2: Scatter plot of first two features of QDA training data. Classes are distinguished by color and shape.

Figure 4.2 displays the resulting scatter plot.

This data set has $n = 10,000$ training samples with $p = 10$ features and equal class sizes. It is normally distributed with class means equal to $\pm\mathbf{1}$. The class 1 covariance matrix has entries $\Sigma^1_{i,j} = (0.5)^{|i-j|}$, and the class 2 covariance matrix has entries $\Sigma^2_{i,j} = (-0.5)^{|i-j|}$ The test data was independently generated from the same class distributions and proportions, but it has only $n = 1,000$ samples.

A modification of Quadratic Discriminant Analysis is well-suited to such data. The package comes with a function `QDA` for such purposes.

```
test_pred <- QDA(TrainData = QDA_Data$TrainData,
                 TrainCat = QDA_Data$TrainCat,
                 TestData = QDA_Data$TestData,
                 Method = "Compressed")
mean(test_pred != QDA_Data$TestCat)
[1] 0
```

Compressed QDA gives perfect class prediction.

4.2.3 Quick Sparse Kernel Optimal Scoring Example

What happens if the data is not well-suited to either Linear or Quadratic Discriminant Analysis? Moreover, what happens if, in addition to a non-linear decision boundary between classes, there also appear to be variables which do not contribute to group separation?

For example, consider the KOS Data shown below.

```
data(KOS_Data)

par(mfrow = c(1,2))
plot(KOS_Data$TrainData[,2]~KOS_Data$TrainData[,1],
     col = c("orange","blue")[KOS_Data$TrainCat],
     pch = c("1","2")[KOS_Data$TrainCat],
```

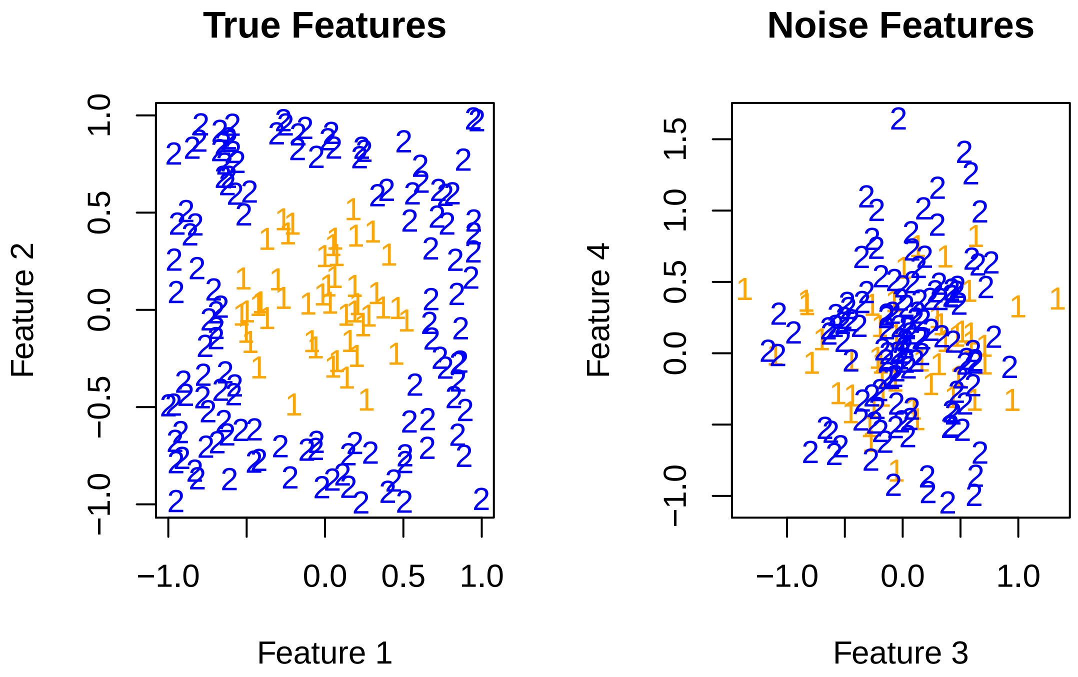

Figure 4.3: Scatter plot of the KOS training data. Classes are distinguished by color and shape. Only the first two features contribute to class separation.

```
     xlab = "Feature 1",
     ylab = "Feature 2",
     main = "True Features")

plot(KOS_Data$TrainData[,4]~KOS_Data$TrainData[,3],
     col = c("orange","blue")[KOS_Data$TrainCat],
     pch = c("1","2")[KOS_Data$TrainCat],
     xlab = "Feature 3",
     ylab = "Feature 4",
     main = "Noise Features")
     par(mfrow = c(1,1))
```

