DISCRIMINANT ANALYSIS BASED FEATURE EXTRACTION FOR PATTERN RECOGNITION

WEI WU

A THESIS

IN

THE DEPARTMENT

OF

ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

SEPTEMBER 2009

Your file Votre référence
ISBN: 978-0-494-63403-5
Our file Notre référence
ISBN: 978-0-494-63403-5

ABSTRACT

Discriminant Analysis Based Feature Extraction for Pattern Recognition

Wei Wu, Ph.D.
Concordia University, 2009

Fisher's linear discriminant analysis (FLDA) has been widely used in pattern recognition applications. However, this method cannot be applied for solving the pattern recognition problems if the within-class scatter matrix is singular, a condition that occurs when the number of the samples is small relative to the dimension of the samples. This problem is commonly known as the small sample size (SSS) problem and many of the FLDA variants proposed in the past to deal with this problem suffer from excessive computational load because of the high dimensionality of patterns or lose some useful discriminant information. This study is concerned with developing efficient techniques for discriminant analysis of patterns while at the same time overcoming the small sample size problem. With this objective in mind, the work of this research is divided into two parts.

In part 1, a technique by solving the problem of generalized singular value decomposition (GSVD) through eigen-decomposition is developed for linear discriminant analysis (LDA). The resulting algorithm referred to as modified GSVD-LDA (MGSVD-LDA) algorithm is thus devoid of the singularity problem of the scatter matrices of the traditional LDA methods. A theorem enunciating certain properties of the discriminant

subspace derived by the proposed GSVD-based algorithms is established. It is shown that if the samples of a dataset are linearly independent, then the samples belonging to different classes are linearly separable in the derived discriminant subspace; and thus, the proposed MGSVD-LDA algorithm effectively captures the class structure of datasets with linearly independent samples.

Inspired by the results of this theorem that essentially establishes a class separability of linearly independent samples in a specific discriminant subspace, in part 2, a new systematic framework for the pattern recognition of linearly independent samples is developed. Within this framework, a discriminant model, in which the samples of the individual classes of the dataset lie on parallel hyperplanes and project to single distinct points of a discriminant subspace of the underlying input space, is shown to exist. Based on this model, a number of algorithms that are devoid of the SSS problem are developed to obtain this discriminant subspace for datasets with linearly independent samples.

For the discriminant analysis of datasets for which the samples are not linearly independent, some of the linear algorithms developed in this thesis are also kernelized.

Extensive experiments are conducted throughout this investigation in order to demonstrate the validity and effectiveness of the ideas developed in this study. It is shown through simulation results that the linear and nonlinear algorithms for discriminant analysis developed in this thesis provide superior performance in terms of the recognition accuracy and computational complexity.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deep gratitude to my supervisor, Professor M. Omair Ahmad, for his constant support, encouragement, patience, and invaluable guidance during this research. I am grateful to him for spending many hours with me in discussion about my research. The useful suggestions and comments provided by the members of the supervisory committee, Dr. M. N. S. Swamy, and Dr. Weiping Zhu, the committee member, external to the program, Dr. Terry Fancott, and the external examiner, Dr. Wu-Sheng Lu, as well as those of the anonymous reviewers of my journal papers are also greatly appreciated.

My sincere thanks go to my parents, family members, and relatives for their support and encouragement during my research. Special thanks to my husband, Weimin Zhu, whose patience, love, and unlimited support have made the successful completion of this thesis a reality. I would like also to mention my little son, Wuhua, whose beautiful face always refreshes my mind.

I am indebted to my friend, Mr. Jijun He, for fruitful discussions, suggestions and encouragement during the course of my doctoral study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SSS: Small sample size

LDA: Linear discriminant analysis

FLDA: Fisher's linear discriminant analysis

RLDA: Regularized linear discriminant analysis

PCA: Principal component analysis

SVD: Singular value decomposition

GSVD: Generalized singular value decomposition

MGSVD: Modified generalized singular value decomposition

KDA: Kernelized discriminant analysis

KPCA: Kernelized principal component analysis

KFD: Kernel Fisher discriminant

GDA: Generalized kernel discriminant analysis

DS: Discriminant subspace

LS: Linearly separable

NLS: Nonlinearly separable

PR: Pattern recognition

LIST OF ACRONYMS

$\mathbf{A}_w$: Within-class scatter matrix of linearly independent samples

$\mathbf{A}_b$: Between-class scatter matrix of linearly independent samples

$\mathbf{A}_c$: Total scatter matrix of linearly independent samples

$\mathbf{X}$: Input matrix consists of m-dimensional samples $\mathbf{x}_{ij}$

m: Dimensionality of samples

n: Number of input samples

N: Number of classes

i: Class number

n_i: Number of samples of ith class

$\mathbf{x}_l$: Sample vector

$\mathbf{x}_{ij}$: jth sample in ith class,

$\mathbf{c}$: Global centroid vector

$\mathbf{c}^{(i)}$: Centroid vector of the ith class

$\mathbf{S}_t$: Total scatter matrix

$\mathbf{S}_w$: Within-class scatter matrix

$\mathbf{S}_b$: Between-class scatter matrix

λ_i: Eigenvalues

$\mathbf{g}_i$: Eigenvectors

$\mathbf{G}$: Transformation matrix

$\Re^m$: m-dimensional input sample space

$\mathbf{\Gamma}$: Kernel feature space

$\mathbf{K}$: Kernel matrix

$k(\bullet)$: Kernel function

$\mathbf{I}$: Identity matrix

$\boldsymbol{\psi}_l$: Mapped sample in kernel feature space

$\boldsymbol{\psi}_{ij}$: jth mapped sample of ith class in kernel feature space

$\boldsymbol{\psi}$: Global centroid of mapped samples in feature space

$\boldsymbol{\psi}^{(i)}$: Centroid vector of ith mapped class

$\tilde{\mathbf{S}}_w$: Within-class scatter matrix in kernel feature space

$\tilde{\mathbf{S}}_b$: Between-class scatter matrix in kernel feature space

$\tilde{\mathbf{S}}_t$: Total scatter matrix in kernel feature space

$\tilde{\mathbf{G}}$: Transformation matrix in kernel feature space

$\mathbb{R}(\cdot)$: Range of associated matrix

$\mathbb{N}(\cdot)$: Null space of associated matrix

$\langle \bullet \rangle$: Inner product of two vectors

$|\bullet|$: Determinant of the associated square matrix

Chapter 1

Introduction

1.1 Background

Pattern recognition is the discipline that studies how a machine can observe the environment, learn to distinguish patterns (samples), and make reasonable decisions on the classes of new patterns [1]. Pattern is a quantitative or structural description of an object or some other entity of interest. Depending on the applications, patterns can be handwritten cursive words, speech signals, odor signals, fingerprint images, animal footprints, human faces or any type of measurements that need to be classified.

One of the widely used pattern recognition approaches is the statistical pattern recognition. In the statistical approach, each pattern is represented in terms of m features. Depending on the measurements of an object, features in a pattern can be either discrete numbers or real continuous values. The requirement on features is that the features can reflect the characteristics of desired objects and differ from those other objects to the largest extent. For example, a face image, being a $d \times d$ array of 8 bit intensity values, can be represented as a vector of dimension $m = d^2$. Thus, each pattern can be viewed as an m-dimensional feature vector or a point in an m-dimensional space, that is,

$$\mathbf{x} = [x_1, x_2, \cdots, x_m]^T$$

where $x_1, x_2, \cdots, x_m$ are the features. This space is called *sample feature, feature space* or *input space*. For example, an image of size 256×256 becomes a 65536-dimensional vector or equivalently, a point in a 65536-dimensional space.

The procedure of a statistical pattern recognition system has two main steps: training (learning) and classification (testing). In the training step, feature extraction creates a set of representative features based on transformations or combinations of the given patterns. The set of representative features is considered to be the most important and effective attributes in distinguishing the patterns from different classes. The classification step is to assign a class label to each new pattern.

Patterns, being similar in overall configuration, are not randomly distributed in the input space and thus can be described by a relatively low-dimensional subspace. The idea is to find appropriate features for representing the samples with enhanced discriminatory power for the purpose of recognition. This process is known as *feature extraction*. A commonly used feature extraction technique is to transform the original sample space into a lower-dimensional discriminant subspace, in which a transformed sample of the dataset is easily distinguished. The objective is to find a set of transformation vectors spanning over the discriminant subspace, on which the projections of the samples within each class condense into a compact and separated region.

1.2 Motivation

Two classical linear feature extractors are principal component analysis (PCA) [2] - [3], [7] - [8] and Fisher's linear discriminant analysis (FLDA) [4], [5], [6]. Both these methods extract features by projecting the original sample vectors onto a new feature

space through a linear transformation matrix. However, the goal of optimizing the transformation matrix in the two methods is different. In PCA, the transformation matrix is optimized by finding the largest variations in the original feature space [2] - [3], [7] - [8]. On the other hand, in FLDA, the ratio of the between-class and within-class variations is maximized by projecting the original features to a subspace [4], [5], [6]. PCA is effective in restructuring the dataset, but it is weak in providing the class structure. FLDA formulates the class boundaries by finding a discriminant subspace in which different classes occupy compact and disjoint regions using the Fisher criterion [5], [6]

$$\mathbf{G}_{FLDA} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T \mathbf{S}_b \mathbf{G}|}{|\mathbf{G}^T \mathbf{S}_w \mathbf{G}|}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are, respectively, the between-class and within-class scatter matrices, $\mathbf{G}$ is the transformation matrix whose columns are the projection vectors that span the discriminant subspace, and $|\bullet|$ denotes the determinant of the associated matrix. The solution to this maximization problem is the set of eigenvectors corresponding to the non-zero eigenvalues of the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$.

The Fisher linear discriminant analysis cannot be applied to solve pattern recognition problems if the within-class scatter matrix is singular, a situation that occurs when the number of the samples is small relative to the dimension of the samples. This is so-called the small sample size (SSS) problem [4]. Small sample size data with high dimensionality are often encountered in real applications, such as in human face recognition. Many FLDA variants have been proposed to address this singularity issue [9] - [46]. Tian et al. [15] have used the pseudoinverse method by replacing $\mathbf{S}_w^{-1}$ with its pseudoinverse. Cheng

et al. [16] have proposed a rank decomposition method based on successive eigen-decomposition of the total scatter matrix $\mathbf{S}_t$ and the between-class scatter matrix $\mathbf{S}_b$. However, the above methods are typically computationally expensive since the scatter matrices are very large [17]. In [18], a two-stage FLDA method has been introduced, in which a principal component analysis is carried out for dimension reduction prior to applying the Fisher criterion. However, this dimension reduction step eliminates some useful discriminant information, since some of the eigenvectors of the total scatter matrix are discarded in order to make $\mathbf{S}_w$ non-singular [19] – [24]. In the direct LDA (D-LDA) method [19], the null space of $\mathbf{S}_b$ is first removed, and then the discriminant vectors in the range of $\mathbf{S}_b$ are found by simultaneously diagonalizing $\mathbf{S}_b$ and $\mathbf{S}_w$ [4]. A drawback of this method is that some significant discriminant information in the null space of $\mathbf{S}_w$ gets eliminated due to the removal of the null space of $\mathbf{S}_b$ [20], [22] - [24]. In the regularized FLDA (RLDA) [21], [29] - [33], the singularity problem is solved by adding a perturbation to the scatter matrix. The optimal perturbation parameter is normally estimated adaptively from the training samples through cross-validation, a process which is very time consuming. Some FLDA variants have attempted to overcome the SSS problem by using the generalized singular value decomposition (LDA/GSVD) [34], [35]. However, these methods suffer from excessive computational load because of the large dimension of the samples [36]. Chen et al. [41] proposed the null space method based on the modified FLDA criterion

$$\mathbf{G}_{MFLDA} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T \mathbf{S}_b \mathbf{G}|}{|\mathbf{G}^T \mathbf{S}_t \mathbf{G}|}$$

where $\mathbf{S}_t$ is the total scatter matrix [42], [43]. However, the authors did not give an efficient algorithm for applying this method to solve the singularity problem of the Fisher criterion [47].

