2-1-2015

Quadratic Discriminant Analysis Revisited

Wenbo Cao
Graduate Center, City University of New York

How does access to this work benefit you? Let us know!
Follow this and additional works at: http://academicworks.cuny.edu/gc_etds

Part of the Computer Sciences Commons

Recommended Citation
Cao, Wenbo, "Quadratic Discriminant Analysis Revisited" (2015). CUNY Academic Works.
http://academicworks.cuny.edu/gc_etds/536

This Dissertation is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Graduate Works by Year: Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
Quadratic Discriminant Analysis Revisited

by

Wenbo Cao

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2015
This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

Robert Haralick

Date

Chair of Examining Committee

Robert Haralick

Date

Executive Officer

Robert Haralick

Susan Epstein

Andrew Rosenberg

Changhe Yuan

Craig Friedman

Supervisory Committee
Abstract

Quadratic Discriminant Analysis Revisited

by

Wenbo Cao

Advisor: Robert Haralick

In this thesis, we revisit quadratic discriminant analysis (QDA), a standard classification method. Specifically, we investigate the parameter estimation and dimension reduction problems for QDA.

Traditionally, the parameters of QDA are estimated generatively; that is the parameters are estimated by maximizing the joint likelihood of observations and their labels. In practice, classical QDA, though computationally efficient, often underperforms discriminative classifiers, such as SVM, Boosting methods, and logistic regression. Motivated by recent research on hybrid generative/discriminative learning [12, 80, 91], we propose to estimate the parameters of QDA by minimizing a convex combination of negative joint log-likelihood and negative conditional log-likelihood of observations and their labels. For this purpose, we propose an iterative majorize-minimize (MM) algorithm for classifiers of which conditional distributions are from the exponential family; in each iteration of the MM algorithm, a convex optimization problem needs to be solved. To solve the convex problem specially
derived for QDA, we propose a block-coordinate descent algorithm that sequentially updates the parameters of QDA; in each update, we present a trust region method for solving optimal estimations, of which we have closed form solutions in each iteration. Numerical experiments show: 1) the hybrid approach to QDA is competitive with, and in some cases significant better than other approaches to QDA, SVM with polynomial kernel \((d = 2)\) and logistic regression with linear and quadratic features; 2) in many cases, our optimization method converges faster to equal or better optimums than the conjugate gradient method used in [12].

Dimension reduction methods are commonly used to extract more compact features in the hope to build more efficient and possibly more robust classifiers. It is well known that Fisher’s discriminant analysis generates optimal lower dimensional features for linear discriminant analysis. However, “...for QDA, where so far there has been no universally accepted dimension-reduction technique in the literature” [65], though considerable efforts have been made. To construct a dimension reduction method for QDA, we generalize the Fukunaga-Koontz transformation, and propose novel affine feature extraction (AFE) methods for binary QDA. The proposed AFE methods have closed-form solutions and thus can be solved efficiently. We show that 1) the AFE methods have desired geometrical, statistical and information-theoretical properties; and 2) the AFE methods generalize dimension reduction methods for LDA and QDA with equal means. Numerical experiments show that the new proposed AFE method is competitive with, and in some cases significantly better than some commonly used linear dimension reduction techniques for QDA in the literature.
Acknowledgements

First and foremost I would like to thank my advisor, Professor Robert Har- alick. His wisdom and knowledge, patience and encouragement have helped me to finish this long journey, and will continue to influence me in the future.

I would like to thank Dr. Craig Friedman, who introduced me to the field of machine learning, provided financial support for my study at CUNY, and has a great influence on my career.

Specially, I would like to thank Professor George Veronis at Yale University. George was my advisor when I studied Atmospheric Science at Yale. His trust, encouragement and support are always sources of power that help me to survive in difficult time.

I would like to thank my other committee members, Professor Susan Epstein, Professor Andrew Rosenberg, and Professor Changhe Yuan, for their good questions and comments which help me to refine this dissertation.
I would also like to thank Professor Ted Brown for his support.

I would like to thank my friends who helped and supported me in this long journey: Zhiming Yin, Min Wu, Juan Zhou, Yi Wei, Ruquan Wang, Lei Fang, Yilong Ni, Huajun Huang, Wenxuan Du. I would like to thank Jin Chen and Shan Wei for their warm accommodation when I first arrived at New York; they made me feel like a family member. I would like to thank Yuqing Tang, Ou Liu, Rave Harpaz and Ingrid Montealegre for the great times and fruitful discussions we shared.

Finally, I would like to thank my long-suffering family for their love and support: my parents and aunt, my brother, my wife and two sons and the soon-to-be-born third one. This thesis is dedicated to them.
## Contents

1. Introduction  
   1.1. Classification  2  
   1.2. Examples: LDA and QDA  5  
      1.2.1. Parameter Estimation  7  
      1.2.2. Dimension Reduction  8  
   1.3. Thesis Summary and Organization  10  
   1.4. Notations and Conventions  11  

2. A Review of Classification  16  
   2.1. Decision Theory for Classification  17  
   2.2. Generative vs. Discriminative Classifiers  20  
   2.3. Learning for Classifications  21  

3. QDA Revisited  26
### CONTENTS

3.1. Learning For Exponential Family .......................... 28

3.2. QDA Revisited ........................................... 32
   3.2.1. Updating $v_k$’s .................................. 36
   3.2.2. Updating $\theta_k$’s ................................ 37
   3.2.3. Updating $\Omega_k$’s ................................. 40
   3.2.4. Initial Points, Early Stopping and $\beta$ .............. 44

3.3. Numerical Experiments ..................................... 46
   3.3.1. Data and Benchmark Methods ......................... 46
   3.3.2. Effect of $\beta$ on Classification Error .............. 50
   3.3.3. Effect of Training Sample Size on Classification Error 50
   3.3.4. Comparison on Classification Error ................... 52
   3.3.5. Comparison with Conjugate Gradient Method ........... 56

3.4. Conclusion .................................................. 58

4. A Review of Discriminative LSDR Methods ................. 63

4.1. Criteria for Dimension Reduction .......................... 67
   4.1.1. Information-theoretic error exponents for Bayes error . 68
   4.1.2. Sufficient Dimension Reduction ........................ 73

4.2. Discriminative LSDR: When $\mu_k = \text{arbitrary}$ and $\Sigma_k = \Sigma$ .. 75

4.3. Discriminative LSDR When $\mu_k = \mu$ and $\Sigma_k = \text{arbitrary}$ .. 80
CONTENTS

4.4. Discriminative LSDR When $\mu_k = \text{arbitrary}$ and $\Sigma_k = \text{arbitrary}$ 87

4.4.1. FKT ................................. 88

4.4.2. SAVE ................................. 90

4.4.3. The Chernoff Criterion ................................. 93

4.4.4. The KL Criterion ................................. 96

5. Affine Feature Extraction 99

5.1. Affine Feature Extraction ................................. 101

5.1.1. Background ................................. 101

5.1.2. Problem Formulation ................................. 103

5.1.3. Basic Algorithms ................................. 108

5.1.4. Computational Issues ................................. 112

5.2. Discussion ................................. 116

5.2.1. Information theoretical property of AFE ................................. 117

5.2.2. Connection with FDA and MBD($\Delta \mu = 0$) ................................. 120

5.2.3. Connection with QDA ................................. 122

5.3. Experiments ................................. 125

5.3.1. Two Simple Examples ................................. 125

5.3.2. Visualization on Simulated Data Sets ................................. 131

5.3.3. Classification with Benchmark Data Sets ................................. 131
CONTENTS

5.4. Conclusions ........................................... 141

6. Conclusions and Future Directions 144

A. Appendices for Chapter 3 150

A.1. Optimization problem (3.24) ............................ 150
A.2. Optimization sub-problem for $\theta_k$’s .................. 152
A.3. Optimization sub-problem for $\Omega_k$’s .................. 153

B. Appendices for Chapter 5 155

B.1. Augmented Second Moment Matrix ...................... 155
B.2. Proof of (5.29) ........................................... 156
B.3. Proof of (5.41) and (5.42) .............................. 158
B.4. Proof of Theorem 5.1 ................................. 158
B.5. Proof of (B.10) − (B.14) ............................. 163

C. Software 168

C.1. Compile Codes ............................................. 168
C.1.1. Matlab .................................................. 168
C.1.2. R ......................................................... 169
C.2. Usage ....................................................... 170
C.2.1. set_mod_opt .......................................... 170
CONTENTS

C.2.2. train_model_cv ........................................ 175
C.2.3. train_model ............................................. 176
C.2.4. test_model ............................................... 177

Bibliography ................................................. 178
List of Tables

1.1. Mathematical symbols used in this thesis ........................................ 14
1.2. Abbreviations used in this thesis ..................................................... 15

3.1. Descriptions of 25 benchmark datasets ($K$: number of classes, $d$: dimension, $N$: number of observations) ........................................... 47
3.2. Average test set error rates for 25 datasets (Shown in percentage) ................................................................. 57
3.3. Standard Deviation of test set error rates for 25 datasets
   (Shown in percentage) ........................................................................... 59
3.4. CPU time (seconds) and final objective function values of
   training sets ($\beta = 0.5$): CG v.s. our method (Smaller is better) 60

4.1. Bayes Error for decision rules on two SAVE directions ................. 93

5.1. Two kinds of AFE problems ............................................................... 108
LIST OF TABLES

5.2. Description of 20 artificial and real data sets ($K$: number of classes, $D$: dimensionality of data, and $N$: number of observations) .................................................. 134

5.3. Optimal observed mean test error rates and the corresponding dimensionality (shown in parentheses) for each method. Error rates are shown in percentage. Best error rates are displayed in bold. So are error rates that are not statistically significantly different from the best ones........................................... 141
List of Figures

3.1. Influence of coefficient of $\beta$ on test set error rates: $\beta = 0$ is discriminative, and $\beta = 1$ is generative. Red bars are the median test error rates over 20 runs, while blue circles is the mean test set error rates over 20 runs. .......................... 51

3.2. Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods. .......................... 53

3.3. Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods. .......................... 54

3.4. Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods. .......................... 55
LIST OF FIGURES

3.5. Ranks of various quadratic classifiers over 23 datasets. (Smaller
rank means smaller test set error rates) .......................... 58

3.6. Boxplots of speedup (defined in (3.63)) and improvement (de-
defined in (3.63)) are shown for 25 selected data sets. ............ 62

5.1. We show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0, 1]$. The
order of $\xi \lambda + \frac{1-\xi}{\lambda}$ are used to select projection direction in the
AFE methods. ......................................................... 128

5.2. We show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0, 1]$. The
order of $\xi \lambda + \frac{1-\xi}{\lambda}$ are used to select projection direction in the
AFE methods. ......................................................... 130

5.3. Comparison of features found by PCA, FDA, FKT and Our
method. Blue and red points belong to different classes. . . . 132

5.4. Effect of $\xi$ on error rates when features are extracted by the
$\Xi_t$–Orthogonal method and then classified by regularized dis-
criminant analysis. Red bar shown in the figures are the me-
dian error rate of 20 test sets; while blue circle shown in the
figures are the mean error rates of 20 test sets. See text for
details. ................................................................. 136

5.5. Test set error rates for different dimensionality. ............ 138
5.6. Test set error rates for different dimensionality. . . . . . . . . 139

5.7. The ranks of each method based on mean test error rates (smaller rank means smaller error rate). Red bar and green circle are, respectively the median and mean rank of the 18 real datasets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
List of Algorithms

1. A MM optimization algorithm for estimating parameters of the exponential family .......................... 32
2. An iterative algorithm for finding $\Upsilon^*$ .................................. 37
3. An iterative trust-region algorithm for finding $\theta_m^*$ .................................... 40
4. An trust-region algorithm for finding $\Omega_m^*$ ................................. 45
5. An Algorithm for affine feature extraction ............................... 111
6. Affine feature extraction for undersample problems .............. 117
Chapter 1

Introduction

This thesis is about quadratic discriminant analysis (QDA), which is a standard probabilistic classification method in statistics and machine learning [10, 39, 64, 124]. Roughly speaking, classical QDA assumes class-conditional distributions to be normal, and then classifies given test points by the posterior distributions; we shall formally discuss QDA in Section 1.2. In this thesis we will investigate the problems of parameter estimation and dimension reduction for QDA. In the problem of parameter estimation, we learn model parameters for constructing a QDA classifier. In the problem of dimension reduction, we find a more succinct data representation for a QDA classifier, and thus reduce the dimensionality of input data. The organization of this chapter is as follows: we will give a discussion of classification
problems in Section 1.1; to give concrete examples of classification methods and to state the motivations of this thesis work, we will give an overview of linear and quadratic discriminant analysis in Section 1.2; we will then give a summary of this thesis work and the organization of this thesis in Section 1.3; finally, we will present the conventions of symbols and a list of key notations and abbreviations in Section 1.4.

1.1 Classification

Classification is a fundamental supervised learning problem that has been studied in statistics and machine learning. Some examples of classification problems are: handwritten digit recognition [82], face recognition [134, 140], obstacle detection in autonomous driving [122], tumor classifier with gene expression data [40].

In a classification problem, we are given a list of labelled observations. With the given labelled observations, we need to construct a classifier to assign a previously unseen observation to one of the pre-defined categories. Formally, we assume that a training set $D$ is given for a classification problem as follows

$$D = \{(x_n, y_n)|n = 1, 2, \cdots , N, x_n \in \mathcal{X}, y_n \in \mathcal{Y}\} \quad (1.1)$$
where a sample \((x_n, y_n)\) contains an observation \(x_n\) and its label \(y_n\). \(X\) and \(Y\) are, respectively, the domains for observations and their labels. In this thesis, we assume \(X = \mathbb{R}^D\), and \(Y = \{1, 2, \cdots, K\}\). The task is to learn with the training set \(\mathcal{D}\) a mapping as follows,

\[
f: \mathbb{R}^D \mapsto \{1, 2, \cdots, K\}
\]

Probabilistic classifiers have been widely used for classification problems, for instance logistic regression (LR), linear discriminant analysis (LDA) and QDA. Essentially speaking, probabilistic classifiers either assume posterior probabilities \(p(Y = k|X = x)\) directly, or assume class conditional probabilities \(p(X = x|Y = k)\) and derive posterior probabilities via Bayes theorem as following

\[
p(Y = k|X = x) = \frac{p(X = x|Y = k)\pi_k}{\sum_{k=1}^{K} p(X = x|Y = k)\pi_k}
\]

\(^1\)There are many names associated with observations \(x_n\)'s and labels \(y_n\)'s in the literature. In statistics, \(x_n\)'s are called inputs, explanatory variables, predictor variables, or independent variables; \(y_n\)'s are called outputs, responses, predicted variables, or dependent variables. In machine learning, \(x_n\)'s are called patterns, cases, instances, features, or observations; \(y_n\)'s are called labels, targets, outputs or sometimes also observations. In this thesis, we will call \(x_n\) an observation or sometimes an observation vector, and \(y_n\) a label or a class label; we will call components of \(x_n\) as variables.

\(^2\)Strictly speaking, we should write posterior probabilities as \(p(Y = k|X = x; \Theta)\), where \(\Theta\) is the parameter that determines posterior probabilities. For compactness, we will omit \(\Theta\) and write posterior probabilities as \(p(Y = k|X = x)\); we will, however, explicitly write parameter \(\Theta\) when we discuss parameter estimation problem. We will following the same convention when we discuss class-conditional probabilities \(p(X = x|Y = k)\).
CHAPTER 1. INTRODUCTION

where \( \pi_k = p(Y = k) \) is the prior probability of class \( k \). For simplicity, we use lowercase \( p(\cdot) \) to denote both a probability mass function and a probability density function in this thesis. Then for a simple 0 – 1 loss, we can assign an observation \( \mathbf{x} \) to a class via the following decision rule

\[
\hat{y} = \arg \max_k p(Y = k | \mathbf{X} = \mathbf{x})
\]  

(1.3)

For a specific probabilistic classifier, we have to learn its parameter with a given training set \( \mathcal{D} \). This is the parameter estimation problem for a classifier. In essence, we need to answer the following two questions in a parameter estimation problem:

1. What criterion should be used for estimating the parameter of a given probabilistic classifier?

2. How can we optimize a given optimization criterion to estimate the parameter of a given probabilistic classifier?

To get reliable estimations of parameters for a probabilistic classifier, we need sufficient training observations with respect to the number of unknown parameters. The number of the parameters of a probabilistic classifier often increases as the dimensionality of observations increases. On the other hand, labeled data are generally rare or expensive for many classification
problems. To efficiently obtain robust estimations of the parameters for a given probabilistic classifier, dimension reduction methods are commonly used to pre-process data [59]. This is especially true for modern classification problems in which data are often high dimensional [36, 43]. Dimension reduction methods are also useful in exploratory data analysis, in which more compact representations of raw observations are learned for improved visualization, for example principle component analysis (PCA) [76]. In essence, dimension reduction methods reduce the dimensionality of raw observations by removing irrelevant information for classification. To design a dimension reduction method, we need to answer the following two questions:

1. What criterion should be used for capturing the “relevant information” of classification in given input data?

2. How can we optimize a given criterion to obtain a more compact representation of original input data?

1.2 Examples: LDA and QDA

To illustrate the problems of parameter estimation and dimension reduction described in Section 1.1, we give a brief overview of LDA and QDA in this section. This section also serves as our motivation for the research presented
in this thesis. Good references on LDA and QDA include the books by Duda, Hart and Stork (2000) [39] and Hastie, Tibshirani, and Friedman (2003) [64].

LDA and QDA assume that class conditional probabilities are multivariate Gaussian. The densities of class conditional probabilities can be written as following,

\[
p(X = x | Y = k; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D \det \Sigma_k}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}
\]  

(1.4)

where \( \mu_k \) and \( \Sigma_k \) are the mean and covariance matrix of the class conditional probability for class \( k \). For LDA, the covariance matrices of class conditional probabilities are assumed to be the same, i.e.

\[
\Sigma_k = \Sigma_0
\]

(1.5)

The posterior probabilities can be derived via Bayes theorem; that is

\[
p(Y = k | X = x; \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \Pi) = \frac{\pi_k p(x|k; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k p(x|k; \mu_k, \Sigma_k)}
\]

(1.6)

where \( \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K \) and \( \Pi = (\pi_1, \pi_2, \cdots, \pi_K) \) are unknown parameters that need to be estimated from a given training set \( D \).
1.2.1 Parameter Estimation

In classical LDA and QDA, $\pi_k$’s, $\mu_k$’s and $\Sigma_k$’s are estimated by maximizing the joint likelihood of observations and their labels, which can be formally written as following

$$\max \prod_{k=1}^{K} \prod_{y_n=k} p(X=x_n|Y=k; \Sigma_k, \mu_k) \pi_k$$  \hspace{1cm} (1.7)

s.t. 

$$\Sigma_k \succ 0 \hspace{1cm} k = 1, 2, \cdots, K$$  \hspace{1cm} (1.8)

$$\pi_k > 0 \hspace{1.5cm} k = 1, 2, \cdots, K$$  \hspace{1cm} (1.9)

$$\sum_{k=1}^{K} \pi_k = 1$$  \hspace{1cm} (1.10)

Here we use $\Sigma_k \succ 0$ to mean that $\Sigma_k$ is a positive definite matrix. It can be easily shown that the estimations can be written as follows \(^3\)

$$\hat{\pi}_k = \frac{N_k}{N}$$  \hspace{1cm} (1.11)

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{y_n=k} x_n$$  \hspace{1cm} (1.12)

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{y_n=k} (x_n - \hat{\mu}_k)(x_n - \hat{\mu}_k)^T$$  \hspace{1cm} (1.13)

$$\Sigma_0 = \sum_{k=1}^{K} \frac{N_k}{N} \Sigma_k$$  \hspace{1cm} (1.14)

\(^3\)For the moment, let us assume $\Sigma_k$’s are positive definite.
CHAPTER 1. INTRODUCTION

where $N_k$ is the number of class-$k$ observations in the given training set $\mathcal{D}$.

We note that there are mainly two problems associated with this estimation:

- Estimations $\hat{\mu}_k$ and $\hat{\Sigma}_k$ are obtained by maximizing the joint likelihood of observations and their labels. The joint likelihood is for data generation but is not directly related to classification performance;

- As a consequence of maximizing the joint likelihood, estimations $\hat{\mu}_k$ and $\hat{\Sigma}_k$ only use observations that are in class $k$. Observations that are not in class $k$ have no effect on the estimations of $\mu_k$ and $\Sigma_k$, which is a waste of labelled observations.

Conceivably we can improve the performance of classical QDA by estimating parameters of QDA with an optimization criterion that is closely related to classification performance. This is the motivation of our work presented in chapter 3.

1.2.2 Dimension Reduction

It is well accepted that Fisher discriminant analysis (FDA) is an optimal dimension reduction method for LDA [64]. For illustrating the dimension reduction problem for LDA, let us assume we have binary classification problem, i.e. $K = 2$. For assigning observations into different classes, it
is sufficient to consider the log-ratio of the posterior probabilities

\[
\log \frac{p(Y = 1 | X = x)}{p(Y = 2 | X = x)} = \log \frac{p(X = x | Y = 1; \mu_1, \Sigma_0) \pi_1}{p(X = x | Y = 2; \mu_2, \Sigma_0) \pi_2} \\
= \log \frac{\pi_1}{\pi_2} - \frac{1}{2} (\mu_1 + \mu_2)^T \Sigma_0^{-1} (\mu_1 - \mu_2) \\
+ x^T \Sigma_0^{-1} (\mu_1 - \mu_2)
\]

Note that only the last term of the right hand side of the equation is a function of \( x \). Therefore for a binary classification problem, we can project data onto a subspace defined by \( \Sigma_0^{-1} (\mu_1 - \mu_2) \). In real application, the sample estimations of \( \Sigma_0, \mu_1 \) and \( \mu_2 \) are used. This defines Fisher’s discriminant analysis (FDA), a natural dimension reduction method for LDA. Fisher obtained the same solution by consider the following problem [44]:

Find the linear combination \( Z = (w^*)^T X \) such that the between-class variance is maximized relative to the within-class variance, i.e.

\[
w^* = \arg \max_w \frac{w^T (\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T w}{w^T \Sigma_0 w}
\]

Then a natural question is if there exists a dimension reduction that is optimal for QDA, similar as FDA for LDA? This is the motivation of our work presented in Chapter 5.
1.3 Thesis Summary and Organization

In this thesis, we will revisit QDA. In summary the contributions of this thesis are as follows

- Motivated by the recent research on hybrid learning [12, 80, 91], we propose to estimate parameters of QDA by minimizing a convex combination of negative joint log-likelihood and negative conditional log-likelihood of observations and their labels. For this purpose, we propose an iterative majorize-minimize (MM) for classifiers of which conditional distributions are from the well-known exponential family; in each iteration of the MM algorithm, a convex optimization problem needs to be solved. To solve the convex problem specifically derived for QDA, we propose a block-coordinate descent algorithm that sequentially updates the parameters of QDA; in each update, we present a trust region method for solving optimal estimations, of which we have closed form solutions in each iteration.

- We generalize the Fukunaga-Koontz transformation (FKT), and propose a novel affine feature extraction (AFE) algorithm for binary QDA, i.e. $K = 2$. The proposed AFE method has closed-form solutions and thus can be solved efficiently. We show that AFE has the fol-
lowing properties: 1) it is a sufficient dimension reduction method [57, 24, 119]; that is, it finds $Z$ such that $Y$ is conditionally independent of $X$ given $Z$, i.e. $Y \perp\!\!\!\!\!\!\!\!\perp X|Z$; 2) it finds an affine subspace that best preserves the symmetric Kullback-Leibler (KL) divergence; and 3) FDA is a special case of AFE.

The remainder of this thesis is organized as follows: in Chapter 2, we will give an overview of classification problem and discuss the learning criterion for a probabilistic classifier. In Chapter 3, we will present an optimization algorithm for estimating parameters of QDA; this chapter is mainly based on our work published in ICPR 2014 [18]. In chapter 4, we give an overview of subspace dimension reduction for discriminant analysis. In Chapter 5, we present an affine feature extraction method for QDA, which is based on our works [16] and [17]. In Chapter 6, we conclude this thesis with possible directions that can be followed from the works in this thesis.

1.4 Notations and Conventions

Let us start with some notations and conventions for symbols used in this thesis. For the convenience of future discussion, we define some concepts about data which are used throughout this thesis.
CHAPTER 1. INTRODUCTION

• An index set $C_k$ contains the indices of observations in a given training set $\mathcal{D}$ that belong to class $k$; i.e.

$$C_k = \{n| (x_n, y_n) \in \mathcal{D}, y_n = k \}$$ (1.15)

• Let $N_k$ be the number of observations in a given training set $\mathcal{D}$ that are in class $k$, i.e. $N_k = \text{card}(C_k)$, where $\text{card}(C_k)$ is the cardinality of set $C_k$.

• We have $N = N_1 + N_2 + \cdots + N_k$.

In this thesis, we follow the following conventions for mathematical symbols unless otherwise specified:

1. Vectors are column vectors and are represented as bold small letters, e.g. $\mathbf{x}$;

2. Matrices are represented as bold capital letters, e.g. $\mathbf{X}$;

3. Random variables are represented as italic letters, e.g. $X$;

4. $\mathbf{I}$ is an identity matrix;

5. $\mathbf{0}$ is a vector or matrix of zeros;

6. $\mathbf{1}$ is a vector or matrix of ones;
7. $\mathbf{\Box}^T$ is the transpose of a vector or matrix $\mathbf{\Box}$;

8. $\det(\mathbf{X})$ is the determinant of a square matrix $\mathbf{X}$;

9. $\text{tr}(\mathbf{X})$ is the trace of a matrix $\mathbf{X}$;

10. $\text{diag}(x_1, x_2, \cdots, x_k)$ is a diagonal matrix, whose $(i, i)$-th entry is $x_i$,

    $(i = 1, 2, \cdots, k)$;

11. for symmetric matrices $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \succeq \mathbf{Y}$ means that matrix $\mathbf{X} - \mathbf{Y}$

    is positive semidefinite;

12. and we define the inner product of two matrices $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{Y} \in \mathbb{R}^{m \times n}$ as following

    $$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} Y_{ij}$$

For convenience, we summarize the key mathematical symbols of this
thesis in table 1.1 and the abbreviations of this thesis in table 1.2.
### Chapter 1. Introduction

**Notation**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>random variables for observations</td>
</tr>
<tr>
<td>Y</td>
<td>random variable for class label</td>
</tr>
<tr>
<td>((x_n, y_n))</td>
<td>the (n)-th observation and its label</td>
</tr>
<tr>
<td>(K)</td>
<td>number of groups</td>
</tr>
<tr>
<td>(N)</td>
<td>number of observations</td>
</tr>
<tr>
<td>(N_k)</td>
<td>number of observations in group (k)</td>
</tr>
<tr>
<td>(D)</td>
<td>dimensionality of observations</td>
</tr>
<tr>
<td>(p(X = x</td>
<td>Y = k; \Theta_k))</td>
</tr>
<tr>
<td>(q(X = x</td>
<td>Y = k; \Theta_k, \tilde{\Theta}_k))</td>
</tr>
<tr>
<td>(\Theta_k)</td>
<td>unknown parameters of class-conditional probabilities</td>
</tr>
<tr>
<td>(\pi_k)</td>
<td>prior probabilities</td>
</tr>
<tr>
<td>(v_k)</td>
<td>log prior, (v_k = \log \pi_k)</td>
</tr>
<tr>
<td>(\Theta)</td>
<td>(\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_K))</td>
</tr>
<tr>
<td>(\Pi)</td>
<td>(\Pi = (\pi_1, \pi_2, \ldots, \pi_K))</td>
</tr>
<tr>
<td>(\Upsilon)</td>
<td>(\Upsilon = (v_1, v_2, \ldots, v_K))</td>
</tr>
<tr>
<td>(\mu_k)</td>
<td>the mean of class-(k)</td>
</tr>
<tr>
<td>(\Sigma_k)</td>
<td>the covariance of class-(k)</td>
</tr>
<tr>
<td>(\Omega_k)</td>
<td>the inverse covariance of class-(k), (\Omega_k = \Sigma_k^{-1})</td>
</tr>
<tr>
<td>(\theta_k)</td>
<td>(\theta_k = \Sigma_k^{-1}\mu_k)</td>
</tr>
<tr>
<td>(M_k)</td>
<td>the second order moment (M_k = \Sigma_k + \mu_k\mu_k^T)</td>
</tr>
<tr>
<td>(\Xi_k)</td>
<td>the augmented second moment matrix (\Xi_k = \begin{pmatrix} M_k &amp; \mu_k \ \mu_k &amp; 1 \end{pmatrix})</td>
</tr>
</tbody>
</table>

Table 1.1: Mathematical symbols used in this thesis
**Abbreviation** | **Description**  
---|---  
FDA | Fisher discriminant analysis  
LDA | linear discriminant analysis  
QDA | quadratic discriminant analysis  
RDA | regularized discriminant analysis  
BDA7 | Bayesian quadratical discriminant analysis  
LR | logistic regression  
SVM | support vector machine  
MM algorithm | majorize-minimize optimization algorithm  
PCA | principle component analysis  
FKT | Fukunaga-Koontz transformation  
SAVE | sliced average variance estimation  
KL divergence | Kullback-Leibler divergence  
AFE | affine feature extraction  
SVD | singular value decomposition

Table 1.2: Abbreviations used in this thesis
Chapter 2

A Review of Classification

We assume that, for a specific classification problem, data are generated by a fixed but unknown probability \( p(X = x, Y = y) \). In the literature, data generated this way are called \textit{independent and identically distributed} (i.i.d.). Learning with i.i.d. data is a standard assumption in statistical learning theory.

A typical setting of classification is as following: a training set \( \mathcal{D} \) of \( N \) i.i.d. samples is given; a decision function \( f : \mathbb{R}^D \rightarrow \{1, 2, \cdots, K\} \) needs to be learned with dataset \( \mathcal{D} \), such that previous unseen observations can be assigned into different classes.

In this chapter, we will give an overview of classification. We focus on probabilistic classifiers in this thesis. In Section 2.1, we will introduce deci-
CHAPTER 2. A REVIEW OF CLASSIFICATION

decision theory for classification problems. In Section 2.2, we will present two approaches for specifying probabilistic classifiers: discriminative vs. generative classifiers. Finally, in Section 2.3, we will discuss common approaches for training a probabilistic classifier: generative, discriminative and hybrid generative/discriminative learning.

References for decision theory and its application on classification are [10, 39, 56, 110, 127]. References on parameter estimation methods can be found in [10, 19, 39, 110, 127].

2.1 Decision Theory for Classification

Decision problems are inherent for classification: after all a classifier has to predict class labels for unseen observations. The decision of assigning a given observation to a class has consequence, whose value is measured by loss or equivalently economic gain. Essentially speaking, we want to construct a classifier that achieves minimum loss for a given classification problem. Statistical decision theory provides a fundamental statistical approach for us to choose optimal decision rules, and moreover to understand the nature of classification.