Figure 4.2 displays the resulting scatter plot.

For this data set, neither LDA or QDA would suffice. The function KOS is the sparse kernel optimal scoring algorithm presented in Section 2.4. It is particularly well-suited to such problems, as can be seen from the following.

```
output <- KOS(TrainData = KOS_Data$TrainData,
              TrainCat = KOS_Data$TrainCat,
              TestData = KOS_Data$TestData)
print(output$Weight)
[1] 1 1 0 0


mean(output$Predictions != KOS_Data$TestCat)
[1] 0


plot(output$Dvec,
     main = "Discriminant Vector Coefficients",
     xlab = "Feature Index",
     ylab = "Discriminant Coefficient Value")
```

Figure 4.4 displays the resulting plot of the discriminant vector coefficients.

The output Weight is how much weight the kernel classifier gives to each feature. The weight values lie in $[-1, 1]$, and zero weight means that the feature does not contribute to computing the discriminant function. The KOS function correctly identifies that the first two features are important for class separation, and gives them full weight. It also correctly identifies Features 3 and 4 as being “noise”, and it gives them zero weight.

The output Predictions are the predicted class labels for the test data. As we can see, KOS has perfect classification.

The output Dvec are the coefficients of the kernel discriminant vector (2.3).

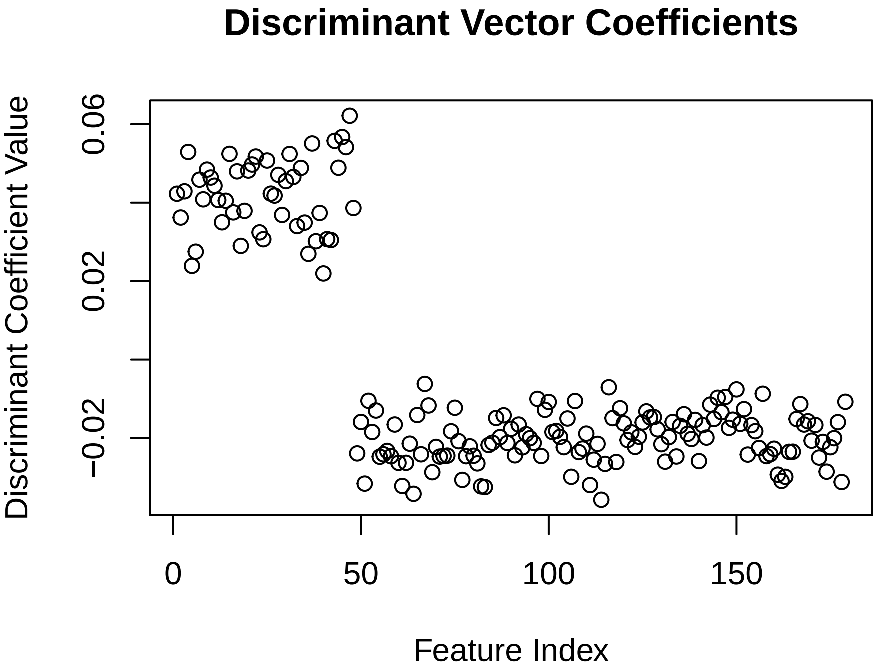

Figure 4.4: Plot of the Discriminant Vector Coefficients generated by the KOS function on the KOS training data.

4.3 Compressed Linear Discriminant Analysis

This Section provides a more in-depth treatment to the Linear Discriminant methods available in biClassify.