Since the linear feature extraction methods cannot capture the nonlinear class boundaries, which exist in many patterns and affect the recognition accuracy of the patterns, kernel machines [49] - [56] are used to map the patterns into a high-dimensional kernel feature space where the patterns are linearly separable, and thus, the linear feature extraction techniques can be applied in the mapped space. The integration of the kernel machine with a linear discriminant method provides a nonlinear algorithm with improved recognition accuracy [57] - [77]. However, nonlinear algorithms also suffer from the same problems as that inherent in the corresponding linear versions. Hence, the choice of a good linear algorithm is crucial to obtaining an efficient kernelized algorithm.

From the foregoing discussion, it is clear that the existing discriminant analysis techniques, in general, suffer from the excessive computational load in dealing with the high dimensionality of patterns or lose some useful discriminant information in order to overcome the singularity problem associated with the Fisher criterion. It is, therefore, necessary to conduct an in-depth study of the mechanism of the discriminant analysis leading to designs of efficient low computational complexity algorithms for feature extraction without having to deal with the SSS problem.

1.3 Scope of the Thesis

The objective of this research is to devise efficient techniques for discriminant analysis of patterns and to apply them for developing feature extraction algorithms that

are devoid of the small sample size problem. With this unifying theme, the work of this study is carried out in two parts.

In part 1, a low-complexity algorithm that overcomes the singularity problem of the scatter matrices of the traditional FLDA methods is developed for linear discriminant analysis (LDA) by solving the problem of generalized singular value decomposition (GSVD) through eigen-decomposition. A theorem providing the distance between samples in the discriminant subspace derived from this GSVD-based algorithm is established to address the class structure and separability of linearly independent samples.

In part 2, a new systematic framework for the feature extraction of datasets with linearly independent samples is developed. Within this framework, a discriminant model is first established. It is shown that if the samples of a dataset are linearly independent, then the samples of the individual classes of the dataset lie on parallel hyperplanes and the samples of the entire class can be projected onto a unique point of a discriminant subspace of the underlying input space. A number of algorithms that are devoid of the SSS problem are developed to determine the discriminant subspace for datasets with linearly independent samples.

1.4 Organization of the Thesis

The thesis is organized as follows.

In Chapter 2, a brief review of the linear and nonlinear techniques for feature extraction is presented. This review is intended to facilitate the understanding of the development of the techniques for feature extraction presented in the thesis. This chapter

also includes some preliminaries on the commonly used techniques for dealing with the singularity problem associated with the Fisher criterion.

In Chapter 3, a new technique [37], referred to as the MGSVD-LDA algorithm that can effectively deal with the SSS problem, is presented by applying eigen-decomposition to solve the problem of the generalized singular value decomposition. A scheme is developed to kernelize the proposed linear algorithm to deal with the discriminant analysis of datasets in which samples are not linearly separable and a direct application of a linear algorithm fails to separate the classes of the datasets. In order to improve the recognition accuracy of the proposed linear and nonlinear algorithms further, a method is devised to take care of the over-fitting problem by orthogonalizing the basis of the discriminative subspace [38], [39]. Extensive simulation results are also presented in this chapter to demonstrate the effectiveness of the proposed linear, kernelized and orthogonalized algorithms and compare their performance with that of other existing algorithms.

In Chapter 4, a theorem that establishes the class structure and separability of linearly independent samples in the discriminant subspace derived from the proposed MGSVD-LDA algorithm is developed. This theorem is then used to develop a method to estimate the numerical errors of the proposed algorithms and also to control the kernel parameters to maximize the recognition accuracy of the kernelized algorithm.

In Chapter 5, a systematic framework for the pattern recognition of datasets with linearly independent samples is developed [40]. A discriminant model, in which the samples of the individual classes of a dataset lie on parallel hyperplanes and project to single distinct points of the discriminant subspace of the underlying input space, is shown

to exist. In conformity with this model, three new algorithms are developed to obtain the discriminant subspace for datasets with linearly independent samples. A kernelized algorithm is also developed for the discriminant analysis of datasets for which the samples are not linearly independent. Simulation results are also provided in this chapter to examine the validity of the proposed discriminant model and to demonstrate the effectiveness of the linear and nonlinear algorithms designed based on the proposed model.

Finally, in Chapter 6, concluding remarks highlighting the contributions of the thesis and suggestions for some further investigation of the topics related to the work of this thesis are provided.

Chapter 2

Literature Review

2.1 Introduction

Feature extraction is one of the central and critical issues to solving pattern recognition problems. It is the process of generating a representative set of data from the measurements of an object, which are considered to be the most important and effective descriptors or characteristic attributes in distinguishing the object to belong to one class from another class. The main objective here is to find techniques that can introduce low-dimensional feature representation of objects, i.e., reduce the amount of data needed in representing objects, while achieving the best discriminatory power.

Feature extraction techniques, in general, can be classified into two categories: linear and nonlinear methods [1], [4]. Linear methods can be applied when the samples are linearly separable. Two subsets U and V of $\Re^m$ are said to be linearly separable (LS) if there exists a hyperplane P in $\Re^m$ such that the samples of U and those of V lie on its opposite sides. On the contrary, if they are nonlinearly separable (NLS), then a single hyperplane cannot be used to classify them [84], [87]. Figure 2.1 shows an example of LS and NLS set of sample points. In some cases, linear methods may not provide a sufficient discriminating power for nonlinearly separable samples. Nonlinear techniques, such as

kernel methods [49] - [56], can be used to transform the input samples into a higher dimensional kernel feature space by a nonlinear kernel mapping where samples become linearly separable so that the linear discriminant analysis can be applied in that high dimensional kernel feature space.

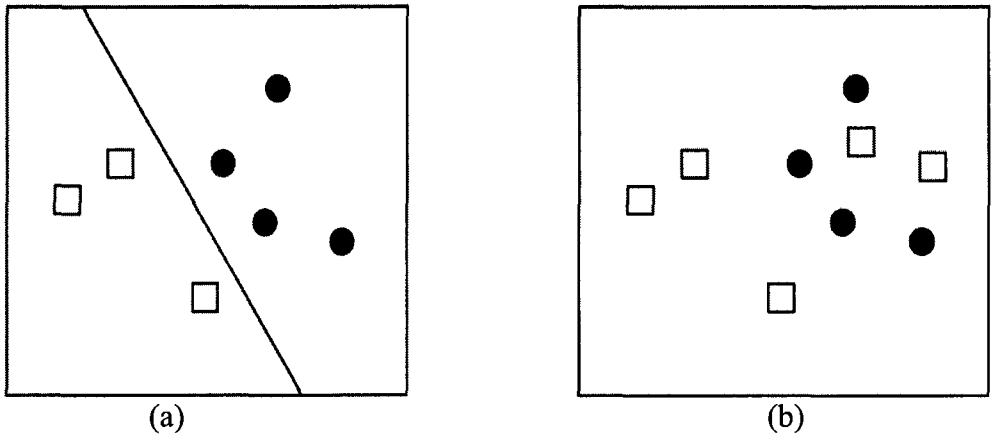

Figure 2.1: Examples of (a) LS set of sample points and (b) NLS set of sample points

In this chapter, the linear feature extractors --- principle component analysis (PCA) [2], [3], [7], [8], Fisher's linear discriminant analysis (FLDA) [5], [6] and some of the representative FLDA variants --- are reviewed. Techniques to deal with the singularity problem of the scatter matrices of the traditional FLDA are explained in detail. Some nonlinear discriminant methods that can effectively deal with nonlinearly distributed patterns are also briefly discussed.

2.2 Linear Feature Extraction Techniques

Many linear methods have been proposed for feature extraction during the last two decades [2], [3], [5] - [46]. Among these methods, PCA and FLDA are the two most well-known and frequently used techniques. In PCA, the projection axes, along which the

variance of the projected components of all the sample vectors is maximized, are found. In FLDA, the optimal directions to project input samples in high-dimensional space onto a lower-dimensional space are searched with an objective of finding a discriminant subspace where different class categories occupy compact and disjoint regions.

2.2.1 Principle Component Analysis

Given a set of *m*-dimensional samples, the total covariance matrix can be formed as

$$\mathbf{S}_t = \frac{1}{n}\sum_{l=1}^{n}(\mathbf{x}_l - \mathbf{c})(\mathbf{x}_l - \mathbf{c})^T \tag{2.1}$$

where $\mathbf{x}_l$ is the *l*th sample vector, *n* the sample size, and $\mathbf{c}$ the global centroid given by

$$\mathbf{c} = \frac{1}{n}\sum_{l=1}^{n}\mathbf{x}_l \tag{2.2}$$

PCA finds the set of projection directions, $\mathbf{G}_{PCA}$, in the sample space that maximize the total scatter across all the samples:

$$\mathbf{G}_{PCA} = \arg\max_{\mathbf{G}} |\mathbf{G}^T\mathbf{S}_t\mathbf{G}| \tag{2.3}$$

where $\mathbf{G}$ is the transformation matrix, $\mathbf{G}_{PCA}$ is the optimal transformation matrix whose columns are the orthonormal projection (transformation) vectors that can maximize the total scatter, and $|\bullet|$ denotes the determinant of the associated matrix. Essentially, this is an eigenvalue problem. If the eigenvectors are sorted in the order of descending eigenvalues, the variance of the projected samples along any eigenvector is larger than that along the next eigenvector in the sorted sequence. When the number of non-zero

eigenvalues is less than the dimension of the original sample space, PCA can be used to project the samples from the original high-dimensional sample space to a subspace of the sample space to reduce the dimensionality of the samples and to set them as apart as possible.

Generally, if the original sample space is low-dimensional, the eigenvectors and eigenvalues of the matrix $\mathbf{S}_t$ can be calculated directly. However, for a problem, such as face recognition using holistic whole-image based approach, the dimensionality of a face sample vector is always very high. A direct calculation of the eigenvectors of $\mathbf{S}_t$ is computationally expensive, or even infeasible on computers with low cache memory.