For the purpose of understanding classification, let us assume joint prob-
ability \( p(X = x, Y = y) \) is known in this section. By Bayes theorem, we can derive posterior probabilities as following

\[
p(Y = y|X = x) = \frac{p(X = x, Y = y)}{\sum_{k=1}^{K} p(X = x, Y = k)} \tag{2.1}
\]

We have the marginal probability

\[
p(X = x) = \sum_{k=1}^{K} p(X = x, Y = k) \tag{2.2}
\]

For compactness, we ignore parameters in \( p(X = x, Y = y) \), \( p(X = x) \) and \( p(Y = y|X = x) \) in this section.

Let \( g(x) \) be a decision rule such that, for any observation \( x \) of a classification problem, a decision \(^1\) of assigning it to a class can be made. Let \( c(y, g(x)) \) be a cost function, or more commonly a loss function when the true label is \( y \) and the assigned label is \( g(x) \). Without losing generality, we assume loss functions are nonnegative and finite, i.e. \( 0 \leq c(y, g(x)) < \infty \); moreover we assume no loss is incurred for a correct decision, i.e. \( c(y, y) = 0 \). A commonly used loss function in the literature is \( 0-1 \) loss, which assumes a loss of 1 whenever a wrong decision is made. Formally, we can define a

\(^1\)Decisions are often called actions in decision-theoretical literature.
0–1 loss function as following

\[
c(y, g(x)) = \mathbb{I}(y \neq g(x)) = \begin{cases} 
0, & \text{if } g(x) = y \\
1, & \text{otherwise}
\end{cases}
\]  \tag{2.3}

In this thesis, we consider 0–1 loss for its simplicity. Nevertheless, our argument is ready to be extended for cost-sensitive classifications, of which loss functions depend on true class labels \(y\) [35].

For a classification problem, we can define the expected risk of a decision rule \(g(x)\) as follows

\[
\mathcal{R}(g) = \int_x \sum_{k=1}^{K} c(k, g(x)) p(Y = k | X = x) p(X = x) dx
\]  \tag{2.4}

Note that quantity \(\sum_{k=1}^{K} c(k, g(x)) p(Y = k | X = x)\) is the expected loss associated with a particular decision \(g(x)\) for a specific observation \(x\). Therefore the expected risk \(\mathcal{R}(g)\) can be thought of as an overall risk/loss for all possible observations generated from \(p(X)\). An optimal decision rule, which is often called a Bayes rule, is a decision rule that minimizes the expected risk \(\mathcal{R}(g)\).

To minimize the expected risk \(\mathcal{R}(g)\), it suffices that, for each observation \(x\), we minimize quantity \(\sum_{k=1}^{K} c(k, g(x)) p(Y = k | X = x)\). Therefore a Bayes
rule, \( g^*(x) \), can be defined

\[
g^*(x) = \arg \min_{1 \leq m \leq K} \sum_{k=1}^{K} c(k, m)p(Y = k | X = x)
\] (2.5)

For 0–1 loss function, it is obvious that a Bayes rule can be defined as follows,

\[
g^*(x) = \arg \max_k p(k|x)
\] (2.6)

That is to assign an observation \( x \) to a class \( k \) such that the posterior probability \( p(Y = k | X = x) \) is maximum.

### 2.2 Generative vs. Discriminative Classifiers

We have seen that knowing posterior probabilities \( p(Y | X = x) \) is sufficient for making predictions in classification problems. There are essentially two approaches to specify posterior probabilities \( p(Y | X = x) \): discriminative vs. generative classifiers.

In a discriminative classifier, posterior probabilities \( p(Y | X = x) \) are directly specified, for example logistic regression. In a generative classifier, class conditional probabilities \( p(X | Y = k) \) are assumed, either parametric or non-parametric; posterior probabilities are then derived via Bayes theo-
rem as following

\[
p(Y = k | X = x) = \frac{p(X = x | Y = k)p(Y = k)}{\sum_{k=1}^{K} p(X = x | Y = k)p(Y = k)} \tag{2.7}
\]

where \(p(Y = k)\) is the prior probability for class \(k\). LDA and QDA are, for example, two widely used generative classifiers.

The debate of using generative or discriminative classifiers has been one of the key questions in machine learning literature. Generally speaking, discriminative classifiers model \(p(Y | X = x)\) directly, and thus may need fewer parameters; this alleviates the difficulty of parameter estimation problems. Generative classifiers take an indirect approach for classification and often need more parameters; consequently parameter estimation problem for generative classifier is more difficult. On the other hand, because class conditional probabilities are known, generative classifiers have unique advantages in dealing with missing data and outliers, and in incorporating experts’ beliefs and unlabelled observations in learning procedure.

### 2.3 Learning for Classifications

As we have seen in Section 2.1, expected risk plays important role in classification. Therefore it is tempting to estimate parameters by minimizing
expected risk, since then we would use the same principle in estimation and
decision. However expected risk is quite difficult to calculate. In practice,
empirical risk is often used as an approximation to expected risk [111, 127].
The empirical risk for a given training data set $D$ is defined as follows,

$$\mathcal{R}_D = \frac{1}{N} \sum_{n=1}^{N} c(y_n, g(x_n)) \quad (2.8)$$

For the $0 - 1$ loss function, we can write empirical risk as

$$\mathcal{R}_D = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_n \neq g(x_n)) \quad (2.9)$$

where $\mathbb{I}$ is the indicator function, defined as: $\mathbb{I}(\text{TRUE}) = 1$ and $\mathbb{I}(\text{FALSE}) = 0$.

Optimizing problem 2.9 directly is generally hard, because of the dis-
continuity of $0 - 1$ loss. Therefore a common approach is to use a smooth
surrogate function for $0 - 1$ loss, which results in an upper bound of empir-
ical risk. Convex surrogate functions have been extensively used in binary
classification problems, for example, hinge loss for support vector machine
($SVM$) and exponential loss for boosting algorithms [7]. Often these kinds
of classifiers are called large margin classifiers. It is known that minimiz-
ing these surrogate functions are Bayes-risk consistent [7]. However, it is
not trivial to extend from binary classification to multi-class classification.
Generally either pairwise, one vs. all, or error code coding approaches are used for constructing multi-class classifiers [64, 39, 10, 111].

In this thesis, we focus on probabilistic classification methods and an upper bound surrogate function derived from probabilities. As we shall see later, this kind of surrogate function might not be convex. Nevertheless, this kind of surrogate function can be naturally extended to multi-class classification; they can be also derived from the maximum-likelihood principle, which is one of the most widely used principles for parameter estimation. For the $0 - 1$ loss function, the empirical risk for $\mathcal{D}$ can be written as

$$R_{\mathcal{D}} = 1 - \frac{1}{N} \sum_{n=1}^{N} p(Y = y_n|X = x_n)$$

(2.10)

Directly optimizing $R_{\mathcal{D}}$ is difficult, therefore the upper bound of $R_{\mathcal{D}}$ is often used in practice. We have $p(Y = y_n|X = x_n) \geq \log p(X = y_n|X = x_n)$ for $p(Y = y_n|X = x_n) \geq 0$. Therefore, we can see that

$$R_{\mathcal{D}} = 1 - \frac{1}{N} \sum_{n=1}^{N} p(Y = y_n|X = x_n)$$

$$\leq 1 - \frac{1}{N} \sum_{n=1}^{N} \log p(Y = y_n|X = x_n)$$
CHAPTER 2. A REVIEW OF CLASSIFICATION

Define the scaled negative conditional log-likelihood of $\mathcal{D}$ as follows,

$$
\mathcal{L}_D(\mathcal{D}) = -\frac{1}{N} \sum_{n=1}^{N} \log p(Y = y_n | X = x_n)
$$  \hspace{1cm} (2.11)

Therefore, we can estimate unknown parameters by minimizing $\mathcal{L}_D(\mathcal{D})$.

For other loss functions, we can follow [35] and bound empirical risk by weighted log-likelihood. This approach can also be derived directly from the maximum-likelihood principle. This approach is called discriminative learning or discriminative estimator in the literature. Minimizing $\mathcal{L}_D$ is closely related to minimizing classification error. As a discriminative approach, this method is robust to model mis-specification ([97, 85]).

On the other hand, it is intuitive that if we can estimate $p(X, Y)$ correctly, we can then calculate $p(Y | X = x)$ correctly. Besides, estimating $p(X, Y)$ is often more computationally efficient. We can define the scaled negative joint log-likelihood of $\mathcal{D}$ as follows,

$$
\mathcal{L}_G(\mathcal{D}) = -\frac{1}{N} \sum_{n=1}^{N} \log p(X = x_n, Y = y_n)
$$  \hspace{1cm} (2.12)

We can estimate unknown parameters by minimizing $\mathcal{L}_G(\mathcal{D})$. This approach is called generative learning or generative estimator in the literature. As a generative approach, this method is often computationally efficient, but
sensitive to model mis-specification. When models are well-specified, this method achieves better performance with limited training data ([97, 85]).

To take the advantage of both generative and discriminative estimation approaches, some researchers proposed to blend generative and discriminative estimator together and estimate model parameters by a *hybrid* way ([12, 80, 91, 102, 38, 51, 11]). One convenient approach is to minimize the convex combination of $L_G$ and $L_D$. The objective function to be minimized can be defined formally as following,

$$L(\Theta, \Upsilon) = \beta L_D(\Theta, \Upsilon) + (1 - \beta) L_G(\Theta, \Upsilon)$$  \hspace{1cm} (2.13)

where $0 \leq \beta \leq 1$. As $\beta$ varies from 0 to 1, we can interpolate between the two approaches mentioned above. We will minimize $L$ to estimate parameters $\Theta$ and $\Upsilon$ in this thesis.
Chapter 3

Quadratic Discriminant Analysis Revisited

Quadratic discriminant analysis (QDA) is a classical generative probabilistic method for classification problems. Essentially speaking, conditional distributions of QDA are assumed to be multivariate Gaussian; posterior distributions are derived via Bayes theorem, and used to classify observations into different classes [39, 64]. Traditionally, parameters of QDA are estimated by maximizing the joint likelihood of observations and their associate class labels. Though computational efficient, this generative approach does not aim at reducing classification error, and is not robust to model mis-specification.

For recent studies on QDA, we refer to [117] and references thereof.

Motivated by recent works on hybrid learning, we propose to estimate
parameters of QDA by maximizing a convex combination of the joint log-likelihood and conditional log-likelihood of given observations and their class labels. Our main contributions of this study are:

- We present a general iterative algorithm for parameter estimation when conditional distributions of classifiers are from the well known exponential family; in each iteration, a convex optimization problem needs to be solved;

- We revisit QDA, and present a block coordinate descent algorithm for solving a specific optimization problem derived for QDA. The algorithm is easy to implement, and is computation and storage efficient.

The remainder of the chapter is organized as follows. In section 3.1, we present an iterative optimization algorithm for parameter estimation when conditional distribution are from the exponential family. Multivariate Gaussian is a specific distribution of the exponential family; we revisit QDA and present a specific block-coordinate descent algorithm for estimating parameters of QDA in section 3.2. We present experimental results in section 3.3, and conclude the study with the summary of our work, and possible future directions in section 3.4.
3.1 Learning For Exponential Family

In this study, we assume conditional distributions are from the exponential family. That is,

\[
p(x|\theta_k) = \exp\{h_k(x) + \langle \Theta_k, T_k(x) \rangle - A_k(\Theta_k)\} \tag{3.1}
\]

where \(T_k(x)\) is the potential function or sufficient statistics; \(h_k(x)\) is a function that does not depend on \(\Theta_k\); \(A_k(\Theta_k)\) is the log partition function. For details about the exponential family, we refer to [130]. Canonical parameter \(\Theta_k \in \mathcal{P}_k\) needs to be estimated; \(\mathcal{P}_k\) is the domain of \(\Theta_k\), which we assume to be convex in this study. We intend to keep \(h_k(x)\)’s and \(T_k(x)\)’s, \(A_k(\Theta_k)\)’s and \(\mathcal{P}_k\)’s abstract at this moment, and will specify them when needed.

Following the definition in Equation (2.13), we can write the objective function to be minimized as follows

\[
\mathcal{L}(\Theta, \Upsilon) = -\frac{1}{N} \sum_{k=1}^{K} \sum_{n \in c_k} \log [p(x_n|k; \Theta_k) \frac{e^{\upsilon_k}}{\sum_{m=1}^{K} e^{\upsilon_m}}] + \beta \frac{1}{N} \sum_{n=1}^{N} \log \sum_{k=1}^{K} [p(x_n|k; \Theta_k) \frac{e^{\upsilon_k}}{\sum_{m=1}^{K} e^{\upsilon_m}}] \tag{3.2}
\]

The optimization problem for estimating parameters \(\Theta\) and \(\Upsilon\) can be writ-
CHAPTER 3. QDA REVISITED

\[ \min_{\Theta, \Upsilon} \mathcal{L}(\Theta, \Upsilon) \]  
\[ \text{s.t. } \Theta_k \in \mathcal{P}_k\quad k = 1, \cdots, K \]  

We note that the optimization problem is not convex, and thus we will be satisfied to find local optima. In the following, we shall present a majorize-minimize (MM) algorithm in which we iteratively minimize a convex upper bound of the original objective function. For an introduction of the MM optimization method, we refer to [71] and references therein.

We note that the nonconvexity of \( \mathcal{L} \) is due to the second term of the right hand side of Equation (3.2). Several lower bounds of it have been proposed in the literature, for example, [131, 74]. As global bounds, they are complex and loose. Note that one important property of log-partition functions \( A_k(\Theta_k), (k = 1, 2, \cdots, K) \), is that they are convex functions of \( \Theta_k \). Therefore we have

\[ A_k(\Theta_k) \geq A_k(\tilde{\Theta}_k) + \left\langle \frac{\partial A_k}{\partial \Theta_k}, \Theta_k - \tilde{\Theta}_k \right\rangle \]  

\[ (3.5) \]
CHAPTER 3. QDA REVISITED

Define an auxiliary function for \( p(x|k; \Theta_k) \) at \( \tilde{\Theta}_k \) as follows,

\[
q(x|k; \Theta_k, \tilde{\Theta}_k) = e^{h_k(x) + \langle \Theta_k, T_k(x) \rangle - A_k(\Theta_k) - \langle \frac{\partial A_k}{\partial \Theta_k} \mid \Theta_k \rangle} \tag{3.6}
\]

Consequently, we have \( p(x|k; \Theta_k) \leq q(x|k; \Theta_k, \tilde{\Theta}_k) \). When \( \Theta_k = \tilde{\Theta}_k \), \( p(x|k; \Theta_k) = q(x|k; \Theta_k, \tilde{\Theta}_k) \). It is then obvious that

\[
\log \sum_{k=1}^{K} e^{\nu_k} p(x_n|k; \Theta_k) \leq \log \sum_{k=1}^{K} e^{\nu_k} q(x_n|k; \Theta_k, \tilde{\Theta}_k) \tag{3.7}
\]

Note that \( \log \sum_{k=1}^{K} e^{\nu_k} q(x_n|k; \Theta_k, \tilde{\Theta}_k) \) is a log-sum-exp function of \( \Theta \), and thus a convex function of \( \Theta \) [13]. Define

\[
\overline{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) = -\frac{1}{N} \sum_{k=1}^{K} \sum_{n \in C_k} \log [p(x_n|k; \Theta_k) \frac{e^{\nu_k}}{\sum_{m=1}^{K} e^{\nu_m}}] \tag{3.8}
\]

\[
+ \frac{\beta}{N} \sum_{n=1}^{N} \log \sum_{k=1}^{K} [q(x_n|k; \Theta_k, \tilde{\Theta}_k) \frac{e^{\nu_k}}{\sum_{m=1}^{K} e^{\nu_m}}]
\]

Then we have \( \overline{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) \) majorizes \( \mathcal{L}(\Theta, \Upsilon) \) at \( \tilde{\Theta} \): \( \mathcal{L}(\Theta, \Upsilon) \geq \overline{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) \), and \( \mathcal{L}(\tilde{\Theta}, \Upsilon) = \overline{\mathcal{L}}(\tilde{\Theta}, \Upsilon; \tilde{\Theta}) \). Moreover, \( \overline{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) \) is a convex function.

Thus we have the following convex optimization problem

\[
\min_{\Theta, \Upsilon} \overline{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) \tag{3.9}
\]

\[
s.t. \quad \Theta_k \in \mathcal{P}_k, \quad k = 1, \cdots, K \tag{3.10}
\]
Let $Θ^*$ and $Υ^*$ be optimal solutions of optimization problem (3.9). We note that, for any $\tilde{Θ}$ and $\tilde{Υ}$,

$$L(Θ^*, Υ^*) \leq \bar{L}(Θ^*, Υ^*; \tilde{Θ}) \leq \bar{L}(\tilde{Θ}, \tilde{Υ}; \tilde{Θ}) = L(\tilde{Θ}, \tilde{Υ})$$  

That is $L(Θ^*, Υ^*)$ is less than or equal to $L(\tilde{Θ}, \tilde{Υ})$. In principal, we can iteratively find a new point $(Θ, Υ)$ to reduce the function value of $L$ until a local optimum is reached. We present in Algorithm 1 the pseudo-code of the MM algorithm in which the hybrid objective function for the exponential family is minimized. We note that

- This algorithm produces a sequence of points that monotonically reduce the original function $L(Θ, Υ)$, until it converges to a local optimum;

- Instead of finding optimums of the optimization problem (3.9) in each iteration, we can find a feasible point that reduces the objective function;

- Since this algorithm can only find local optimums, we may need try several initial points of $Θ$ and $Υ$ in order to get better solutions.
Algorithm 1 A MM optimization algorithm for estimating parameters of the exponential family

1: \( \tau \leftarrow 0 \) \{iteration count\}
2: Initialize feasible \( \Theta^{(\tau)} \), and \( Y^{(\tau)} \) for problem (3.3)
3: while not converge do
4: \( \Theta \leftarrow \Theta^{(\tau)} \)
5: Obtain \( (\Theta^{(\tau+1)}, Y^{(\tau+1)}) \) by solving convex optimization problem (3.9) and (3.10);
6: \( \tau \leftarrow \tau + 1 \)
7: end while

3.2 QDA Revisited

In this section, we shall investigate how to apply algorithm 1 for estimation parameters of QDA. Define

\[
\Theta_k = (\Omega_k, \theta_k) = (\Sigma_k^{-1}, \Sigma_k^{-1} \mu_k) \quad (3.12)
\]

where \( \Sigma_k \) and \( \mu_k \) are, respectively, the covariance matrix and the mean for the \( k \)-th conditional distribution. In the following discussion, we shall use \( \Omega_k, \theta_k \) and \( \Sigma_k, \mu_k \) interchangeably. In QDA, conditional distributions are specified as multivariate Gaussian; thus for \( k = 1, 2, \cdots, K \), we have the densities of conditional distribution expressed in the standard form of the exponential family

\[
p(x|k; \Theta_k) = e^{h_k(x) + (\Theta_k, T_k(x)) - A_k(\Theta)} \quad (3.13)
\]
where

\[ h_k(x) = -\frac{d}{2} \log(2\pi) \quad T_k(x) = \left( -\frac{1}{2}xx^T, \ x \right) \] (3.14)

and

\[ A_k(\Theta_k) = \frac{1}{2} \theta_k^T \Omega_k^{-1} \theta_k - \frac{1}{2} \log \det \Omega_k \] (3.15)

Note that

\[ \frac{\partial}{\partial \Theta_k} A_k(\Theta_k) = \left( \frac{\partial}{\partial \Omega_k} A_k(\Theta_k), \frac{\partial}{\partial \theta_k} A_k(\Theta_k) \right) \] (3.16)

\[ = \left( -\frac{1}{2} \Omega_k^{-1} \theta_k \theta_k^T \Omega_k^{-1} - \frac{1}{2} \Omega_k^{-1}, \mu_k \right) \] (3.17)

We thus have the auxiliary function for \( p(x|k; \Theta_k) \) at \( \tilde{\Theta}_k = (\tilde{\Omega}_k, \tilde{\theta}_k) \) as following,

\[ q(x|k; \Theta_k, \tilde{\Theta}_k) = e^{\frac{1}{2} (\tilde{M}_k - xx^T, \Omega_k)} + (x - \tilde{\mu}_k, \tilde{\theta}_k) + c_k \] (3.18)

where

\[ \tilde{M}_k = \tilde{\Omega}_k^{-1} + \Omega_k^{-1} \theta_k \theta_k^T \Omega_k^{-1} \] (3.19)

\[ c_k = -\frac{d}{2} \log(2\pi) - A_k(\tilde{\Theta}_k) + \left( \tilde{\mu}_k, \tilde{\theta}_k \right) - \frac{1}{2} \left( \tilde{M}_k, \tilde{\Omega}_k \right) \] (3.20)
The convex upper bound of \( \mathcal{L}(\Theta, \Upsilon) \) is as follows

\[
\bar{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) = \frac{\beta}{N} \sum_{n=1}^{N} \log \left[ \sum_{k=1}^{K} e^{v_k} q(x_n|k; \Theta_k, \tilde{\Theta}_k) \right] \\
- \frac{1}{N} \sum_{k=1}^{K} \sum_{n \in C_k} \left[ \langle x_n, \theta_k \rangle - \frac{1}{2} \langle x_n x_n^T, \Omega_k \rangle \right] + (1 - \beta) \log \sum_{k=1}^{K} e^{v_k} \\
- \sum_{k=1}^{K} \frac{N_k}{N} [v_k - \frac{1}{2} \theta_k^T \Omega_k^{-1} \theta_k + \frac{1}{2} \log \det \Omega_k] 
\]

(3.21)

We require covariance matrices being positive definite; i.e. \( \Sigma_k \succ 0 \). Equivalently, we can have \( \Omega_k \succ 0 \). In practice, we can have \( \Omega_k \succeq \alpha I \), where \( \alpha > 0 \).

We can apply algorithm 1 to estimate \( \Theta \) and \( \Upsilon \). In each iteration of Algorithm 1, we need solve the following convex optimization problem

\[
\min_{\Theta, \Upsilon} \bar{\mathcal{L}}(\Theta, \Upsilon; \tilde{\Theta}) \tag{3.22}
\]

\[
s.t. \quad \Omega_k \succeq \alpha I, \quad k = 1, 2, \ldots, K \tag{3.23}
\]

where \( \alpha > 0 \) is given. To the best of our knowledge, there is no existing optimization method to solve this problem. \(^1\)

\(^1\)In a linear semidefinite programming problem (SDP), one optimizes a linear objective function of matrix variables, which are constrained to be positive (semi)definite. Linear SDPs have been extensively studied in the optimization literature, as many types of convex optimization problems can be expressed as linear SDPs [126, 13]. Many machine learning problems can be expressed or approximated as linear SDPs, e.g. [79, 105, 113, 132]. Recently, there has been increased interest on quadratic SDPs, of which a convex quadratic objective function is optimized with positive (semi)definite constraints on matrix variables, for example [67, 89, 14, 101, 118]. However, we need to optimize, with positive
In the following, we will present a block-coordinate descent algorithm for optimization problem (3.22) and (3.23) that sequentially updates $\Upsilon$, $\theta_1$, $\theta_2$, $\ldots$, $\theta_K$ and $\Omega_1$, $\Omega_2$, $\ldots$, $\Omega_K$. In each update we analytically define a sequence of feasible points to decrease the objective function. We designed such updates for the following two reasons:

- In each update, the sequence of feasible points, though not the shortest path to reach optimum, can be efficiently determined. We hope that for practical problems the computational efficiency of each iteration can compensate for many iterations required to obtain an optimum.

- More importantly it is well understood that optimization with many small steps and early stopping helps boosting methods to achieve better classification results [139, 50]. The updates particularly enable us to tune for classification performance.

In summary the algorithm can be implemented with BLAS and LAPACK routines, which are numerical robust, efficient and reliable. Moreover, as parameters are sequentially updated, the memory requirement of the algorithm is much less than algorithms that update parameters altogether.

definite constraints (3.23), objective function (3.22) which is neither linear nor quadratic. Therefore, we cannot directly utilize existing optimization methods for linear or quadratic SDPs.
We will discuss the details of optimization algorithms for $\Upsilon$ in subsection 3.2.1, for $\theta_k$ in subsection 3.2.2, and for $\Omega_k$ in subsection 3.2.3; we will discuss selecting initial points, early stopping and $\beta$ in subsection 3.2.4.

3.2.1 Updating $v_k$'s

In this subsection, we assume $\Theta_k$'s are fixed: for $k = 1, 2, \cdots, K$, $\theta_k = \tilde{\theta}_k$, and $\Omega_k = \tilde{\Omega}_k$. Since $q(x_n|k; \tilde{\Theta}_k, \tilde{\Theta}_k) = p(x_n|k; \tilde{\Theta}_k)$, we have the optimization problem for $\Upsilon = (v_1, v_2, \cdots, v_K)$ as follows

$$\min_{\Upsilon} - \sum_{k=1}^{K} \frac{N_k}{N} v_k + \frac{\beta}{N} \sum_{n=1}^{N} \log \sum_{k=1}^{K} p(x_n|k; \tilde{\Theta}_k) e^{v_k} + (1 - \beta) \log \sum_{k=1}^{K} e^{v_k}$$ (3.24)

Let $\Upsilon^* = (v_1^*, v_2^*, \cdots, v_K^*)$ be the optimal solution of optimization problem (3.24). We therefore must have

$$\frac{\beta}{N} \sum_{k=1}^{N} p(m|x_n; \tilde{\Theta}, \Upsilon^*) = \frac{N_m}{N} - (1 - \beta) \frac{e^{v_m^*}}{\sum_{k=1}^{K} e^{v_k^*}}$$ (3.25)

where $m = 1, 2, \cdots, K$. Specifically, when $\beta = 1$, $\sum_{k=1}^{N} p(k|x_n; \tilde{\Theta}, \Upsilon^*) = N_k$; that is the sum of posterior densities for class $k$ is equal to the number of observations of class $k$. When $\beta = 0$, $\frac{e^{v_m^*}}{\sum_{k=1}^{K} e^{v_k^*}} = \frac{N_m}{N}$, which is widely used for classical QDA [39, 64].

In Appendix A.1, we show an iterative optimization procedure for (3.24),
of which we have closed-form solutions in each iteration. The pseudo-code of an iterative algorithm to find $\Upsilon^*$ is described in Algorithm 2.

**Algorithm 2** An iterative algorithm for finding $\Upsilon^*$

1: $\tau \leftarrow 0 \{\tau$: iteration count}\n2: $v_k^{(\tau)} \leftarrow \log \frac{N_k}{N}$, $k = 1, 2, \cdots, K$
3: **while** not converge **do**
4: **for** $k = 1, 2, \cdots, K$ **do**
5: $p(k|x_n; \Theta_k, \Upsilon^{(\tau)}) = \frac{p(x_n|k; \Theta_k) e^{v_k^{(\tau)}}}{\sum_{k=1}^{K} p(x_n|k; \Theta_k) e^{v_k^{(\tau)}}}$
6: $\pi_k^{(\tau)} = \frac{e^{v_k^{(\tau)}}}{\sum_{k=1}^{K} e^{v_k^{(\tau)}}}$
7: $\lambda_k = \frac{1}{N} \sum_{n=1}^{N} p(k|x_n; \Theta_k, \Upsilon^{(\tau)})$, $k = 1, 2, \cdots, K$
8: $\phi_k^* = \log \frac{N_k}{\lambda_k} (1 - \beta \pi_k^{(\tau)})$
9: $v_k^{(\tau+1)} \leftarrow v_k^{(\tau)} + \phi_k^*$
10: **end for**
11: $\tau \leftarrow \tau + 1$
12: **end while**

3.2.2 Updating $\theta_k$'s

In this subsection, we consider $v_k$’s and $\Omega_k$’s are fixed; that is, for $k = 1, \cdots, K$, $v_k = v_k^*$, and $\Omega_k = \bar{\Omega}_k$. Furthermore, we consider all $\theta_k$’s but $\theta_m$ are fixed; that is $\forall k < m, \theta_k = \theta_k^*$; and $\forall k > m$, $\theta_k = \bar{\theta}_k$. For the convenience of discussion, we define $f(\theta_m)$ in this subsection as follows,

$$f(\theta_m) = \bar{L}(\theta_1^*, \cdots, \theta_{m-1}^*, \theta_m, \bar{\theta}_{m+1}, \cdots, \bar{\theta}_K, \bar{\Omega}_1, \cdots, \bar{\Omega}_K, \Upsilon^*)$$ (3.26)
CHAPTER 3. QDA REVISITED

Then the optimization problem for $\theta_m$ is as follows

$$\min_{\theta_m} f(\theta_m)$$  \hspace{1cm} (3.27)

This optimization problem is convex, and can be solved by a standard convex optimization software, such as IPOPT [129].

In the following, we will present a trust region method to solve optimization problem (3.27), which is easy to implement and computationally efficient. We refer to [22, 137] for details about trust region method. For the convenience of future discussion, we define

$$\phi_m = \theta_m - \theta_m^{(\tau)}$$  \hspace{1cm} (3.28)

and

$$q^{(\tau)}(m|x_n) = \frac{e^{\nu^*_m q(x_n|m; \Theta_m^{(\tau)}, \tilde{\theta}_m)}}{\sum_{k=1}^{K} e^{\nu^*_k q(x_n|k; \Theta_k^{(\tau)}, \tilde{\theta}_k)}}$$

where $\theta_m^{(\tau)}$ is an estimation of $\theta_m$ at $\tau$-th iteration, and

$$\Theta_k^{(\tau)} = \begin{cases} (\bar{\Omega}_k, \theta_k^*), & k = 1, 2, \ldots, m - 1 \\ (\bar{\Omega}_k, \theta_k^{(\tau)}), & k = m \\ (\bar{\Omega}_k, \tilde{\theta}_k), & k = m + 1, m + 2, \ldots, K \end{cases}$$  \hspace{1cm} (3.29)
The second order Taylor expansion of $f(\theta_m)$ around $\theta_m^{(\tau)}$ is

$$f(\theta_m) \approx f(\theta_m^{(\tau)}) + g_m^T \phi_m + \frac{1}{2} \phi_m^T H_m \phi_m$$  \hspace{1cm} (3.30)

where

$$H_m = \frac{\beta}{N} \sum_{n=1}^{N} q^{(\tau)}(m|x_n)[1 - q^{(\tau)}(m|x_n)](x_n - \tilde{\mu}_m)(x_n - \tilde{\mu}_m)^T$$  \hspace{1cm} (3.31)

$$g_m = \frac{1}{N} \left\{ \sum_{n=1}^{N} \beta q^{(\tau)}(m|x_n)(x_n - \tilde{\mu}_m) - \sum_{n \in C_m} (x_n - \mu_m^{(\tau)}) \right\}$$  \hspace{1cm} (3.32)

Let $\lambda$ be the 1-norm of matrix $H_m$, which is the maximum absolute column sum of $H_m$. Therefore, we have $H_m \preceq \lambda I$. We solve the following trust region subproblem in the $\tau$-th iteration,

$$\min_{\phi_m} \frac{1}{2} \phi_m^T H_m \phi_m + g_m^T \phi_m$$ \hspace{1cm} (3.33)

$$\text{s.t.} \quad \phi_m^T \phi_m \leq \epsilon^{(\tau)}$$  \hspace{1cm} (3.34)

We show in Appendix A.2 that the optimal solution of problem (3.33) and (3.34) is

$$\phi_m^* = -\frac{g_m}{\max(\sqrt{\frac{g_m^T g_m}{\epsilon^{(\tau)}}, \lambda})}$$ \hspace{1cm} (3.35)

To summarize, we present the pseudo code of the trust region algorithm.
for finding \( \theta_m^* \) in algorithm 3.