There are five separate linear discriminant methods available through the LDA wrapper function:

1. Full Linear Discriminant Analysis, which is LDA trained on the full data [2, Section 11.5].
2. Compressed Linear Discriminant Analysis of Section 3.2.
3. Projected LDA of Section 3.4.1.
4. Subsampled LDA, where LDA is trained on data which is sub-sampled uniformly from both classes.
5. Fast Random Fisher Discriminant Analysis of [62].

The individual methods are invoked by setting the `Method` argument. Let us first load the data for convenience.

```
TrainData <- LDA_Data$TrainData
TrainCat <- LDA_Data$TrainCat
TestData <- LDA_Data$TestData
TestCat <- LDA_Data$TestCat
```

4.3.1 Full LDA

This method is the result of setting `Method` equal to `"Full"`. This method is traditional Linear Discriminant Analysis, as presented in [2, Section 11.5]. No additional parameters need to be supplied, and the code will run as stated.

```
test_pred <- LDA(TrainData, TrainCat, TestData)
table(test_pred)
```

```
test_pred
  1   2
700 300


mean(test_pred != TestCat)
[1] 0
```

The above code produces a list containing a vector of predicted class labels for `TestData`.

4.3.2 Compressed LDA

Compressed LDA seeks to solve the LDA problem on reduced-size data. The details of compressed LDA are contained in Section 3.2. This method is the result of setting `Method` equal to `"Compressed"`. Compressed LDA reduces the group sample amounts from n_1 and n_2 to m_1 and m_2 respectively. It requires the parameters `m1`, `m2`, `s`.

The easiest way to run Compressed LDA is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Compressed", Mode = "Automatic")
table(test_pred)
test_pred
  1   2
700 300


mean(test_pred != TestCat)
[1] 0
```

`Automatic` is the default value for `Mode`, and so one could run

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Compressed")
table(test_pred)
test_pred
  1   2
700 300

mean(test_pred != TestCat)
[1] 0
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the compression amounts m_1, m_2, and sparsity level s to be used in compression. The user will type in the amounts:

```
output <- LDA(TrainData, TrainCat, TestData,
              Method = "Compressed", Mode = "Interactive")
"Please enter the number m1 of group 1 compression samples: "700
"Please enter the number m2 of group 2 compression samples: "300
"Please enter sparsity level s used in compression: "0.01
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to provide all necessary parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Compressed", Mode = "Research",
                 m1 = 700, m2 = 300, s = 0.01)


table(test_pred)
test_pred
  1   2
```

```
700 300
```

```
mean(test_pred != TestCat)
[1] 0
```

WARNING: The argument `Mode` will override any supplied parameters if its value is `Automatic` or `Research`.

4.3.3 Sub-Sampled LDA

Sub-sampled LDA is trains LDA on data sub-sampled uniformly from both classes. To run sub-sampled LDA, set `Method` equal to `Subsampled`. It requires the additional parameters `m1` and `m2`.

The easiest way to run Compressed LDA is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Subsampled", Mode = "Automatic")
table(test_pred)
test_pred
  1   2
700 300
```

`Automatic` is the default value for `Mode`, and so one could simply run

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Subsampled")
table(test_pred)
test_pred
  1   2
700 300
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the sub-sample amounts m_1, m_2 for each group to be used. The user will type in the amounts:

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Subsampled", Mode = "Interactive")
"Please enter the number m1 of group 1 sub-samples: "700
"Please enter the number m2 of group 2 sub-samples: "300
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to give the `LDA` function all parameters.

```
output <- LDA(TrainData, TrainCat, TestData,
              Method = "Subsampled",  Mode = "Research",
              m1 = 700, m2 = 300)


table(output)
output
  1   2
700 300


mean(output != TestCat)
[1] 0
```

WARNING: The argument `Mode` will override any supplied parameters if its value is `Automatic` or `Research`.

4.3.4 Projected LDA

This method is the result of setting `Method` equal to `"Projected"`. It is Projected LDA, as presented in Section 3.4.1. Projected LDA creates the discriminant vector on compressed data and then projects the full training data onto the discriminant vector. Projected LDA requires the parameters `m1`, `m2`, `s`.