The Eigenface technique [25] that determines the required eigenvectors has been proposed to deal with this problem. These eigenvectors are also called eigenfaces. In this method, $\mathbf{S}_t$ is first expressed as

$$\mathbf{S}_t = \frac{1}{n}\sum_{l=1}^{n}(\mathbf{x}_l - \mathbf{c})(\mathbf{x}_l - \mathbf{c})^T = \mathbf{H}_t\mathbf{H}_t^T \tag{2.4}$$

where $\mathbf{H}_t = \frac{1}{\sqrt{n}}[\mathbf{x}_1 - \mathbf{c}, ..., \mathbf{x}_n - \mathbf{c}]$, and then an $n \times n$ matrix $\mathbf{R} = \mathbf{H}_t^T\mathbf{H}_t$ is formed. In case that the number of samples n is much smaller than the dimension m of the samples, the size of $\mathbf{R}$ is much smaller than that of $\mathbf{S}_t$, and hence, it is much easier to obtain its eigenvectors. Let $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n-1}$ be the orthonormal eigenvectors of $\mathbf{R}$, corresponding to the n-1 largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1}$. Then, the corresponding orthonormal eigenvectors of $\mathbf{S}_t$ are given by

$$\mathbf{g}_j = \frac{1}{\sqrt{\lambda_j}}\mathbf{H}_t\mathbf{u}_j, \quad j=1, \ldots, n\text{-}1 \tag{2.5}$$

The projection of the sample $\mathbf{x}_l$ on the eigenvector $\mathbf{g}_j$ is given by

$$y_j = \mathbf{g}_j{}^T \mathbf{x} = \frac{1}{\sqrt{\lambda_j}} \mathbf{u}_j{}^T \mathbf{H}_t^T \mathbf{x}, \quad j=1, \ldots, n\text{-}1 \tag{2.6}$$

The resulting features, $y_1, y_2, \ldots, y_{n-1}$, form a PCA-transformed feature vector $\mathbf{y}_l = [y_1, y_2, \ldots, y_{n-1}]^T$ for the samples $\mathbf{x}_l$, $l = 1, \ldots, n$.

2.2.2 Fisher's Linear Discriminant Analysis and its Variants

Fisher's Linear Discriminant analysis is one of the most prevalent linear feature extraction techniques for discriminant analysis. Similar to PCA, in FLDA, the optimal directions are obtained to project input high-dimensional samples onto a lower-dimensional subspace. However, while the key idea behind PCA is to find the directions along which the data variance is the largest, that behind FLDA is to search for the projection directions that simultaneously maximize the distance between the samples of different classes and minimize the distance between the samples of the same class. The class separability in low-dimensional representation is maximized in the FLDA method while it is not in PCA [1], [4]. FLDA-based algorithms usually outperform PCA-based ones because of the more rational objective and optimality criterion of the former.

Given a set of m-dimensional samples that consisting of N classes with the ith class having n_i samples, the global centroid is given by (2.2) and the centroid of ith class is given by

$$\mathbf{c}^{(i)} = \frac{1}{n_i} \sum_{l=k_{i-1}+1}^{k_i} \mathbf{x}_l \tag{2.7}$$

where $k_i = n_1 + n_2 + \ \ldots \ + n_i$, and $\mathbf{x}_l$ is the lth ($l = 1, 2, \ldots, n$) sample vector, n the sample size, $n = \sum_{i=1}^{N} n_i$, and i the class number, $i = 1, \ldots, N$. By using the three matrices, $\mathbf{H}_b$, $\mathbf{H}_w$ and $\mathbf{H}_t$, given by

$$\mathbf{H}_b = \frac{1}{\sqrt{n}}\left[\sqrt{n_1}(\mathbf{c}^{(1)} - \mathbf{c}), \ldots, \sqrt{n_N}(\mathbf{c}^{(N)} - \mathbf{c})\right]$$

$$\mathbf{H}_w = \frac{1}{\sqrt{n}}\left[(\mathbf{x}_1 - \mathbf{c}^{(1)}), \ldots, (\mathbf{x}_{n_1} - \mathbf{c}^{(1)}), (\mathbf{x}_{n_1+1} - \mathbf{c}^{(2)}), \ldots, (\mathbf{x}_{n_1+n_2} - \mathbf{c}^{(2)}), \ldots, (\mathbf{x}_{n-n_N+1} - \mathbf{c}^{(N)}), \ldots, (\mathbf{x}_n - \mathbf{c}^{(N)})\right] \tag{2.8}$$

$$\mathbf{H}_t = \frac{1}{\sqrt{n}}\left[(\mathbf{x}_1 - \mathbf{c}), \ldots, (\mathbf{x}_n - \mathbf{c})\right]$$

the between-class and within-class and total scatter matrices can be defined as

$$\mathbf{S}_w = \mathbf{H}_w\mathbf{H}_w^T, \ \mathbf{S}_b = \mathbf{H}_b\mathbf{H}_b^T, \ \mathbf{S}_t = \mathbf{H}_t\mathbf{H}_t^T \tag{2.9}$$

respectively. The linear discriminant analysis employs the Fisher criterion given by

$$\mathbf{G}_{FLDA} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T\mathbf{S}_b\mathbf{G}|}{|\mathbf{G}^T\mathbf{S}_w\mathbf{G}|} = [\mathbf{g}_1, \ \mathbf{g}_2, \ \ldots, \ \mathbf{g}_d] \tag{2.10}$$

where $\mathbf{G}$ is the transformation matrix and $\mathbf{G}_{FLDA}$ is the optimal transformation matrix whose columns, $\mathbf{g}_i$, $i = 1, 2, \ldots, d$, are the set of generalized eigenvectors of $\mathbf{S}_b$ with regard to $\mathbf{S}_w$ [5] corresponding to the $d \leq N - 1$ largest generalized eigenvalues λ_i that is,

$$\mathbf{S}_b\mathbf{g}_i = \lambda_i\mathbf{S}_w\mathbf{g}_i, \ i = 1, 2, \ldots, d. \tag{2.11}$$

When the inverse of $\mathbf{S}_w$ does exist, the generalized eigenvectors can be obtained by the eigen-decomposition of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The new feature vectors $\mathbf{y}_l$ are defined by $\mathbf{y}_l = \mathbf{G}^T \mathbf{x}_l, \quad l = 1, 2, \ldots, n$.

Through the above process of FLDA, the set of the transformation vectors is found to map the high-dimensional samples onto a low-dimensional space, the discriminant subspace, and at the same time, all the samples projected along this set of transformation vectors have the maximum between-class and minimum within-class scatters simultaneously.

It is seen that the Fisher linear discriminant analysis has a limitation in that it requires the within-class scatter matrix $\mathbf{S}_w$ to be non-singular, which is not the case in practice when the SSS problem occurs, i.e., when the number of the samples is smaller than the dimension of the samples. Small sample size datasets with high dimensionality widely exist in real applications such as human face recognition and analysis of micro-array data. To deal with this limitation of the FLDA technique, a number of variants to this technique have been proposed in the literature.

1) PCA + FLDA Method

Swets and Weng [18] have proposed the PCA + FLDA method, also known as the Fisherface method, in order to solve the singularity problem of the Fisher criterion. In this method, in order to make $\mathbf{S}_w$ nonsingular, PCA is first applied to reduce the sample dimension from m to n-N and the transformation matrix $\mathbf{G}_{PCA}$ is obtained. Then, the dimension is further reduced to N-1 for obtaining a lower-dimensional feature

representation of the samples and the transformation matrix $\hat{\mathbf{G}}_{FLDA}$. The overall transformation matrix of the PCA + FLDA method is given by

$$\mathbf{G}_{PCA+FLDA}^{T} = \hat{\mathbf{G}}_{FLDA}^{T}\mathbf{G}_{PCA}^{T} \tag{2.12}$$

where $\mathbf{G}_{PCA}$ can be obtained from (2.3), which rewritten here as

$$\mathbf{G}_{PCA} = \arg\max_{\mathbf{G}} |\mathbf{G}_1^T\mathbf{S}_t\mathbf{G}_1| \tag{2.13}$$

and $\hat{\mathbf{G}}_{FLDA}$ is given by

$$\hat{\mathbf{G}}_{FLDA} = \arg\max_{\mathbf{G}_2} \frac{|\mathbf{G}_2^T\mathbf{G}_{PCA}^T\mathbf{S}_b\mathbf{G}_{PCA}\mathbf{G}_2|}{|\mathbf{G}_2^T\mathbf{G}_{PCA}^T\mathbf{S}_w\mathbf{G}_{PCA}\mathbf{G}_2|} = \arg\max_{\mathbf{G}_2} \frac{|\mathbf{G}_2^T\mathbf{S}_b'\mathbf{G}_2|}{|\mathbf{G}_2^T\mathbf{S}_w'\mathbf{G}_2|} \tag{2.14}$$

with $\mathbf{G}_1$ and $\mathbf{G}_2$ being the matrices whose columns are the projection vectors in the PCA and FLDA transformed spaces, respectively, and $\mathbf{S}_b' = \mathbf{G}_{PCA}^T\mathbf{S}_b\mathbf{G}_{PCA}$ and $\mathbf{S}_w' = \mathbf{G}_{PCA}^T\mathbf{S}_w\mathbf{G}_{PCA}$.

A problem with this algorithm is that the discarded eigenvectors in its PCA part may contain some discriminant information, very useful to the FLDA part. Later, Chen et al. [41] have proved that the null space of $\mathbf{S}_w$, as a matter of fact, contains the most discriminative information. To avoid the loss of significant discriminant information due to the PCA preprocessing step, an algorithm, referred to as direct LDA (D-LDA), without a separate PCA step, has been proposed in [19].

2) Direct LDA (D-LDA) Method

In the D-LDA method [19], the idea of "simultaneous diagonalization" [4], [78] of $\mathbf{S}_b$ and $\mathbf{S}_w$ is employed to deal with the SSS problem. The matrix $\mathbf{S}_b$ is first diagonalized and scaled, and then $\mathbf{S}_w$ is diagonalized.

The eigenvector matrix $\mathbf{V}$ that diagonalizes $\mathbf{S}_b$ is obtained by using eigen-decomposition of $\mathbf{S}_b$ such that

$$\mathbf{V}^T\mathbf{S}_b\mathbf{V} = \mathbf{\Lambda} \tag{2.15}$$

where $\mathbf{V}$ is the eigenvector matrix and $\mathbf{\Lambda}$ is the eigenvalue matrix of $\mathbf{S}_b$. By keeping only the non-zero eigenvalues of $\mathbf{\Lambda}$, (2.15) is re-written as

$$\bar{\mathbf{V}}^T\mathbf{S}_b\bar{\mathbf{V}} = \mathbf{\Lambda}_b \tag{2.16}$$

where in this equation, the diagonal elements of $\mathbf{\Lambda}_b$, with the non-zero eigenvalues only, are arranged in a non-increasing order and $\bar{\mathbf{V}}$ is the eigenvector matrix with the eigenvectors corresponding to the non-zero eigenvalues only. Next, using the matrix

$$\mathbf{Z} = \bar{\mathbf{V}}\Lambda_b^{-\frac{1}{2}} \tag{2.17}$$

the $\mathbf{S}_b$ is diagonalized as

$$\mathbf{Z}^T\mathbf{S}_b\mathbf{Z} = \mathbf{I} \tag{2.18}$$

and the matrix $\mathbf{Z}^T\mathbf{S}_w\mathbf{Z}$ is diagonalized using eigen-decomposition as

$$\mathbf{U}^T(\mathbf{Z}^T\mathbf{S}_w\mathbf{Z})\mathbf{U} = \mathbf{D}_w \tag{2.19}$$

where $\mathbf{U}$ is the eigenvector matrix and $\mathbf{D}_w$ is the eigenvalue matrix of $\mathbf{Z}^T\mathbf{S}_w\mathbf{Z}$. Finally, the transformation matrix is given by

$$\mathbf{G} = \mathbf{ZUD}_w^{-\frac{1}{2}} \tag{2.20}$$

In this algorithm, the null space of $\mathbf{S}_b$, which the authors of [19] claim to contain no useful information for discrimination, is ignored in the first step. However, Gao et al. [28] have pointed out that the D-LDA algorithm has a shortcoming in that ignoring of the null space of $\mathbf{S}_b$ for dimension reduction would also neglect part of the null space of $\mathbf{S}_w$ and would thus result in the loss of some useful discriminant information contained in the null space of $\mathbf{S}_w$.