**Algorithm 3** An iterative trust-region algorithm for finding \( \theta_m^* \)

1. Specify \( 0 < \eta_3 < \eta_4 < 1 < \eta_1, 0 \leq \eta_0 \leq \eta_2 < 1, \eta_2 > 0 \), and \( \epsilon^{(0)} \) \{See [137] \}
2. \( \tau \leftarrow 0 \) \{\( \tau \):iteration count\}
3. \( \theta_m^{(\tau)} \leftarrow \tilde{\theta}_m \)
4. \textbf{while} not converge \textbf{do}
5. \( \phi_m^* = -g_m / \max(\sqrt{\frac{g_m^T g_m}{\epsilon^{(\tau)}}, \lambda}) \)
6. \( r = \frac{f(\theta_m + \phi_m^*) - f(\theta_m)}{\|\phi_m^*\|^2 + 0.5\lambda \phi_m^T \phi_m} \)
7. if \( r > \eta_0 \), \( \theta_{m}^{(\tau + 1)} \leftarrow \theta_{m}^{(\tau)} + \phi_m^* \); otherwise, \( \theta_{m}^{(\tau + 1)} \leftarrow \theta_{m}^{(\tau)} \)
8. if \( r \geq \eta_2 \), update \( \epsilon^{(\tau + 1)} \in [\epsilon^{(\tau)}, \eta_1 \epsilon^{(\tau)}] \); otherwise, update \( \epsilon^{(\tau + 1)} \in [\eta_3 \|\phi_m^*\|^2, \eta_4 \epsilon^{(\tau)}] \);
9. \( \tau \leftarrow \tau + 1 \)
10. \textbf{end while}

### 3.2.3 Updating \( \Omega_k \)'s

In this subsection, we consider \( \upsilon_k \)'s and \( \theta_k \)'s are fixed; that is, for \( k = 1, \cdots, K \), \( \upsilon_k = \upsilon_k^* \), and \( \theta_k = \theta_k^* \). Furthermore, consider all \( \Omega_k \)'s but \( \Omega_m \) are fixed; that is \( \forall k < m, \Omega_k = \Omega_k^*; \forall k > m, \Omega_k = \tilde{\Omega}_k \). We define \( h(\Omega_m) \) in this subsection as follows,

\[
h(\Omega_m) = \mathcal{T}(\theta_1^*, \cdots, \theta_K^*, \Omega_1^*, \cdots, \Omega_{m-1}^*, \Omega_m, \tilde{\Omega}_{m+1}, \cdots, \tilde{\Omega}_K, \Upsilon^*) \quad (3.36)
\]

So the optimization problem for \( \Omega_m \) is

\[
\min_{\Omega_m} h(\Omega_m) \quad (3.37)
\]
s.t. $\Omega_m \succeq \alpha I$ \hfill (3.38)

To our best knowledge, there is no existing method to solve this optimization problem. We will present an iterative trust region method to solve this optimization problem in the remaining of this subsection.

Let $\Omega_m^{(\tau)}$ be an estimation of $\Omega_m$ at $\tau$-th iteration. We define

$$
\Phi_m = (\Omega_m^{(\tau)})^{-\frac{1}{2}}(\Omega_m - \Omega_m^{(\tau)})(\Omega_m^{(\tau)})^{-\frac{1}{2}} \hfill (3.39)
$$

Then by $\Omega_m \succeq \alpha I$, we have

$$
\Phi_m \succeq \alpha (\Omega_m^{(\tau)})^{-1} - I \hfill (3.40)
$$

Define

$$
q^{(\tau)}(m|x_n) = \frac{e^{v^*_m q(x_n|\Theta_m^{(\tau)}, \tilde{\Theta}_m)}}{\sum_{k=1}^{K} e^{v^*_k q(x_n|\Theta_k^{(\tau)}, \tilde{\Theta}_k)}} \hfill (3.41)
$$

where

$$
\Theta_k^{(\tau)} = \begin{cases} 
(\Omega_k, \theta_k^*), & k = 1, 2, \ldots, m \\
(\Omega_k^{(\tau)}, \theta_k^*), & k = m \\
(\tilde{\Omega}_k, \theta_k^*), & k = m + 1, m + 2, \ldots, K
\end{cases} \hfill (3.42)
$$

The second order Taylor expansion of $h(\Omega_m)$ around $\Omega_m^{(\tau)}$ is as follows

$$
h(\Omega_m) \approx h(\Omega_m^{(\tau)}) + \langle G_m, \Phi_m \rangle + \frac{1}{2} \langle H(\Phi_m), \Phi_m \rangle \hfill (3.43)
$$
where

\[
\mathcal{H}(\Phi_m) = \frac{\beta}{4N} B_m(r) + \frac{N_m}{2N} \{ S_m(r) \Phi_m + \Phi_m S_m(r) - \Phi_m \} \quad (3.44)
\]

\[
G_m = \frac{1}{2N} \{ \beta \sum_{n=1}^{N} q(r)(m|x_n) \tilde{X}_n(r) - \sum_{n \in C_m} X_n(r) \} \quad (3.45)
\]

and

\[
S_m(r) = (\Omega(r))^{\frac{1}{2}} \{ (\Omega(r))^{-1} + (\Omega(r))^{-1} \theta_m^*(\theta_m^*)^T (\Omega(r))^{-1} \} (\Omega(r))^{\frac{1}{2}} \quad (3.46)
\]

\[
X_n(r) = (\Omega(r))^{\frac{1}{2}} \{ (\Omega(r))^{-1} + (\Omega(r))^{-1} \theta_m^*(\theta_m^*)^T (\Omega(r))^{-1} \} (\Omega(r))^{\frac{1}{2}} - x_n x_n^T \quad (3.47)
\]

\[
\tilde{X}_n(r) = (\Omega(r))^{\frac{1}{2}} \{ \Omega^{-1} + \tilde{\Omega}^{-1} \theta_m^* \tilde{\Omega}^{-1} \} (\Omega(r))^{\frac{1}{2}} \quad (3.48)
\]

\[
B_m(r) = \sum_{n=1}^{N} q(r)(m|x_n) [1 - q(r)(m|x_n)] \langle \tilde{X}_n(r), \Phi_m \rangle \tilde{X}_n(r) \quad (3.49)
\]

For an iterative optimization algorithm that uses Hessian information, the Hessian matrix needs to be updated at each iteration. We note that the first term \( B_m(r) \) of the right hand side of (3.44) is computationally expensive to update. We also note that

\[
S_m(r)(\Omega(r))^{-\frac{1}{2}} \theta_m^* = (1 + (\theta_m^*)^T (\Omega_m(r))^{-1} \theta_m^*) (\Omega(r))^{-\frac{1}{2}} \theta_m^* \quad (3.50)
\]
That is \((\Omega^{(\tau)})^{-\frac{1}{2}}\theta^*_m\) is the eigenvector of \(S_m^{(\tau)}\) with corresponding eigenvalue \(1 + (\theta^*_m)^T(\Omega_m^{(\tau)})^{-1}\theta^*_m\). It can be easily verified that \(1 + (\theta^*_m)^T(\Omega_m^{(\tau)})^{-1}\theta^*_m\) is indeed the largest eigenvalue of \(S_m^{(\tau)}\). Thus we have that, \(\forall \Phi_m\),

\[
\frac{N_m}{2N}\{S_m^{(\tau)}\Phi_m + \Phi_m S_m^{(\tau)} - \Phi_m\} \preceq \lambda \Phi_m
\]

(3.51)

where

\[
\lambda = \frac{N_m}{N}\{(\theta^*_m)^T(\Omega_m^{(\tau)})^{-1}\theta^*_m + \frac{1}{2}\}
\]

(3.52)

We therefore approximate the Hessian matrix (3.44) by \(\lambda I\), and \(\lambda\) is easy to calculate.

We solve the following trust region subproblem in the \(\tau\)-th iteration

\[
\min_{\Phi_m} \frac{\lambda}{2} \langle \Phi_m, \Phi_m \rangle + \langle G_m, \Phi_m \rangle
\]

s.t. \(\Phi_m \succeq \alpha(\Omega_m^{(\tau)})^{-1} - I\)

(3.54)

\[
\langle \Phi_m, \Phi_m \rangle \leq \epsilon^{(\tau)}
\]

(3.55)

where \(\lambda\) is defined in Equation (3.52). In Appendix A.3, we show that we
can approximately solve this problem by \(^2\)

\[
\Phi_m^* = \sqrt{\min\left(\frac{e^{(r)}}{\rho}, 1\right)}\{C^{(r)}_\tau + \alpha(\Omega_m^{(r)})^{-1} - I\} \tag{3.56}
\]

where

\[
C^{(r)} = \frac{1}{\lambda} G_m + \alpha(\Omega_m^{(r)})^{-1} - I \tag{3.57}
\]

\[
\rho = \|C^{(r)}_\tau + \alpha(\Omega_m^{(r)})^{-1} - I\|_F^2 \tag{3.58}
\]

and \(C^{(r)}_\tau\) is the negative semidefinite parts of \(C^{(r)}\); let \(C^{(r)} = \sum_{i=1}^d \lambda_i e_i e_i^T\) be the eigen-decomposition of \(C^{(r)}\), then \(C^{(r)}_\tau = \sum_{i=1}^d \max(-\lambda_i, 0)e_i e_i^T\), (see [14]).

In summary, we present in Algorithm 4 the pseudo-code of the trust region algorithm for finding \(\Omega_m^*\).

### 3.2.4 Initial Points, Early Stopping and \(\beta\)

We use the following as our initial points,

\[
\Omega_k^{(0)} = \text{diag}(\text{diag}(\hat{\Sigma}_k))^+ + \alpha I \tag{3.59}
\]

\[
\theta_k^{(0)} = \frac{1}{N_k} \sum_{n \in C_k} \Omega_k^{(0)} x_n \tag{3.60}
\]

\(^2\)To be precise, \(\Phi_m^*\) is a solution of the following feasibility problem: \(\frac{1}{2} \langle \Phi_m, \Phi_m \rangle + \langle G_m, \Phi_m \rangle \leq 0\), \(\Phi_m \succeq \alpha(\Omega_m^{(r)})^{-1} - I\), and \(\langle \Phi_m, \Phi_m \rangle \leq \epsilon\).
CHAPTER 3. QDA REVISITED

Algorithm 4 An trust-region algorithm for finding $\Omega^*_m$

1: Specify $0 < \eta_3 < \eta_4 < 1 < \eta_1$, $0 \leq \eta_0 \leq \eta_2 < 1$, $\eta_2 > 0$, and $\epsilon^{(0)}$ \{see \cite{137}\}
2: $\tau \leftarrow 0$ \{iteration count\}
3: $\Omega^{(\tau)}_m \leftarrow \Omega_m$
4: while not converge do
5: calculate $\Phi^*_m$ as described in equation (3.56)
6: $\Omega_m = \Omega^{(\tau)}_m + (\Omega^{(\tau)}_m)^{1/2} \Phi^*_m (\Omega^{(\tau)}_m)^{1/2}$
7: $r = \frac{h(\Omega_m) - h(\Omega^{(\tau)}_m)}{0.5 \lambda (\Phi^*_m)^{\top} (\Phi^*_m) + (\Omega_m)^{\top} \Omega_m}$
8: if $r > \eta_0$, $\Omega^{(\tau+1)}_m \leftarrow \Omega_m$; otherwise, $\Omega^{(\tau+1)}_m \leftarrow \Omega^{(\tau)}_m$
9: if $r \geq \eta_2$, update $\epsilon^{(\tau+1)} \in [\epsilon^{(\tau)}, \eta_1 \epsilon^{(\tau)}]$; otherwise, update $\epsilon^{(\tau+1)} \in [\eta_0 \|\Phi^*_m\|_2, \eta_4 \epsilon^{(\tau)}]$;
10: $\tau \leftarrow \tau + 1$
11: end while

$$v_k^{(0)} = \log \frac{N_k}{N}. \quad (3.61)$$

where $\hat{\Sigma}_k$'s are sample covariance matrices for class $k$; $\Box^+$ is the pseudo-inverse of a matrix $\Box$.

Early stopping has been shown to be an effective regularization method for boosting algorithms \cite{139, 108}. Motivated by their works, we use early stopping in Algorithm 1. We use cross-validation to determine $\beta$ and the early stopping step in the iterative MM algorithm.
3.3 Numerical Experiments

3.3.1 Data and Benchmark Methods

We selected 2 artificial data sets used in [15] and 23 real data sets from UCI machine learning repository [47]. The statistics of the selected data sets are listed in table 3.1. The two artificial data sets are as follows:

Dataset twonorm It is a 20-dimensional binary classification data. Each class is drawn from a multivariate normal distribution with identity covariance matrix. Class 1 has mean \((2, 2, \cdots, 2)/\sqrt{20}\), and class 2 has mean \(-(2, 2, \cdots, 2)/\sqrt{20}\). It is reported in [15] that this dataset has an expected error rate of 2.3%.

Dataset ringnorm It is a 20-dimensional binary classification data. Class 1 is drawn from a multivariate normal distribution with zero mean and covariance \(4I\). Class 2 is drawn from a multivariate normal distribution with mean \((2, 2, \cdots, 2)/\sqrt{20}\) and identity covariance matrix. It is reported in [15] that this dataset has an expected error rate of 1.3%.

In our numerical experiments, we repeated the following procedure 20 times for each dataset. For all datasets, we randomly selected 40% of the
TABLE 3.1: Descriptions of 25 benchmark datasets ($K$: number of classes, $d$: dimension, $N$: number of observations)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$d$</th>
<th>$N$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twonorm</td>
<td>20</td>
<td>7400</td>
<td>2</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>20</td>
<td>7400</td>
<td>2</td>
</tr>
<tr>
<td>Australian</td>
<td>14</td>
<td>690</td>
<td>2</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>4</td>
<td>748</td>
<td>2</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>30</td>
<td>569</td>
<td>2</td>
</tr>
<tr>
<td>Breast Tissue</td>
<td>9</td>
<td>106</td>
<td>6</td>
</tr>
<tr>
<td>BUPA</td>
<td>6</td>
<td>345</td>
<td>2</td>
</tr>
<tr>
<td>Climate</td>
<td>20</td>
<td>540</td>
<td>2</td>
</tr>
<tr>
<td>Diabetes</td>
<td>8</td>
<td>768</td>
<td>2</td>
</tr>
<tr>
<td>German</td>
<td>24</td>
<td>1000</td>
<td>2</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>270</td>
<td>2</td>
</tr>
<tr>
<td>ILPD</td>
<td>10</td>
<td>583</td>
<td>2</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>18</td>
<td>2310</td>
<td>7</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>33</td>
<td>351</td>
<td>2</td>
</tr>
<tr>
<td>Iris</td>
<td>4</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>Magic</td>
<td>10</td>
<td>19020</td>
<td>2</td>
</tr>
<tr>
<td>Mammographic</td>
<td>5</td>
<td>830</td>
<td>2</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>22</td>
<td>195</td>
<td>2</td>
</tr>
<tr>
<td>Sat. Images</td>
<td>36</td>
<td>6435</td>
<td>6</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Splice</td>
<td>60</td>
<td>3175</td>
<td>3</td>
</tr>
<tr>
<td>Thyroid</td>
<td>5</td>
<td>215</td>
<td>3</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>6</td>
<td>310</td>
<td>3</td>
</tr>
<tr>
<td>Vowel</td>
<td>13</td>
<td>990</td>
<td>11</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>178</td>
<td>3</td>
</tr>
</tbody>
</table>

data as training sets, and the remaining data are used as test sets; the only exception is when we illustrated test error rates for various percentage training sample size at Subsection 3.3.3. We standardized training data using the sample mean and standard deviation of training data. All methods are trained with training sets, and evaluated on corresponding test sets.
For the convenience of following discussion, we denote our method by MM QDA. We benchmarked MM QDA against related approaches to QDA: naive Bayes, classical QDA (QDA), regularized discriminant analysis (RDA) [49] Bayesian quadratical discriminant analysis (BDA7) [117], and hybrid learning QDA with a conjugate gradient method (CG QDA) [12]. For reference, we also benchmarked with two discriminative quadratic classifiers: L2-regularized logistic regression with linear and quadratic terms (LR), and SVM with polynomial kernel \((d = 2)\) (QSVM). Unless otherwise specified, we used the following settings for the methods in this study:

- Naive Bayes and classical QDA are two widely used classification methods [64, 39]. We use matlab built-in function classify for naive Bayes and QDA. For some datasets, function classify failed because covariance matrices are not positive definite; for these cases, we reported test error rates as −.

- RDA is a convex combination between linear and quadratic discriminant analysis with proper shrinkage regularization [49]. It is perhaps the most popular approach to statistical discriminant analysis, especially for undersampled problems [64]. We used the original code for
RDA, which is available from http://www-stat.stanford.edu/~jhf. We used leave-one-out approach to select optimal parameters for RDA on a \((0,0.1,\cdots,1) \times (0,0.1,\cdots,1)\) grid.

- Srivastava, Gupta and Frigyik (2007) proposed BDA7, a Bayesian distribution-based approach to QDA, to minimize the expected misclassification cost [117]. We used the original code for BDA7, which is available from http://www.ee.washington.edu/research/guptalab. For BDA7, we used the same settings as described in [117], and used leave-one-out approach to select optimal parameter from 42 parameter choices.

- For our method, we let \(\alpha = 10^{-6}\) and \(\beta = (0,0.1,\cdots,1.0)\), and used 10 fold cross validation to select optimal \(\beta\) and early stopping iteration.

- Bouchard and Triggs (2004) [12] used conjugate gradient method to optimize the same objective function as ours. However, an important drawback is that there is no regularization in their approach, as we observed that proper regularization is often critical to achieve better classification results. To make comparison as fair as possible, we used the same initial point and the optimal \(\beta\) selected from our method; we trained CG QDA with \(\frac{1}{2}\)-, 1- and 2-times of iterations determined
from our method, which we denoted by CG QDA (1/2), CG QDA (1), and CG QDA (2), respectively.

- For LR and QSVM, we used Weka and searched regularization coefficients in $[0, 10]$ by 10-fold cross validation [60].

### 3.3.2 Effect of $\beta$ on Classification Error

We examined the influence of $\beta$ on test set error rates by varying $\beta$ from 0 (discriminative) to 1 (generative). For Datasets twonorm, ringnorm, diabetes, breast cancer ionosphere and sonar, we showed boxplots of test set error rates in figure 3.1. For artificial datasets twonorm and ringnorm, generative learning ($\beta = 1$) gives the smallest mean error rates. For diabetes and breast cancer, generative learning ($\beta = 1$) gives the smallest mean and median error rates. For ionosphere, discriminative learning ($\beta = 0$) gives the smallest mean and median error rates. And for sonar, combinations of discriminative and generative learning ($\beta = 0.2$) gives the best performance.

### 3.3.3 Effect of Training Sample Size on Classification Error

We examined the effect of different training sample size on test set error rates. For this purpose, we randomly chose 10%, 20%, ..., 90% samples of a given dataset as training samples, and the remaining as test samples.
Figure 3.1: Influence of coefficient of $\beta$ on test set error rates: $\beta = 0$ is discriminative, and $\beta = 1$ is generative. Red bars are the median test error rates over 20 runs, while blue circles is the mean test set error rates over 20 runs.
Figures 3.2, 3.3, and 3.4 show mean test error rates over 20 runs for datasets twonorm, ringnorm, breast cancer, diabetes, ionosphere and sonar. Except for diabetes with 10% training data, our method is competitive or superior than BDA7 and RDA; this shows that our method is as statistically efficient as BDA7 and RDA. Generally speaking, our method is better than CG QDA methods. By comparing the performance of CG QDA methods we also observed that early stopping does affect classification performance, as often CG QDA (1/2) and/or CG QDA (1) have smaller error rates than CG QDA (2).

3.3.4 Comparison on Classification Error

Mean and standard deviation error rates for 25 benchmark datasets are reported in tables 3.3.4 and 3.3.4, respectively. For compactness, we reported the smallest mean test set error rates and the corresponding standard deviation of CG(1/2), CG(1) and CG(2) under CG QDA in tables 3.3.4 and 3.3.4.. In table 3.3.4, the smallest mean error rates are emphasized with bold numbers. We also reported other mean error rates in bold, which are not statistically significantly different from the best ones; for this purpose, we used Wilcoxon signed-rank test with significant level 0.05 to compare error rates. As shown in table 3.3.4, in 15 of the 25 datasets our method is either
Figure 3.2: Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods.
Figure 3.3: Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods.
Figure 3.4: Influence of training sample size on test set error rates. Green, blue, red and black lines are, respectively, for RDA, BDA7, our method and CG QDA methods.
the best one, or not statistically significantly different from the best ones.

For each dataset, we calculated the rank for each method based on mean test error rates: a method with the smallest mean error rate has rank 1, while a method with the largest error rate has rank 8. In Figure 3.5, we showed statistical boxplots of the ranks of these methods (red bars and green circles are, respectively, the median and mean ranks over the 25 datasets). Figure 3.5 shows that our method has the best rank when compared with other methods. The experiments show that our method is competitive with, and in some cases significantly better than other widely used quadratic classifiers in the literature.

### 3.3.5 Comparison with Conjugate Gradient Method

Bouchard and Triggs [12] used conjugate gradient (CG) method to optimize the same objective function as ours. We compared the new proposed method against CG for training QDA. For this purpose, we disabled early stopping in our method, and performed numerical experiments on an Intel Xeon (3.30GHz) computer with 16GB RAM. We restricted the maximum number of iterations to be 3000 for both methods. CG method reported local optimums within 3000 iterations for all datasets; our methods reported local optimums within 3000 iterations for all datasets except ionosphere. Computational speed (mean and std) and objective function values (mean and std) for both methods are reported in table 3.3.5. In 22 of 25 datasets, our method is as fast as or faster than CG method; in 16 of 25 datasets, our method terminates with equal or better local optimums. To further
Table 3.2: Average test set error rates for 25 datasets (Shown in percentage)

<table>
<thead>
<tr>
<th>Class</th>
<th>Naive Bayes</th>
<th>QDA</th>
<th>RDA</th>
<th>BDAT</th>
<th>LR</th>
<th>QSVM</th>
<th>CG QDA</th>
<th>MM QDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwoNorm</td>
<td>2.19</td>
<td>2.37</td>
<td>2.38</td>
<td>2.37</td>
<td>2.74</td>
<td>2.37</td>
<td>2.36</td>
<td>2.20</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.33</td>
<td>1.44</td>
<td>1.34</td>
<td>1.48</td>
<td>2.05</td>
<td>2.30</td>
<td>1.48</td>
<td>1.35</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>7.05</td>
<td>5.37</td>
<td>4.86</td>
<td>4.63</td>
<td>3.41</td>
<td>3.28</td>
<td>4.02</td>
<td>3.65</td>
</tr>
<tr>
<td>Breast Tissue</td>
<td>38.03</td>
<td>–</td>
<td>41.91</td>
<td>39.75</td>
<td>42.83</td>
<td>41.15</td>
<td>37.00</td>
<td>37.49</td>
</tr>
<tr>
<td>BUPA</td>
<td>46.78</td>
<td>42.37</td>
<td>34.75</td>
<td>39.35</td>
<td>30.59</td>
<td>31.89</td>
<td>34.36</td>
<td>36.88</td>
</tr>
<tr>
<td>Climate</td>
<td>10.25</td>
<td>–</td>
<td>6.81</td>
<td>6.83</td>
<td>7.08</td>
<td>8.03</td>
<td>6.66</td>
<td>6.78</td>
</tr>
<tr>
<td>Diabetes</td>
<td>25.61</td>
<td>26.68</td>
<td>23.86</td>
<td>25.75</td>
<td>23.52</td>
<td>23.79</td>
<td>24.39</td>
<td>24.09</td>
</tr>
<tr>
<td>German</td>
<td>24.98</td>
<td>–</td>
<td>25.92</td>
<td>25.41</td>
<td>29.77</td>
<td>26.28</td>
<td>25.99</td>
<td></td>
</tr>
<tr>
<td>Heart</td>
<td>17.44</td>
<td>22.68</td>
<td>16.56</td>
<td>17.87</td>
<td>18.69</td>
<td>20.70</td>
<td>20.38</td>
<td>18.49</td>
</tr>
<tr>
<td>ILPD</td>
<td>43.76</td>
<td>43.23</td>
<td>30.81</td>
<td>38.20</td>
<td>29.25</td>
<td>29.80</td>
<td>34.29</td>
<td>33.75</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>–</td>
<td>–</td>
<td>8.77</td>
<td>9.71</td>
<td>5.26</td>
<td>3.89</td>
<td>5.19</td>
<td>6.26</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>–</td>
<td>–</td>
<td>6.75</td>
<td>7.45</td>
<td>13.36</td>
<td>10.57</td>
<td>5.64</td>
<td>4.90</td>
</tr>
<tr>
<td>Iris</td>
<td>5.25</td>
<td>2.94</td>
<td>4.70</td>
<td>2.95</td>
<td>5.46</td>
<td>5.41</td>
<td>3.60</td>
<td>3.79</td>
</tr>
<tr>
<td>Mammographic</td>
<td>19.38</td>
<td>20.34</td>
<td>18.91</td>
<td>20.16</td>
<td>17.77</td>
<td>17.42</td>
<td>18.00</td>
<td>18.10</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>30.28</td>
<td>–</td>
<td>17.67</td>
<td>12.96</td>
<td>15.81</td>
<td>14.28</td>
<td>13.69</td>
<td>14.68</td>
</tr>
<tr>
<td>Sonar</td>
<td>32.25</td>
<td>–</td>
<td>25.10</td>
<td>22.65</td>
<td>23.67</td>
<td>20.27</td>
<td>20.75</td>
<td>19.85</td>
</tr>
<tr>
<td>Splice</td>
<td>–</td>
<td>–</td>
<td>11.06</td>
<td>7.21</td>
<td>10.81</td>
<td>11.94</td>
<td>8.93</td>
<td>8.16</td>
</tr>
<tr>
<td>Thyroid</td>
<td>3.92</td>
<td>5.65</td>
<td>6.01</td>
<td>4.87</td>
<td>5.02</td>
<td>5.54</td>
<td>4.35</td>
<td>4.00</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>19.17</td>
<td>18.35</td>
<td>19.15</td>
<td>16.81</td>
<td>15.23</td>
<td>16.59</td>
<td>17.22</td>
<td>16.60</td>
</tr>
<tr>
<td>Vowel</td>
<td>38.68</td>
<td>18.24</td>
<td>18.87</td>
<td>17.11</td>
<td>14.70</td>
<td>9.40</td>
<td>10.80</td>
<td>11.24</td>
</tr>
<tr>
<td>Wine</td>
<td>3.76</td>
<td>–</td>
<td>2.67</td>
<td>3.75</td>
<td>4.52</td>
<td>3.94</td>
<td>3.51</td>
<td>2.85</td>
</tr>
</tbody>
</table>

compare CG QDA vs. MM QDA, we define relative speedup and relative improvement of objective function as follows.

\[
\text{Relative Speedup} = \frac{\text{CG CPU time} - \text{Our CPU time}}{\text{CG CPU time}} \times 100 \quad (3.62)
\]

\[
\text{Relative Improvement} = \frac{|\text{CG Obj. value} - \text{Our obj. value}|}{\text{CG Obj. value}} \times 100. \quad (3.63)
\]

For the 25 selected data sets, We showed boxplots of relative speed up and relative improvement for CG vs. our method. It clearly shows that as an optimization algorithm, our method outperforms CG method on most of the selected data sets.
3.4 Conclusion

Quadratic discriminant analysis is a standard tool for classification problems. Classical QDA sacrifices classification accuracy for computation efficiency. Sacrificing classification accuracy is not desired for classification problems. Motivated by the recent studies in hybrid generative/discriminative learning, we argue that, in order to obtain better classification accuracy, parameters of QDA can be estimated by maximizing a convex combination.
CHAPTER 3. QDA REVISITED