The easiest way to run Projected LDA is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
output <- LDA(TrainData, TrainCat, TestData,
              Method = "Projected", Mode = "Automatic")
table(output)
output
  1   2
700 300

mean(output != TestCat)
[1] 0
```

`Automatic` is the default value for `Mode`, and so one could simply run

```
output <- LDA(TrainData, TrainCat, TestData,
              Method = "Projected")
table(output)
output
  1   2
700 300

mean(output != TestCat)
[1] 0
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the compression amounts m_1, m_2, and sparsity level s to be used in compression. The user will type in the amounts:

```
output <- LDA(TrainData, TrainCat, TestData,
             Method = "Projected", Mode = "Interactive")
"Please enter the number m1 of group 1 compression samples: "700
"Please enter the number m2 of group 2 compression samples: "300
"Please enter sparsity level s used in compression: "0.01
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to give the `LDA` function all parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "Projected", Mode = "Research",
                 m1 = 700, m2 = 300, s = 0.01)


table(test_pred)
test_pred
  1   2
700 300


mean(output != TestCat)
[1] 0
```

WARNING: The argument `Mode` will override any supplied parameters if its value is `Automatic` or `Research`.

4.3.5 Fast Random Fisher Discriminant Analysis

This method is the result of setting `Method` equal to `"fastRandomFisher"`. It is the Fast Random Fisher Discriminant Analysis algorithm, as presented in [62]. Fast Random Fisher creates the discriminant vector on reduced sample amounts m, and then projects the full training data onto the learned discriminant vector. The difference between Fast Random Fisher Discriminant

Analysis and Projected LDA is that Fast Random Fisher mixes the groups together when forming the discriminant vector, but Projected LDA does not. Fast Random Fisher requires the parameters `m`, and `s`.

The easiest way to run Fast Random Fisher is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "fastRandomFisher", Mode = "Automatic")
table(test_pred)
test_pred
  1   2
700 300

mean(test_pred != TestCat)
[1] 0
```

`Automatic` is the default value for `Mode`, and so one could simply run

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "fastRandomFisher")
table(test_pred)
output
  1   2
700 300

mean(test_pred != TestCat)
[1] 0
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the total amount of compressed samples m and sparsity level s to be used in compression. The user will type in the amounts:

```
output <- LDA(TrainData, TrainCat, TestData,
              Method = "fastRandomFisher", Mode = "Interactive")
"Please enter the number m of total compressed samples: "1000
"Please enter sparsity level s used in compression: "0.01
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to give the `LDA` function all parameters.

```
test_pred <- LDA(TrainData, TrainCat, TestData,
                 Method = "fastRandomFisher",
                 Mode = "Research", m = 1000, s = 0.01)


table(test_pred)
test_pred
  1   2
700 300


mean(test_pred != TestCat)
[1] 0
```

WARNING: The argument `Mode` will override any supplied parameters if its value is `Automatic` or `Research`.

4.4 Compressed Quadratic Discriminant Analysis

This section provides a more in-depth treatment to the Linear Discriminant methods available in `biClassify`.

There are three seperate quadratic discriminant methods avilable through the `QDA` wrapper function:

1. `Full` Quadratic Discriminant Analyses, which is QDA trained on the full data [33, Section 4.3].
2. `Compressed` Linear Discriminant Analysis of Section 3.4.2.
3. `Subsampled` QDA, where QDA is trained on data which is sub-sampled uniformly from both classes.

The individual methods are invoked by setting the `Method` argument. Let us first load the data for notational convenience.

```
TrainData <- QDA_Data$TrainData
TrainCat <- QDA_Data$TrainCat
TestData <- QDA_Data$TestData
TestCat <- QDA_Data$TestCat
```

4.4.1 Full QDA

This method is the result of setting `Method` equal to `"Full"`. This method is traditional Quadratic Discriminant Analysis, as presented in [33, Section 4.3]. No additional parameters need to be supplied, and the code will run as stated. The function `QDA` produces predicted class labels for `TestData`.

```
Predictions <- QDA(TrainData, TrainCat, TestData,
                   Method = "Full")


table(Predictions)


Predictions
```

```
  1   2
700 300
```

4.4.2 Compressed QDA

This method is the result of setting `Method` equal to `"Compressed"`. It is compressed QDA, as presented in Section 3.4.2. Compressed QDA reduces the group sample amounts from n_1 and n_2 to m_1 and m_2 respectively via compression and trains QDA on the reduced samples.