3) Regularized LDA Method

To deal with the singularity problem of the Fisher criterion, a regularized FLDA (RLDA) has been introduced in [29], [29]. The basic idea of the regularization technique is to add a constant $\alpha > 0$, known as the regularization parameter, to the diagonal elements of the scatter matrices. This parameter is estimated via cross-validation.

The way to deal with the singularity of scatter matrix $\mathbf{S}_w$ in the classical or $\mathbf{S}_t$ in the modified Fisher criterion [42] is to apply regularization by adding a constant to the diagonal elements of $\mathbf{S}_w$ or $\mathbf{S}_t$, i.e., $\hat{\mathbf{S}}_w = \mathbf{S}_w + \alpha\mathbf{I}$ or $\hat{\mathbf{S}}_t = \mathbf{S}_t + \alpha\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of size $m \times m$.

The classical Fisher criterion giving $\mathbf{G}_{FLDA}$ is defined by (2.10), and the modified Fisher linear discriminant criterion [4] is given by

$$\mathbf{G}_{MFLDA} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T\mathbf{S}_b\mathbf{G}|}{|\mathbf{G}^T\mathbf{S}_t\mathbf{G}|} \tag{2.21}$$

where

$$\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b \tag{2.22}$$

Since $\mathbf{S}_w$ and $\mathbf{S}_t$ are positive semi-definite, $\hat{\mathbf{S}}_w$ and $\hat{\mathbf{S}}_t$ are also positive definite, and hence, nonsingular. Then, the transformation matrix for the RLDA method, $\mathbf{G}_{RLDA1}$ or $\mathbf{G}_{RLDA2}$, can be obtained by using the following optimality criteria:

$$\mathbf{G}_{RLDA1} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T \mathbf{S}_b \mathbf{G}|}{|\mathbf{G}^T (\mathbf{S}_w + \alpha \mathbf{I}) \mathbf{G}|} \tag{2.23}$$

$$\mathbf{G}_{RLDA2} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T \mathbf{S}_b \mathbf{G}|}{|\mathbf{G}^T (\mathbf{S}_t + \alpha \mathbf{I}) \mathbf{G}|} \tag{2.24}$$

The solution to (2.23) or (2.24) can be obtained by computing the eigen-decomposition of $(\mathbf{S}_w + \alpha \mathbf{I})^{-1} \mathbf{S}_b$ or $(\mathbf{S}_t + \alpha \mathbf{I})^{-1} \mathbf{S}_b$.

This method has a high computational load when the samples have a large dimension. Also, an adaptive estimation of the optimal regularization parameter from the training samples using cross-validation is very time-consuming. To overcome the shortcomings of the RLDA method, a number of improved RLDA algorithms have been proposed in the literature [21], [31] - [33].

4) Null Space Method

In order to overcome the singularity problem of the Fisher criterion, Chen et al. [41] have proposed the null space method based on the modified criterion [4] for Fisher's linear discriminant analysis.

In this method, a preprocessing step is employed to extract the geometric features and to reduce the dimension of the original sample space. All the training samples are then

projected onto the null space of $\mathbf{S}_w$. The projection vectors thus obtained are finally transformed into the projection vectors by applying PCA.

The algorithm given in [41] for applying the null space method in the original sample space is not efficient. A pixel grouping method is applied to extract geometric features so as to reduce the dimension of the sample space. It has been pointed out that the performance of this method depends on the dimension of the null space of $\mathbf{S}_w$ in the sense that a larger dimension provides a better performance. Thus, a preprocessing step to reduce the original sample dimension should be avoided [47], [88], [89].

2.3 Nonlinear Feature Extraction Techniques

Although the linear discriminant methods described in the previous section are successful when the samples in datasets are linearly separable, they do not provide good performance when the samples do not follow such a pattern, since it is difficult to capture a nonlinear distribution of samples with linear mapping. As the distributions of most patterns in real world are nonlinear and very complicated, problem of pattern recognition of nonlinearly separable samples should be addressed using nonlinear methods. Kernel machine techniques [49] - [56] are a category of such nonlinear methods. The main idea behind these techniques is to transform the input space into a higher dimensional feature space by using a nonlinear kernel mapping where patterns become linearly separable so that the principles of linear discriminant analysis can be applied in the kernel feature space. The kernel functions allow such nonlinear extensions without explicitly forming a

nonlinear mapping, as long as the problem formulation involves the inner products between the mapped data points.

A kernel is a nonlinear mapping $\mathbf{\Phi}$, designed to map the samples of the input space $\Re^m$ into a higher-dimensional feature space $\mathbf{\Gamma}$:

$$\mathbf{\Phi}: \ \Re^m \rightarrow \mathbf{\Gamma}$$

$$\mathbf{x}_l \rightarrow \boldsymbol{\psi}_l$$

Correspondingly, the samples $\mathbf{x}_l$'s in the original input space $\Re^m$ are mapped into the kernel feature space $\mathbf{\Gamma}$, where the classes of the resulting higher dimensional feature vectors $\boldsymbol{\psi}_l$'s become linearly separable. However, the high dimensionality of the feature space makes the feature extraction computationally infeasible. This problem is overcome by using the so-called "kernel trick" [54], in which the inner product of the mapped vectors in the feature space can be implicitly derived from the inner products between the input samples, such that

$$\langle \boldsymbol{\psi}_l, \boldsymbol{\psi}_h \rangle = k\left(\langle \mathbf{x}_l, \mathbf{x}_h \rangle\right) = k_{lh} \tag{2.25}$$

where $\langle \bullet \rangle$ denotes the inner product of the two associated vectors, $k(\bullet)$ denotes a kernel function, and k_{lh} is a scalar. The key to a successful kernelization of a linear algorithm is in its ability to construct inner products in the input space and then to reformulate these products in the feature space. A number of kernelized discriminant analysis algorithms have been proposed with enhanced recognition accuracy [57] - [77]. In the next two subsections, the method of kernelizing linear algorithms is demonstrated using the linear principle component analysis [2] and Fisher's discriminant analysis [4] methods.

2.3.1 Kernelization of the Principle Component Analysis Method

The basic idea of the kernelized principle component analysis (KPCA) [57] is to map the input data into a new feature space $\mathbf{\Gamma}$ where the samples become linearly separable so that the linear PCA can be performed in that feature space.

Given a set of m-dimensional training samples $\mathbf{x}_l$, $l = 1, 2, \ldots, n$, the matrices $\mathbf{\Phi}_t$ and $\mathbf{\Phi}$ are defined as

$$\mathbf{\Phi}_t = \frac{1}{\sqrt{n}}\left[(\psi_1 - \psi), \ldots, (\psi_n - \psi)\right] \tag{2.26}$$

$$\mathbf{\Phi} = \left[\psi_1, \psi_2, \ldots, \psi_n\right] \tag{2.27}$$

where ψ_l is the mapped sample vector corresponding to sample vector $\mathbf{x}_l$ and ψ is the global centroid of the mapped sample vectors in the kernel feature space.

Similar to the definition of the total scatter matrix $\mathbf{S}_t$ in the input sample space, by using the matrix $\mathbf{\Phi}_t$ the total scatter matrix $\tilde{\mathbf{S}}_t$ in the feature space is given as

$$\tilde{\mathbf{S}}_t = \mathbf{\Phi}_t \mathbf{\Phi}_t^T \tag{2.28}$$

The elements of the matrix $\tilde{\mathbf{R}} = \mathbf{\Phi}^T \mathbf{\Phi}$ are then determined by using the "kernel trick":

$$\tilde{\mathbf{R}}_{ij} = \psi_l^T \psi_k = \langle \psi_l, \psi_k \rangle = k\left(\langle \mathbf{x}_l, \mathbf{x}_k \rangle\right) \tag{2.29}$$

The mapped samples are centered around the global centroid by replacing the matrix $\tilde{\mathbf{R}}$ by

$$\hat{\mathbf{R}} = \tilde{\mathbf{R}} - \mathbf{1}_n \tilde{\mathbf{R}} - \tilde{\mathbf{R}} \mathbf{1}_n + \mathbf{1}_n \tilde{\mathbf{R}} \mathbf{1}_n \tag{2.30}$$

where the matrix $\mathbf{1}_n = (1/n)_{n\times n}$.

The orthonormal eigenvectors $\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \ldots, \tilde{\mathbf{u}}_{n-1}$ of $\hat{\mathbf{R}}$, corresponding to the n-1 largest non-zero eigenvalues, $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_{n-1}$, can be obtained by using the eigen-decomposition of $\hat{\mathbf{R}}$ and the corresponding orthonormal eigenvectors of $\tilde{\mathbf{S}}_t$, $\tilde{\mathbf{g}}_1, \tilde{\mathbf{g}}_2, \ldots, \tilde{\mathbf{g}}_{n-1}$, given by

$$\tilde{\mathbf{g}}_j = \frac{1}{\sqrt{\tilde{\lambda}_j}} \mathbf{\Phi} \tilde{\mathbf{u}}_j, \quad j = 1, 2, \ldots n-1 \tag{2.31}$$

form the kernelized transformation matrix.

2.3.2 Kernelization of the Fisher Linear Discriminant Analysis Method

FLDA is designed for linear pattern recognition applications. However, it fails to perform well for the recognition of patterns that are not linearly separable. To deal with this problem, nonlinear versions of FLDA have been proposed. First, Mika [58] formulated a kernelized Fisher discriminant (KFD) analysis method for a two-class case, and then Baudat [59] proposed a generalized kernel discriminant analysis (GDA) for datasets with multiple classes.