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Naive Bayes</th>
<th>QDA</th>
<th>RDA</th>
<th>BDAT</th>
<th>LR</th>
<th>Q SVM</th>
<th>CG QDA</th>
<th>MM QDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwoNorm</td>
<td>0.16</td>
<td>0.14</td>
<td>0.14</td>
<td>0.15</td>
<td>0.18</td>
<td>0.12</td>
<td>0.22</td>
<td>0.14</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
<td>0.09</td>
<td>0.17</td>
<td>0.18</td>
<td>0.17</td>
<td>0.10</td>
</tr>
<tr>
<td>Australian</td>
<td>1.89</td>
<td>2.14</td>
<td>1.86</td>
<td>1.88</td>
<td>1.45</td>
<td>1.59</td>
<td>1.62</td>
<td>1.48</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>5.67</td>
<td>—</td>
<td>1.75</td>
<td>1.82</td>
<td>1.39</td>
<td>1.89</td>
<td>2.10</td>
<td>1.62</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>0.95</td>
<td>0.94</td>
<td>1.01</td>
<td>0.86</td>
<td>0.65</td>
<td>1.17</td>
<td>1.18</td>
<td>0.85</td>
</tr>
<tr>
<td>Breast Tissue</td>
<td>5.55</td>
<td>—</td>
<td>5.95</td>
<td>6.45</td>
<td>6.23</td>
<td>4.36</td>
<td>5.41</td>
<td>6.49</td>
</tr>
<tr>
<td>BUPA</td>
<td>5.26</td>
<td>4.04</td>
<td>3.45</td>
<td>3.72</td>
<td>2.40</td>
<td>3.08</td>
<td>4.32</td>
<td>5.74</td>
</tr>
<tr>
<td>Climate</td>
<td>2.71</td>
<td>—</td>
<td>1.82</td>
<td>1.25</td>
<td>1.54</td>
<td>1.25</td>
<td>1.13</td>
<td>1.36</td>
</tr>
<tr>
<td>Diabetes</td>
<td>1.44</td>
<td>1.88</td>
<td>1.37</td>
<td>1.21</td>
<td>1.50</td>
<td>1.09</td>
<td>1.11</td>
<td>1.06</td>
</tr>
<tr>
<td>German</td>
<td>—</td>
<td>—</td>
<td>1.02</td>
<td>1.33</td>
<td>1.78</td>
<td>1.11</td>
<td>1.23</td>
<td>1.14</td>
</tr>
<tr>
<td>Heart</td>
<td>2.68</td>
<td>3.39</td>
<td>1.93</td>
<td>2.31</td>
<td>2.52</td>
<td>2.85</td>
<td>3.66</td>
<td>2.53</td>
</tr>
<tr>
<td>ILPD</td>
<td>2.58</td>
<td>1.79</td>
<td>2.61</td>
<td>3.61</td>
<td>1.99</td>
<td>1.58</td>
<td>2.69</td>
<td>2.71</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>—</td>
<td>—</td>
<td>0.55</td>
<td>1.01</td>
<td>0.70</td>
<td>0.63</td>
<td>1.14</td>
<td>1.01</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>—</td>
<td>—</td>
<td>1.60</td>
<td>2.07</td>
<td>2.20</td>
<td>2.22</td>
<td>1.89</td>
<td>1.29</td>
</tr>
<tr>
<td>Iris</td>
<td>1.75</td>
<td>1.30</td>
<td>3.55</td>
<td>1.26</td>
<td>1.38</td>
<td>2.52</td>
<td>1.78</td>
<td>1.41</td>
</tr>
<tr>
<td>Magic</td>
<td>0.28</td>
<td>0.23</td>
<td>0.22</td>
<td>0.23</td>
<td>0.19</td>
<td>0.16</td>
<td>0.17</td>
<td>0.26</td>
</tr>
<tr>
<td>Mammographic</td>
<td>2.01</td>
<td>1.97</td>
<td>1.62</td>
<td>2.07</td>
<td>1.47</td>
<td>1.91</td>
<td>1.50</td>
<td>1.45</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>3.04</td>
<td>—</td>
<td>3.29</td>
<td>3.18</td>
<td>3.40</td>
<td>1.86</td>
<td>2.26</td>
<td>4.07</td>
</tr>
<tr>
<td>Sat. Images</td>
<td>0.45</td>
<td>0.57</td>
<td>0.68</td>
<td>0.54</td>
<td>0.38</td>
<td>0.38</td>
<td>0.57</td>
<td>0.47</td>
</tr>
<tr>
<td>Sonar</td>
<td>5.01</td>
<td>—</td>
<td>4.47</td>
<td>4.90</td>
<td>3.64</td>
<td>3.49</td>
<td>3.59</td>
<td>4.01</td>
</tr>
<tr>
<td>Splice</td>
<td>—</td>
<td>—</td>
<td>1.00</td>
<td>1.86</td>
<td>0.73</td>
<td>0.74</td>
<td>2.55</td>
<td>3.22</td>
</tr>
<tr>
<td>Thyroid</td>
<td>1.00</td>
<td>2.27</td>
<td>2.16</td>
<td>1.54</td>
<td>2.30</td>
<td>2.38</td>
<td>0.97</td>
<td>1.27</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>2.80</td>
<td>2.77</td>
<td>3.72</td>
<td>2.04</td>
<td>1.65</td>
<td>2.86</td>
<td>2.42</td>
<td>2.18</td>
</tr>
<tr>
<td>Vowel</td>
<td>2.56</td>
<td>2.37</td>
<td>2.63</td>
<td>2.38</td>
<td>2.03</td>
<td>1.97</td>
<td>2.20</td>
<td>2.17</td>
</tr>
<tr>
<td>Wine</td>
<td>1.88</td>
<td>—</td>
<td>1.37</td>
<td>2.56</td>
<td>1.70</td>
<td>1.60</td>
<td>2.44</td>
<td>1.62</td>
</tr>
</tbody>
</table>

Table 3.3: Standard Deviation of test set error rates for 25 datasets (Shown in percentage)

joint log-likelihood and conditional log-likelihood of given observations and their labels. For this purpose, we presented a MM optimization algorithm to estimate parameters for generative classifiers, of which conditional distributions are from the well known exponential family. Furthermore, we proposed a block-coordinate descent algorithm to sequentially update parameters of QDA in each iteration of the MM algorithm. For each update, we used a trust region method, of which each iteration has a simple closed form solution. Our numerical experiments show that our method is competi-
<table>
<thead>
<tr>
<th></th>
<th>CG QDA</th>
<th>MM QDA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU Time</td>
<td>Obj. Value</td>
</tr>
<tr>
<td>Twonorm</td>
<td>0.30 ± 0.05</td>
<td>13.62 ± 0.02</td>
</tr>
<tr>
<td>Ringsnorm</td>
<td>0.42 ± 0.00</td>
<td>13.31 ± 0.02</td>
</tr>
<tr>
<td>Australian</td>
<td>0.40 ± 0.18</td>
<td>8.85 ± 0.16</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>1.50 ± 0.60</td>
<td>0.83 ± 0.25</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>11.40 ± 1.39</td>
<td>−0.66 ± 0.33</td>
</tr>
<tr>
<td>Breast Tissue</td>
<td>10.89 ± 0.67</td>
<td>−3.86 ± 0.62</td>
</tr>
<tr>
<td>BUPA</td>
<td>0.18 ± 0.07</td>
<td>4.33 ± 0.07</td>
</tr>
<tr>
<td>Climate</td>
<td>2.23 ± 0.98</td>
<td>13.30 ± 0.23</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0.19 ± 0.08</td>
<td>5.70 ± 0.05</td>
</tr>
<tr>
<td>German</td>
<td>0.35 ± 0.32</td>
<td>15.65 ± 0.13</td>
</tr>
<tr>
<td>Heart</td>
<td>0.18 ± 0.01</td>
<td>8.32 ± 0.10</td>
</tr>
<tr>
<td>ILPD</td>
<td>2.27 ± 0.91</td>
<td>5.45 ± 0.34</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>222.23 ± 15.56</td>
<td>−4.57 ± 0.56</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>10.99 ± 2.18</td>
<td>4.88 ± 1.03</td>
</tr>
<tr>
<td>Iris</td>
<td>0.52 ± 0.10</td>
<td>0.88 ± 0.12</td>
</tr>
<tr>
<td>Magic</td>
<td>33.87 ± 4.54</td>
<td>5.28 ± 0.03</td>
</tr>
<tr>
<td>Mammographic</td>
<td>0.24 ± 0.08</td>
<td>3.48 ± 0.13</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>10.46 ± 1.12</td>
<td>−2.47 ± 0.60</td>
</tr>
<tr>
<td>Sat. Images</td>
<td>117.52 ± 10.07</td>
<td>−2.19 ± 0.21</td>
</tr>
<tr>
<td>Sonar</td>
<td>23.24 ± 2.04</td>
<td>−17.29 ± 4.80</td>
</tr>
<tr>
<td>Splice</td>
<td>67.59 ± 31.53</td>
<td>39.38 ± 0.34</td>
</tr>
<tr>
<td>Thyroid</td>
<td>1.13 ± 0.33</td>
<td>1.04 ± 0.22</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>4.01 ± 1.49</td>
<td>1.54 ± 0.13</td>
</tr>
<tr>
<td>Vowel</td>
<td>12.01 ± 3.13</td>
<td>5.05 ± 0.10</td>
</tr>
<tr>
<td>Wine</td>
<td>1.73 ± 0.96</td>
<td>4.67 ± 0.24</td>
</tr>
</tbody>
</table>

Table 3.4: CPU time (seconds) and final objective function values of training sets ($\beta = 0.5$): CG v.s. our method (Smaller is better)

...utive with, in some cases significantly better than other well-known quadratic classification methods in the literature.

Our method can be easily adapted for linear discriminant analysis (LDA).

Moreover in spirit, our MM algorithm and block-coordinate descent algorithm can be applied for any generative classifiers of which conditional distributions are from the exponential family.

Sparse parameters are desired when handling small sample problems, e.g. [4, 48]. Though early stopping provides regularization in our algorithm,
we think explicitly adding sparse constraint via $L_1$ or $L_0$ regularization in the block-coordinate descent algorithm might provide additional robustness for QDA.

Recently, there is a growing interest in semi-supervised learning; that is to use unlabelled data to help training classification methods [20]. Our future work will include unlabelled data and missing data for estimating parameters for LDA and QDA.

Our method has advantage in handling large scale ($N \gg 1$) but small or medium dimensional ($D < 100$) problems, since we observed that the main computation bottleneck of our method is that we have to do full eigen-decompositions in Equation (3.56) to update inverse covariance matrices. We implemented it with LAPACK routine SYEVD. In [81], Laue (2012) proposed a global convergent algorithm for linear SDPs, in which a full eigen-decomposition is replaced by an efficient rank-1 Hazan update [66] and a nonlinear update (which can be done by CG or other methods). We think our method can be further speeded by applying the same technique to update inverse covariance matrices. Nevertheless, the implementation would be more difficult as it involves more complex third party softwares.
Figure 3.6: Boxplots of speedup (defined in (3.63)) and improvement (defined in (3.63)) are shown for 25 selected data sets.
Chapter 4

Discriminative Linear Subspace Dimension Reduction Method: A Review

High dimensional data are more and more common in modern machine learning and pattern classification problems. High dimensionality of data causes two serious challenges in learning problems: computational efficiency and the curse of dimensionality [36, 43]. That is: (1) as the dimensionality of data increases, the computational time and storage cost increase for any learning algorithm; and (2) more seriously, as the dimensionality of data
increases, learning becomes more difficult because data are essentially sparse in any region of interest.

Because of the curse of dimensionality and the concern of computational efficiency, dimension reduction methods are often used to extract more sufficient features from high dimensional raw variables\textsuperscript{1}. Some examples are

- In computer vision community, principal component analysis (PCA) and Fisher’s discriminant analysis (FDA) are applied to construct, respectively, *eigenfaces* and *Fisherfaces* for face recognition problems [8, 134];

- In Brain-Computer Interface, common spatial pattern (CSP) are used to extract lower dimensional features for electroencephalogram (EEG) signal classification problems [37, 103];

- In microarray data analysis, PCA are used to construct *eigengenes* for gene expression classification problems [1, 29].

Linear subspace dimension reduction (LSDR) methods have been widely used for the purpose of dimension reduction, for example principal compo-

\textsuperscript{1}In the literature, dimension reduction is also called as feature extraction, feature construction. In this and the next chapter, we use term “variables” for original raw input variables, and term “features” or “transformed variables” for variables constructed from raw input variables. We do not distinguish “variables” and “features” when there is no impact on the discussion. We sometimes use “vector” instead of ”variables” or ”features” in our discussion.
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

Principal component analysis (PCA) [76] and independent component analysis (ICA) [73]. PCA and ICA are unsupervised linear subspace methods for dimension reduction. PCA tries to find linear subspaces such that the variance of the projected data are maximally preserved. ICA is a way of finding linear subspaces in which the second- and higher-order statistical dependencies of the data are minimized; that is the features are as statistically independent from each other as possible. Note that, as unsupervised methods, neither PCA nor ICA use label information, which is crucial for classification problems. Consequently, PCA and ICA are optimal for pattern description, but not optimal for pattern discrimination.

In this chapter, we focus on LSDR methods for classification. To differentiate with PCA and ICA, we call these methods discriminative LSDR methods. In essence, discriminative LSDR methods apply linear transformations on original raw variables, and construct more compact features by extracting only classification-relevant information of original raw variables.

Formally, consider a classification problem of which data are generated by a fixed probability $p(X, Y)$, where a random vector $X$ takes value in $\mathbb{R}^D$ and a random variable $Y$ takes value in $\{1, 2, \cdots, K\}$. To simplify our task, we focus on binary classification problems in this chapter, i.e. $K = 2$. In
a discriminative LSDR method, one tries to find a linear transformation matrix $\mathbf{W} = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^{D \times d}$, $(d \leq D)$, such that a transformed random vector $\mathbf{Z} = \mathbf{W}^T \mathbf{X} \in \mathbb{R}^d$ can capture most, if not all, classification-relevant information in $Y|\mathbf{X}$.

In this chapter, we give an overview of commonly used discriminant LSDR methods when class conditional distributions are normal. That is, we assume class conditional distribution to be multivariate normal, i.e.

$$\mathbf{X}|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$  \hspace{1cm} (4.1)

For the convenience of theoretical discussion, we choose to present the population versions of these discriminative LSDR methods, instead of their sample versions which are more commonly presented in the machine learning and pattern classification literature. We also emphasize that, though we focus on class conditional distributions to be normal, our discussion can be similarly extended to ellipsoidally symmetric class conditional distributions [63].

This chapter is organized as follows. In Section 4.1, we summarize some widely used criteria for motivating and evaluating discriminative dimension reduction methods in the literature. We then give an overview of discriminative LSDR methods when class-conditional distributions are assumed to
be normal. Specifically, in Section 4.2, we give an introduction of Fisher’s discriminant analysis for LDA, i.e. $\Sigma_k = \Sigma$. In Section 4.3, we give an introduction of discriminative LSDR methods for a special case of QDA in which class conditional distributions share the same mean, i.e. $\mu_k = \mu$. In Section 4.4, we introduces several discriminative LSDR methods for QDA of which class conditional distributions have arbitrary mean $\mu_k$’s and covariance $\Sigma_k$’s.

4.1 Criteria for Dimension Reduction

In principle, a discriminative dimension reduction method constructs a lower dimensional feature vector $Z$ such that $Z$ keeps only classification-relevant information of a raw input vector $X$ about class label $Y$. A critical question for discriminative dimension reduction methods is how to measure the effectiveness of $Z$ in terms of preserving classification-relevant information of raw variable $X$ about class label variable $Y$? In the literature, there are two commonly used approaches summarized as follows:

- The ultimate goal of a dimension reduction method is to build efficient and possibly more effective classifiers with a feature vector $Z$.

Therefore, theoretically speaking, the Bayes error is an ultimate mea-
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

surement of the effectiveness of $Z$: more effective $Z$ means smaller Bayes error. However, the Bayes error is often hard to calculate for practical problems. Therefore upper bounds of the Bayes error are often used to construct and evaluate $Z$. In subsection 4.1.1, we discuss various information-theoretic error exponents that can be used to obtain upper bounds of the Bayes error.

- As we discussed in Chapter 2, it suffices to know $p(Y|X)$ for classification purpose. Therefore from a classification viewpoint, $Z$ capture all information for class label $Y$ of $X$ if $Y|Z$ has the same distribution as $Y|X$, i.e. $Y|Z \sim Y|X$. This is the sufficient dimension reduction approach, which is originally developed for regression problems in the statistics community. We discuss this approach in detail in subsection 4.1.2.

4.1.1 Information-theoretic error exponents for Bayes error

With a dimension reduction method, one can construct lower dimensional features $Z$ from raw input variables $X$, and then use $Z$ to build an efficient, and possibly more effective classifier. Intuitively, more effective features result in smaller classification error. The Bayes error is the lowest achievable error rate for a given statistical classifier. Therefore, the Bayes error can
be used as an optimum measurement of feature effectiveness. However, the Bayes error is often difficult to calculate in practice. Consequently, various upper bounds of the Bayes error have been used for measuring feature effectiveness in the literature. Two commonly used information-theoretical quantities play important roles in constructing upper bounds of the Bayes error: the Chernoff distance and the Kullback-Leibler (KL) distance.

For two continuous distributions $p(X)$ and $q(X)$, the Chernoff distance is defined as

$$D_C(p||q; \beta) = -\ln \int p(x)^\beta q(x)^{1-\beta} dx$$

(4.2)

where $0 \leq \beta \leq 1$. When $\beta = \frac{1}{2}$, we have the so-called Bhattacharyya distance, i.e.

$$D_B(p||q) = -\ln \int \sqrt{p(x)q(x)} dx$$

(4.3)

For the simplicity of discussion, we will use, respectively, $D_C(\beta)$ and $D_B$ for $D_C(p||q; \beta)$ and $D_B(p||q)$ in the following discussion.

Now let us see the connection between the Bayes error and the Chernoff distance. As discussed in Chapter 2, we have the following decision rule for

---

2For simplicity, we assume distributions $p(X)$ and $q(X)$ are continuous. The Chernoff distance, and the K-L distance can be defined similarly when $p(X)$ and $q(X)$ are discrete.
a 0 − 1 loss function:

\[ \hat{y} = \arg \max_k p(Y = k|X = x) \]

The probability of error can be calculated as following

\[
P(error) = \int_{p(Y=1|X=x)>p(Y=2|X=x)} p(X = x|Y = 2)p(Y = 2)dx \\
+ \int_{p(Y=2|X=x)>p(Y=1|X=x)} p(X = x|Y = 1)p(Y = 1)dx \\
= \int \min\{p(X = x|Y = 1)p(Y = 1), p(X = x|Y = 2)p(Y = 2)\}dx
\]

For \( a \geq 0, b \geq 0, \) and \( 1 \geq \beta \geq 0, \) we have the following inequality,

\[
\min(a, b) \leq a^\beta b^{1-\beta}
\]

By applying the above inequality, we can get an upper bound of probability of error as following

\[
P(error) = \int \min\{p(X = x|Y = 1)p(Y = 1), p(X = x|Y = 2)p(Y = 2)\}dx \\
\leq p^\beta(Y = 1)p^{1-\beta}(Y = 2) \int p^\beta(X = x|Y = 1)p^{1-\beta}(X = x|Y = 2)dx \\
= p^\beta(Y = 1)p^{1-\beta}(Y = 2)e^{-D_C(p(X|Y=1)||p(X|Y=2);\beta)}
\]

where \( 1 \geq \beta \geq 0. \)
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

For two multivariate normal class conditional distributions \( p(X|Y = k) = \mathcal{N}(\mu_k, \Sigma_k) \), \((k = 1, 2)\), we can analytically write the Chernoff distance and the Bhattacharyya distance as follows [28, 39, 54]

\[
D_C(\beta) \equiv D_C(p(X|Y = 1)||p(X|Y = 2); \beta) = \frac{\beta(1-\beta)}{2}(\mu_2 - \mu_1)^T[\beta\Sigma_1 + (1-\beta)\Sigma_2]^{-1}(\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{\det[\beta\Sigma_1 + (1-\beta)\Sigma_2]}{\det\beta(\Sigma_1)\det(1-\beta)(\Sigma_2)} \quad (4.6)
\]

and

\[
D_B \equiv D_B(p(X|Y = 1)||p(X|Y = 2)) = \frac{1}{8}(\mu_2 - \mu_1)^T[\frac{\Sigma_1 + \Sigma_2}{2}]^{-1}(\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{\det[\frac{\Sigma_1 + \Sigma_2}{2}]}{\sqrt{\det(\Sigma_1)\det(\Sigma_2)}} \quad (4.7)
\]

The KL distance is another information-theoretical quantity which is commonly used for evaluating feature effectiveness. The KL distance, also known as the KL divergence or relative entropy, is a non-symmetric measure of the difference between two distributions \( p(X) \) and \( q(X) \), which is defined as

\[
D_{KL}(p||q) = \int p(x) \ln \frac{p(x)}{q(x)} dx \quad (4.8)
\]

For two normal class conditional distributions \( p(X|Y = k) = \mathcal{N}(\mu_k, \Sigma_k) \),
(k = 1, 2), we have the KL distance as follows,

\[
L_{12} \equiv D_{KL}(p(X = x| Y = 1)||p(X = x| Y = 2))
= \frac{1}{2} \left\{ \ln \frac{\det(\Sigma_2)}{\det(\Sigma_1)} + \text{tr}\left(\Sigma_2^{-1}\Sigma_1\right) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2) \right\}
\]

(4.9)

A drawback of the KL distance is its non-symmetry. The so-called J-divergence is the arithmetic mean of the two possible KL distances between two distributions. Specifically, for two normal class conditional distributions \(p(X|Y)\), we have the J-divergence as follows

\[
J_a = \frac{1}{2}(L_{12} + L_{21})
= \frac{1}{4} \text{tr}\left\{ \Sigma_2^{-1}\Sigma_1 + \Sigma_1^{-1}\Sigma_2 + (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \right\}
\]

(4.10)

In addition to the J-divergence, other ways to symmetrize the KL distance are the following [75, 121]

Geometric Mean \( J_g = \sqrt{L_{12}L_{21}} \)

Harmonic Mean \( J_h = \frac{2L_{12}L_{21}}{L_{12} + L_{21}} \)

(4.11) (4.12)

The Chernoff distance, the Bhattacharyya distance, and the KL divergence are from a more general class, the Csiszár \( f \)-divergences. Nguyen,
Wainwright and Jordan (2009) show that \( f \)-divergence has a close relationship with surrogate loss functions, which are used in designing large margin classifiers [98]. Moreover, the Chernoff distance and the KL divergence provides the asymptotic bound of probability of error in hypothesis testing [28]. A more detailed discussion about these measurements and their relationship is out of the scope of this dissertation. We refer to [98, 104, 28, 128] for a more thorough and rigorous discussion on this subject.

Note that the larger these specific information-theoretical distances are, the smaller corresponding upper bounds of the Bayes error are. Therefore, for discriminative LSDR methods, one would like to choose a subspace in which these information-theoretic distances are maximized. This is a common way to design LSDR methods in the literature, for example [54, 87, 78, 121].

### 4.1.2 Sufficient Dimension Reduction

The concept *sufficient dimension reduction* has long been proposed for regression problems in the statistics literature, e.g. [24, 25, 23, 84], and recently gained attention in the machine learning literature, e.g. [30, 52, 53, 120, 115].

Formally speaking, consider a \( D \times 1 \) random vector \( X \) and a random
variable $Y$ with a joint distribution $p(X, Y)$; for simplicity let us assume random variables $X$ take value in $\mathbb{R}^D$, and random variable $Y$ is continuous or discrete. The goal of a sufficient dimension reduction method is to find a mapping $R : \mathbb{R}^D \to \mathbb{R}^d$, $d < D$, such that the conditional distribution of $Y|X$ is the same as the conditional distribution of $Y|R(X)$, i.e. $Y|X \sim Y|R(X)$. As pointed out in [23], the following three statements are equivalent in determining if a dimension reduction method is a sufficient:

$$Y|X \sim Y|R(X) \quad (4.13)$$
$$X|(Y, Z) \sim X|R(X) \quad (4.14)$$
$$Y \perp \!\!\!\!\perp X|R(X) \quad (4.15)$$

We can compare sufficient dimension reduction with the well-established concept of sufficient statistics as follows: a statistic $T$ summarizes data samples $\mathcal{D}$, and is sufficient with respect to a given statistical model if $\theta|\mathcal{D} \sim \theta|T$, where $\theta$ are model parameters; on the other hand, a map $R$ summarizes raw input variables $X$ about class label $Y$, and is a sufficient with respect to a given statistical model if $Y|X \sim Y|R(X)$.

As we discussed in Chapter 2, it suffices to know posterior distributions $p(Y|X)$ for classification purposes. Therefore, if a new reduced dimensional
random vector $Z = R(X)$ satisfies $Y|X \sim Y|Z$, then all classification relevant information of $Y|X$ are saved by $Z$. Therefore sufficiency is a desired property in designing and evaluating a dimension reduction method. In principle one would like to construct a random vector $Z$ with minimal dimensionality by a sufficient dimension reduction method.

For discriminative LSDR methods, we require $Z = W^TX$, where $W \in \mathbb{R}^{D \times d}$ is a transformation matrix. The columns of transformation matrix $W$ span the so-called effective dimension reduction subspace.

4.2 Discriminative LSDR: When $\mu_k = \text{arbitrary}$ and $\Sigma_k = \Sigma$

In this section, we consider LDA of which class conditional distributions share the same covariance, i.e. $\Sigma_k = \Sigma$. That is we have class conditional distributions as follows,

$$X|Y = k \sim \mathcal{N}(\mu_k, \Sigma)$$

It is well known that Fisher’s discriminant analysis (FDA) is an optimal dimension reduction technique for LDA [54, 39, 64].

FDA determines a linear subspace in which the distance between the
means of the classes is maximized and the variance of each class is minimized at the same time \[44\]. Formally, for a binary classification problem, an optimal projection vector \( \mathbf{w}^* \) is found by solving the following optimization problem,

\[
\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{\mathbf{w}^T (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \mathbf{w}}{\mathbf{w}^T \Sigma \mathbf{w}}
\]  

It is easy to show that \( \mathbf{w}^* \) is the largest eigenvector of matrix \( \Sigma^{-1}(\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \). As matrix \( \Sigma^{-1}(\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \) is a rank-one matrix, there is only one eigenvector with positive eigenvalue, which can be written as

\[
\mathbf{w}^* = c \Sigma^{-1}(\mu_1 - \mu_2)
\]  

where \( c \) is a normalization constant such that \( \langle \mathbf{w}^*, \mathbf{w}^* \rangle = 1 \).

Define a random variable as follows

\[
\mathbf{Z} = \mathbf{w}^T \mathbf{X}
\]  

Since we have

\[
\mathbf{X|Y} = k \sim \mathcal{N}(\mu_k, \Sigma)
\]  

we have

\[
\mathbf{Z|Y} = k \sim \mathcal{N}(\mathbf{w}^T \mu_k, \mathbf{w}^T \Sigma \mathbf{w})
\]
We can write, respectively, the Chernoff distance, the Bahattacharyya distance, and the J-divergence in the projected subspace spanned by \( w \) as follows

\[
D_C(w) = \frac{\beta(1 - \beta)}{2} \frac{w^T(\mu_2 - \mu_1)^T(\mu_2 - \mu_1)w}{w^T\Sigma w}
\]

\[
D_B(w) = \frac{1}{8} \frac{w^T(\mu_2 - \mu_1)^T(\mu_2 - \mu_1)w}{w^T\Sigma w}
\]

\[
J_a(w) = \frac{1}{4} \frac{w^T(\mu_2 - \mu_1)^T(\mu_2 - \mu_1)w}{w^T\Sigma w} + \text{constant}
\]

Then it is easy to see that, in the subspace spanned by \( w^* \), the Chernoff distance, the Bahattacharyya distance and the J-divergence of class conditional distributions are maximized.

Let \( z = (w^*)^T x \), \( \omega_k = (w^*)^T \mu_k \), and \( \sigma^2 = (w^*)^T \Sigma_k w^* \). We have the following:

\[
(\mu_1 - \mu_2)^T \Sigma^{-1} x = \frac{(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)(\mu_1 - \mu_2) \Sigma^{-1} x}{(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)} = \frac{(\omega_1 - \omega_2)z}{\sigma^2}
\]

and

\[
-\frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2
\]

\[
= -\frac{(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)((\mu_1^T \Sigma^{-1} (\mu_1 - \mu_2) + \mu_2^T \Sigma^{-1} (\mu_1 - \mu_2))}{2(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)}
\]
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

\[
= - \frac{(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 \mu_1^T - \mu_2 \mu_2^T) \Sigma^{-1} (\mu_1 - \mu_2)}{2(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)}
\]

\[
= - \frac{\omega_1^2 - \omega_2^2}{2}\sigma^2
\]

Therefore, for a binary classification problem, the log ratio of posterior probabilities can be written as follows

\[
\log \frac{p(Y = 1 | X = x)}{p(Y = 2 | X = x)} = (\mu_1 - \mu_2)^T \Sigma^{-1} x - \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

\[
= \frac{(\omega_1 - \omega_2)z - \omega_1^2 - \omega_2^2}{2\sigma^2} + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

\[
= \log \frac{p(Y = 1 | Z = z)}{p(Y = 2 | Z = z)}
\]

We also have that, for a binary classification problems \(\sum_{k=1}^2 p(Y = 2 | X = x) = \sum_{k=1}^2 p(Y = 2 | Z = z) = 1\). Therefore \(p(Y | X) = p(Y | Z)\). That is, for the case of class conditional distributions being multivariate normal with the same covariance, FDA is a sufficient dimension reduction method. Therefore carrying out LDA classification in the subspace spanned by \(w^*\) is equivalent to doing LDA in the original space.

FDA assumes that class conditional distributions share the same covariance, which is rarely true for real data. FDA constructs lower dimensional features by focusing on the difference of class conditional means, and ig-
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

nore the possible difference of class conditional covariances. Consequently, an important drawback of FDA is that, for $K$-class classification problems, it can only find $K - 1$ dimensional subspaces. This becomes more serious when binary classification problems are considered, for which FDA can only extract one optimal feature.

Canonical correlation analysis (CCA) is a method for finding linear subspaces to maximize the correlation of the observation vectors and their labels. It has been known for a long time that FDA and CCA indeed give identical subspaces for dimension reduction purpose [6].

Recently there has been some interest in partial least squares (PLS) [107]. Only recently, it has been shown that PLS has a close connection with FDA [5]. PLS finds linear subspaces by iteratively maximizing the covariance of the deflated observation vectors and their labels. In one mode, PLS can be used to extract more than one feature for binary classification. The main concern in PLS is the efficiency issue, since in each iteration one has to subtract the observation matrix by its rank-one estimation found in the previous iteration, and generate deflated observation vectors [107].
4.3 Discriminative LSDR When $\mu_k = \mu$ and $\Sigma_k = \text{arbitrary}$

In this section, we consider a special case of QDA: class conditional distributions share the same mean, i.e. $\mu_k = \mu$. That is, we have class conditional distributions as follows

$$X|Y = k \sim \mathcal{N}(\mu, \Sigma_k) \quad (4.25)$$

When class conditional distributions have the same mean, we can simplify their Bhattacharyya distance defined in Equation (4.7) as follows

$$D_B = \frac{1}{2} \ln \frac{\det[\Sigma_1 + \Sigma_2]}{\sqrt{\det(\Sigma_1) \det(\Sigma_2)}} \quad (4.26)$$

$$= \frac{1}{4} \ln \det[\frac{1}{4}(\Sigma_1^{-1}(\Sigma_1 + \Sigma_2)\Sigma_2^{-1}(\Sigma_1 + \Sigma_2))] \quad (4.27)$$

$$= \frac{1}{4} \ln \det[\frac{1}{4}(\Sigma_2^{-1}\Sigma_1 + \Sigma_1^{-1}\Sigma_2 + 2I)] \quad (4.28)$$

Let $W = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^{D \times d}$ be a linear transformation matrix.

Define a random vector $Z = W^T X$. Then we have

$$Z|Y = k \sim \mathcal{N}(W^T \mu, W^T \Sigma_k W) \quad (4.29)$$

Therefore, from Equation (4.28) we can write the Bhattacharyya distance
between \( p(Z|Y = 1) \) and \( p(Z|Y = 2) \) as follows,

\[
D_B(W) = \frac{1}{4} \ln \det \left\{ \frac{1}{4} \left[ (W^T \Sigma_2 W)^{-1} W^T \Sigma_1 W + (W^T \Sigma_1 W)^{-1} W^T \Sigma_2 W + 2I \right] \right\}
\]

(4.30)

We can find an optimal transformation matrix \( W^* \) such that in the subspace spanned by the columns of \( W^* \) the Bhattacharyya distance is maximized. That is, we can find \( W^* \) by solving the following optimization problem

\[
W^* = \arg \max_W D_B(W)
\]

(4.31)

From the discussion in subsection 4.1.1, we know that maximizing the Bhattacharyya distance provides an approximate but efficient way to minimize the Bayes error. For the convenience of discussion, we call this method \( MBD (\Delta \mu = 0) \).