Compressed QDA requires the parameters `m1`, `m2`, `s`.

The easiest way to run Compressed QDA is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Compressed", Mode = "Automatic")
table(output)


output
  1   2
700 300
```

`Automatic` is the default value for `Mode`, and so one could simply run

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Compressed")
table(output)


output
  1   2
700 300
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the compression amounts m_1, m_2, and sparsity level s to be used in compression. The user will type in the amounts:

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Compressed", Mode = "Interactive")
"Please enter the number m1 of group 1 compression samples: "700
"Please enter the number m2 of group 2 compression samples: "300
"Please enter sparsity level s used in compression: "0.01

table(output)

output
  1   2
700 300
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to give the `QDA` function all parameters.

```
output <- QDA(TrainData, TrainCat,
              TestData, Method = "Compressed", Mode = "Research",
              m1 = 700, m2 = 300, s = 0.01)

table(output)

output
  1   2
700 300
```

4.4.3 Sub-Sampled QDA

Sub-sampled QDA is just QDA trained on data sub-sampled uniformly from both classes. To run sub-sampled QDA, set `Method` equal to `Subsampled`. It requires the additional parameters `m1` and `m2`.

The easiest way to run sub-sampled QDA is to set `Mode` to `Automatic` and not worry about supplying additional parameters.

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Subsampled", Mode = "Automatic")
table(output)


output
  1   2
700 300
```

`Automatic` is the default value for `Mode`, and so one could simply run

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Subsampled")
table(output)


output
  1   2
700 300
```

and obtain the same output.

When `Mode` is set to `Interactive`, prompts will appear asking for the sub-sample amounts m_1, m_2 for each group to be used. The user will type in the amounts:

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Subsampled", Mode = "Interactive")
"Please enter the number m1 of group 1 sub-samples: "700
"Please enter the number m2 of group 2 sub-samples: "300

table(output)

output
  1   2
700 300
```

and the output is produced.

If the user is interested in running simulation studies or has mastery over the functionality, they may wish to give the `QDA` function all parameters.

```
output <- QDA(TrainData, TrainCat, TestData,
              Method = "Subsampled", Mode = "Research",
              m1 = 700, m2 = 300)

table(output)

output
  1   2
700 300
```

WARNING: The argument `Mode` will override any supplied parameters if its value is `Automatic` or `Research`.

4.5 Sparse Kernel Optimal Scoring

This section presents the kernel optimal scoring method available in the `biClassify` package. Kernel optimal scoring is presented in Section 2.2.2. Sparse kernel optimal scoring finds the kernel discriminant coefficients $\alpha \in \mathbb{R}^n$ of (2.3) and feature weights $w \in [-1, 1]^p$.

Let us load the data set used in kernel optimal scoring

```
TrainData <- KOS_Data$TrainData

TrainCat <- KOS_Data$TrainCat

TestData <- KOS_Data$TestData

TestCat <- KOS_Data$TestCat
```

4.5.1 Parameter Selection

This subsection details how `KOS` selects the parameters σ^2, γ, and λ.

The gaussian kernel parameter σ^2 is selected based on the $\{.05, .1, .2, .3, .5\}$ quantiles of the set of squared distances between the classes

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

The ridge parameter γ is selected by adapting a kernel matrix shrinkage technique of [14] to the setting of ridge regression. The sparsity parameter λ is selected using 5-fold cross-validation to minimize the error rate over a grid of 20 equally-spaced values. More details of parameter selection are contained in Section 2.5.

The function `SelectParams` implements these methods automatically.

```
SelectParams(TrainData, TrainCat)


$Sigma

[1] 0.7390306


$Gamma

[1] 0.137591
```

```
$Lambda
[1] 0.02902946
```

If parameters are not supplied to `KOS`, the function first invokes `SelectParams` to generate any missing parameters.