In basic idea of the GDA method is to first perform the centering in the kernel feature space by shifting each mapped sample vector using the global centroid, and then to apply the discriminant analysis in the centralized kernel feature space. In this method, given a

set of m-dimensional training samples $\mathbf{x}_l$, $l = 1, 2, \ldots, n$, consisting of N classes where the ith class has n_i samples(thus, $n = \sum_{i=1}^{N} n_i$), first the following three matrices are defined:

$$\mathbf{\Phi}_b = \frac{1}{\sqrt{n}}\left[\sqrt{n_1}(\boldsymbol{\psi}^{(1)} - \boldsymbol{\psi}), \ldots, \sqrt{n_N}(\boldsymbol{\psi}^{(N)} - \boldsymbol{\psi})\right]$$

$$\mathbf{\Phi}_w = \frac{1}{\sqrt{n}}\left[(\boldsymbol{\psi}_1 - \boldsymbol{\psi}^{(1)}), \ldots, (\boldsymbol{\psi}_{n_1} - \boldsymbol{\psi}^{(1)}), (\boldsymbol{\psi}_{n_1+1} - \boldsymbol{\psi}^{(2)}), \ldots, (\boldsymbol{\psi}_{n_1+n_2} - \boldsymbol{\psi}^{(2)}), \ldots,\right.$$

$$\left.(\boldsymbol{\psi}_{n-n_N+1} - \boldsymbol{\psi}^{(\mathrm{N})}), \ldots, (\boldsymbol{\psi}_n - \boldsymbol{\psi}^{(N)})\right] \tag{2.32}$$

$$\mathbf{\Phi}_t = \frac{1}{\sqrt{n}}\left[(\boldsymbol{\psi}_1 - \boldsymbol{\psi}), \ldots, (\boldsymbol{\psi}_n - \boldsymbol{\psi})\right]$$

where $\boldsymbol{\psi}_l$ is the mapped sample vector corresponding to the input sample vector $\mathbf{x}_l$, $\boldsymbol{\psi}^{(i)}$ is the centroid of the mapped samples of the ith class, and $\boldsymbol{\psi}$ is the global centroid of the mapped samples in the kernel feature space. The Fisher criterion can then be expressed as

$$\tilde{\mathbf{G}}_K = \arg\max_{\mathbf{G}} \frac{\left|\mathbf{G}^T \tilde{\mathbf{S}}_b \mathbf{G}\right|}{\left|\mathbf{G}^T \tilde{\mathbf{S}}_w \mathbf{G}\right|} = [\tilde{\mathbf{g}}_{K1}, \tilde{\mathbf{g}}_{K2}, \ldots, \tilde{\mathbf{g}}_{Kd}] \tag{2.33}$$

where $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ are, respectively, the between-class and within-class scatter matrices defined in the kernel feature space $\mathbf{\Gamma}$ as

$$\tilde{\mathbf{S}}_b = \mathbf{\Phi}_b \mathbf{\Phi}_b^T \tag{2.34}$$

$$\tilde{\mathbf{S}}_w = \mathbf{\Phi}_w \mathbf{\Phi}_w^T \tag{2.35}$$

and $\mathbf{G} = [\mathbf{g}_{K1}, \mathbf{g}_{K2}, \ldots, \mathbf{g}_{Kd}]^T$ is the transformation matrix. The matrix $\tilde{\mathbf{G}}_K$ is the optimal transformation matrix with its columns $\tilde{\mathbf{g}}_{Ki}$'s as the eigenvectors corresponding to the d

largest eigenvalues obtained by solving the generalized eigenvalue problem: $\tilde{\mathbf{S}}_b\tilde{\mathbf{g}}_{Ki} = \tilde{\lambda}_i\tilde{\mathbf{S}}_w\tilde{\mathbf{g}}_{Ki}$. This generalized eigenvalue problem cannot be solved due to the high dimensionality of the mapped sample vectors. This problem is solved by formulating an alternate generalized eigenvalue problem.

Let $\boldsymbol{\alpha}_i = (\alpha_{i1},\ldots,\alpha_{in})^T$, $i = 1,\ 2,\ \ldots,\ d$, such that

$$\tilde{\mathbf{g}}_i = \sum_{l=1}^{n} \alpha_{il}\boldsymbol{\psi}_l = \boldsymbol{\Phi}\boldsymbol{\alpha}_i \tag{2.36}$$

where $\boldsymbol{\Phi}$ is defined by (2.27). The transformation matrix $\mathbf{G}$ can be expressed as

$$\mathbf{G} = \boldsymbol{\Phi}[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_d] = \boldsymbol{\Phi}\boldsymbol{\Theta} \tag{2.37}$$

where $\boldsymbol{\Theta} = [\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_d]$. Substituting (2.37) into (2.33), the Fisher criterion can be expressed as [57]

$$\boldsymbol{\Theta}_K = \arg\max_{\boldsymbol{\Theta}} \frac{\left|\boldsymbol{\Theta}^T(\hat{\mathbf{R}}\mathbf{W}\hat{\mathbf{R}})\boldsymbol{\Theta}\right|}{\left|\boldsymbol{\Theta}^T(\hat{\mathbf{R}}\hat{\mathbf{R}})\boldsymbol{\Theta}\right|} \tag{2.38}$$

where the matrix $\hat{\mathbf{R}}$ is given by (2.30) and $\mathbf{W} = \mathrm{diag}(\mathbf{W}_1,\ldots\mathbf{W}_i,\ldots,\mathbf{W}_N)$ is an $n\times n$ block diagonal matrix with $\mathbf{W}_i$ being an $n_i\times n_i$ diagonal matrix with all its diagonal elements equal to $1/n_i$.

Conducting an eigen-decomposition of the matrix $\hat{\mathbf{R}}$ yields

$$\hat{\mathbf{R}} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T \tag{2.39}$$

where $\mathbf{P}$ is the eigenvector matrix of $\hat{\mathbf{R}}$ with $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, and $\mathbf{\Lambda}$ is the eigenvalue matrix with non-zero eigenvalues as its diagonal elements. Substituting (2.39) into (2.38), (2.38) can be expressed as

$$\mathbf{\Theta}_K = \arg\max_{\mathbf{\Theta}} \frac{\left|(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{\Theta})^T(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{W}\mathbf{P}\mathbf{\Lambda}^{\frac{1}{2}})(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{\Theta})\right|}{\left|(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{\Theta})^T\mathbf{\Lambda}(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{\Theta})\right|} \tag{2.40}$$

By letting

$$\mathbf{B} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{P}^T\mathbf{\Theta} \tag{2.41}$$

Eqn. (2.40) can be expressed as

$$\mathbf{B}_K = \arg\max_{\mathbf{B}} \frac{\left|\mathbf{B}^T\tilde{\mathbf{S}}_b\mathbf{B}\right|}{\left|\mathbf{B}^T\tilde{\mathbf{S}}_w\mathbf{B}\right|} \tag{2.42}$$

where $\tilde{\mathbf{S}}_b = \mathbf{\Lambda}^{1/2}\mathbf{P}^T\mathbf{W}\mathbf{P}\mathbf{\Lambda}^{1/2}$ is semi-positive definite and $\tilde{\mathbf{S}}_w = \mathbf{\Lambda}$ is positive definite. The columns of the optimal transformed coefficient matrix $\mathbf{B}_K = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_d]$ are actually the eigenvectors of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$ corresponding to the d ($d \le N-1$) largest eigenvalues, and can be obtained by eigen-decomposition of $\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b$. Once the optimal transformed coefficient matrix $\mathbf{B}_K$ is determined, the corresponding optimal coefficient matrix $\mathbf{\Theta}_K$ can be obtained as $\mathbf{\Theta}_K = \mathbf{P}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{B}_K$. Finally, based on Fisher's optimality criterion given by (2.33), the optimal transformation matrix $\mathbf{G}_K$ is obtained as

$$\mathbf{G}_K = \mathbf{\Phi}\mathbf{\Theta}_K = \mathbf{\Phi}\mathbf{P}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{B}_K \tag{2.43}$$

2.4 Summary

In this chapter, feature extraction techniques have been reviewed for solving pattern recognition problems. Techniques for feature extraction can be classified into linear and nonlinear categories.

Linear methods are applied when the samples are linearly separable. In linear category, the principle component analysis (PCA), Fisher's linear discriminant analysis (FLDA) and some typical FLDA variants have been reviewed. Both the PCA and FLDA methods extract features by projecting the original sample vectors onto a new lower dimensional feature space through a linear transformation. However, the goal of optimizing the transformation matrix in the two methods is different. The FLDA-based algorithms usually outperform the PCA-based ones because of the use of more rational and objective optimality criteria in the former. The singularity problem of the scatter matrices of the traditional FLDA has been explained and techniques used in the FLDA variants for solving this problem have been described in detail.

In the nonlinear category, the kernel technique has been discussed to deal with the nonlinearly distributed patterns. The main idea behind the kernel techniques is to transform the input data into a higher dimensional space by using a nonlinear mapping function, so that the samples become linearly separable and hence, the principles of linear discriminant analysis can be applied in that space. The method of kernelizing linear algorithms has been demonstrated using the methods of the linear principle component analysis and Fisher's discriminant analysis.

Motivations behind the development of the various discriminant analysis techniques discussed in this chapter and their limitation have been point out.

Chapter 3

Discriminant Analyses Based on Modified Generalized Singular Value Decomposition

3.1 Introduction

An alternative to the FLDA algorithm [5] is the LDA/GSVD algorithm [34], [35], in which the GSVD [82], [83] is adapted to the FLDA algorithm for pattern recognition problems. GSVD not only provides a framework for finding the feature vectors with high recognition accuracy, but more importantly, it relaxes the requirement of the within-class scatter matrix to be non-singular. Thus, the LDA/GSVD algorithm is an effective approach to overcome the SSS problem. However, this algorithm has a drawback in that it cannot provide a practical solution to a pattern recognition problem with a large sample dimension. An important area is the face recognition problem, in which the sample dimension is almost invariably very high. Thus, in such a case, the LDA/GSVD algorithm experiences a memory overflow problem and fails to carry out the task of face recognition. The memory overflow occurs in conducting the SVD of a high-dimensional matrix associated with large dimension patterns. In the same paper [35], Ye et al. have presented yet another method, known as approximate LDA/GSVD method, in which the K-Means algorithm is introduced to reduce the computational complexity. However, it does not effectively address the problem of high computational complexity related to the

high dimensionality of the samples. This method also results in losing some useful discriminant information while dealing with the computational complexity problem.

This LDA/GSVD algorithm may also suffer from the over-fitting problem in some applications, since the samples in the derived discriminant subspace may get corrupted with some random features that are unrelated to the actual discriminatory features and adversely impact the recognition accuracy. Ye et al. [86] have proposed a method to deal with the over-fitting problem of the LDA/GSVD algorithm. However, the proposed orthogonalization technique achieved through QR decomposition is not computationally efficient for high dimensional data and also it cannot be subjected to kernel methods.

In this chapter, an algorithm, referred to as MGSVD-LDA algorithm, which overcomes the singularity problem in the Fisher criterion and deals effectively with the excessive computational load problem of the LDA/GSVD algorithm, is developed by using the eigen-decomposition to conduct the generalized singular value decomposition in the discriminant analysis. Schemes are given to kernelize the proposed linear algorithm and to deal with the over-fitting problem.

Section 3.2 gives a brief review of the LDA/GSVD algorithm. The development of the proposed MGSVD-LDA [37] is carried out in Section 3.3. The scheme to kernelize the proposed linear algorithm is given in Section 3.4. A method that orthogonalizes the basis of the discriminant subspace derived from the GSVD-based algorithms is given in Section 3.5 to deal with the over-fitting problem [38], [39]. Experimental results demonstrating the performance of the proposed algorithms and their comparisons with other existing algorithms are presented in Section 3.6.

3.2 An Review of the LDA/GSVD Algorithm

The objective of the Fisher linear discriminant analysis is to find an optimal transformation matrix **G** that consists of a set vectors **g**'s given by

$$\mathbf{G}_{FLDA} = \arg\max_{\mathbf{G}} \frac{|\mathbf{G}^T\mathbf{S}_b\mathbf{G}|}{|\mathbf{G}^T\mathbf{S}_w\mathbf{G}|} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_d] \tag{3.1}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are, respectively, the between-class and within-class scatter matrices. This criterion is equivalent to the generalized eigenvalue problem, $\mathbf{S}_b\mathbf{g} = \lambda\mathbf{S}_w\mathbf{g}$, in which λ is the generalized eigenvalue and $\mathbf{g}$ is the corresponding eigenvector of $\mathbf{S}_b$ respect to $\mathbf{S}_w$. The solution of this generalized eigenvalue problem has an important property that the matrix consisting of **g**'s diagnolizes $\mathbf{S}_b$, $\mathbf{S}_w$, and the total scatter matrix $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$ simultaneously [4]. Because of this property, the generalized singular value decomposition based LDA [34], [35] tries to find an optimal transformation matrix **G** that consists of **g**'s.