Denote

\[
F(W) = (W^T \Sigma_2 W)^{-1} W^T \Sigma_1 W + (W^T \Sigma_1 W)^{-1} W^T \Sigma_2 W + 2I
\]

(4.32)

Taking the derivative of Equation (4.30) with respect to \( W \), we have the following

\[
\frac{\partial D_B(W)}{\partial W}
\]
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

\[
-2\{\Sigma_2 W(W^T \Sigma_2 W)^{-1}W^T \Sigma_1 W(W^T \Sigma_2 W)^{-1} - \Sigma_1 W(W^T \Sigma_2 W)^{-1}\} [F(W)]^{-1}
-2\{\Sigma_1 W(W^T \Sigma_1 W)^{-1}W^T \Sigma_2 W(W^T \Sigma_1 W)^{-1} - \Sigma_2 W(W^T \Sigma_1 W)^{-1}\} [F(W)]^{-1}
\]

We have the following lemma for a generalized eigenvalue problem [58]:

**Lemma 4.1.** If \(A \in \mathbb{R}^{k \times k}\) is symmetric, and \(B \in \mathbb{R}^{k \times k}\) is symmetric positive definite, then there exists a nonsingular matrix \(U = (u_1, u_2, \cdots, u_k) \in \mathbb{R}^{k \times k}\) such that \(U^T B U = I\) and \(U^T A U = \Lambda\), where \(\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{m+1})\).

Moreover, \(A u_i = \lambda_i B u_i\), i.e. \(\lambda_i\) and \(u_i\) are the generalized eigenvalue and eigenvector of \((A, B)\). Furthermore, if \(A\) is also positive definite, then \(\lambda_i > 0\).

Let \(\lambda_i\) and \(v_i, \ i = 1, 2, \cdots, D,\) be the eigenvalue and eigenvector of \((\Sigma_1, \Sigma_2)\), respectively. As shown in [54], the eigenvectors of \((\Sigma_1, \Sigma_2)\) make \(\frac{\partial D_B(W)}{\partial W} = 0\), and for the optimization problem (4.31) we must select the \(d\) eigenvectors \(v_i\)'s corresponding to the \(d\) largest \(\ln \frac{1}{1}(\lambda_i + \frac{1}{\lambda_i} + 2)\).

Furthermore, we have the following theorem:

**Theorem 4.1.** We assume that, for a binary classification problem, class conditional distributions are multivariate normal with mean \(\mu\) and covariance \(\Sigma_k\), i.e., \(X|Y = k \sim \mathcal{N}(\mu, \Sigma_k)\), where \(k = 1, 2\). Let \(\lambda_i\) and \(v_i, \ i = 1, 2, \cdots, D\), be the eigenvalue and eigenvector of \((\Sigma_1, \Sigma_2)\), respectively.
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

For simplicity, let us assume that, for \( i > m \), \( \lambda_i = 1 \). Define

\[
W = (v_1, v_2, \cdots, v_m) \quad (4.33)
\]

\[
W^\perp = (v_{m+1}, v_{m+2}, \cdots, v_D) \quad (4.34)
\]

Then \( Z = W^T X \) defines a sufficient dimension reduction for \( X \) about \( Y \).

Proof. Let \( x \) be a realization of random vector \( X \). Define \( z = W^T x \). To prove the theorem, it suffices to show that

\[
\log \frac{p(Y = 1 | X = x)}{p(Y = 2 | X = x)} = \log \frac{p(Y = 1 | Z = z)}{p(Y = 2 | Z = z)}
\]

By Lemma 4.1, we have the following

\[
\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m) \quad (4.35)
\]

\[
(W, W^\perp)^T \Sigma_1(W, W^\perp) = \begin{pmatrix} \Lambda & 0 \\ 0^T & I \end{pmatrix} \quad (4.36)
\]

\[
(W, W^\perp)^T \Sigma_2(W, W^\perp) = I \quad (4.37)
\]

It can be easily verified that we have

\[
\Sigma_1^{-1} = (W, W^\perp) \begin{pmatrix} \Lambda^{-1} & 0 \\ 0^T & I \end{pmatrix} (W, W^\perp)^T \quad (4.38)
\]

\[
\Sigma_2^{-1} = (W, W^\perp) I (W, W^\perp)^T \quad (4.39)
\]
We then have

\[ \Sigma^{-1}_1 - \Sigma^{-1}_2 = (W, W^\perp) \begin{pmatrix} \Lambda^{-1} - I & 0 \\ 0 & 0 \end{pmatrix} (W, W^\perp)^T \]  

(4.40)

\[ = W(\Lambda^{-1} - I) W^T \]  

(4.41)

We note that the log ratio of posterior probabilities in the original space can be written as follows

\[
\log \frac{p(Y = 1 | X = x)}{p(Y = 2 | X = x)} = -\frac{1}{2} x^T (\Sigma^{-1}_1 - \Sigma^{-1}_2) x + x^T (\Sigma^{-1}_1 - \Sigma^{-1}_2) \mu + \frac{1}{2} \log \frac{\det \Sigma^{-1}_1}{\det \Sigma^{-1}_2} + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

\[
= -\frac{1}{2} x^T W(\Lambda^{-1} - I) W^T x + x^T W(\Lambda^{-1} - I) W^T \mu - \frac{1}{2} \log \det \Lambda + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

Since \( Z = W^T X \), we have

\[ Z | Y = k \sim N(W^T \mu, W^T \Sigma_k W) \]

By Equations (4.36) and (4.37), we have

\[ W^T \Sigma_1 W = \Lambda \]

\[ W^T \Sigma_2 W = I \]

Then we have the log ratio of posterior probabilities in the subspace spanned...
by the columns of \( \mathbf{W} \) as follows

\[
\log \frac{p(Y = 1 | Z = \mathbf{z})}{p(Y = 2 | Z = \mathbf{z})} = \log \frac{p(Z = \mathbf{z} | Y = 1) p(Y = 1)}{p(Z = \mathbf{z} | Y = 2) p(Y = 2)}
\]

\[
= -\frac{1}{2} \mathbf{z}^T (\Lambda^{-1} - \mathbf{I}) \mathbf{z} + \mathbf{z}^T (\Lambda^{-1} - \mathbf{I}) \mathbf{W}^T \mathbf{\mu} - \frac{1}{2} \log \det \Lambda + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

Therefore we have

\[
\log \frac{p(Y = 1 | \mathbf{X} = \mathbf{x})}{p(Y = 2 | \mathbf{X} = \mathbf{x})} = \log \frac{p(Y = 1 | Z = \mathbf{z})}{p(Y = 2 | Z = \mathbf{z})}
\]

\[
\square
\]

We note that by Equations (4.36) and (4.37) we have \((\mathbf{W}^\perp)^T \Sigma_k \mathbf{W}^\perp = \mathbf{I}\), where \(k = 1, 2\); that is, in the subspace spanned by the columns of \(\mathbf{W}^\perp\) two class conditional distributions have the same covariance, and therefore such a subspace contains no discrimination information.

When class conditional distributions share the same mean, another commonly used dimension reduction method is common spatial pattern (CSP). CSP has been successfully applied in EEG classification problems. CSP has a close relationship to the MBD \((\Delta \mathbf{\mu} = 0)\) method: they use generalized eigenvectors of \((\Sigma_1, \Sigma_2)\) as bases for a subspace, but order them differently.
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

Formally speaking, the CSP optimization problems are

$$\max \quad \text{tr}(W^T\Sigma_1 W) \tag{4.42}$$

$$\text{s.t.} \quad W^T(\Sigma_1 + \Sigma_2)W = I \tag{4.43}$$

or

$$\max \quad \text{tr}(W^T\Sigma_2 W) \tag{4.44}$$

$$\text{s.t.} \quad W^T(\Sigma_1 + \Sigma_2)W = I \tag{4.45}$$

It can be easily shown that the solution of the optimization problems is the $d$ generalized eigenvectors of $(\Sigma_1, \Sigma_2)$ whose associated generalized eigenvalue are farthest away from $\frac{1}{2}$. In practice, one often selects pairs of generalized eigenvectors: one with associated generalized eigenvalue that are largest than 1, the other with associated generalized eigenvalue that are smallest than 1. Interested readers can find the statistical property of this method in [72].
4.4 Discriminative LSDR When $\mu_k = \text{arbitrary}$ and $\Sigma_k = \text{arbitrary}$

Finally, we consider the more general QDA case: class-conditional means and covariances are arbitrary. That is we have class conditional distributions as follows

$$X | Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Considerable efforts have been made to construct features for the general QDA case, for example, Fukunaga and Koontz (1970) [55], Young, Marco and Odell (1987) [135], Fukunaga (1990) [54], Schott (1993) [112], Flury et. al. (1997) [45], Röhl and Weihs (1999) [106], Cook and Yin (2001) [27], Zhu and Hastie (2004) [141], Loog and Duin (2004) [87], Cook and Ni (2005) [25], Pardoe et. al. (2007) [100], Cook and Forzani (2009) [24], Tao, Li, Wu and Maybank (2009) [121]. Unfortunately, as Hastie and Zhu pointed out, “...for QDA, where so far there has been no universally accepted dimension-reduction technique in the literature” [65]. We summarize some of the important ones in the following discussion.
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

4.4.1 FKT

The Fukunaga-Koontz transformation (FKT) can be seen as an extension of CSP. FKT considers second moment matrices instead of covariance matrices.

Let $M_i$'s are class-conditional second moment matrices, defined as

$$M_i = \Sigma_i + \mu_i\mu_i^T$$  \hspace{1cm} (4.46)

The FKT method considers the following optimization problems,

$$\max \quad \text{tr}(W^T M_1 W)$$ \hspace{1cm} (4.47)

$$\text{s.t.} \quad W^T(M_1 + M_2)W = I$$ \hspace{1cm} (4.48)

or

$$\max \quad \text{tr}(W^T M_2 W)$$ \hspace{1cm} (4.49)

$$\text{s.t.} \quad W^T(M_1 + M_2)W = I$$ \hspace{1cm} (4.50)

The solution of the FKT optimization problems is the $d$ generalized eigenvector of $(M_1, M_2)$ whose associated generalized eigenvalue are farthest away from $\frac{1}{2}$.

It is well known that FKT can miss important classification information.
Example 4.1. In [46], Foley and Sammon constructed the following example. Consider a binary classification problem in which class-conditional distributions are multivariate Gaussian. Specifically, we have

\[
\begin{align*}
\mathbf{\mu}_1 &= \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix}, \\
\mathbf{\Sigma}_1 &= \begin{pmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \\
\pi_1 &= 0.5 \\
\mathbf{\mu}_2 &= \begin{pmatrix} 0 \\ -5 \\ 0 \end{pmatrix}, \\
\mathbf{\Sigma}_2 &= \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \\
\pi_2 &= 0.5
\end{align*}
\] (4.51)

It can be easily shown that we have the following

\[
\begin{align*}
\mathbf{M}_1 &= \mathbf{\Sigma}_1 + \mathbf{\mu}_1\mathbf{\mu}_1^T = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 26 & 0 \\ 0 & 0 & 8 \end{pmatrix} \\
\mathbf{M}_2 &= \mathbf{\Sigma}_2 + \mathbf{\mu}_2\mathbf{\mu}_2^T = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 26 & 0 \\ 0 & 0 & 2 \end{pmatrix} \\
\mathbf{M}_t &= \pi_1\mathbf{M}_1 + \pi_2\mathbf{M}_2 = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 26 & 0 \\ 0 & 0 & 5 \end{pmatrix}
\end{align*}
\]
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

Then we have eigenvectors and their corresponding eigenvalues as follows

\[
v_1 = \begin{pmatrix} 5^{\frac{-2}{7}} \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \lambda_1 = 0.8
\]

\[
v_2 = \begin{pmatrix} 0 \\ 26^{\frac{-1}{7}} \\ 0 \\ 0 \end{pmatrix} \quad \lambda_2 = 0.5
\]

\[
v_3 = \begin{pmatrix} 0 \\ 0 \\ 5^{\frac{-2}{7}} \\ 0 \end{pmatrix} \quad \lambda_3 = 0.8
\]

The FKT transformation then identifies \( v_1 \) and \( v_3 \). However, the most important classification direction \( v_2 \) is missed. Indeed, for class conditional distributions being multivariate normal, projecting data to \( v_2 \) results in Bayes error 0.0003 percent, while projecting data to \( v_1 \) or \( v_3 \) has Bayes error 35 percent.

4.4.2 SAVE

Sliced average variance estimation (SAVE) is a dimension reduction method that was originally proposed for regression problems [26, 25, 23], and later was shown to have a close relationship with QDA [27, 100].

In this subsection, we follow the standard practice in the literature of
SAVE: assume data are whitened; that is

$$X \leftarrow \Sigma_t^{-\frac{1}{2}}(X - \mu_t)$$  \hspace{1cm} (4.53)$$

where $\Sigma_t$ and $\mu_t$ is the data variance and mean.

SAVE defines a kernel matrix as follows

$$K = \sum_{i=1}^{K} \pi_k(I - \Sigma_k)^2$$  \hspace{1cm} (4.54)$$

where $\Sigma_k$ is the covariance matrix of $k$-th class conditional distributions (on whitened data), and $\pi_k = p(Y = k)$. Then the optimization problem for SAVE is as follows,

$$\max \ \text{tr}(W^T K W)$$ 
$$\text{s.t.} \ \ W^T W = I$$  \hspace{1cm} (4.55, 4.56)$$

The solution is the $d$ eigenvectors of $K$ with largest eigenvalues.

It is known that SAVE finds subspaces that contain the first- and second-order difference in class-conditional distributions \[27\]; that is, for a binary classification problem, we have

$$\text{span}(K) = \text{span} (\mu_1, \mu_2, \Sigma_2 - \Sigma_1)$$  \hspace{1cm} (4.57)$$
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LNSDR METHODS

Let \( \lambda_i \) and \( v_i \), \( i = 1, 2, \cdots, D \), be the eigenvalues and eigenvectors of \( K \), respectively. Assume that for \( i > m \), \( \lambda_i = 0 \); for \( i \leq m \), \( \lambda_i > 0 \). Define

\[
W = (v_1, v_2, \cdots, v_m)
\]

In [100], Pardoe, Yin and Cook show that

\[
\log \frac{p(X = x|Y = 1)}{p(X = x|Y = 2)} = \log \frac{p(Z = z|Y = 1)}{p(Z = z|Y = 2)}
\]

where \( Z = W^T X \) and \( z = W^T x \). That is classification can be equivalently done in the subspace spanned by the columns of \( W \) as QDA in the original space.

However, it is known that SAVE can mis-rank projection directions [142].

Example 4.2. In [142], Zhu constructed a binary classification problem in which class-conditional distributions are multivariate Gaussian. Specifically, we have

\[
\mu_1 = \begin{pmatrix} -\sqrt{0.5} \\ 0 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.3 \end{pmatrix}
\]

\[
\mu_2 = \begin{pmatrix} \sqrt{0.5} \\ 0 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 1.7 \end{pmatrix}
\]

In this case, SAVE calculates the linear transformation matrix as \((v_1, v_2) = \cdots\).
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

<table>
<thead>
<tr>
<th>Decision Rule</th>
<th>Bayes Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 4.62</td>
<td>15.9%</td>
</tr>
<tr>
<td>Rule 4.63</td>
<td>30.2%</td>
</tr>
</tbody>
</table>

Table 4.1: Bayes Error for decision rules on two SAVE directions

\[
egin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix},
\]

and rank then as \( \mathbf{v}_2 \) and \( \mathbf{v}_1 \) (with score 0.49 and 0.25). Now consider building classifiers with the two variables separately. We have

\[
\hat{y} = \begin{cases} 1 & \text{if } x_1 \leq 0 \\ 2 & \text{if } x_1 > 0 \end{cases}
\]

(4.62)

and

\[
\hat{y} = \begin{cases} 1 & \text{if } |x_2| \leq 0.795 \\ 2 & \text{if } |x_2| > 0.795 \end{cases}
\]

(4.63)

We then can calculate the Bayes error for the two decision rules, as shown in table 4.1. So in this example, SAVE ranks the projection directions wrongly.

4.4.3 The Chernoff Criterion

As we discussed in Subsection 4.1.1, the Chernoff distance provides an upper bound of the Bayes error. Define

\[
\Sigma_t = \beta \Sigma_1 + (1 - \beta) \Sigma_2
\]

(4.64)
where $0 \geq \beta \geq 1$. Let $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d) \in \mathbb{R}^{D \times d}$ be a transformation matrix. Following Equation (4.6), we can write the Chernoff distance between two normal distributions in the subspace spanned by $\mathbf{W}$ as

$$D_c(\mathbf{W}; \beta) = \frac{\beta(1-\beta)}{2} (\mu_2 - \mu_1)^T \mathbf{W} (\mathbf{W}^T \Sigma_t \mathbf{W})^{-1} \mathbf{W}^T (\mu_2 - \mu_1) + \frac{1}{2} \log \frac{\det \mathbf{W}^T \Sigma_t \mathbf{W}}{\det^{\beta} (\mathbf{W}^T \Sigma_1 \mathbf{W}) \det^{1-\beta} (\mathbf{W}^T \Sigma_2 \mathbf{W})} \quad (4.65)$$

An optimal transformation matrix can be found as following,

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} D_c(\mathbf{W}; \beta) \quad (4.66)$$

Optimization problem (4.66) has no closed-form solutions; Consequently a time-consuming iterative algorithm has to be used to solve $\max_{\mathbf{W}} D_c(\mathbf{W}; \beta)$ for $\mathbf{W}^*$ [109].

Define matrix $\mathbf{S}$ as following

$$\mathbf{S} = \frac{\beta(1-\beta)}{2} \Sigma_t^{-\frac{1}{2}} (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \Sigma_t^{-\frac{1}{2}}$$

$$-\frac{1}{2} \left[ \beta \log \Sigma_t^{-\frac{1}{2}} \Sigma_1 \Sigma_t^{-\frac{1}{2}} + (1-\beta) \log \Sigma_t^{-\frac{1}{2}} \Sigma_2 \Sigma_t^{-\frac{1}{2}} \right] \quad (4.67)$$

Note that we can write the Chernoff distance of two normal distributions as
follows

\[ D_c(\beta) = \frac{\beta(1-\beta)}{2} (\mu_2 - \mu_1)^T \Sigma_t^{-1}(\mu_2 - \mu_1) \]

\[ + \frac{1}{2} \log \frac{\det \Sigma_t}{\det^\beta(\Sigma_1) \det^{1-\beta}(\Sigma_2)} \]

\[ = \text{tr} \mathbf{S} \]

Specifically, when \( \Sigma_1 = \Sigma_2 = \Sigma \), we have

\[ S_E = \frac{\beta(1-\beta)}{2} \Sigma^{-\frac{1}{2}}(\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \Sigma^{-\frac{1}{2}} \]

\[ D_c(\beta) = \text{tr} S_E \]

In [87], Loog and Duin noted that, when \( \Sigma_1 = \Sigma_2 = \Sigma \), the objective function of FDA can be reformulate as following

\[ \max \text{tr}\{(\mathbf{W}^T \Sigma_t \mathbf{W})^{-1}(\mathbf{W}^T \Sigma_t^{\frac{1}{2}} \mathbf{S}_E \Sigma_t^{\frac{1}{2}} \mathbf{W})\} \quad (4.68) \]

Loog and Duin then proposed the heteroscedastic FDA method (HFDA) by generalizing optimization problem (4.68) for heteroscedastic data as follows [87]:

\[ \max \text{tr}\{(\mathbf{W}^T \Sigma_t \mathbf{W})^{-1}(\mathbf{W}^T \Sigma_t^{\frac{1}{2}} \mathbf{S} \Sigma_t^{\frac{1}{2}} \mathbf{W})\} \quad (4.69) \]
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

This is a generalized eigenvalue problem. The solution of optimization problem (4.69) is the $d$ generalized eigenvector of $(\Sigma_{1}^{\frac{1}{2}}S\Sigma_{2}^{\frac{1}{2}}, \Sigma_{1})$ with largest eigenvalues. One drawback of the HFDA method is that its objective function in Equation (4.69) does not have a clear relationship to the Chernoff distance defined in Equation (4.65), which is theoretically undesirable.

4.4.4 The KL Criterion

Let $W = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^{D \times d}$ be a linear transformation matrix. Then the KL divergence of two normal class-conditional distributions in the subspace spanned by the columns of $W$ can be written as

$$L_{12}(W) = \frac{1}{2} \left\{ \ln \frac{\det(W^T \Sigma_2 W)}{\det(W^T \Sigma_1 W)} + \text{tr}((W^T \Sigma_2 W)^{-1}(W^T \Sigma_1 W)) \right\}$$

$$+ \frac{1}{2}(\mu_1 - \mu_2)^T W(W^T \Sigma_2 W)^{-1} W^T (\mu_1 - \mu_2)$$

(4.70)

Decell and Mayekar [34] and De La Torre and Kanade [33] maximize the arithmetic mean of the KL distance to find an optimal transformation matrix $W^*$; that is for binary classification problems, the optimization problem is defined as following:

$$W^* = \arg \max_W L_{12}(W) + L_{21}(W)$$

(4.72)
Tao, Wu, and Maybank (2009) consider the geometric mean of the KL distances, and the normalized distances, defined respectively as follows

\[ J_g(W) = \sqrt{L_{12}(W)L_{21}(W)} \] (4.73)

\[ J_{gn}(W) = \sqrt{\frac{L_{12}(W)L_{21}(W)}{L_{12}(W) + L_{21}(W)}} \] (4.74)

Specifically, for binary classification problems, Tao, Wu and Maybank consider solving the convex combination of the log of the geometric mean of the KL distances and the log of the geometric mean of the normalized KL distances as follows [121]

\[ \mathbf{W}^* = \arg\max_{\mathbf{W}} \{\alpha \log J_{gn}(\mathbf{W}) + (1 - \alpha) \log J_g(\mathbf{W})\} \] (4.75)

where \(0 \leq \alpha \leq 1\).

Bian and Tao (2008) maximize the harmonic mean of the KL distance to find \(\mathbf{W}^*\); formally for a binary classification problem, the optimization problem can be written as following [9]:

\[ \mathbf{W}^* = \arg\max_{\mathbf{W}} \frac{2L_{12}(\mathbf{W})L_{21}(\mathbf{W})}{L_{12}(\mathbf{W}) + L_{21}(\mathbf{W})} \] (4.76)

Unfortunately, optimization problems (4.72), (4.75), and (4.76) do not have closed-form solutions, and therefore have to be solved by more compu-
CHAPTER 4. A REVIEW OF DISCRIMINATIVE LSDR METHODS

tationally expensive optimization methods.
Chapter 5

Affine Feature Extraction for Quadratic Discriminant Analysis

In this chapter, we still focus on dimension reduction methods when class conditional distributions are assumed to be multivariate normal; i.e.

\[ X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k) \] (5.1)

In the previous chapter, we have seen that, for special cases of \( \Sigma_k = \Sigma \) or \( \mu_k = \mu \), there are discriminative linear subspace dimension reduction methods that are computationally efficient and theoretically elegant. However, “...for QDA, where so far there has been no universally accepted
CHAPTER 5. AFFINE FEATURE EXTRACTION 100
dimension-reduction technique in the literature” [65], though considerable
efforts have been made to design dimension reduction methods for this case:
[55, 135, 54, 112, 45, 27, 141, 87, 121].

Linear subspaces are affine spaces that contain the origin. In this chap-
ter, we discuss a novel affine feature extraction (AFE) method to find affine
subspaces ¹ for QDA. Our method can be seen as a generalization of the
Funkunaga-Koontz transformation (FKT). We show that our method has a
closed-form solution, which can be efficiently found by solving a generalized
eigenvalue problem. We also show that our method has desirable geometric,
information-theoretical and statistical properties.

This chapter is organized as follows. In Section 5.1, we present an AFE
method for QDA. Specifically, we discuss the motivation of the study in Sub-
section 5.1.1, the AFE problem formulation in Subsection 5.1.2, the closed-
form solutions in Subsections 5.1.3 and computational issues in Subsection
5.1.4. In Section 5.2, we investigate the properties of the AFE methods and
its relationship with other linear subspace dimension reduction methods.
We present experimental results in Section 5.3, and conclude this chapter
with the summary of our work and possible future directions in Section 5.4.

¹Affine subspaces are also called linear manifolds in the literature.
5.1 Affine Feature Extraction

Consider a binary classification problem for which the data are generated by a fixed probability \( p(X, Y) \), where a random vector \( X \) takes value in \( \mathbb{R}^D \) and a random variable \( Y \) takes value in \( \{1, 2\} \). In this section, we consider the dimension reduction problem for the general QDA case; that is class conditional distributions are multivariate normal with arbitrary mean \( \mu_k \) and covariance \( \Sigma_k \).

5.1.1 Background

In this subsection, we take a closer look at the FKT method. This subsection also serves as our motivation of this study. For simplicity, we consider how to find a 1-d transformation matrix via the FKT method. As we discussed the FKT method in Subsection 4.4.1, we can write the optimization problems of the FKT method as follows:

\[
\begin{align*}
\max & \quad w^T M_1 w \\
\text{s.t.} & \quad w^T (M_1 + M_2) w = 1
\end{align*}
\]
or

\[
\begin{align*}
\text{max} & \quad w^T M_2 w \\
\text{s.t.} & \quad w^T (M_1 + M_2) w = 1
\end{align*}
\] (5.4)

where \( M_k, \ (k = 1, 2), \) is the second moment matrix of class \( k, \) defined as

\[
M_k = E_{X|Y=k}(XX^T)
\] (5.6)

We note that by Equation (5.6), we have

\[
w^T M_k w = w^T E_{X|Y=k}(XX^T)w = E_{X|Y=k}(w^T XX^T w) = E_{Z|Y=k}(Z^2)
\]

where feature \( Z = w^T X. \) That is \( w^T M_k w \) is the mean of feature square \( Z^2 \) of class \( k. \) Thus we can equivalently define FKT as follows

\[
\begin{align*}
\text{max} & \quad E_{Z|Y=1}(Z^2) \\
\text{s.t.} & \quad E_{Z|Y=1}(Z^2) + E_{Z|Y=2}(Z^2) = 1
\end{align*}
\] (5.7)
or

\[
\begin{align*}
\max & \quad \mathbb{E}_{Z|Y=2}(Z^2) \\
\text{s.t.} & \quad \mathbb{E}_{Z|Y=1}(Z^2) + \mathbb{E}_{Z|Y=2}(Z^2) = 1
\end{align*}
\]  (5.9)

\[
\begin{align*}
\max & \quad \mathbb{E}_{Z|Y=2}(Z^2) \\
\text{s.t.} & \quad \mathbb{E}_{Z|Y=1}(Z^2) + \mathbb{E}_{Z|Y=2}(Z^2) = 1
\end{align*}
\]  (5.10)

where \( Z = W^TX \). Therefore, FKT can be interpreted as finding a linear subspace in which one can maximize the distance of the means of squared feature \( Z^2 \). However, as discussed in Example (4.1), FKT may ignore important discriminant information for some cases.

We note that we can equivalently write optimization problems (5.7) and (5.8) as the following

\[
\max \quad \frac{\mathbb{E}_{Z|Y=1}(Z^2)}{\mathbb{E}_{Z|Y=2}(Z^2)}
\]  (5.11)

Similarly we can write optimization problems (5.9) and (5.10) as following

\[
\max \quad \frac{\mathbb{E}_{Z|Y=2}(Z^2)}{\mathbb{E}_{Z|Y=1}(Z^2)}
\]  (5.12)

### 5.1.2 Problem Formulation

We consider an affine transformed random variable

\[
Z = v_1^TX + v_0
\]  (5.13)
where \( \mathbf{v}_1 \in \mathbb{R}^D \) is a vector and \( v_0 \) is a scalar. Linear transformations are a special form of affine transformations, where \( v_0 = 0 \). Now denoting \( \mathbf{w}^T = (\mathbf{v}_1^T, v_0) \), we have

\[
\mathbf{Z} = \mathbf{w}^T \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}
\]  

(5.14)

Note that we have abused the notation of \( \mathbf{w} \). From now on, we shall use \( \mathbf{w} \) for affine transformations unless specified otherwise.

Define an augmented second moment matrix for class \( k \), \( (k = 1, 2) \), as follows

\[
\Xi_k = \mathbb{E}_{X|Y=k} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}^T \\ 1 \end{bmatrix}
\]  

(5.15)

Define a total augmented second moment matrix as follows,

\[
\Xi_t = \mathbb{E}_X \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}^T \\ 1 \end{bmatrix} = \pi_1 \Xi_1 + \pi_2 \Xi_2
\]  

(5.17)

where \( \pi_k = p(Y = k) \) is the prior probability for class \( k \). The relation of augmented second moment matrix, covariance matrix and mean can be found in Appendix B.1.
Then we note that the mean of squared feature $Z^2$ is

$$E_{Z|Y=k}(Z^2) = E_{X|Y=k} \left\{ w^T \begin{pmatrix} X \\ 1 \end{pmatrix} \begin{pmatrix} X^T & 1 \end{pmatrix} \right\} w$$ \hspace{1cm} (5.18)$$

$$= w^T \Xi_k w$$ \hspace{1cm} (5.19)$$

Therefore, motivated by FKT, we maximize the following objective function to find the optimal one dimensional affine subspace

$$C(w; \xi) = \xi \frac{E_{Z|Y=1}(Z^2)}{E_{Z|Y=2}(Z^2)} + (1 - \xi) \frac{E_{Z|Y=2}(Z^2)}{E_{Z|Y=1}(Z^2)}$$ \hspace{1cm} (5.20)$$

$$= \xi \frac{w^T \Xi_1 w}{w^T \Xi_2 w} + (1 - \xi) \frac{w^T \Xi_2 w}{w^T \Xi_1 w}$$ \hspace{1cm} (5.21)$$

where $0 \leq \xi \leq 1$. We use the sum of ratios to measure the importance of $w$ instead of two separated optimization problems in FKT. The parameter $\xi$ can be used to balance the importance of different classes and thus is useful for asymmetric learning problems.