4.5.2 Hierarchical Parameters

Sparse kernel optimal scoring has three parameters: a Gaussian kernel parameter `Sigma`, a ridge parameter `Gamma`, and a sparsity parameter `Lambda`. They have a hierarchical dependency, in that `Sigma` influences `Gamma`, and both influence `Lambda`. The ordering is

Top `Sigma`
Middle `Gamma`
Bottom `Lambda`

When using either of the functions, the user is only allowed to specify parameter combinations which adhere to the hierarchical ordering above. That is, they can only input parameters which go from Top to Bottom. For example, they could specify both `Sigma` and `Gamma`, but leave `Lambda` as the default `NULL` value. On the other hand, the user would not be allowed to specify only `Lambda` while leaving `Sigma` and `Gamma` as their default `NULL` values.

```
SelectParams(TrainData, TrainCat, Sigma = 1, Gamma = 0.1)


$Sigma
[1] 1


$Gamma
[1] 0.1


$Lambda
[1] 0.01078724
```

If the user supplies parameter values which violate the hierarchical ordering, the error message `Hierarchical order of parameters violated.` will be returned.

```
SelectParams(TrainData, TrainCat, Gamma = 0.1)


Error in SelectParams(TrainData, TrainCat, Gamma = 0.1) :
  Hierarchical order of parameters violated.
```

4.5.3 KOS

This package comes with an all-purpose function for running kernel optimal scoring.

```
Sigma <- 1.325386
Gamma <- 0.07531579
Lambda <- 0.002855275


output <- KOS(TestData, TrainData, TrainCat, Sigma = Sigma,
              Gamma = Gamma, Lambda = Lambda)
print(output$Weight)
[1] 1 1 0 0


table(output$Predictions)
 1  2
26 68
```

5. CONCLUSIONS

Linear Discriminant Analysis is a common classification tool. However, it has several disadvantages. The first is that it underfits the data when the true decision boundary is non-linear. Secondly, it uses all data features which constructing the classification rule, and thus can overfit in the high-dimensional setting where not all features are important for class separation. Lastly, Linear Discriminant Analysis is expensive to train on large sample data.

In this dissertation, we propose several adaptations of Linear Discriminant Analysis which address the above limitations. In particular, Chapter 2 proposes a kernel discriminant classifier with simultaneous sparse feature selection. Chapter 3 proposes a sample reduction scheme for discriminant analysis based on compression using sparse random matrices. Chapter 4 presents an R package 'biClassify' containing implements of the proposed methods.

Future directions for research could investigate how to effectively compute the compressed kernel matrix presented in Chapter 3. An additional avenue of future research is investigating the effects of compressing a data matrix in both the sample space and feature space.

REFERENCES

[1] A. F. Lapanowski and I. Gaynanova, "Sparse feature selection in kernel discriminant analysis via optimal scoring," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1704–1713, 2019.

[2] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*. Orlando, FL: Academic Press, 1979.

[3] T. Hastie, R. Tibshirani, and A. Buja, "Flexible discriminant analysis by optimal scoring," *Journal of the American statistical association*, vol. 89, no. 428, pp. 1255–1270, 1994.

[4] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *The Annals of Statistics*, pp. 73–102, 1995.

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

[6] Y. Eidelman, V. D. Milman, and A. Tsolomitis, *Functional analysis: an introduction*, vol. 66. American Mathematical Soc., 2004.

[7] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

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

[9] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," in *Advances in Neural Information Processing Systems*, pp. 568–574, 2000.

[10] G. I. Allen, "Automatic feature selection via weighted kernels and regularization," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 284–299, 2013.

[11] T. Cai and W. Liu, “A direct estimation approach to sparse linear discriminant analysis,” *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1566–1577, 2011.

[12] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, “Sparse discriminant analysis,” *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[13] I. Gaynanova, J. G. Booth, and M. T. Wells, “Simultaneous sparse estimation of canonical vectors in the p n setting,” *Journal of the American Statistical Association*, vol. 111, no. 514, pp. 696–706, 2016.