Given a set of m-dimensional training samples that consists of N classes, where the ith class has n_i images, the global centroid and the class centroid are given by (2.2) and (2.7), respectively. We define $\mathbf{H}_b$ and $\mathbf{H}_w$ as given by (2.8) and a matrix **C** as

$$\mathbf{C} = \begin{bmatrix} \mathbf{H}_b^T \\ \\ \mathbf{H}_w^T \end{bmatrix}_{((N+n)\times m)} \tag{3.2}$$

Then, SVD of **C** can be obtained as

$$\mathbf{C} = \mathbf{P}\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{Q}^T \tag{3.3}$$

where $\mathbf{R} \in \Re^{k\times k}$, k being rank($\mathbf{C}$), is a diagonal matrix whose diagonal components are the non-zero singular values of the matrix $\mathbf{C}$ sorted in a non-increasing order, and $\mathbf{P} \in \Re^{(N+n)\times(N+n)}$ and $\mathbf{Q} \in \Re^{m\times m}$ are orthogonal eigenvector matrices. The matrix $\mathbf{P}$ can be partitioned as

$$\mathbf{P} = \begin{bmatrix} \underbrace{\mathbf{P}_1}_{k} & \underbrace{\mathbf{P}_2}_{n+N-K} \end{bmatrix} \tag{3.4}$$

where $\mathbf{P}_1 \in \Re^{(N+n)\times k}$ and $\mathbf{P}_2 \in \Re^{(N+n)\times(N+n-k)}$. The sub-matrix $\mathbf{P}_1$ can be further partitioned as $\begin{bmatrix} \mathbf{P}_{11} \\ \mathbf{P}_{12} \end{bmatrix}$, where $\mathbf{P}_{11} \in \Re^{N\times k}$ and $\mathbf{P}_{12} \in \Re^{n\times k}$. Now, using SVD of $\mathbf{P}_1$, we have

$$\mathbf{U}^T\mathbf{P}_{11}\mathbf{W} = \mathbf{\Sigma}_b = \begin{bmatrix} \mathbf{I}_b & & \\ & \mathbf{D}_b & \\ & & \mathbf{0}_b \end{bmatrix}_{(N\times k)} \tag{3.5}$$

and

$$\mathbf{V}^T\mathbf{P}_{12}\mathbf{W} = \mathbf{\Sigma}_w = \begin{bmatrix} \mathbf{0}_w & & \\ & \mathbf{D}_w & \\ & & \mathbf{I}_w \end{bmatrix}_{(n\times k)} \tag{3.6}$$

where matrix $\mathbf{U}$,$\mathbf{V}$ and $\mathbf{W}$ are orthogonal eigenvector matrices and $\mathbf{\Sigma}_b$ and $\mathbf{\Sigma}_w$ are eigenvalue matrices. In $\mathbf{\Sigma}_b$ and $\mathbf{\Sigma}_w$, $\mathbf{I}_b \in \Re^{u\times u}$ and $\mathbf{I}_w \in \Re^{(k-u-s)\times(k-u-s)}$ are the identity matrices, where

$$s = \text{rank}(\mathbf{H}_b) + \text{rank}(\mathbf{H}_w) - \text{rank}(\mathbf{C}) \tag{3.7}$$

and $u = \text{rank}(\mathbf{C}) - \text{rank}(\mathbf{H}_w)$, $\mathbf{0}_b \in \Re^{(N-u-s)\times(N-u-s)}$ and $\mathbf{0}_w \in \Re^{(n-k+u)\times u}$ are zero matrices, and $\mathbf{D}_b = \text{diag}(\alpha_{u+1}, \ldots, \alpha_{u+s})$ and $\mathbf{D}_w = \text{diag}(\beta_{u+1}, \ldots, \beta_{u+s})$ satisfying

$$\begin{aligned} &1 > \alpha_{u+1} \geq \cdots \geq \alpha_{u+s} > 0 \\ &0 < \beta_{u+1} \leq \cdots \leq \beta_{u+s} < 1 \\ &\alpha_i^2 + \beta_i^2 = 1, \quad i = u+1, \cdots, u+s \end{aligned} \tag{3.8}$$

Combining (3.3), (3.5) and (3.6) gives

$$\begin{bmatrix} \mathbf{H}_b^T \\ \mathbf{H}_w^T \end{bmatrix} \mathbf{Q} = \begin{bmatrix} \mathbf{P}_1\mathbf{R}, & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{U}\mathbf{\Sigma}_b\mathbf{W}^T\mathbf{R} & \mathbf{0} \\ \mathbf{V}\mathbf{\Sigma}_w\mathbf{W}^T\mathbf{R} & \mathbf{0} \end{bmatrix} \tag{3.9}$$

which can be expressed equivalently as

$$\mathbf{U}^T\mathbf{H}_b^T\mathbf{Q} = \mathbf{\Sigma}_b \begin{bmatrix} \mathbf{W}^T\mathbf{R} & \mathbf{0} \end{bmatrix} \tag{3.10}$$

and

$$\mathbf{V}^T\mathbf{H}_w^T\mathbf{Q} = \mathbf{\Sigma}_w \begin{bmatrix} \mathbf{W}^T\mathbf{R} & \mathbf{0} \end{bmatrix} \tag{3.11}$$

Let

$$\mathbf{Y} = \mathbf{Q}\begin{bmatrix} \mathbf{R}^{-1}\mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3.12}$$

Then, (3.10) and (3.11) can be transformed into

$$\mathbf{H}_b^T\mathbf{Y} = \mathbf{U}\begin{bmatrix} \boldsymbol{\Sigma}_b & \mathbf{0} \end{bmatrix}, \quad \mathbf{H}_w^T\mathbf{Y} = \mathbf{V}\begin{bmatrix} \boldsymbol{\Sigma}_w & \mathbf{0} \end{bmatrix} \tag{3.13}$$

from which we have

$$\mathbf{Y}^T\mathbf{S}_b\mathbf{Y} = \begin{bmatrix} \boldsymbol{\Sigma}_b^T\boldsymbol{\Sigma}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{D}_1 \tag{3.14}$$

$$\mathbf{Y}^T\mathbf{S}_w\mathbf{Y} = \begin{bmatrix} \boldsymbol{\Sigma}_w^T\boldsymbol{\Sigma}_w & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{D}_2 \tag{3.15}$$

Thus, both $\mathbf{S}_b$ and $\mathbf{S}_w$ are diagonalized by the matrix $\mathbf{Y}$. As the null space of $\mathbf{D}_1$ has little discrimination information [35], the only columns of the matrix $\mathbf{Y}$ that correspond to the range of $\mathbf{S}_b$ need to be maintained during the feature extraction, and they collectively form the optimal transformation matrix $\mathbf{G}$.

A limitation of the above LDA/GSVD algorithm is the excessive computation involved with the SVD of $\mathbf{C}$ whose size is $(N+n)\times m$. In the case, when the sample dimension m is higher than the sample size n, the computational complexity depends mainly on m, and very little on n or on the number of classes N. This algorithm is found to suffer from the over-fitting problem in some applications. This is because all the singular vectors of the matrix $\mathbf{C}$ are maintained, and as the singular vectors are divided by their associated singular values, the impact of the small singular vectors gets amplified in the classification.

In order to reduce the computational complexity, Ye et al. [35] have presented an approximation method, where the K-Means algorithm has been introduced to somewhat reduce the size of $\mathbf{C}$. The samples within each class are grouped to generate K clusters, and the centroid of each cluster is used to replace all the samples of the cluster. For instance, if $K = 2$, the class size is reduced to 2. Unfortunately, there are two drawbacks of this approximation method: First, it loses some of the information to be used for discrimination because of the approximation of the samples of a cluster by their centroid. Second, it does not effectively address the problem of high computational complexity that is caused mainly due to the large size of $\mathbf{Q}$. The size of $\mathbf{Q}$ depends on the sample dimension m, which is not affected by the clustering of the class samples.

Park et al. [36] have recently proposed a method to reduce the computational load of the LDA/GSVD algorithm. They replace the two SVDs in the conventional GSVD framework with two eigen-decompositions. The first eigen-decomposition is carried out on the total scatter matrix to find its range. To reduce the computational load of the eigen-decomposition, the total scatter matrix is transformed into its inner product form. The second eigen-decomposition is carried out on the between-class scatter matrix in the range found above. This method does not address the over-fitting problem.

Ye et al. [86] have addressed the over-fitting problem of the LDA/GSVD algorithm through an orthogonalization of the basis of the discrimination subspace, and used QR decomposition to achieve orthogonalization. However, QR decomposition is not efficient when the data is of high dimension, and moreover this method cannot be combined with kernelization.

3.3 MGSVD-LDA: Linear Discriminant Analysis Based on Modified GSVD

In order to reduce the computational complexity, we now propose a method in which for the SVD of **C**, as given by (3.3), the explicit computation of **Q** is avoided. The singular vector matrices, **P** and **Q**, are the eigenvector matrices of $\mathbf{C}\mathbf{C}^T$ and $\mathbf{C}^T\mathbf{C}$, respectively. Therefore, we can first evaluate **P** from $\mathbf{C}\mathbf{C}^T$ whose size is $(n+N)\times(n+N)$. In order to compute **Q** whose size is $m \times m$, we proceed as follows instead of computing it explicitly by using SVD.

Just as **P** is partitioned in a form given by (3.4), where $\mathbf{P}_2$ corresponds to the null space of $\mathbf{C}\mathbf{C}^T$, **Q** is partitioned in the form

$$\mathbf{Q} = \left[\underbrace{\mathbf{Q}_1}_{k} \quad \underbrace{\mathbf{Q}_2}_{m-k} \right] \tag{3.16}$$

where k = rank(**C**), and $\mathbf{Q}_2$ corresponds to the null space of $\mathbf{C}\mathbf{C}^T$. Since both $\mathbf{P}_2$ and $\mathbf{Q}_2$ correspond to the null space, removal of these sub-matrices from the SVD of **C** in the proposed scheme has no influence on the discrimination effectiveness. The matrix **C** can now be rewritten as

$$\mathbf{C} = [\mathbf{P}_1, \mathbf{P}_2] \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} = \mathbf{P}_1 \mathbf{R} \mathbf{Q}_1^T \tag{3.17}$$

From this equation, we have

$$\mathbf{Q}_1 = \mathbf{C}^T \mathbf{P}_1 \mathbf{R}^{-1} \tag{3.18}$$

Using the above equation in (3.12), $\mathbf{Y}_k$, the matrix consisting of the first k columns of $\mathbf{Y}$, can be expressed as

$$\mathbf{Y}_k = \mathbf{Q}_1\mathbf{R}^{-1}\mathbf{W} = \mathbf{C}^T\mathbf{P}_1\mathbf{R}^{-2}\mathbf{W} \tag{3.19}$$

Thus, without an explicit computation of $\mathbf{Q}$, $\mathbf{Y}_k$ is obtained, which can be used to diagonalize the scatter matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ simultaneously by employing (3.12) into (3.14) as

$$\mathbf{Y}^T\mathbf{S}_b\mathbf{Y}_k = \mathbf{\Sigma}_b^T\mathbf{\Sigma}_b \tag{3.20}$$

$$\mathbf{Y}^T\mathbf{S}_w\mathbf{Y}_k = \mathbf{\Sigma}_w^T\mathbf{\Sigma}_w \tag{3.21}$$

The leftmost r columns of $\mathbf{Y}_k$ form the optimal transformation matrix $\mathbf{G}$, where r = rank($\mathbf{H}_b$). The proposed MGSVD-LDA algorithm is summarized in Table 3.1.