Now let us consider how to find higher dimensional affine subspaces. Let $W = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^{(D+1) \times d}$ be a low-rank affine transformation matrix. In principle, we can iteratively find $w_i$'s by maximizing an optimization function similar as (5.21), provided $w_i$'s are conjugate to each other. That is, we find an optimal affine transformation matrix $W^*$ by solving the
following optimization problem:

$$\max \xi \sum_{i=1}^{d} \frac{w_i^T \Xi_{1} w_i}{w_i^T \Xi_{2} w_i} + (1 - \xi) \sum_{i=1}^{d} \frac{w_i^T \Xi_{2} w_i}{w_i^T \Xi_{1} w_i} \quad (5.22)$$

$$\text{s.t.} \quad w_i^T \Xi_{i} w_j = \delta_{ij}, i, j = 1, 2, \cdots, d \quad (5.23)$$

where $\delta_{ij}$ is the Kronecker delta, defined as

$$\delta_{ij} = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{otherwise} 
\end{cases} \quad (5.24)$$

Constraint (5.23) guarantees $w_i$’s are conjugate (with respect to $\Xi_i$) to each other. Let $\Pi_i = W^T \Xi_i W$. It is easy to recognize that $\Pi_i$’s are indeed the second moment matrices of class $k$ in the lower dimensional space. It can be shown that we can write optimization problems (5.22) and (5.23) equivalently as follows:

$$\max \xi \text{tr}(\Pi_1^{-1} \Pi_2) + (1 - \xi) \text{tr}(\Pi_2^{-1} \Pi_1) \quad (5.25)$$

$$\text{s.t.} \quad W^T \Xi_i W = I \quad (5.26)$$

Generally speaking, we want to generate compact representations of the original observations. Therefore it is desirable to encourage finding lower dimensional affine subspaces. Motivated by the Akaike information crite-
CHAPTER 5. AFFINE FEATURE EXTRACTION

rion and Bayesian information criterion, we propose the following objective function that is to be maximized:

\[
C(W; \xi, d) = (1 - \xi) \text{tr}(\Pi_2^{-1}\Pi_1) + \xi \text{tr}(\Pi_1^{-1}\Pi_2) - d
\]

(5.27)

where \(0 \leq \xi \leq 1\), \(d (1 \leq d \leq m)\) is the number of features we want to generate. We see that high dimensional solutions are penalized by the term \(-d\). Hyperparameter \(\xi\) may be tuned via standard cross-validation methods [65]. In principal, the optimum \(d\) can also be determined by cross-validation procedures. However such a procedure is often computationally expensive. One alternative is: define \(C_0(\xi) = C(I; \xi, m)\); we select the smallest \(d\) such that \(C\) is large enough, i.e. \(d^* = \inf\{d|C(W; \xi, d) \geq \gamma C_0\}\), where \(\gamma\) is a constant.

Note if \(W^T \Xi W = I\), we have \(E[Z_i^T Z_j] = \delta_{ij}\), where \(Z_i = \begin{pmatrix} X \\ 1 \end{pmatrix}\), and \(\delta_{ij}\) is the Kronecker delta. That is the constraint \(W^T \Xi W = I\) guarantees that transformed random variables \(Z_i\)'s are orthogonal.

The constraint \(W^T \Xi W = I\) is necessary in our generalization from the one dimensional to the high dimensional formulation, but it does not generate an mutually orthogonal basis. Obtaining an orthogonal discriminant vector basis is geometrically desirable. Therefore we introduce another
orthogonality constraint $W^TW = I$.

To summarize, we are interested in two different kinds of constraints as follows:

1. $\Xi_t$-orthogonal: $W^T\Xi_t W = I$;

2. Orthogonal: $W^TW = I$.

When data are whitened, i.e. $\Xi_t = I$, these two constraints are equivalent.

We summarize two kinds of AFE problems in Table 5.1

<table>
<thead>
<tr>
<th>AFE1</th>
<th>AFE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\max C(W;\xi,d)$</td>
<td>$\max C(W;\xi,d)$</td>
</tr>
<tr>
<td>$W^T\Xi_t W = I$</td>
<td>$W^TW = I$</td>
</tr>
</tbody>
</table>

Table 5.1: Two kinds of AFE problems

5.1.3 Basic Algorithms

In this subsection, we show how to solve the proposed optimization problems.

Define a function $f$ as:

$$f(x;\xi) = \xi x + (1 - \xi) \frac{1}{x}$$

Let $0 < a \leq x \leq b$. Note that $f$ is a convex function, and thus achieves its maximum at the boundary of $x$, i.e. either $a$ or $b$.

Define $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{D+1})$, and $\lambda_i$’s are the eigenvalues of $(\Xi_1, \Xi_2)$.
(i = 1, 2, · · · , D + 1), i.e. \( \Xi_1 u_i = \lambda_i \Xi_2 u_i \). Let \( \lambda_i(\xi) \)'s be the ordered eigenvalues of \( (\Xi_1, \Xi_2) \) with respect to \( f(\lambda; \xi) \). That is: \( f_1(\xi) \geq f_2(\xi) \geq \cdots \geq f_{D+1}(\xi) \), where \( f_i(\xi) = f(\lambda_i(\xi); \xi) \). The following lemma for nonsingular symmetric \( \Xi_1 \) and \( \Xi_2 \) can be found in [58]:

**Lemma 5.1.** If \( A \in \mathbb{R}^{k \times k} \) is symmetric, and \( B \in \mathbb{R}^{k \times k} \) is symmetric positive definite, then there exists a nonsingular matrix \( U = (u_1, u_2, \cdots, u_k) \in \mathbb{R}^{k \times k} \) such that \( U^T B U = I \) and \( U^T A U = \Lambda \), where \( \Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{m+1}) \).

Moreover, \( Au_i = \lambda_i Bu_i \), i.e. \( \lambda_i \) and \( u_i \) are the generalized eigenvalue and eigenvector of \((A, B)\). Furthermore, if \( A \) is also positive definite, then \( \lambda_i > 0 \).

In Appendix B.2, we show that:

\[
C(W; \xi, d) \leq \sum_{i=1}^{d} f_i(\xi) - d \tag{5.29}
\]

**Proposition 5.1.** Let \( U_\xi = (u_1^\xi, u_2^\xi, \cdots, u_d^\xi) \), where \( u_1^\xi \) is the eigenvector of \((\Xi_1, \Xi_2)\) and has eigenvalue \( \lambda_1(\xi) \). Let \( R \) be a nonsingular matrix. Then \( W = U_\xi R \) maximize \( C(W; \xi, d) \).

**Proof.** It is enough to show \( U_\xi \) maximizes \( C(W; \xi, d) \). Note \( U_\xi^T \Xi_2 U_\xi = I \) and \( U_\xi^T \Xi_1 U_\xi = \text{diag}(\lambda_1(\xi), \lambda_2(\xi), \cdots, \lambda_d(\xi)) \). Then it is easy to affirm the proposition. \( \square \)
Remark 5.1. Let \( U_\xi = (u_1^{\xi}, u_2^{\xi}, \cdots, u_d^{\xi}) \) maximize \( C(W; \xi, d) \); Let \( u_{d+1}^{\xi} \) be an eigenvector of \((\Xi_1, \Xi_2)\) whose eigenvalue is 1. Then it is straightforward to show that \( C(U_\xi; \xi, d) = C((U_\xi, u_{d+1}^{\xi}); \xi, d + 1) \). We prefer \( U_\xi \) to \((U_\xi, u_{d+1}^{\xi})\), because of the lower dimensionality. In other words, we can safely ignore the eigenvectors of \((\Xi_1, \Xi_2)\) whose eigenvalues are 1.

Remark 5.2. If \( W_1 \) maximizes \( C(W; \xi, d) \), then \( W_1R \) also maximizes \( C(W; \xi, d) \), where \( R \in \mathbb{R}^{d \times d} \) is a nonsingular matrix. The proof is straightforward and therefore is omitted.

Let \( U_\xi = QR \), where \( Q \) and \( R \) are the thin QR factorization of \( U_\xi \); then \( W_1 = U_\xi R^{-1} \) maximizes \( C(W; \xi, d) \) and satisfies the orthogonal constraint. Let \( W_2 = U_\xi \Gamma^{-\frac{1}{2}} \), where

\[
\Gamma = \{ \text{diag}(\pi_1\lambda_1(\xi) + \pi_2, \pi_1\lambda_2(\xi) + \pi_2, \cdots, \pi_1\lambda_d(\xi) + \pi_2) \} \quad (5.30)
\]

It can be easily shown that \( W_2 \) maximizes \( C(W; \xi, d) \) and satisfies the \( \Xi_t \)-orthogonal constraint. In practice, we only need to check the largest \( d \) and the smallest \( d \) eigenvalues and eigenvectors of \((\Xi_1, \Xi_2)\) in order to generate \( d \) features. The pseudo-code of an algorithm for AFE is given in Algorithm 5.

Remark 5.3. It can be easily verified that our method is invariant to invertible linear transformation of \( X \). That is let \( W \) be the AFE transformation
Algorithm 5: An Algorithm for affine feature extraction

**Input:** Estimation of augmented second matrices and prior probability: \( \hat{\Xi}_k \) and \( \hat{\pi}_k \), \( k = 1, 2 \)

1. Compute the largest \( d \) and the smallest \( d \) eigenvalues and eigenvectors of \( (\hat{\Xi}_1, \hat{\Xi}_2) \);
2. Sort \( 2d \) eigenvalues and eigenvectors with respect to Eq. (5.28);
3. Selected the largest \( d \) eigenvectors to form \( U_\xi \);
4. **if** Orthogonal Constraint **then**
   5. Apply the thin QR factorization on \( U_\xi \), i.e. \( U_\xi = QR \);
   6. Let \( W = Q \);
5. **else if** \( \hat{\Xi}_t \)-Orthogonal Constraint **then**
   8. Calculate \( \Gamma \) as Eq. (5.30);
   9. Let \( W = U_\xi \Gamma^{-\frac{1}{2}} \);
7. **end if**
11. **return** \( W \)

Matrix for \( X \), and \( \hat{W} \) be the AFE transformation matrix for \( A^T X \), where \( A \) is a nonsingular matrix. Then we have

\[
W^T \begin{pmatrix} X \\ 1 \end{pmatrix} = \hat{W}^T \begin{pmatrix} A & 0 \\ 0^T & 1 \end{pmatrix}^T \begin{pmatrix} X \\ 1 \end{pmatrix} \quad (5.31)
\]

**Remark 5.4.** For \( \hat{\Xi}_t \)-orthogonal AFE, we indeed have \( E[Z_i^T Z_j | Y = k] = \delta_{ij} \), where \( Z_i = w_i^T \begin{pmatrix} X \\ 1 \end{pmatrix} \), and \( \delta_{ij} \) is the Kronecker delta. That is transformed variables \( Z_i \)'s are conditional orthogonal given class label \( Y \). This guarantees that extracted features provide complementary classification-relevant information about each class.
CHAPTER 5. AFFINE FEATURE EXTRACTION

5.1.4 Computational Issues

To apply AFE for real problems, we need to estimate $\Xi_k$ and $\pi_k$, where $k = 1, 2$. In Chapters 2 and 3, we discussed parameter estimation problems. Nevertheless, for computational efficiency consideration, sample estimations of mean, covariance and prior probabilities are often used in the dimension reduction literature. In this chapter, we follow the majority in the literature: we use sample augmented second moment matrix and empirical prior probability in the AFE method. Let \( \{(x_j, y_j) \in \mathbb{R}^m \times \{1, 2\} | n = 1, 2, \ldots, N\} \) be a training set, where $x_j$ and $y_j$ are the observation vector and the corresponding class label. For simplicity, we assume the training set is permuted such that observations 1 to $N_1$ have label 1, and observations $N_1 + 1$ to $N_1 + N_2$ have label 2. Define a data matrix as

\[
X = (x_1, x_2, \ldots, x_N) = (X_1, X_2)
\]

where $X_1 = (x_1, x_2, \ldots, x_{N_1})$, and $X_2 = (x_{N_1+1}, x_{N_1+2}, \ldots, x_N)$. The empirical prior probabilities are

\[
\hat{\pi}_k = \frac{N_k}{N}
\]
where \( k = 1, 2 \). Define an augmented data matrix \( \tilde{X}^T = (X^T, 1) \). Define an augmented data matrix \( \tilde{X}_k \) for class \( k \) as \( \tilde{X}_k^T = (X_k^T, 1) \). Then a sample augmented second moment matrix can be calculated as follows

\[
\hat{\Xi}_k = \frac{1}{N_k} \tilde{X}_k \tilde{X}_k^T
\]

(5.34)

where \( k = 1, 2 \), and

\[
\hat{\Xi}_t = \frac{1}{N} \tilde{X} \tilde{X}^T
\]

(5.35)

In our derivation, we assume the positive definiteness of \( \Xi_k \)'s. However the estimations of \( \Xi_k \)'s may not satisfy this assumption in real applications. The deficiency can be fixed by adding a small regularization matrix to \( \hat{\Xi}_k \)'s; that is

\[
\hat{\Xi}_k \leftarrow \hat{\Xi}_k + \alpha I
\]

(5.36)

where \( k = 1, 2 \) and \( \alpha \) is a small positive constant.

Many real machine learning problems are undersampled; that is the dimensionality of observations is larger than the sample size, i.e. \( D > N \). For undersampled problems, especially when \( D >> 1 \), solving the generalized eigenvalue problem \((\hat{\Xi}_1, \hat{\Xi}_2)\) is not only computationally expensive, but also memory intensive. In the remainder of this subsection, we show an efficient
algorithm to overcome the handicap.

With a given estimation of $\Xi_t$, $\hat{\Xi}_t$, we can find its Choleskey decomposition via incomplete Cholesky decomposition\(^2\) such that

$$\hat{\Xi}_t = YY^T$$

(5.37)

where $Y \in \mathbb{R}^{(D+1) \times m}$, and $m \leq D + 1$. When $\hat{\Xi}_t$ is a sample estimation as defined in Equation (5.35), we can conveniently set

$$Y = \frac{1}{\sqrt{N}} \tilde{X}$$

(5.38)

Let $Y = U \text{diag}(\Delta, 0) V^T$ be the SVD of matrix $Y$, where $\Delta \in \mathbb{R}^{s \times s}$ is a diagonal matrix that contains non-zero singular values of $Y$, and $U$ and $V$ are orthonormal, i.e. $UU^T = VV^T = I$. Then we have

$$\hat{\Xi}_t = U \text{diag}(\Delta^2, 0) U^T$$

(5.39)

$$U^T \hat{\Xi}_t U = \text{diag}(\Delta^2, 0)$$

(5.40)

Let $U = (u_1, u_2, \cdots, u_{D+1}) = (U_1, U_2)$, such that $U_1 \in \mathbb{R}^{(D+1) \times s}$ contains singular vectors with nonzero singular values, and $U_2$ be the remaining part

\(^2\)We refer to [3] for a detailed description of incomplete Cholesky decomposition algorithm.
of $U$. Since $NU^T\hat{\Xi}_1 U = U^T (N_1 \hat{\Xi}_1 + N_2 \hat{\Xi}_2) U$, we know by the positive semidefinite properties of $\hat{\Xi}_k$'s that

$$U^T \hat{\Xi}_k U = \text{diag}(\Phi_k, 0)$$  
(5.41)

$$\hat{\Xi}_k = U \text{diag}(\Phi_k, 0) U^T$$  
(5.42)

where $\Phi_k = U_1^T \hat{\Xi}_k U_1$, i.e. $\Phi_k$ is the second moment of class $k$ in the span of $U_1$ (see Appendix B.3).

Now consider the regularized generalized eigenvalue problem,

$$U \begin{pmatrix} \Phi_1 + \alpha I & 0 \\ 0 & \alpha I \end{pmatrix} U^T w_i = \lambda_i U \begin{pmatrix} \Phi_2 + \alpha I & 0 \\ 0 & \alpha I \end{pmatrix} U^T w_i$$  
(5.43)

Let $w_i = \sum_{j=1}^{m+1} c_{i,j} u_j = Uc_i$, where $c_i^T = (c_{i,1}, c_{i,2}, \cdots, c_{i,m+1})^T$. Then the problem can be simplified as

$$\begin{pmatrix} \Phi_1 + \alpha I & 0 \\ 0 & \alpha I \end{pmatrix} c_i = \lambda_i \begin{pmatrix} \Phi_2 + \alpha I & 0 \\ 0 & \alpha I \end{pmatrix} c_i$$

Denote the $i$-th canonical vector by $e_i$; that is the $i$-th component of $e_i$ is 1 and the others are zero. Note $e_{s+1}, e_{s+2}, \cdots, e_{D+1}$ are eigenvectors with eigenvalue 1, and therefore can be safely removed. Hence we only need to consider $c_i$ with the form of $c_i^T = (d_i^T, 0^T)$. It is easy to verify that $d_i$ is
the generalized eigenvector of \((\Phi_2 + \alpha I, \Phi_I + \alpha I)\), i.e.

\[
(\Phi_1 + \alpha I)d_i = \lambda(\Phi_2 + \alpha I)d_i
\]  \hspace{1cm} (5.44)

Since \(w_i = U_1 d_i\), we can get \(W_1\) as

\[
W_1 = U_1 D
\]  \hspace{1cm} (5.45)

where \(D = (d_1, d_2, \cdots, d_d)\).

To summarize, for data sets with high dimensionality, we can carry on the calculation in two levels. In the first level, we apply SVD on the (incomplete) Cholesky decomposition of \(\hat{\Xi}_t\); we then select singular vectors to form \(U_1\), whose singular values are larger than a predefined threshold value. In the second level, we project data in the span of \(U_1\) and calculate the second moments \(\Phi_i\)'s; finally we solve the generalized eigenvalue problem (5.44) and obtain the solution as defined in Equation (5.45). The pseudo-code of AFE for undersampled problems is presented in Algorithm 6.

5.2 Discussion

In this section, we investigate the properties of our proposed method, and study the relationship of the new proposed method with other dimension
Algorithm 6 Affine feature extraction for undersample problems

Input: Estimation of augmented second moment matrix and prior probabilities: \( \hat{\Xi}_k \) and \( \hat{\pi}_k \);
1: Apply (incomplete) Cholesky decomposition on \( \hat{\Xi}_t \): \( \hat{\Xi}_t = Y Y^T \)
2: Apply (thin) SVD on \( Y \) such that \( Y = U_1 \Delta V_1^T \)
3: Get a transformation matrix \( D \) by Algorithm 5 with inputs \( U_1^T \hat{\Xi}_k U_1 + \alpha I \) and \( \hat{\pi}_k \), \( (k = 1, 2) \)
4: return \( W = U_1 D \)

5.2.1 Information theoretical property of AFE

Connection with the Chernoff criterion

As discussed in subsection 4.1.1, the Chernoff distance of two multivariate normal distributions \( p_i \) and \( p_j \) has a closed expression as:

\[
D_c(\beta) = \frac{\beta(1-\beta)}{2} (\mu_2 - \mu_1)^T [\beta \Sigma_1 + (1-\beta) \Sigma_2]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{\det[\beta \Sigma_1 + (1-\beta) \Sigma_2]}{\det(\beta \Sigma_1) \det^{1-\beta}(\Sigma_2)}
\]

where \( p_i = \mathcal{N}(\mu_i, \Sigma_i) \).

We note that

\[
\det[\beta \Xi_1 + (1-\beta) \Xi_2]
= \det \begin{pmatrix}
\beta \Sigma_1 + \beta \mu_1 \mu_1^T + (1-\beta) \Sigma_2 + (1-\beta) \mu_2 \mu_2^T & \beta \mu_1 + (1-\beta) \mu_2 \\
\beta \mu_1^T + (1-\beta) \mu_2^T & 1
\end{pmatrix}
= \beta(1-\beta) \{1 + (\mu_2 - \mu_1)[\beta \Sigma_1 + (1-\beta) \Sigma_2]^{-1}(\mu_2 - \mu_1)\} \det[\beta \Xi_1 + (1-\beta) \Xi_2]
\]
and

\[ \det \Xi_k = \det \Sigma_k \]

where \( k = 1, 2 \). Therefore we have

\[ \ln \frac{\det[\beta \Xi_1 + (1-\beta) \Xi_2]}{\det^\beta(\Xi_1) \det^{1-\beta}(\Xi_2)} \]

\[ = \ln \beta (1-\beta) \{1 + (\mu_2 - \mu_1)[\beta \Sigma_1 + (1-\beta) \Sigma_2]^{-1}(\mu_2 - \mu_1)\} + \ln \frac{\det[\beta \Xi_1 + (1-\beta) \Xi_2]}{\det^\beta(\Sigma_1) \det^{1-\beta}(\Sigma_2)} \]

\[ \leq (1-\beta)(\mu_2 - \mu_1)[\beta \Sigma_1 + (1-\beta) \Sigma_2]^{-1}(\mu_2 - \mu_1) + \ln \frac{\det[\beta \Xi_1 + (1-\beta) \Xi_2]}{\det^\beta(\Sigma_1) \det^{1-\beta}(\Sigma_2)} \]

\[ = 2D_c(\beta) \]

That is \( \ln \frac{\det[\beta \Xi_1 + (1-\beta) \Xi_2]}{\det^\beta(\Xi_1) \det^{1-\beta}(\Xi_2)} \) is a lower bound of the Chernoff distance between two normal distributions. When \( \beta = \frac{1}{2} \), \( \ln \frac{\det[\frac{1}{2}(\Xi_1 + \Xi_2)]}{\sqrt{\det(\Xi_1) \det(\Xi_2)}} \) is a lower bound of the Bhattacharyya distance.

For arbitrary \( \mu_k \) and \( \Sigma_k \), directly maximizing the Chernoff distance or the Bhattacharyya distance is time consuming as there is no closed-form solution. To find a lower dimensional transformation matrix \( \mathbf{W} \), we can instead maximize a lower bound of the Bhattacharyya distance; that is

\[ \max \frac{\det\left[\frac{1}{2}(\mathbf{W}^T \Xi_1 \mathbf{W} + \mathbf{W}^T \Xi_2 \mathbf{W})\right]}{\sqrt{\det(\mathbf{W}^T \Xi_1 \mathbf{W}) \det(\mathbf{W}^T \Xi_2 \mathbf{W})}} \]  \hspace{1cm} (5.46)

Similar as we discussed in Section 4.3, the solution of optimization problem
(5.46) is the generalized eigenvectors \( w_i \) of \((\Xi_1, \Xi_2)\) with \( d \) largest \( \ln(\lambda_i + \frac{1}{\lambda_i} + 2) \), where \( \Xi_1 w_i = \lambda_i \Xi_2 w_i \). It is obvious that optimization problem (5.46) is equivalent as maximizing \( C(W; \frac{1}{2}, d) \). Therefore the AFE method can be thought as maximizing a lower bound of the Bhattacharyya distance when \( \xi = \frac{1}{2} \).

**Connection with the KL criterion**

As discussed in subsection 4.1.1, the KL distance of two multivariate normal distributions \( p_i \) and \( p_j \) has a closed expression as:

\[
L_{ij} = \frac{1}{2} \left\{ \log(|\Sigma_i^{-1} \Sigma_j|) + \text{tr}(\Sigma_i \Sigma_j^{-1}) + (\mu_i - \mu_j)^T \Sigma_j^{-1} (\mu_i - \mu_j) \right\} \quad (5.47)
\]

where \( p_i = \mathcal{N}(\mu_i, \Sigma_i) \). The J divergence is defined as

\[
J_a = \frac{1}{2} (L_{12} + L_{21})
\]

\[
= \frac{1}{4} \text{tr}\{\Sigma_2^{-1} \Sigma_1 + \Sigma_1^{-1} \Sigma_2 + (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_2 - \mu_1)(\mu_2 - \mu_1)^T\}
\]

Using formulas in Appendix B.1, one can easily get that

\[
J_a = \frac{1}{2} (C_0(\frac{1}{2}) - 1); \quad (5.48)
\]
That is, when $\xi$ is $1/2$, $C_0$ is equivalent to the $J$ divergence (up to a constant) of two normal distributions. The solution of maximizing $C$ can be seen as finding an affine subspace that maximally preserves $C_0$, i.e. an optimal truncated spectrum of $J_a$.

The KL divergence can be seen as a distance measure between two distributions, and therefore a measure of separability of classes. Traditional viewpoints aim at maximizing the KL divergence between classes in lower dimensional linear subspaces, see [54] for an introduction and [78, 121] for the recent development. It is easy to show that maximizing the lower-dimensional KL divergence in [54, 78, 121] is equivalent to our proposed problem with an additional constraint

$$\mathbf{W}^T = (\mathbf{V}^T, \mathbf{e})$$  \hspace{1cm} (5.49)

where $\mathbf{V} \in \mathbb{R}^{D \times d}$, and $\mathbf{e}^T = (0, 0, \cdots, 1)$. With the additional constraint, a closed-form solution cannot be found. By relaxing $\mathbf{e} \in \mathbb{R}^{D \times 1}$, we can find closed-form solutions.

5.2.2 Connection with FDA and MBD($\Delta \mu = 0$)

Without losing generality, let us consider the one dimensional case in this subsection. Let $\mathbf{w}^T = (\mathbf{v}_1^T, v_0)$. Then we have $Z = \mathbf{v}_1^T X + v_0$, where $X$ and $Z$
are random covariate in higher- and lower-dimensional spaces. Displacement \( v_0 \) is the same for both classes, and therefore plays no important role for final classifications. In other words, the effectiveness of the generated feature is solely determined by \( v_1 \). Let \( v_1^* \) be an optimal solution.

Consider maximizing \( C(W; 1/2, d) \). We know that \( w^* \) is the eigenvector of \( \Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 \) with the largest eigenvalue.

First, let us consider \( \mu_1 = \mu_2 = \mu \). Using formulas in Appendix B.1, we can simplify \( \Xi_1^{-1}\Xi_2 + \Xi_2^{-1}\Xi_1 \) as

\[
\Xi_1^{-1}\Xi_2 + \Xi_2^{-1}\Xi_1 = \begin{pmatrix}
\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 & 0 \\
2\mu^T - \mu^T(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1) & 1
\end{pmatrix}
\]

Then by simple linear algebra, we can show that \( v_1^* \) is also the eigenvector of \( \Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 \) with the largest eigenvalue.

Second, let us consider \( \Sigma_1 = \Sigma_2 = \Sigma \). In this case, it is easy to verify the following:

\[
\Xi_1^{-1}\Xi_2 + \Xi_2^{-1}\Xi_1 = \begin{pmatrix}
A & 0 \\
0 & B
\end{pmatrix} + 2I
\]

where \( A = \Sigma^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \) and \( B = (\mu_1 - \mu_2)^T\Sigma^{-1}(\mu_1 - \mu_2)^T \). It is then not hard to show that \( v_1^* \) is the eigenvector of \( A \) with the largest
eigenvalue.

In summary, we show that FDA and MBD(∆µ = 0) are special cases of our proposed AFE for normally distributed data. Therefore, theoretically speaking AFE is more flexible than FDA and MBD(∆µ = 0).

5.2.3 Connection with QDA

Now let us see the connection of the new proposed AFE method with QDA. As in [64], we define the quadratic discriminant score for an observation vector \( x \) as

\[
\delta(x) = (x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) - (x - \mu_2)^T \Sigma_2^{-1}(x - \mu_2)
\]  

(5.50)

For the moment, we assume \( \Sigma_i \)'s are nonsingular. We have the log-ratio of the posterior probabilities as

\[
\ln \frac{p(Y = 2 | X = x)}{p(Y = 1 | X = x)} = \frac{1}{2}(\delta(x) - \theta)
\]  

(5.51)

where \( \theta \) is a constant [39, 65]. Therefore, for 0–1 loss function, we have the following Bayesian decision rule: if \( \delta(x) > \theta \), \( x \) is classified into class 2; if \( \delta(x) < \theta \), \( x \) is classified into class 1; if \( \delta(x) = \theta \), \( x \) is on the decision boundary. By formulas in Appendix B.1, we can simplify the quadratic
discriminant score as

$$
\delta(x) = (x^T, 1)((\Xi_1^{-1} - \Xi_2^{-1}) \begin{pmatrix} x \\ 1 \end{pmatrix})
$$

(5.52)

It is then easy to recognize that the null space of $((\Xi_1^{-1} - \Xi_2^{-1})$ contributes zero to the quadratic discriminant score, and therefore plays no role in classifications. Therefore, roughly speaking, a natural reduced-rank QDA can be obtained by projecting observations into the range of $(\Xi_1^{-1} - \Xi_2^{-1})$ and discarding the null space of $(\Xi_1^{-1} - \Xi_2^{-1})$.

Let $w_i$ and $\lambda_i$ be the generalized eigenvector and eigenvalue of $((\Xi_1, \Xi_2)$ such that $\Xi_1 w_i = \lambda_i \Xi_2 w_i$, where $\lambda_i > 0$, $i = 1, 2, \cdots, D + 1$. Denote $W = (w_1, w_2, \cdots, w_{D+1})$. Then by Lemma 5.1, we have

$$
W^T \Xi_k W = \Lambda_k
$$

(5.53)

where $\Lambda_1 = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{D+1})$ and $\Lambda_2 = I$. Then by simple linear algebra, we have the following

$$
W^T \Xi_k W = \Lambda_k \Rightarrow \Xi_k = (W^{-1})^T \Lambda_k W^{-1} \Rightarrow \Xi_k^{-1} = W \Lambda_k^{-1} W^T
$$
Therefore the quadratic discriminant score can be written as

$$\delta(x) = \begin{pmatrix} x^T & 1 \end{pmatrix} W (\Lambda_1^{-1} - I) W^T \begin{pmatrix} x \\ 1 \end{pmatrix}$$

For any \( \lambda_i \approx 1 \), we can remove the corresponding \( w_i \) in \( W \) without losing discriminative power too much. Assume that \( \lambda_i \)'s and \( w_i \)'s are permuted such that, for a positive integer \( n \), we have \(|1 - \lambda_i| \geq \tau\) for \( i \in \{1, 2, \cdots, n\} \), and \(|1 - \lambda_i| < \tau\) for \( i \in \{n+1, n+2, \cdots, D+1\} \), where \( \tau > 0 \) is a predefined threshold value. Let \( \Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n) \) and \( W_1 = (w_1, w_2, \cdots, w_n) \).

Then we can write the reduced-rank quadratic discriminant score as

$$\delta(x) \approx z^T (\Lambda^{-1} - I) z \quad (5.54)$$

where \( z = W_1^T \begin{pmatrix} x \\ 1 \end{pmatrix} \). Therefore, AFE naturally defines reduced-rank QDA, similar as FDA for reduced-rank LDA [64].

Furthermore, we have the following theorem:

**Theorem 5.1.** We suppose that class conditional distributions are multivariate normal with mean \( \mu_k \) and covariance \( \Sigma_k \), i.e., \( X \mid Y = k \sim \mathcal{N}(\mu_k, \Sigma_k) \), where \( k = 1, 2 \). Let \( \lambda_i \) and \( w_i \), \( i = 1, 2, \cdots, D \), be the eigenvalue and eigenvector of \((\Xi_1, \Xi_2)\), respectively. For simplicity, let us assume that, for \( i > m \),
\[ \lambda_i = 1. \] Define
\[
W = (w_1, w_2, \cdots, w_m) \quad (5.55)
\]
\[
W^\perp = (w_{m+1}, w_{m+2}, \cdots, w_{D+1}) \quad (5.56)
\]
Then \( Z = W^T \begin{pmatrix} X \\ 1 \end{pmatrix} \) defines a sufficient dimension reduction for \( X \) about \( Y \).

**Proof.** See Appendix B.4. \( \Box \)

### 5.3 Experiments

#### 5.3.1 Two Simple Examples

We first consider two examples discussed in Chapter 4. For compactness, we only presented results for the AFE1 method. In Example 4.1, Foley and Sammon showed that FKT can miss important discriminant directions [46]. Now let us see how AFE works on the example.