[14] T. Lancewicki, “Regularization of the kernel matrix via covariance matrix shrinkage estimation,” *arXiv preprint arXiv:1707.06156*, 2017.

[15] P. Craven and G. Wahba, “Smoothing noisy data with spline functions,” *Numerische Mathematik*, vol. 31, pp. 377–403, Dec 1978.

[16] G. H. Golub, M. Heath, and G. Wahba, “Generalized cross-validation as a method for choosing a good ridge parameter,” *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[17] D. Xiang and G. Wahba, “A generalized approximate cross validation for smoothing splines with non-gaussian data,” *Statistica Sinica*, pp. 675–692, 1996.

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

[19] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 6, ACM, 2004.

[20] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1531–1565, 2006.

[21] F. R. Bach, “Consistency of the group lasso and multiple kernel learning,” *Journal of Machine Learning Research*, vol. 9, no. Jun, pp. 1179–1225, 2008.

[22] S. Sun, M. Kolar, and J. Xu, “Learning structured densities via infinite dimensional exponential families,” in *Advances in Neural Information Processing Systems*, pp. 2287–2295, 2015.

[23] O. Chapelle and V. Vapnik, “Model selection for support vector machines,” in *Advances in Neural Information Processing Systems*, pp. 230–236, 2000.

[24] C.-H. Li, C.-T. Lin, B.-C. Kuo, and H.-S. Chu, “An automatic method for selecting the parameter of the rbf kernel function to support vector machines,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pp. 836–839, IEEE, 2010.

[25] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, “Kernlab-an s4 package for kernel methods in r,” *Journal of Statistical Software*, vol. 11, no. 9, pp. 1–20, 2004.

[26] G. C. Cawley and N. L. Talbot, “Reduced rank kernel ridge regression,” *Neural Processing Letters*, vol. 16, no. 3, pp. 293–302, 2002.

[27] G. C. Cawley, N. L. Talbot, R. J. Foxall, S. R. Dorling, and D. P. Mandic, “Heteroscedastic kernel ridge regression,” *Neurocomputing*, vol. 57, pp. 105–124, 2004.

[28] P. Exterkate, “Model selection in kernel ridge regression,” *Computational Statistics & Data Analysis*, vol. 68, pp. 1–16, 2013.

[29] S. An, W. Liu, and S. Venkatesh, “Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression,” *Pattern Recognition*, vol. 40, no. 8, pp. 2154–2162, 2007.

[30] Y. Zhang, J. Duchi, and M. Wainwright, “Divide and conquer kernel ridge regression,” in *Conference on Learning Theory*, pp. 592–617, 2013.

[31] L. H. Dicker, D. P. Foster, D. Hsu, *et al.*, “Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators,” *Electronic Journal of Statistics*, vol. 11, no. 1, pp. 1022–1047, 2017.

[32] G. S. Kimeldorf and G. Wahba, "A correspondence between bayesian estimation on stochastic processes and smoothing by splines," *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. 495–502, 1970.

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

[34] A. Nosedal-Sanchez, C. B. Storlie, T. C. Lee, and R. Christensen, "Reproducing kernel hilbert spaces for penalized regression: A tutorial," *The American Statistician*, vol. 66, no. 1, pp. 50–60, 2012.

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

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

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

[38] B. Caputo, K. Sim, F. Furesjo, and A. Smola, "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*, vol. 2002, 2002.

[39] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[40] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[41] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*. New York: Springer, fourth ed., 2002.

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

[43] I.-C. Yeh, K.-J. Yang, and T.-M. Ting, "Knowledge discovery on rfm model using bernoulli sequence," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5866–5871, 2009.

[44] D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, and Y. Zhang, "Failure analysis of parameter-induced simulation crashes in climate models," *Geoscientific Model Development*, vol. 6, no. 4, pp. 1157–1171, 2013.

[45] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

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

[47] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Advanced lectures on machine learning*, pp. 169–207, Springer, 2004.

[48] I. Gaynanova and T. Wang, "Sparse quadratic classification rules via linear dimension reduction," *arXiv preprint arXiv:1711.04817*, 2017.