Table 3.1: MGSVD-LDA algorithm

Input: Training sample $\mathbf{x}_l$
Output: Transformation matrix $\mathbf{G}$
1. Use Equation (2.8) and (3.2) to obtain $\mathbf{H}_b$, $\mathbf{H}_w$ and $\mathbf{C}$;
2. Find $\mathbf{P}$ and $\mathbf{R}$ from $\mathbf{C}\mathbf{C}^T = \mathbf{P}\begin{bmatrix}\mathbf{R}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{P}^T$;
3. $\mathbf{P}_1 \leftarrow \mathbf{P}(:, 1:k), \mathbf{P}_{11} \leftarrow \mathbf{P}_1(1:N, :), k = \text{rank}(\mathbf{C})$;
4. Find $\mathbf{W}$ through SVD of $\mathbf{P}_{11}$: $\mathbf{P}_{11} = \mathbf{U}\mathbf{\Sigma}_b\mathbf{W}^T$;
5. $\mathbf{Y}_k \leftarrow \mathbf{C}^T\mathbf{P}_1\mathbf{R}^{-2}\mathbf{W}$;
6. $\mathbf{G} \leftarrow \mathbf{Y}_k(:, 1:r)$, $r = \text{rank}(\mathbf{H}_b)$.

Computing SVD of **C** requires approximately $2m^2(N+n) + 11(N+n)^3$ flops, if Chan's algorithm [78], which is efficient for a matrix with the column dimension much higher than the row dimension (the case of small sample size datasets), is used. On the other hand, computing the eigen-decomposition of $\mathbf{C}\mathbf{C}^T$ needs only $3/4\,(N+n)^3 + O((N+n)^2)$ flops [78], which is much smaller than the flop count required form computing the SVD of **C**. In addition, computing of the SVD of **C** requires a memory space of approximately $(m + N + n)\,(N + n)$ locations, whereas computing the eigen-decomposition of matrix $\mathbf{C}\mathbf{C}^T$ requires only $(N + n)^2$ locations. In the case of small sample size datasets, that is, $m >> (N+n)$, the proposed MGSVD-LDA algorithm uses much less memory space than the LDA/GSVD algorithm does. From the above discussion, we conclude that, in the case of small sample size datasets, the proposed MGSVD-LDA algorithm has much lower levels of time and space complexities than that of the LDA/GSVD algorithm.

3.4 Kernelization of MGSVD-LDA

In the preceding section, we have presented a linear discriminant algorithm, which, like most other linear discrimination approaches, assumes that the classes are linearly separable in the input space. However, the distributions of many patterns in the real world are nonlinear, and linear methods of discriminant analysis may not provide sufficient recognition accuracy. Fortunately, in this case one can establish the linear separable condition [87] by using appropriate kernels [55] and then apply the linear discriminant analysis techniques in that space. We now present a scheme to kernelize the proposed MGSVD-LDA algorithm. The new kernel discriminant analysis algorithm is hereafter

referred to as the MGSVD-KDA algorithm.

As it was done for the development of the MGSVD-LDA algorithm, we first define the following matrix:

$$\begin{aligned}\boldsymbol{\Phi}_b &= \tfrac{1}{\sqrt{n}}\left[\sqrt{n_1}(\boldsymbol{\psi}^{(1)}-\boldsymbol{\psi}),\cdots,\sqrt{n_i}(\boldsymbol{\psi}^{(i)}-\boldsymbol{\psi}),\cdots,\sqrt{n_N}(\boldsymbol{\psi}^{(N)}-\boldsymbol{\psi})\right] \\ \boldsymbol{\Phi}_w &= \tfrac{1}{\sqrt{n}}\left[\boldsymbol{\psi}_1-\boldsymbol{\psi}^{(1)},\cdots,\boldsymbol{\psi}_l-\boldsymbol{\psi}^{(l)},\cdots,\boldsymbol{\psi}_N-\boldsymbol{\psi}^{(N)}\right]\end{aligned} \tag{3.22}$$

and

$$\boldsymbol{\Gamma} = \begin{bmatrix}\boldsymbol{\Phi}_b^T \\ \boldsymbol{\Phi}_w^T\end{bmatrix} \tag{3.23}$$

where $\boldsymbol{\psi}^{(i)}$ is the centroid of the ith embedding class, and $\boldsymbol{\psi}$ is the global centroid of the embedding samples in the f-dimensional kernel feature space. The SVD of $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} = \tilde{\mathbf{P}}\begin{bmatrix}\tilde{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{Q}}^T \tag{3.24}$$

where $\tilde{\mathbf{P}} \in \Re^{(N+n)\times(N+n)}$ and $\tilde{\mathbf{Q}} \in \Re^{f\times f}$ are orthogonal matrices, and $\tilde{\mathbf{R}} \in \Re^{z\times z}$ with z = rank($\boldsymbol{\Gamma}$) is a diagonal matrix with its elements being equal to non-zero singular values of $\boldsymbol{\Gamma}$ sorted in a non-increasing order. Due to the high dimensionality of $\boldsymbol{\Gamma}$, it would be practically not feasible to conduct the SVD directly. However, the smaller dimension singular vector matrix $\tilde{\mathbf{P}}$ and singular value matrix $\tilde{\mathbf{R}}$ can be evaluated separately by using the kernel method. We form a symmetric matrix as

$$\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T = \begin{bmatrix}\boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_b & \boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_w \\ \boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_b & \boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w\end{bmatrix} \tag{3.25}$$

where each of the four sub-matrices is in an inner product form. The matrix $\tilde{\mathbf{P}}$ is exactly the eigenvector matrix of $\mathbf{\Gamma}\mathbf{\Gamma}^T$ and the matrix $\tilde{\mathbf{R}}$ is the square root of its eigenvalue matrix. The main problem here is as how to evaluate the matrix $\mathbf{\Gamma}\mathbf{\Gamma}^T$, or equivalently, the four sub-matrices. We construct a kernel matrix

$$\mathbf{K} = (k_{lh})_{l,h=1,\cdots,n} \tag{3.26}$$

whose elements are the inner products in the feature space determined through a kernel function. Then, we can express the sub-matrices in (3.25) in terms of $\mathbf{K}$ as

$$\begin{aligned}
\mathbf{\Phi}_b^T\mathbf{\Phi}_b &= \mathbf{D}(\mathbf{B}-\mathbf{L})^T\mathbf{K}(\mathbf{B}-\mathbf{L})\mathbf{D} \\
\mathbf{\Phi}_w^T\mathbf{\Phi}_w &= (\mathbf{I}-\mathbf{A})\mathbf{K}(\mathbf{I}-\mathbf{A}) \\
\mathbf{\Phi}_b^T\mathbf{\Phi}_w &= \mathbf{D}(\mathbf{B}-\mathbf{L})^T\mathbf{K}(\mathbf{I}-\mathbf{A})
\end{aligned} \tag{3.27}$$

where

$$\mathbf{A} = diag(\mathbf{A}_1,\cdots,\mathbf{A}_N),\ \mathbf{A}_i = (1/n_i)_{n_i\times n_i}$$

$$\mathbf{B} = diag(\mathbf{B}_1,\cdots,\mathbf{B}_N),\ \mathbf{B}_i = (1/n_i)_{n_i\times 1}$$

$$\mathbf{D} = diag(\mathbf{D}_1,\cdots,\mathbf{D}_N),\ \mathbf{D}_i = (\sqrt{n_i})_{n_i\times n_i}$$

$i = 1,\cdots,N$, $\mathbf{L} = (1/n)_{n\times N}$, and $\mathbf{I}$ is an $n\times n$ identity matrix.

Derivation of this set of formulas is presented in Appendix A.

Similar to the MGSVD-LDA algorithm, the eigen-decomposition of $\mathbf{\Gamma}\mathbf{\Gamma}^T$ generates the eigenvector matrix $\tilde{\mathbf{P}}$ and the non-zero eigenvalue matrix $\tilde{\mathbf{R}}$. The leftmost z columns of $\tilde{\mathbf{P}}_1$, where z = rank($\mathbf{\Gamma}\mathbf{\Gamma}^T$), form the matrix $\tilde{\mathbf{P}}_1$, and the first N rows of $\tilde{\mathbf{P}}_1$ form the

matrix $\tilde{\mathbf{P}}_{11}$. The SVD of $\tilde{\mathbf{P}}_{11}$ provides the orthogonal matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{W}}$ such that $\tilde{\mathbf{P}}_{11} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}_b\tilde{\mathbf{W}}$. Letting $\tilde{\mathbf{Y}}_z = \mathbf{\Gamma}^T\tilde{\mathbf{P}}_1\tilde{\mathbf{R}}^{-2}\tilde{\mathbf{W}}$ and $\mathbf{\Lambda} = \tilde{\mathbf{W}}^T\tilde{\mathbf{R}}^{-2}\tilde{\mathbf{P}}_1^T$, we have

$$\tilde{\mathbf{Y}}_z^T = \mathbf{\Lambda}\mathbf{\Gamma} \tag{3.28}$$

Further, let

$$\tilde{\mathbf{G}}^T = \mathbf{\Lambda}_v\mathbf{\Gamma} \tag{3.29}$$

where v = rank($\mathbf{\Phi}^T\mathbf{\Phi}_b$), and $\mathbf{\Lambda}_v$ consists of the first v rows of $\mathbf{\Lambda}$. The columns of $\tilde{\mathbf{G}}$ are the extracted feature vectors of the feature space.

Given a test image $\mathbf{x}_t$ with its mapping in the feature space being $\boldsymbol{\psi}_t$, the kernel function is applied again to obtain

$$q_l = k(\langle \mathbf{x}_l, \mathbf{x}_t \rangle) = \langle \boldsymbol{\psi}_l, \boldsymbol{\psi}_t \rangle \tag{3.30}$$

and subsequently to form the vectors

$$\begin{aligned} \mathbf{Q}_b &= \tfrac{1}{\sqrt{n}}\left[\sqrt{n_1}(\mathbf{q}^{(1)} - \mathbf{q}), \cdots, \sqrt{n_i}(\mathbf{q}^{(i)} - \mathbf{q}), \cdots, \sqrt{n_N}(\mathbf{q}^{(N)} - \mathbf{q})\right] \\ \mathbf{Q}_w &= \tfrac{1}{\sqrt{n}}\left[\mathbf{q}_1 - \mathbf{q}^{(1)}, \cdots, \mathbf{q}_l - \mathbf{q}^{(l)}, \cdots, \mathbf{q}_N - \mathbf{q}^{(N)}\right] \end{aligned} \tag{3.31}$$

where $\mathbf{q}^{(i)} = \dfrac{1}{n_i}\sum_{l=n_1+n_2+\ldots+n_{i-1}+1}^{n_1+n_2+\ldots+n_{i-1}+n_i}\mathbf{q}_l$ and $\mathbf{q} = \frac{1}{n}\sum_{l=1}^{n}\mathbf{q}_l$. Since $\mathbf{\Gamma}\boldsymbol{\psi}_t = \begin{bmatrix}\mathbf{Q}_b^T \\ \mathbf{Q}_\omega^T\end{bmatrix}$, the projection of $\boldsymbol{\psi}_t$ on the feature vectors can be found as

$$w = \tilde{\mathbf{G}}^T\boldsymbol{\psi}_t = \mathbf{\Lambda}_v\begin{bmatrix}\mathbf{Q}_b^T \\ \mathbf{Q}_w^T\end{bmatrix}. \tag{3.32}$$

The proposed MGSVD-KDA algorithm is summarized as in Table 3.2

This algorithm, like many other kernelized algorithms, has a computational complexity determined approximately by the accumulated effects of implementing the

kernel operator and the associated linear discriminant algorithm with the cost of computing the kernel matrices depending mainly on kernel function chosen.