**Example 5.1.** In the example constructed by Foley and Sammon [46], we have a binary classification problem in which class-conditional distributions
are multivariate Gaussian. Specifically, we have

\[ \begin{align*}
\mu_1 &= \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix}, \quad 
\Sigma_1 = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \quad \pi_1 = 0.5 \\
\mu_2 &= \begin{pmatrix} 0 \\ -5 \\ 0 \end{pmatrix}, \quad 
\Sigma_2 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \pi_2 = 0.5
\end{align*} \] (5.57) (5.58)

It can be easily verified that, for this case, projecting data to \( x_2 \) results in Bayes error 0.0003 percent, while projecting data to \( x_1 \) or \( x_3 \) has Bayes error 35 percent.

We have

\[ \Xi_1 = \begin{pmatrix} \Sigma_1 + \mu_1 \mu_1^T & \mu_1 \\ \mu_1^T & 1 \end{pmatrix} = \begin{pmatrix} 8 & 0 & 0 & 0 \\ 0 & 26 & 0 & 5 \\ 0 & 0 & 8 & 0 \\ 0 & 5 & 0 & 1 \end{pmatrix} \]

\[ \Xi_2 = \begin{pmatrix} \Sigma_2 + \mu_2 \mu_2^T & \mu_2 \\ \mu_2^T & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 26 & 0 & -5 \\ 0 & 0 & 2 & 0 \\ 0 & -5 & 0 & 1 \end{pmatrix} \]
CHAPTER 5. AFFINE FEATURE EXTRACTION

Ξ_t = π_1Ξ_1 + π_2Ξ_2 = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 26 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}

Then we have eigenvectors and their corresponding eigenvalues of (Ξ_1, Ξ_2) as follows

\begin{align*}
\mathbf{v}_1 &= \begin{pmatrix} 0 \\ 0.1387 \\ 0 \\ -0.7071 \end{pmatrix} & \lambda_1 &= 0.0098 \\
\mathbf{v}_2 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -0.4472 \end{pmatrix} & \lambda_2 &= 4 \\
\mathbf{v}_3 &= \begin{pmatrix} 0.4472 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \lambda_3 &= 4 \\
\mathbf{v}_4 &= \begin{pmatrix} 0 \\ 0.1387 \\ 0 \\ 0.7071 \end{pmatrix} & \lambda_4 &= 101.9902
\end{align*}
Figure 5.1: We show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0,1]$. The order of $\xi \lambda + \frac{1-\xi}{\lambda}$ are used to select projection direction in the AFE methods.

In Figure 5.1, we show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0,1]$. We select projection directions based on the orders of $\xi \lambda + \frac{1-\xi}{\lambda}$. Therefore, using AFE, we always select $v_1$, $v_4$, or both, before we project data to $v_2$ and $v_3$. That is the AFE methods correctly rank projection directions in this example.

In Example 4.2, Zhu showed that SAVE can mis-rank projection directions [142]. Let us show how AFE works on the example.

Example 5.2. In the example constructed by Zhu, [142], we have a binary classification problem in which class-conditional distributions are multivari-
ate Gaussian. Specifically, we have

\[
\mu_1 = \begin{pmatrix} -\sqrt{0.5} \\ 0 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.3 \end{pmatrix} \quad (5.59)
\]

\[
\mu_1 = \begin{pmatrix} \sqrt{0.5} \\ 0 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 1.7 \end{pmatrix} \quad (5.60)
\]

It can be verified that projecting data to \(x_1\) and \(x_2\) results in Bayes error 15.9\% and 30.2\%, respectively.

In this case, we have

\[
\Xi_1 = \begin{pmatrix} 
\Sigma_1 + \mu_1\mu_1^T & \mu_1 \\
\mu_1^T & 1
\end{pmatrix} = \begin{pmatrix} 
1 & 0 & -\sqrt{0.5} \\
0 & 0.3 & 0 \\
-\sqrt{0.5} & 0 & 1
\end{pmatrix}
\]

\[
\Xi_2 = \begin{pmatrix} 
\Sigma_2 + \mu_2\mu_2^T & \mu_2 \\
\mu_2^T & 1
\end{pmatrix} = \begin{pmatrix} 
1 & 0 & \sqrt{0.5} \\
0 & 0.3 & 0 \\
\sqrt{0.5} & 0 & 1
\end{pmatrix}
\]

Then we have eigenvectors and their corresponding eigenvalues of \((\Xi_1, \Xi_2)\) as follows

\[
v_1 = \begin{pmatrix} -0.5412 \\ 0 \\ -0.5412 \end{pmatrix}, \quad \lambda_1 = 0.1716
\]
Figure 5.2: We show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0,1]$. The order of $\xi \lambda + \frac{1-\xi}{\lambda}$ are used to select projection direction in the AFE methods.

\[ v_2 = \begin{pmatrix} 0 \\ 0.7670 \\ 0 \end{pmatrix} \quad \lambda_2 = 0.1765 \]
\[ v_3 = \begin{pmatrix} -1.3066 \\ 0 \\ 1.3066 \end{pmatrix} \quad \lambda_3 = 5.8284 \]

In Figure 5.2, we show the values of $\xi \lambda + \frac{1-\xi}{\lambda}$ for various $\xi \in [0,1]$.

Therefore, using AFE, we always select $v_1$, $v_3$, or both, before we project data to $v_2$. That is the AFE methods correctly rank projection directions in this example.
5.3.2 Visualization on Simulated Data Sets

In order to compare our method with PCA, FDA, and FKT, we generated several 7-dimensional toy data sets. The toy data sets contain 3-dimensional relevant components, while the others are merely random noise. The 3 relevant components form two concentric cylinders. The generated data are spread along the surfaces of the cylinders. The cylinders are of elliptic, parabolic and hyperbolic forms. Figure 5.3 illustrates the first two features found by PCA, FDA, FKT and our new approach AFE. FDA fails to separate the two classes. PCA and FKT separate the two classes for some cases; but they fail to show the inherent discriminant structure of data for most cases. On the other hand, our method correctly captures the discriminant information in the data.

5.3.3 Classification with Benchmark Data Sets

We selected two artificial data sets used in [15]: twonorm and ringnorm. We summarize these two datasets as follows:

**Dataset twonorm** This is a 20-dimensional binary classification data. Each class conditional distribution is a multivariate normal distribution with identity covariance matrix. The mean of class 1 is \((2, 2, \cdots, 2)/\sqrt{20}\);
Figure 5.3: Comparison of features found by PCA, FDA, FKT and Our method. Blue and red points belong to different classes.
and the mean of class 2 is $-(2, 2, \cdots, 2)/\sqrt{20}$. This dataset has an expected error rate of 2.3% [15].

**Dataset ringnorm** This is a 20-dimensional binary classification data. The class one conditional distribution is a multivariate normal distribution with zero mean and covariance $4I$. The class two conditional distribution is a multivariate normal distribution with mean $(2, 2, \cdots, 2)/\sqrt{20}$ and identity covariance matrix. This dataset has an expected error rate of 1.3% [15].

We also selected 18 binary classification data sets from UCI Machine Learning Repository [47]. The statistics of these benchmark datasets are listed in Table 5.2

In our numerical experiments, we followed a procedure as follows. For each dataset, we randomly selected 80% of the data as a training set, the remaining data are used as a test set; we repeated 20 times to create 20 training and test sets. We compared our new approach with FDA, MBD ($\Delta \mu = 0$) [54], FKT [55], HLDA [87] \(^3\), SAVE [26, 100] \(^4\). For convenience, AFE1 and AFE2 are used for $\Xi_t$-orthogonal and orthogonal AFE algorithms,

\(^3\)We used code PRTools [125].
\(^4\)We used R package dr [133].
<table>
<thead>
<tr>
<th>Dataset</th>
<th>$D$</th>
<th>$N$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twonorm</td>
<td>20</td>
<td>7400</td>
<td>2</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>20</td>
<td>7400</td>
<td>2</td>
</tr>
<tr>
<td>Australian</td>
<td>14</td>
<td>690</td>
<td>2</td>
</tr>
<tr>
<td>Banknote</td>
<td>4</td>
<td>1372</td>
<td>2</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>30</td>
<td>569</td>
<td>2</td>
</tr>
<tr>
<td>BUPA</td>
<td>6</td>
<td>345</td>
<td>2</td>
</tr>
<tr>
<td>Climate</td>
<td>20</td>
<td>540</td>
<td>2</td>
</tr>
<tr>
<td>Diabetes</td>
<td>8</td>
<td>768</td>
<td>2</td>
</tr>
<tr>
<td>German</td>
<td>24</td>
<td>1000</td>
<td>2</td>
</tr>
<tr>
<td>Heart</td>
<td>13</td>
<td>270</td>
<td>2</td>
</tr>
<tr>
<td>ILPD</td>
<td>10</td>
<td>583</td>
<td>2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>33</td>
<td>351</td>
<td>2</td>
</tr>
<tr>
<td>Magic</td>
<td>10</td>
<td>19020</td>
<td>2</td>
</tr>
<tr>
<td>Mammographic</td>
<td>5</td>
<td>830</td>
<td>2</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>22</td>
<td>195</td>
<td>2</td>
</tr>
<tr>
<td>Planning</td>
<td>12</td>
<td>182</td>
<td>2</td>
</tr>
<tr>
<td>QSAR</td>
<td>41</td>
<td>1055</td>
<td>2</td>
</tr>
<tr>
<td>Sonar</td>
<td>60</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Splice</td>
<td>60</td>
<td>3175</td>
<td>2</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>6</td>
<td>310</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.2: Description of 20 artificial and real data sets ($K$: number of classes, $D$: dimensionality of data, and $N$: number of observations)
CHAPTER 5. AFFINE FEATURE EXTRACTION

while MBD is used for MBD ($\Delta \mu = 0$). We used these dimension reduction methods to generate lower-dimensional features; the features are then used by regularized discriminant analysis (RDA)\(^5\) to do classifications. To measure the discriminant information of the data set, we also classified the original data set via RDA, which we denote FULL in the following discussion. Feature extraction and classification are trained on training sets, and error rates are calculated with predictions on corresponding test sets.

We examined the influence of $\xi$ of the AFE methods on classification performance by varying $\xi$ from 0 to 1. For Datasets Twonorm, Ringnorm, Ionosphere, and Sonar, we showed boxplots of test set error rates in Figure 5.4. For Ringnorm, the best performance is attained when $\xi = 0$, while for Ionosphere, the best performance is attained when $\xi = 1$; For Twonorm and Sonar, the best performance is attained when $\xi$ is between 0 and 1. For optimal classification performance, we therefore recommend using a cross-validation procedure to determine the best $\xi$ for the AFE methods. Nevertheless, for simplicity, we set $\xi$ to be the class empirical prior in the following numerical experiments, i.e. $\xi = \frac{N_1}{N}$.

---

\(^5\)RDA is a convex combination between linear and quadratic discriminant analysis with proper shrinkage regularization [49]. It is perhaps the most popular approach to statistical discriminant analysis, especially for undersampled problems [64]. In all our experiments, we used leave-one-out approach to select optimal parameters for RDA on a $(0, 0.1, \cdots, 1) \times (0, 0.1, \cdots, 1)$ grid.
Figure 5.4: Effect of $\xi$ on error rates when features are extracted by the $\Xi_t$-Orthogonal method and then classified by regularized discriminant analysis. Red bar shown in the figures are the median error rate of 20 test sets; while blue circle shown in the figures are the mean error rates of 20 test sets. See text for details.
To compare the effectiveness of these discriminative dimension reduction methods, we extracted $d$-dimensional features as input for RDA, and calculated test set error rates. For each dataset of dimensionality $D$, we set $d$ to be even numbers between 2 and $D - 1$. In Figures 5.5 and 5.6, we showed the test set error rates of different dimensionality for datasets: Twonorm, Ringnorm, Ionosphere, and Sonar. Theoretically speaking, the optimal discriminative feature for Dataset Twonorm is determined by FDA; we observed that all methods except FKT have similar classification performance. For Dataset Ringnorm, we observed that all methods except SAVE have similar error rates; we believe that the worse performance of SAVE is because SAVE sometimes mis-ranks extracted features, as illustrated in Example 4.2. For Dataset Ionosphere, HLDA achieves better performance than the AFE methods when dimensionality $d$ is small; however when $d > 18$, the AFE methods attain smaller error rates. For Dataset Sonar, AFE1 and AFE2 outperform others when $d < 40$, while for $d \geq 40$, AFE1 attain the best performance. We think the deteriorating performance of AFE2 on Dataset Sonar is that: unlike AFE1, AFE2 extracts features that are not mutually orthogonal; thus when $d$ is large, features may be highly correlated and result in worse classification performance.
Figure 5.5: Test set error rates for different dimensionality.

(1) Two Norm

(2) Ringnorm
Figure 5.6: Test set error rates for different dimensionality.
We compared our new approach with FDA, MBD, FKT, HLDA, SAVE, and FULL on the 20 selected benchmark datasets. For MBD, FKT, HLDA, SAVE and the AFE methods, we set $d$ to be even numbers between 2 and $D - 1$ for each dataset of dimensionality $D$. Following [87], we reported the best attainable classification performance and the optimal dimensionality for each method in Table 5.3. In Table 5.3, smallest mean test error rates are displayed as bold numbers. We also reported other mean error rates in bold, which are not statistically significantly different from the best ones; for this purpose, we used Wilcoxon signed-rank test with significant level 0.05 to compare error rates. For each dataset, we calculated the rank for each method based on mean test error rates: a method with the smallest mean error rate has rank 1, while a method with the largest error rate has rank 8. In Figure 5.7, we showed statistical boxplots of the ranks of these methods (red bars and green circles are, respectively, the median and mean ranks over the 18 real datasets). The poor performance of FDA, MBD and FKT affirms that first-order or second-order statistics alone cannot capture discriminant information contained in the data sets. As shown in Table 5.3, in 17 of the 20 datasets, AFE1 is either the best one, or not statistically significantly different from the best ones; in 15 of the 20 datasets, AFE2 is
CHAPTER 5. AFFINE FEATURE EXTRACTION

Table 5.3: Optimal observed mean test error rates and the corresponding dimensionality (shown in parentheses) for each method. Error rates are shown in percentage. Best error rates are displayed in bold. So are error rates that are not statistically significantly different from the best ones.

<table>
<thead>
<tr>
<th>DataSet</th>
<th>FDA</th>
<th>MBD</th>
<th>FKT</th>
<th>HLDA</th>
<th>SAVE</th>
<th>AFE1</th>
<th>AFE2</th>
<th>FULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwoNorm</td>
<td>2.09 (1)</td>
<td>2.72 (18)</td>
<td>6.08 (18)</td>
<td>2.05 (14)</td>
<td>2.06 (12)</td>
<td>2.05 (18)</td>
<td>2.05 (6)</td>
<td>2.06 (20)</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>22.71 (1)</td>
<td>1.91 (18)</td>
<td>1.83 (18)</td>
<td>1.78 (18)</td>
<td>2.22 (18)</td>
<td>1.84 (18)</td>
<td>1.84 (18)</td>
<td>1.40 (20)</td>
</tr>
<tr>
<td>Australian</td>
<td>14.54 (1)</td>
<td>14.38 (12)</td>
<td>14.60 (10)</td>
<td>13.69 (6)</td>
<td>13.44 (8)</td>
<td>13.64 (6)</td>
<td>13.93 (6)</td>
<td>13.62 (14)</td>
</tr>
<tr>
<td>Banknote</td>
<td>1.93 (1)</td>
<td>1.57 (2)</td>
<td>1.70 (2)</td>
<td>1.64 (2)</td>
<td>1.71 (2)</td>
<td>1.57 (2)</td>
<td>1.62 (2)</td>
<td>1.71 (4)</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>36.90 (1)</td>
<td>4.48 (20)</td>
<td>4.46 (14)</td>
<td>3.01 (2)</td>
<td>4.96 (24)</td>
<td>4.57 (28)</td>
<td>4.58 (26)</td>
<td>5.21 (30)</td>
</tr>
<tr>
<td>BUPA</td>
<td>33.17 (1)</td>
<td>33.02 (4)</td>
<td>33.30 (4)</td>
<td>33.79 (2)</td>
<td>35.99 (4)</td>
<td>30.75 (2)</td>
<td>31.34 (2)</td>
<td>32.99 (6)</td>
</tr>
<tr>
<td>Climate</td>
<td>5.51 (1)</td>
<td>5.95 (18)</td>
<td>6.35 (18)</td>
<td>6.21 (16)</td>
<td>5.93 (18)</td>
<td>5.75 (18)</td>
<td>6.11 (16)</td>
<td>6.07 (20)</td>
</tr>
<tr>
<td>Diabetes</td>
<td>24.11 (1)</td>
<td>27.43 (6)</td>
<td>25.14 (6)</td>
<td>24.21 (6)</td>
<td>24.74 (6)</td>
<td>23.94 (2)</td>
<td>24.64 (6)</td>
<td>24.38 (8)</td>
</tr>
<tr>
<td>German</td>
<td>23.09 (1)</td>
<td>23.03 (18)</td>
<td>23.66 (4)</td>
<td>22.75 (22)</td>
<td>23.71 (22)</td>
<td>22.64 (14)</td>
<td>22.37 (12)</td>
<td>23.03 (24)</td>
</tr>
<tr>
<td>Heart</td>
<td>16.48 (1)</td>
<td>16.39 (12)</td>
<td>15.15 (10)</td>
<td>16.19 (10)</td>
<td>15.68 (10)</td>
<td>15.69 (2)</td>
<td>16.68 (2)</td>
<td>16.29 (13)</td>
</tr>
<tr>
<td>ILPD</td>
<td>28.68 (1)</td>
<td>27.04 (4)</td>
<td>28.53 (4)</td>
<td>27.08 (2)</td>
<td>26.99 (2)</td>
<td>27.04 (2)</td>
<td>26.99 (2)</td>
<td>28.68 (10)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>13.61 (1)</td>
<td>5.78 (28)</td>
<td>6.15 (32)</td>
<td>6.55 (28)</td>
<td>6.09 (26)</td>
<td>5.64 (28)</td>
<td>5.52 (26)</td>
<td>6.02 (33)</td>
</tr>
<tr>
<td>Magic</td>
<td>32.08 (1)</td>
<td>21.13 (8)</td>
<td>20.86 (4)</td>
<td>20.58 (8)</td>
<td>20.57 (6)</td>
<td>20.73 (8)</td>
<td>20.73 (8)</td>
<td>20.67 (10)</td>
</tr>
<tr>
<td>Mammographic</td>
<td>19.18 (1)</td>
<td>18.90 (2)</td>
<td>19.89 (4)</td>
<td>19.30 (4)</td>
<td>19.01 (2)</td>
<td>19.29 (4)</td>
<td>19.43 (4)</td>
<td>19.29 (5)</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>23.27 (1)</td>
<td>13.56 (20)</td>
<td>11.92 (14)</td>
<td>10.21 (10)</td>
<td>11.21 (16)</td>
<td>13.67 (20)</td>
<td>14.08 (20)</td>
<td>11.91 (22)</td>
</tr>
<tr>
<td>Planning</td>
<td>29.28 (1)</td>
<td>27.80 (8)</td>
<td>28.06 (2)</td>
<td>28.32 (2)</td>
<td>29.80 (2)</td>
<td>28.01 (6)</td>
<td>27.67 (10)</td>
<td>29.18 (12)</td>
</tr>
<tr>
<td>QSAR</td>
<td>33.65 (1)</td>
<td>14.82 (40)</td>
<td>13.85 (40)</td>
<td>14.07 (38)</td>
<td>14.27 (36)</td>
<td>14.02 (32)</td>
<td>14.00 (32)</td>
<td>14.46 (41)</td>
</tr>
<tr>
<td>Sonar</td>
<td>49.34 (1)</td>
<td>14.68 (54)</td>
<td>19.68 (32)</td>
<td>18.54 (30)</td>
<td>18.50 (30)</td>
<td>15.64 (24)</td>
<td>16.20 (32)</td>
<td>15.85 (60)</td>
</tr>
<tr>
<td>Splice</td>
<td>15.24 (1)</td>
<td>10.68 (50)</td>
<td>10.26 (8)</td>
<td>8.93 (4)</td>
<td>9.03 (4)</td>
<td>9.04 (6)</td>
<td>9.12 (6)</td>
<td>10.78 (60)</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>32.67 (1)</td>
<td>16.81 (4)</td>
<td>19.04 (4)</td>
<td>16.16 (4)</td>
<td>16.62 (4)</td>
<td>15.93 (4)</td>
<td>16.60 (4)</td>
<td>15.91 (6)</td>
</tr>
</tbody>
</table>

either the best one, or not statistically significantly different from the best ones. By comparing AFE1 and AFE2 with FULL, we see that AFE1 and AFE2 are capable of extracting the discriminant information of the chosen data.

5.4 Conclusions

In this study, we proposed a novel dimension reduction method for binary classification problems. Unlike traditional linear subspace methods, the new proposed method finds lower-dimensional affine subspaces for data observa-
Figure 5.7: The ranks of each method based on mean test error rates (smaller rank means smaller error rate). Red bar and green circle are, respectively the median and mean rank of the 18 real datasets.
CHAPTER 5. AFFINE FEATURE EXTRACTION

...itions. We presented the closed-form solutions of our new approach, and investigated its information-theoretical and statistical properties. We showed that our method has close connections with FDA, MBD ($\Delta mean = 0$) and FKT methods in the literature. Numerical experiments show the competitiveness of our method as a preliminary data-exploring tool for data visualization and classification.

Though we focus on binary classification problems in this study, it is always desirable to handle multi-class problems. One can extend AFE to multi-class problems by following the work presented in [37]. Here we propose another way to extend AFE to multi-class. Let $J_{ij}$ be the symmetric KL distance of classes $i$ and $j$, and assume class $i$, ($i = 1, 2, \ldots, K$), can be modeled by multivariate normal distribution. Then we have $\sum_{i=1}^{K} \Xi_i^{-1} \Xi_t \propto \sum_{i,j=1}^{K} J_{ij}$, where $\Xi_i$ is the augmented second moment matrix for class $i$ and $N \Xi_t = \sum_{i=1}^{K} N_i \Xi_i$. Therefore we may calculate the truncated spectrum of $\sum_{i=1}^{K} \Xi_i^{-1} \Xi_t$ for the lower-dimensional representations.
Chapter 6

Conclusions and Future Directions

Classical LDA and QDA are two widely used statistical classification methods, which are referred by Hastie, Tibshirani and Friedman in [64] as: “it seems that whatever exotic tools are the rage of the day, we should always have available these two simple tools”. In this thesis, we focused on QDA. Specifically, we investigated the parameter estimation and dimension reduction problems for QDA.

In Chapter 2, we briefly reviewed statistical decision theory for classification, and introduced generative, discriminative and hybrid generative/discriminative learning for parameter estimation.

In Chapter 3, we are motivated by the recent research on hybrid learning
and proposed to estimate parameters of QDA by maximizing a convex combination of joint log-likelihood and conditional log-likelihood of observations and their labels.

We first considered more general case: class-conditional distributions are from the exponential family. We constructed an iterative majorize-minimie algorithm to solve the hybird learning for the exponential family. In each iteration of the proposed MM algorithm, we need to (approximately) solve a convex optimization problem.

The Gaussian distribution is a special case of the exponential family. When applying for QDA, we have a general semidefinite problem, which, to our best knowledge, has no known optimization method. To design an optimization algorithm, we have two concerns:

- It is well understood that optimization with many small steps and early stopping helps boosting methods to achieve better classification results [139, 50]. We therefore would like to have an algorithm that searches parameters with small steps and possibly early stopping;

- We would like each iteration is computationally efficient and easy to implement, which hopefully compensates more iterations required to reach local optiumns.
With these two considerations, we constructed a block-coordinate descent algorithm that sequentially update parameters of QDA; in each update, we have a trust region method for solving optimal estimations, of which we have closed form solutions in each iteration. Our algorithm can be implemented with BLAS and LAPACK routines, which are numerical robust, efficient and reliable. Moreover, as parameters are sequentially updated, memory requirement of the algorithm is much less than algorithms that update parameters altogether. Numerical experiments show that our method is competitive with, in some cases significantly better than some widely-used quadratic classification methods in the literature. Moreover, as an optimization method, our method is competitive with, in some cases significantly better than the conjugate gradient method used in [12].

In Chapter 4, we reviewed linear subspace dimension reduction methods for class-conditional distribution being multivariate normal. Especially, we see that, when covariance matrices and means of class conditional distributions are arbitrary, known methods all have limitations. That is, “... for QDA, where so far there has been no universally accepted dimension-reduction technique in the literature” [65], though considerable efforts have been made to design dimension reduction methods for the case, for example
In Chapters 5, we investigated the dimension reduction problem for QDA. We generalized the Fukunaga-Koontz transformation \((FKT)\), and proposed a novel affine feature extraction \((AFE)\) algorithm for binary QDA, i.e. \(K = 2\). The proposed AFE method has closed-form solutions and thus can be solved efficiently. We show that AFE has the following properties: 1) it is a sufficient dimension reduction method; that is, it finds \(Z\) such that \(Y\) is conditional independent of \(X\) given \(Z\), i.e. \(Y \perp \perp X \mid Z\); 2) it finds an affine subspace that best preserves symmetric Kullback-Leibler (KL) divergence; and 3) FDA is a special case of AFE. Numerical experiments show that the AFE methods is competitive with, and in some cases outperforms other dimension reduction techniques designed for discriminant analysis.

The new proposed hybrid learning algorithm and feature extraction method have difficulties when data are very high dimensional. We think the problem can be alleviated with sparse estimation. Sparse estimation has been proven to be one effective way to deal with high dimension data. In fact, many classical methods have been “sparsified”, for example, sparse PCA \([32, 94, 138, 77, 88]\), sparse FDA \([114, 95, 21, 93]\), sparse regression
we think our the hybrid learning algorithm can benefit from recent works on sparse covariance estimation [48, 31, 4, 83, 70, 90, 99], and our AFE methods can benefit from recent work sparse (generalized) eigenvalue problems [136, 116].

Another serious limitation is that our work is based on the assumption of the class conditional distribution being multivariate normal. To deal with non-Gaussian data, copula, which has been widely used for financial modeling, has gain its attention in machine learning [96, 42], for example copula LDA [62, 86], copula PCA [61], copula Bayesian network [41]. We think combining copula idea with our hybrid learning algorithm and AFE methods can be even more helpful in handling machine learning and pattern classification problems.

In his thesis, Mika argued for the kernel Fisher discriminant analysis (KFD) [92]:

It would certainly be wrong to draw from this work the conclusion that KFD is better than other techniques, but we have demonstrated that KFD yields good results. And as for most algorithms, there are particular situations in which the special
way of solving the learning problem imposed by KFD has its advantages.

We conclude this thesis with similar argument for the new proposed hybrid learning algorithm for exponential family, (QDA in particular), and the AFE method for QDA. We hope that the new proposed hybrid learning algorithm and AFE methods can be useful in machine learning practice, and can be a building block for more advanced learning algorithms.
Appendix A

Appendices for Chapter 3

A.1 Optimization problem (3.24)

Define

\[ s(\Upsilon) = -\sum_{k=1}^{K} \frac{N_k}{N} v_k + \frac{\beta}{N} \sum_{n=1}^{N} \log \sum_{k=1}^{K} p(x_n|k; \hat{\Theta}_k) e^{v_k} + (1 - \beta) \log \sum_{k=1}^{K} e^{v_k} \]  

(A.1)

and

\[ \phi_k = v_k - v_k^{(\tau)} \]  

(A.2)

where \( k = 1, 2, \cdots, K \), and \( \Upsilon^{(\tau)} = (v_1^{(\tau)}, v_2^{(\tau)}, \cdots, v_K^{(\tau)}) \) is the estimation of \( \Upsilon \) at the \( \tau \)-th iteration.