[49] I. Steinwart and C. Scovel, "Fast rates for support vector machines using gaussian kernels," *The Annals of Statistics*, pp. 575–607, 2007.

[50] C. Zhang, Y. Liu, and Y. Wu, "On quantile regression in reproducing kernel hilbert spaces with data sparsity constraint," *Journal of Machine Learning Research*, vol. 17, no. 40, pp. 1–45, 2016.

[51] C. Boutsidis and P. Drineas, "Random projections for the nonnegative least-squares problem," *Linear algebra and its applications*, vol. 431, no. 5-7, pp. 760–771, 2009.

[52] M. Pilanci and M. J. Wainwright, "Randomized sketches of convex programs with sharp guarantees," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 5096–5115, 2015.

[53] M. Pilanci and M. J. Wainwright, "Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1842–1879, 2016.

[54] S. S. Vempala, *The random projection method*, vol. 65. Providence, RI: American Mathematical Society, 2005.

[55] M. W. Mahoney *et al.*, "Randomized algorithms for matrices and data," *Foundations and Trends® in Machine Learning*, vol. 3, no. 2, pp. 123–224, 2011.

[56] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster least squares approximation," *Numerische mathematik*, vol. 117, no. 2, pp. 219–249, 2011.

[57] S. Wang, A. Gittens, and M. W. Mahoney, "Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 8039–8088, 2017.

[58] D. Homrighausen and D. J. McDonald, "Compressed and penalized linear regression," *Journal of Computational and Graphical Statistics*, vol. 00, no. 0, pp. 1–14, 2019.

[59] S. Zhou, L. Wasserman, and J. D. Lafferty, "Compressed regression," in *Advances in Neural Information Processing Systems*, pp. 1713–1720, 2008.

[60] W.-H. Li, Z. Zhong, and W.-S. Zheng, "One-pass person re-identification by sketch online discriminant analysis," *Pattern Recognition*, vol. 93, pp. 237–250, 2019.

[61] B. Tu, Z. Zhang, S. Wang, and H. Qian, "Making fisher discriminant analysis scalable," in *International Conference on Machine Learning*, pp. 964–972, 2014.

[62] H. Ye, Y. Li, C. Chen, and Z. Zhang, "Fast fisher discriminant analysis with randomized algorithms," *Pattern Recognition*, vol. 72, pp. 82–92, 2017.

[63] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 143–152, IEEE, 2006.

[64] G. McLachlan, *Discriminant analysis and statistical pattern recognition*, vol. 544. Hoboken, NJ: John Wiley & Sons, 2004.

[65] J. Shao, Y. Wang, X. Deng, S. Wang, *et al.*, “Sparse linear discriminant analysis by thresholding for high dimensional data,” *The Annals of Statistics*, vol. 39, no. 2, pp. 1241–1265, 2011.

[66] P. J. Bickel, E. Levina, *et al.*, “Some theory for fisher’s linear discriminant function,naive bayes’, and some alternatives when there are many more variables than observations,” *Bernoulli*, vol. 10, no. 6, pp. 989–1010, 2004.

[67] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon, “Squashing flat files flatter,” in *Proceedings Of ACM SIGKDD*, vol. 15, pp. 6–15, 1999.

[68] D. Madigan, N. Raghavan, W. Dumouchel, M. Nason, C. Posse, and G. Ridgeway, “Likelihood-based data squashing: A modeling approach to instance construction,” *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 173–190, 2002.

[69] D. Pavlov, D. Chudova, and P. Smyth, “Towards scalable support vector machines using squashing,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. 20, (Boston, MA), pp. 295–299, 2000.

[70] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[71] R. Bhatt and A. Dhall, “Skin segmentation dataset,” *UCI Machine Learning Repository*, 2010.

[72] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop*, pp. 41–48, IEEE, 1999.

[73] D. Hsu, S. Kakade, T. Zhang, *et al.*, “A tail inequality for quadratic forms of subgaussian random vectors,” *Electronic Communications in Probability*, vol. 17, 2012.

[74] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge University Press, 2018.