Note that \( \forall x > 0, \text{ we have } \log(x) \leq x - 1 \) with equality if and only if
\[ s(\Upsilon) - s(\Upsilon^{(\tau)}) \]
\[
= - \sum_{k=1}^{K} N_k \frac{1}{N} \phi_k + \frac{\beta}{N} \sum_{n=1}^{N} \log \frac{\sum_{k=1}^{K} p(x_n | k; \hat{\Theta}_k) e^{\nu_k}}{\sum_{k=1}^{K} p(x_n | k; \hat{\Theta}_k) e^{\nu_k}} + (1 - \beta) \log \frac{\sum_{k=1}^{K} e^{\nu_k}}{\sum_{k=1}^{K} e^{\nu_k}} \]
\[
\leq - \sum_{k=1}^{K} N_k \frac{1}{N} \phi_k + \frac{\beta}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} p(k | x_n; \hat{\Theta}_k, \Upsilon^{(\tau)}) e^{\phi_k} + (1 - \beta) \sum_{k=1}^{K} \pi_k^{(\tau)} e^{\phi_k} + 1 \]

where for \( k = 1, 2, \cdots, K \),

\[
p(k | x_n; \hat{\Theta}_k, \Upsilon^{(\tau)}) = \frac{p(x_n | k; \hat{\Theta}_k) e^{\nu_k^{(\tau)}}}{\sum_{k=1}^{K} p(x_n | k; \hat{\Theta}_k) e^{\nu_k^{(\tau)}}} \]
\[
\pi_k^{(\tau)} = \frac{e^{\nu_k^{(\tau)}}}{\sum_{k=1}^{K} e^{\nu_k^{(\tau)}}} \]

Instead of minimizing \( s(\Upsilon) \), we can iteratively minimize an upper bound of \( s(\Upsilon) - s(\Upsilon^{(\tau)}) \) until convergence. Consider the following optimization problem

\[
\min_{\phi_1, \phi_2, \cdots, \phi_K} - \sum_{k=1}^{K} N_k \frac{1}{N} \phi_k + \frac{\beta}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} p(k | x_n; \hat{\Theta}_k, \Upsilon^{(\tau)}) e^{\phi_k} + (1 - \beta) \sum_{k=1}^{K} \pi_k^{(\tau)} e^{\phi_k} \]  
(A.3)

It can be easily shown that the optimal solution for this problem is

\[
\phi_k^* = \log \left( \frac{N_k}{\beta \lambda_k + (1 - \beta) \pi_k^{(\tau)}} \right) \]  
(A.4)
where
\[
\lambda_k = \frac{1}{N} \sum_{n=1}^{N} p(k|x_n; \Theta_k, Y^{(r)})
\] (A.5)

A.2 Optimization sub-problem for \( \theta_k \)'s

The Lagrangian of optimization problem (3.33) and (3.34) is
\[
L(\phi_m, \gamma) = \lambda \phi_m^T \phi_m + g_m^T \phi_m + \frac{\gamma}{2} (\phi_m^T \phi_m - \epsilon)
\] (A.6)

where \( \gamma \geq 0 \). We then take the derivative with respect to \( \phi_m \) and set the derivative to be zero; that is \( (\lambda + \gamma) \phi_m^* + g_m = 0 \). Therefore we have
\[
\phi_m^* = -\frac{g_m}{\lambda + \gamma}
\] (A.7)

Then we have
\[
L(\phi_m^*, \gamma) = -\frac{g_m^T g_m}{2(\gamma + \lambda)} - \frac{\gamma \epsilon}{2}
\] (A.8)

We have
\[
\frac{\partial}{\partial \gamma} L(\phi_m^*, \gamma) = \frac{g_m^T g_m}{2(\gamma + \lambda)^2} - \frac{\epsilon}{2}
\] (A.9)
When $\gamma \geq \sqrt{\frac{g^T g}{\epsilon}} - \lambda$, we have $\frac{\partial}{\partial \gamma} L(\phi^*_m, \gamma) \leq 0$; that is $L(\phi^*_m, \gamma)$ is monotonic decreasing function of $\gamma$ when $\gamma \geq \sqrt{\frac{g^T g}{\epsilon}} - \lambda$. Therefore, we have

$$\gamma^* = \max(\sqrt{\frac{g^T g}{\epsilon}} - \lambda, 0) \quad (A.10)$$

Consequently we have

$$\phi^*_m = -\frac{g_m}{\max(\sqrt{\frac{g^T g}{\epsilon}}, \lambda)} \quad (A.11)$$

### A.3 Optimization sub-problem for $\Omega_k$'s

Let us first consider the following relaxed optimization problem,

$$\min_{\Phi_m} \frac{\lambda}{2} \langle \Phi_m, \Phi_m \rangle + \langle G_m, \Phi_m \rangle \quad (A.12)$$

s.t.

$$\Phi_m \succeq \alpha(\Omega^{(\tau)}_m)^{-1} - I \quad (A.13)$$

Following [89, 14, 101], we can show that optimal solution for the relaxed problem is

$$\hat{\Phi}_m = C(\tau) + \alpha(\Omega^{(\tau)}_m)^{-1} - I \quad (A.14)$$

where $C(\tau) = \frac{1}{\lambda} G_m + \alpha(\Omega^{(\tau)}_m)^{-1} - I$. Define

$$\rho = \|C(\tau) + \alpha(\Omega^{(\tau)}_m)^{-1} - I\|^2_F \quad (A.15)$$
If \( \rho \leq \epsilon(\tau) \), then we can let \( \Phi^*_m = \hat{\Phi}_m \); if \( \rho > \epsilon(\tau) \), then we can let \( \Phi^*_m = \sqrt{\frac{\epsilon(\tau)}{\rho}} \hat{\Phi}_m \). Therefore, we can let

\[
\Phi^*_m = \sqrt{\min\left(\frac{\epsilon(\tau)}{\rho}, 1\right)} \left\{ C(\tau) + \alpha(\Omega_m)^{-1} - I \right\}
\]  

(A.16)

Since \( \alpha(\Omega_m)^{-1} \preceq I \), and \( \min\left(\frac{\epsilon(\tau)}{\rho}, 1\right) \leq 1 \), we have \( \sqrt{\min\left(\frac{\epsilon(\tau)}{\rho}, 1\right)} [\alpha \Omega_m^{-1} - I] \geq \alpha \Omega_m^{-1} - I \). Therefore, \( \Phi^*_m \geq \alpha \Omega_m^{-1} - I \).

Note that when \( \rho > \epsilon(\tau) \), \( \Phi^*_m \) is not an optimal solution. Since \( \rho > \epsilon(\tau) \), we have

\[
\frac{\lambda}{2} \langle \Phi^*_m, \Phi^*_m \rangle + \langle G_m, \Phi^*_m \rangle \leq \sqrt{\frac{\epsilon(\tau)}{\rho}} \left( \frac{\lambda}{2} \langle \hat{\Phi}_m, \hat{\Phi}_m \rangle + \langle G_m, \hat{\Phi}_m \rangle \right) \quad \text{(A.17)}
\]

\[
\leq 0
\]

(A.18)

That is \( \Phi^*_m \) does reduce the objective function value.
Appendix B

Appendices for Chapter 5

B.1 Augmented Second Moment Matrix

Let $X$ be a random covariate which has probability distribution $p$. So we have

$$
\mu = E_{X \sim p}X
$$

$$
\Sigma = E_{X \sim p}(X - \mu)(X - \mu)^T
$$

$$
\Xi = E_{X \sim p}\left\{ \begin{pmatrix} X \\ 1 \end{pmatrix} (X^T, 1) \right\}
$$

where $\mu$, $\Sigma$ and $\Xi$ are, respectively, the mean, covariance and augmented second moment of $X$. When $\mu$ and $\Sigma$ are finite, we have

$$
\Xi = \begin{pmatrix} \Sigma + \mu \mu^T & \mu \\ \mu^T & 1 \end{pmatrix}
$$
Assuming $\Sigma$ is positive definite, i.e. $\Sigma \succ 0$, we have the inverse of $\Xi$ as follows:

$$\Xi^{-1} = \begin{pmatrix} \Sigma^{-1} & -\Sigma^{-1}\mu \\ -\mu^T\Sigma^{-1} & 1 + \mu^T\Sigma^{-1}\mu \end{pmatrix}$$

## B.2 Proof of (5.29)

We have the following lemma:

**Lemma B.1.** Let $A$ be an $r \times s$ matrix, $(r \geq s)$, and $A^TA = I$. Let $\Lambda$ be a diagonal matrix. Then

$$\xi tr(A^T\Lambda A) + (1 - \xi) tr([A^T\Lambda A]^{-1}) \leq \sum_{i=1}^{s} f_i(\xi)$$

**Proof.** By the Poincaré separation theorem (c.f. [69]), we know the eigenvalues of $A^T\Lambda A$ interlaces with those of $\Lambda$. That is, for each integer $j$, $(1 \leq j \leq s)$, we have $\lambda_j \leq \tau_j \leq \lambda_{j+r-s}$, where $\tau_j$ is the eigenvalue of $A^T\Lambda A$. Then it is obvious that

$$\xi tr(A^T\Lambda A) + (1 - \xi) tr([A^T\Lambda A]^{-1}) = \sum_{i=1}^{s} [\xi \tau_i + (1 - \xi) \frac{1}{\tau_i}] \leq \sum_{i=1}^{s} f_i(\xi)$$

Now let us prove (5.29). Let $U$ be a nonsingular matrix such that
\[ U^T \hat{\Xi}_2 U = I \text{ and } U^T \hat{\Xi}_1 U = \Lambda. \] Then we have

\[
\begin{align*}
\hat{\Pi}_2 &= W^T (U^{-1})^T U^T \hat{\Xi}_2 U U^{-1} W = V^T V \\
\hat{\Pi}_1 &= W^T (U^{-1})^T U^T \hat{\Xi}_1 U U^{-1} W = V^T \Lambda V
\end{align*}
\]

where \( V = U^{-1} W \in \mathbb{R}^{(m+1) \times k} \). Then we can get

\[
C(W; \xi, d) = (1 - \xi) \text{tr}\left[(V^T \Lambda V)^{-1} V^T V\right] + \xi \text{tr}\left[(V^T V)^{-1} V^T \Lambda V\right]
\]

Applying SVD on \( V \), we get \( V = ADB^T \). Here \( A \) and \( B \) are \((m + 1) \times d \) and \( d \times d \) orthogonal matrices, i.e. \( B^T B = I \), \( BB^T = I \), and \( A^T A = I \). \( D \) is a \( d \times d \) diagonal matrix. Therefore we have:

\[
\begin{align*}
\text{tr}\left[(V^T V)^{-1} V^T \Lambda V\right] &= \text{tr}\left[V(V^T V)^{-1} V^T \Lambda\right] \\
&= \text{tr}\left[V(V^T \Lambda V)^{-1} V^T\right] \\
&= \text{tr}\left[AA^T \Lambda\right] = \text{tr}(A^T \Lambda A) \\
&= \text{tr}\left[(A^T \Lambda A)^{-1} A^T\right] \\
&= \text{tr}\left[(A^T \Lambda A)^{-1}\right]
\end{align*}
\]

Thus by Lemma B.1, we know that

\[
C(W; \xi, d) = \text{tr}[\xi A^T \Lambda A + (1 - \xi)(A^T \Lambda A)^{-1}] - d \\
\leq \sum_{i=1}^{d} f_i(\xi) - d
\]
B.3 Proof of (5.41) and (5.42)

Since \( U^T \Xi_i U = \begin{pmatrix} U_1^T \Xi_i U_1 & U_1^T \Xi_i U_2 \\ U_2^T \Xi_i U_1 & U_2^T \Xi_i U_2 \end{pmatrix} \), we have

\[
\frac{N}{2} \begin{pmatrix} U_1^T \Xi_i U_1 & U_1^T \Xi_i U_2 \\ U_2^T \Xi_i U_1 & U_2^T \Xi_i U_2 \end{pmatrix} + \frac{N}{2} \begin{pmatrix} U_1^T \Xi_2 U_1 & U_1^T \Xi_2 U_2 \\ U_2^T \Xi_2 U_1 & U_2^T \Xi_2 U_2 \end{pmatrix} = \begin{pmatrix} \Delta^2 & 0 \\ 0 & 0 \end{pmatrix}
\]

Since \( \Xi_i \)'s are positive semidefinite, i.e. \( \Xi_i \succeq 0 \), we have \( U_1^T \Xi_i U_1 \succeq 0 \) and \( U_2^T \Xi_i U_2 \succeq 0 \). Therefore we must have \( U_1^T \Xi_2 U_2 = 0 \) and \( U_2^T \Xi_2 U_2 = 0 \), otherwise the above equation will be invalid.

B.4 Proof of Theorem 5.1

To prove the theorem, it suffices to show that

\[
\log \frac{p(Y = 1|X = x)}{p(Y = 2|X = x)} = \log \frac{p(Y = 1|Z = z)}{p(Y = 2|Z = z)}
\]

where \( x \) be a realization of random vector \( X \), and \( z = W^T \begin{pmatrix} x \\ 1 \end{pmatrix} \).

When \( m = D + 1 \), the proof is trivial. So without losing generality, let us assume \( m < D + 1 \).
By Lemma 5.1, we have the following

\[(W, W^\perp)^T \Xi_k(W, W^\perp) = \begin{pmatrix} \Lambda_k & 0 \\ 0^T & I \end{pmatrix} \] (B.1)

where

\[\Lambda_1 = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m) \] (B.2)

\[\Lambda_2 = I \] (B.3)

and \(\lambda_i \neq 1, i = 1, 2, \cdots, m.\)

It can be easily verified that we have

\[\Xi_k^{-1} = (W, W^\perp) \begin{pmatrix} \Lambda_k^{-1} & 0 \\ 0^T & I \end{pmatrix} (W, W^\perp)^T \] (B.4)

We then have

\[\Xi_1^{-1} - \Xi_2^{-1} = (W, W^\perp) \begin{pmatrix} \Lambda_1^{-1} - \Lambda_2^{-1} & 0 \\ 0 & 0 \end{pmatrix} (W, W^\perp)^T \] (B.5)

\[= W(\Lambda_1^{-1} - \Lambda_2^{-1})W^T \] (B.6)

We note that the log ratio of posterior probabilities in the original space
can be written as follows

\[
\log \frac{p(Y = 1 | X = x)}{p(Y = 2 | X = x)} = \log \frac{p(X = x | Y = 1)p(Y = 1)}{p(X = x | Y = 2)p(Y = 2)}
\]

\[
= -\frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1) + \frac{1}{2}(x - \mu_2)^T \Sigma_2^{-1}(x - \mu_2) + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

\[
= -\frac{1}{2}(x^T, 1)(\Xi_1^{-1} - \Xi_2^{-1}) \left( \begin{array}{c} x \\ 1 \end{array} \right) + \frac{1}{2} \log \frac{\det \Xi_1^{-1}}{\det \Xi_2^{-1}} + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

Define

\[
\omega_k = W^T \left( \begin{array}{c} \mu_k \\ 1 \end{array} \right) \quad (B.7)
\]

Since \( Z = W^T \left( \begin{array}{c} X \\ 1 \end{array} \right) \), we have

\[
Z \mid Y = k \sim \mathcal{N}(\omega_k, \Lambda_k - \omega_k \omega_k^T)
\]

We can write the ASM matrix in the affine subspace spanned by the columns of \( W \) as following

\[
\Xi_k^w = \begin{pmatrix} \Lambda_k & \omega_k \\ \omega_k^T & 1 \end{pmatrix}
\]

\quad (B.9)
In Appendix B.5, we show that

\[ \Xi_k^w \succ 0 \] (B.10)

\[ \Lambda_k - \omega_k \omega_k^T \succ 0 \] (B.11)

Furthermore, we show in Appendix B.5 that

\[ \Lambda_1^{-1}\omega_1 = \Lambda_2^{-1}\omega_2 \] (B.12)

\[ \omega_1^T \Lambda_1^{-1}\omega_1 = \omega_2^T \Lambda_2^{-1}\omega_2, \] (B.13)

\[ \omega_k^T \Lambda_k^{-1}\omega_k < 1 \] (B.14)

It can be verified that, when \( \omega_k^T \Lambda_k^{-1}\omega_k < 1 \), we have

\[
\left( \Xi_k^w \right)^{-1} = \begin{pmatrix} \Lambda_k^{-1} & 0 \\ 0^T & 0 \end{pmatrix} + \frac{1}{1 - \omega_k^T \Lambda_k^{-1}\omega_k} \begin{pmatrix} -\Lambda_k^{-1}\omega_k \\ 1 \end{pmatrix} \begin{pmatrix} -\omega_k^T \Lambda_k^{-1} \\ 1 \end{pmatrix} \] (B.15)

By Equations (B.12) and (B.13), we have

\[
\frac{1}{1 - \omega_1^T \Lambda_1^{-1}\omega_1} (1 - z^T \Lambda_1^{-1}\omega_1)^2 = \frac{1}{1 - \omega_2^T \Lambda_2^{-1}\omega_2} (1 - z^T \Lambda_2^{-1}\omega_2)^2 \] (B.16)

Therefore we have

\[
\begin{pmatrix} z & 1 \end{pmatrix} \left[ \left( \Xi_1^w \right)^{-1} - \left( \Xi_2^w \right)^{-1} \right] \begin{pmatrix} z \\ 1 \end{pmatrix} = z^T (\Lambda_1^{-1} - \Lambda_2^{-1}) z \] (B.17)
Note that
\[(\Lambda_k - \omega_k \omega_k^T)^{-1} = \Lambda_k^{-1} + \frac{1}{1 - \omega_k^T \Lambda_k^{-1} \omega_k} \Lambda_k^{-1} \omega_k \omega_k^T \Lambda_k^{-1} \]

Therefore we have
\[
\det(\Lambda_k - \omega_k \omega_k^T)^{-1} = \det\{\Lambda_k^{-1} + \frac{1}{1 - \omega_k^T \Lambda_k^{-1} \omega_k} \Lambda_k^{-1} \omega_k \omega_k^T \Lambda_k^{-1}\}\]
\[
= \frac{1}{1 - \omega_k^T \Lambda_k^{-1} \omega_k} \det \Lambda_k^{-1} \tag{B.19}
\]

Then we have the log ratio of posterior probabilities in the subspace spanned by the columns of \(W\) as follows

\[
\log \frac{p(Y = 1|Z = z)}{p(Y = 2|Z = z)} = \log \frac{p(Y = 1|Z = z)}{p(Y = 2|Z = z)}
= \frac{1}{2} z^T (\Lambda_1^{-1} - \Lambda_2^{-1}) z - \frac{1}{2} \log \frac{\det \Lambda_1}{\det \Lambda_2} + \frac{1}{2} \log \frac{\det (\Lambda_1 - \omega_1 \omega_1^T)^{-1}}{\det (\Lambda_2 - \omega_2 \omega_2^T)^{-1}} + \log \frac{p(Y = 1)}{p(Y = 2)}
\]

Therefore we have
\[
\log \frac{p(Y = 1|X = x)}{p(Y = 2|X = x)} = \log \frac{p(Y = 1|Z = z)}{p(Y = 2|Z = z)}
\]
B.5 Proof of (B.10) – (B.14)

Define

\[ \omega_k = W^T \begin{pmatrix} \mu_k \\ 1 \end{pmatrix} \]  
\[ \tilde{\omega}_k = W^T \perp \begin{pmatrix} \mu_k \\ 1 \end{pmatrix} \]  

We note that

\[ \begin{pmatrix} \mu_k^T \\ 1 \end{pmatrix} \Xi_k^{-1} \begin{pmatrix} \mu_k \\ 1 \end{pmatrix} = 1 \]  

Furthermore, we have

\[ \begin{pmatrix} \mu_k^T \\ 1 \end{pmatrix} \Xi_k^{-1} \begin{pmatrix} \mu_k \\ 1 \end{pmatrix} = (\omega_k^T, \tilde{\omega}_k^T) \begin{pmatrix} \Lambda_k^{-1} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \omega_k \\ \tilde{\omega}_k \end{pmatrix} = \omega_k^T \Lambda_k^{-1} \omega_k + \tilde{\omega}_k^T \tilde{\omega}_k \]

That is we have

\[ \omega_k^T \Lambda_k^{-1} \omega_k = 1 - \tilde{\omega}_k^T \tilde{\omega}_k \]
APPENDIX B. APPENDICES FOR CHAPTER 5

Since $\Xi_1 W \Lambda_1^{-1} = \Xi_2 W \Lambda_2^{-1}$, we have

$$\Lambda_1^{-1} \omega_1 = \Lambda_2^{-1} \omega_2 \quad (B.24)$$

Similarly, we have

$$\ddot{\omega}_1 = \ddot{\omega}_2 \quad (B.25)$$

Therefore, by (B.23) and (B.32), we have

$$\omega_1^T \Lambda_1^{-1} \omega_1 = \omega_2^T \Lambda_2^{-1} \omega_2 \quad (B.26)$$

We prove (B.12) and (B.13).

The following lemma about Schur complement and matrix positive (semi)definiteness can be found in [13]:

**Lemma B.2.** Consider a symmetric block matrix

$$X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

where $A = A^T$ and $C = C^T$. If $A$ is invertible, then matrix $S_A = C - B^T A^{-1} B$ is called the Schur complement of $A$ in $X$. Then we have the following

- $X \succ 0$ if and only if $A \succ 0$ and $S_A \succ 0$. 
• If $A \succ 0$, then $X \succeq 0$ if and only if $S_A \succeq 0$.

If $\tilde{\omega}_k \neq 0$, by Equation (B.23), we have

$$\omega_k^T \Lambda_k^{-1} \omega_k < 1 \quad \text{(B.27)}$$

Note we have $\Lambda_k \succ 0$. Then by Lemma B.2, we have

$$\Lambda_k \succ 0 \quad 1 - \omega_k^T \Lambda_k^{-1} \omega_k > 0 \quad \Rightarrow \Xi_k^w = \begin{pmatrix} \Lambda_k & \omega_k \\ \omega_k^T & 1 \end{pmatrix} \succ 0 \quad \text{(B.28)}$$

and

$$1 > 0 \begin{pmatrix} \Lambda_k & \omega_k \\ \omega_k^T & 1 \end{pmatrix} \succ 0 \quad \Rightarrow \Lambda_k - \omega_k \omega_k^T \succ 0 \quad \text{(B.29)}$$

If $\tilde{\omega}_k \neq 0$, we then prove (B.10), (B.11) and (B.14).

Now let us prove the $\tilde{\omega}_k \neq 0$. Assume $\tilde{\omega}_k = 0$, we have $\omega_k^T \Lambda_k^{-1} \omega_k = 1$.

Therefore by Lemma B.2 we have

$$\Lambda_k - \omega_k \omega_k^T \succeq 0 \quad \text{(B.30)}$$

Note that $\Lambda_1 = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m)$ and $\Lambda_2 = I$. We have $\lambda_i \neq 1$, $i = 1, 2, \cdots, m$. Without losing generality, let us assume $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$. 

APPENDIX B. APPENDICES FOR CHAPTER 5

$\lambda_m$. We note that since $\mathbf{I} - \lambda_1 \mathbf{A}_1^{-1} \succeq \mathbf{0}$, we have

$$\omega_1^T \omega_1 \leq \lambda_1 \omega_1^T \mathbf{A}_1^{-1} \omega_1 = \lambda_1 \quad (B.31)$$

The minimum eigenvalue of $\mathbf{A}_1 - \omega_1 \omega_1^T$ must be 0. It is well known that a lower bound of the minimum eigenvalue of $\mathbf{A}_1 - \omega_1 \omega_1^T$ is $\lambda_1 - \omega_1^T \omega_1$. Therefore we have

$$\lambda_1 - \omega_1^T \omega_1 \leq 0 \quad (B.32)$$

By (B.31) and (B.32), we must have

$$\omega_1^T \omega_1 = \lambda_1 \quad (B.33)$$

Thus we have

$$\omega_1^T (\mathbf{I} - \lambda_1 \mathbf{A}_1^{-1}) \omega_1 = 0 \quad (B.34)$$

Therefore $\omega_1$ must lie in the null space of $\mathbf{I} - \lambda_1 \mathbf{A}_1^{-1}$; that is

$$(\mathbf{I} - \lambda_1 \mathbf{A}_1^{-1}) \omega_1 = \mathbf{0} \Rightarrow \omega_1 = \lambda_1 \mathbf{A}_1^{-1} \omega_1 \quad (B.35)$$

Since $\omega_2 = \mathbf{A}_1^{-1} \omega_1$, we have

$$\omega_1 = \lambda_1 \omega_2 \quad (B.36)$$
APPENDIX B. APPENDICES FOR CHAPTER 5

Therefore we have

\[
\begin{pmatrix}
W^T \\
W_\perp^T
\end{pmatrix}
\begin{pmatrix}
\mu_2 \\
1
\end{pmatrix}
- \lambda_1
\begin{pmatrix}
\mu_1 \\
1
\end{pmatrix}
= 0
\] (B.37)

To satisfy (B.37), we must have

\[
\begin{pmatrix}
\mu_2 \\
1
\end{pmatrix}
- \lambda_1
\begin{pmatrix}
\mu_1 \\
1
\end{pmatrix}
= 0
\] (B.38)

Therefore we must have \( \lambda_1 = 1 \), which is a contradictory. So \( \tilde{\omega}_k \neq 0 \).
Appendix C

Software

For the hybrid generative/discriminative QDA method described in Chapter 3, we developed C codes that are based on BLAS and LAPACK [2]. We also provided Matlab and R interfaces for the C codes. Our experiments in Chapter 3 are based on the Matlab interfaces.

C.1 Compile Codes

C.1.1 Matlab

We tested our codes with Matlab interfaces under Linux and Windows system. To compile Matlab mex files under a Linux system, you need do the following:

1. At file model.h,
2. Run file `compile_mex.sh`.

To compile Matlab mex files under a Windows system, you need do the following

1. At file `model.h`,

```c
#undef _UNIX_
#define _MATLAB_
```

2. Run file `compile_mex.m` under Matlab.

C.1.2 R

We tested our codes with R interfaces under Linux system. To compile R interfaces under a Linux system, you need do the following

1. At file `model.h`,

```c
#define _UNIX_
#undef _MATLAB_
```

2. Run the following commend to build a dynamic library that you can load into R
C.2 Usage

In this section, we explain how to use the codes to train a QDA via hybrid generative/discriminative learning approach described in Chapter 3. For Matlab, you can check the corresponding .m files. For R, you can check hybrid_qda.R. For demo files, you can check demo.m for Matlab and demo.R for R.

C.2.1 set_mod_opt

In this function, we define default values for training a hybrid QDA. We used these default values in the numerical experiments in Chapter 3. You can adjust them if necessary.

- Options.seed: random seed. Default value is 0.
- Options.num_thread: number of threads. This field is saved for future version, in which we plan to use OPENMP to parallelize the codes;
• Options.early_stop: a flag for early stopping: 1 for early stopping, 0 for no early stopping. The default value is 1.

• Options.kFold: K-fold cross validation. The default value is 10.

• Options.init.methods: methods to initialize $\mu_k$'s and $\Sigma_k^{-1}$'s. We initialize $\mu_k$ as sample mean, i.e.

$$\mu_k = \frac{1}{\bar{N}_k} \sum_{n \in \mathcal{C}_k} x_k$$

We initialize $\Sigma_k^{-1}$'s as one of the following

$$\Sigma_k^{-1} = \begin{cases} \frac{1}{\text{tr}(\Sigma_k)} I + \alpha I, & \text{if Options.init.methods}=1 \\ \text{diag}(\text{diag}(\hat{\Sigma}_0)) + \alpha I, & \text{if Options.init.methods}=2 \\ \text{diag}(\text{diag}(\hat{\Sigma}_k)) + \alpha I, & \text{if Options.init.methods}=3 \end{cases}$$

where $\hat{\Sigma}_0$ and $\hat{\Sigma}_k$'s are the pooled-in sample covariance matrix and sample covariance matrices for class $k$, respectively; $\Box^+$ is the pseudo-inverse of a matrix $\Box$. In the numerical experiments of Chapter 3, we use Options.init.methods==3.

• Options.init.method: a scale value 1, 2 or 3. This field has to be set when calling train_model.

• Options.init.prior: one of the following methods to initialize priors:
APPENDIX C. SOFTWARE

− 1: uniform priors;

− 2: empirical priors;

− 3: the Laplace smoothing of empirical priors \((l = 1)\), see for example [97].

• Options.betas: a list of \(\beta\)'s for hybrid generative/discriminative QDA.

The default values are \(\beta = 0.0, 0.1, 0.2, ..., 0.9, 1.0\).

• Options.beta: a scale value \(\beta \in [0, 1]\). This field has to be set when calling \texttt{train\_model}.

• Options.alpha: coefficient for constraints \(\Omega_k \succeq \alpha I, k = 1, 2, ..., K\).

The default value is \(\alpha = 10^{-6}\).

• Options.max\_iter: maximum number of iterations of outer loop. The default value is 3000.

• Options.tol: tolerance of relative change of objective function values between iterations. The default value is \(10^{-6}\).

• Options for optimizing priors \(\Upsilon\):

  − Options.prior.max\_iter: maximum number of iterations for optimizing priors. The default value is 50.
– Options.prior.tolf: tolerance of relative change of objective function for optimizing priors between iterations. The default value is $10^{-6}$.

– Options.prior.tolx: tolerance of the absolute change of priors between iterations. The default value is $10^{-6}$.

– Options.prior.if_optimize: a flag for optimizing priors: 1 for optimizing prior, 0 for not optimizing prior. The default value is to optimize priors.

- Options for optimizing $\theta_k$’s, $k = 1, 2, ..., K$:

  – Options.theta.max_iter: maximum number of iterations for optimizing $\theta_k$. The default value is 10.

  – Options.theta.tol_g: tolerance of gradient for optimizing $\theta_k$. The default value is $10^{-5}$.

  – Options.theta.tol_delta: tolerance of trust region radius. The default value is $10^{-8}$.

  – Options.theta.tol_x: tolerance of absolute change of $\theta_k$ between iterations. The default value is $10^{-5}$.

  – Options.theta.tol_f: tolerance of relative change of the objective function.
function for optimizing $\theta_k$ between iterations. The default value is $10^{-5}$.

- Options for optimizing $\Omega_k$’s, $k = 1, 2, ..., K$
  
  - Options.omega.max_iter: maximum number of iterations for optimizing $\Omega_k$. The default value is 10.
  
  - Options.omega.tol_g: tolerance of gradient for optimizing $\Omega_k$. The default value is $10^{-5}$.
  
  - Options.omega.tol_delta: tolerance of trust region radius. The default value is $10^{-8}$.
  
  - Options.omega.tol_x: tolerance of absolute change of $\Omega_k$ between iterations. The default value is $10^{-5}$.
  
  - Options.omega.tol_f: tolerance of relative change of the objective function for optimizing $\Omega_k$ between iterations. The default value is $10^{-5}$.

- Options for trust region methods: we take the values recommended in [137].
  
  - Options.trust.tao0 = $10^{-8}$
  
  - Options.trust.tao1 = 2
– Options.trust.tao2 = 0.25
– Options.trust.tao3 = 0.25
– Options.trust.tao4 = 0.5

C.2.2 \texttt{train\_model\_cv}

Function \texttt{train\_model\_cv} is used when you want to train a hybrid QDA via cross validation, for example, to determine $\beta$, initial points, and/or the early stopping step. The inputs are:

- X: training observations, $N \times D$ double matrix.
- Y: training labels, $N \times 1$ double matrix
- Options: training options, see \texttt{set\_mod\_opt}.

The output is a trained hybrid QDA $\texttt{Para}$, which has the following fields

- $\texttt{Para.init\_value}$: initial objective function value;
- $\texttt{Para.value}$: final objective function value;
- $\texttt{Para.log\_prior}$: logarithm of priors;
- $\texttt{Para.mu}$: $\mu_k$'s, $k = 1, 2, ..., K$;
- $\texttt{Para.sigma}$: $\Sigma_k$'s, $k = 1, 2, ..., K$;
APPENDIX C. SOFTWARE

- Para.theta: \( \theta_k = \Sigma_k^{-1} \mu_k, \ k = 1, 2, ..., K; \)

- Para.omega: \( \Sigma_k^{-1}, \ k = 1, 2, ..., K; \)

- Para.logdet.omega: \( \log \det \Sigma_k^{-1}, \ k = 1, 2, ..., K; \)

- Para.beta: \( \beta \) for the hybrid generative/discriminative learning;

- Para.alpha: \( \alpha \) for semi-positive definite constraints;

- Para.init.method: initialization methods for \( \mu_k \)'s and \( \Sigma_k \)'s;

- Para.iter: number of iterations in the MM algorithm;

- Para.perf: performance metrics of each iteration in the MM algorithm.

C.2.3 \texttt{train\_model}

Function \texttt{train\_model} is used when you want to train a hybrid QDA with given \( \beta \), initial point, and/or maximum iteration numbers. The inputs are

- \texttt{TrainX}: training observations, \( N_1 \times D \) double matrix;

- \texttt{TrainY}: training labels, \( N_1 \times 1 \) double matrix;

- \texttt{ValidX}: validation observations, \( N_2 \times D \) double matrix;

- \texttt{ValidY}: validation labels, \( N_2 \times 1 \) double matrix;
• Options: training options, see `set_mod_opt`. You may need to set up
the following three additional fields,

  – Options.init_method: initial point,
  
  – Options.beta: $\beta$,
  
  – Options.yPool: possible class labels.

The output is a trained hybrid QDA, (see `train_model_cv`).

C.2.4 test_model

Given new observations, function `test_model` is used for predicting their class
labels and calculating their posterior distributions.

The inputs are:

• X: observations, $N \times D$ double matrix;

• Para: a trained hybrid QDA, see `train_model_cv` and `train_model`.

The outputs are:

• Y: predicted class labels based on $0 - 1$ loss;

• P: posterior distributions for given observations X.
Bibliography


BIBLIOGRAPHY