Table 3.2: MGSVD-KDA algorithm

Training stage
Input: Training sample $\mathbf{x}_l$
Output: $\mathbf{\Lambda}_v$ in (3.29)
1. Form the kernel matrix $\mathbf{K}$ based on (3.26) and (3.26) and the kernel function chosen;
2. Evaluate the matrix $\mathbf{\Gamma\Gamma}^T$ given by (3.25) using (3.27);
3. Find $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{R}}$ from $\mathbf{\Gamma\Gamma}^T = \tilde{\mathbf{P}}\begin{bmatrix}\tilde{\mathbf{R}}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{P}}^T$;
4. $\tilde{\mathbf{P}}_1 \leftarrow \tilde{\mathbf{P}}(:,1:z), \mathbf{P}_{11} \leftarrow \mathbf{P}_1(1:N,:),\ z = \text{rank}(\mathbf{\Gamma\Gamma}^T)$;
5. Find $\tilde{\mathbf{W}}$ through SVD of $\tilde{\mathbf{P}}_{11}$: $\mathbf{P}_{11} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}_b\tilde{\mathbf{W}}^T$;
6. $\mathbf{\Lambda}_v \leftarrow$ the first v rows of $\tilde{\mathbf{W}}^T\tilde{\mathbf{R}}^{-2}\tilde{\mathbf{P}}_1^T, v = \text{rank}(\mathbf{\Phi}_b^T\mathbf{\Phi}_b)$;
Classification stage
Input: Test vector $\mathbf{x}_t$
Output: Weight vector w in (3.32);
7. Evaluate $\mathbf{q}_l$ as in (3.30);
8. Form $\mathbf{Q}_b$ and $\mathbf{Q}_w$ in (3.31);
9. $w \leftarrow \mathbf{\Lambda}_v\begin{bmatrix}\mathbf{Q}_b^T\\ \mathbf{Q}_w^T\end{bmatrix}$.

3.5 Solving the Over-Fitting Problem

The proposed MGSVD-LDA algorithm, like other GSVD based algorithms, is susceptible to the over-fitting problem. As all the eigenvectors computed in the eigen-decomposition of $\mathbf{CC}^T$ are maintained and these eigenvectors are divided by the corresponding eigenvalues, as seen from (3.19), the impact of the random features on the eigenvectors corresponding to small eigenvalues gets amplified.

There are mainly three methods to address the over-fitting problem. The first one is a regularization method [31], [33] in which a small positive perturbation is introduced to a matrix in order to bring small changes to large eigenvalues relative to the changes in the small eigenvalues. Thus, the effect of over-fitting can be reduced when the eigenvectors are divided by the eigenvalues resulting from the perturbed matrix. The optimal perturbation parameter is estimated adaptively from the training samples through cross-validation. However, this process is time consuming. In the second method, the smaller eigenvalues and the corresponding eigenvectors are dropped [85]. But, there is no universal criterion to determine as to how many eigenvalues can be considered small enough to be dropped. Both these methods affect the main idea behind the GSVD technique in that the samples of different classes do not converge into distinct compact regions.

The third approach to fixing the over-fitting problem is to orthogonalize the basis of the discriminant subspace [86]. In this method, the basis is orthogonalized through a QR decomposition of $\mathbf{G}$. However, QR decomposition is computational inefficient for high dimensional data. Also the result linear algorithm cannot be kernelized.

We now propose a novel orthogonalization method to deal with the over-fitting problem of the GSVD-based algorithms. The basis of the discriminant subspace derived from the conventional GSVD mechanism is not orthogonal and the projection of the between-class or total scatters on each of the basis vectors is of unit length. Through orthogonalization, the basis vectors are rescaled so that the larger eigenvectors are assigned more discrimination capacity. The main idea of this method is to orthogonalize the basis of the discriminant subspace by means of the eigen-decomposition of an inner product matrix. Through orthogonalization, the basis vectors are re-scaled so that the larger eigenvectors are assigned more discrimination capacity. Thus, the over-fitting problem is effectively controlled. This method is equally efficient for low and high dimensional data and compatible with the process of kernelization.

First, an eigen-decomposition of $\mathbf{G}^T\mathbf{G}$ carried out as

$$\mathbf{G}^T\mathbf{G} = \mathbf{W}_r^T\mathbf{R}^{-2}\mathbf{W}_r = \boldsymbol{\theta}\boldsymbol{\pi}\boldsymbol{\theta}^T \tag{3.33}$$

where $\mathbf{W}_r \in \Re^{k\times r}$ consists of the left r columns of $\mathbf{W}$, $\boldsymbol{\theta} \in \Re^{r\times r}$ is an orthogonal matrix and $\boldsymbol{\pi}$ is a diagonal matrix. Then,

$$\mathbf{G}_o = \mathbf{G}\boldsymbol{\theta}\boldsymbol{\pi}^{-1/2} \tag{3.34}$$

is the transformation matrix with its columns mutually orthogonal. Since the size of the matrix $\mathbf{W}_r^T\mathbf{R}^{-2}\mathbf{W}_r$ is small, this orthogonalization step is computationally efficient. The proposed orthogonalized algorithm, referred as MGSVD-OLDA algorithm, is summarized as in Table 3.3.

Table 3.3: MGSVD-OLDA algorithm

Input: Training sample $\mathbf{x}_l$
Output: Transformation matrix $\mathbf{G}$
1. Use Equation (2.8) and (3.2) to obtain $\mathbf{H}_b$, $\mathbf{H}_w$ and $\mathbf{C}$;
2. Find $\mathbf{P}$ and $\mathbf{R}$ from $\mathbf{C}\mathbf{C}^T = \mathbf{P}\begin{bmatrix}\mathbf{R}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{P}^T$;
3. $\mathbf{P}_1 \leftarrow \mathbf{P}(:,1:k), \mathbf{P}_{11} \leftarrow \mathbf{P}_1(1:N,:), k = \text{rank}(\mathbf{C})$;
4. Find $\mathbf{W}$ through SVD of $\mathbf{P}_{11}$: $\mathbf{P}_{11} = \mathbf{U}\boldsymbol{\Sigma}_b\mathbf{W}^T$;
5. $\mathbf{Y}_k \leftarrow \mathbf{C}^T\mathbf{P}_1\mathbf{R}^{-2}\mathbf{W}$;
6. $\mathbf{G} \leftarrow \mathbf{Y}_k(:, 1:r)$, $r = \text{rank}(\mathbf{H}_b)$.
7. Obtain $\mathbf{W}_r$, $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ through eigen-decomposition of $\mathbf{G}^T\mathbf{G}$.
8. Find the orthogonalized transformation matrix $\tilde{\mathbf{G}}_o$ using (3.34).

As in the case of the proposed linear algorithm, the over-fitting problem in the proposed kernelized algorithm is taken care through a process of orthogonalization of the basis. To this end, we first obtain the eigen-decomposition of $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}}$ as

$$\tilde{\mathbf{W}}_v^T\tilde{\mathbf{R}}^{-2}\tilde{\mathbf{W}}_v = \tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\pi}}\tilde{\boldsymbol{\theta}}^T \tag{3.35}$$

where $\tilde{\mathbf{W}}_v$ consists of the left most v columns of $\tilde{\mathbf{W}}$, and $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\pi}}$ are, respectively, the eigenvector and eigenvalue matrices. Then, an orthogonalized $\tilde{\mathbf{G}}$, namely $\tilde{\mathbf{G}}_o$, is obtained such that

$$\tilde{\mathbf{G}}_o^T = \tilde{\boldsymbol{\pi}}^{-1/2}\tilde{\boldsymbol{\theta}}^T\tilde{\mathbf{G}}^T \tag{3.36}$$

The proposed orthogonalized algorithm, referred as MGSVD-OKDA algorithm, is summarized as in Table 3.4.

Table 3.4: MGSVD-OKDA algorithm

Training stage
Input:Training sample $\mathbf{x}_l$
Output: $\tilde{\mathbf{G}}_o^T$ in (3.29)
1. Form the kernel matrix $\mathbf{K}$ based on (3.26) and (3.26) and the kernel function chosen;
2. Evaluate the matrix $\mathbf{\Gamma}\mathbf{\Gamma}^T$ given by (3.25) using (3.27);
3. Find $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{R}}$ from $\mathbf{\Gamma}\mathbf{\Gamma}^T = \tilde{\mathbf{P}}\begin{bmatrix}\tilde{\mathbf{R}}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\tilde{\mathbf{P}}^T$;
4. $\tilde{\mathbf{P}}_1 \leftarrow \tilde{\mathbf{P}}(:,1:z), \mathbf{P}_{11} \leftarrow \mathbf{P}_1(1:N,:),\ z = \text{rank}(\mathbf{\Gamma}\mathbf{\Gamma}^T)$;
5. Find $\tilde{\mathbf{W}}$ through SVD of $\tilde{\mathbf{P}}_{11}$: $\mathbf{P}_{11} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}_b\tilde{\mathbf{W}}^T$;
6. $\mathbf{\Lambda}_v \leftarrow$ the first v rows of $\tilde{\mathbf{W}}^T\tilde{\mathbf{R}}^{-2}\tilde{\mathbf{P}}_1^T, v = \text{rank}(\mathbf{\Phi}_b^T\mathbf{\Phi}_b)$;
7. Obtain $\tilde{\mathbf{W}}_v$, $\tilde{\mathbf{\theta}}$ and $\tilde{\pi}$ through eigen-decomposition of $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}}$.
8. Find the orthogonalized transformation matrix $\tilde{\mathbf{G}}_o$ using (3.36).
Classification stage
Input:Test vector $\mathbf{x}_t$
Output: Weight vector w in (3.32);
9. Evaluate $\mathbf{q}_l$ as in (3.30);
10. Form $\mathbf{Q}_b$ and $\mathbf{Q}_w$ in (3.31);
11. $w \leftarrow \mathbf{\Lambda}_v\begin{bmatrix}\mathbf{Q}_b^T\\ \mathbf{Q}_w^T\end{bmatrix}$.

3.6 Experiments

In this section, two sets of experiments are conducted for empirical evaluation of the performance of the proposed MGSVD-LDA and MGSVD-KDA algorithms and their orthogonalized versions. The first set of experiments is designed to evaluate the performances of the proposed linear algorithms, MGSVD-LDA and its orthogonalized version, MGSVD-OLDA, and four other linear algorithms, using small sample size datasets. The four linear algorithms chosen for comparison are LDA/GSVD [34], RLDA [33], PCA+LDA [85], and PCA [25] algorithms. The second set of experiments evaluates the performance of the kernelized versions of the above linear algorithms, namely MGSVD-KDA, MGSVD-OKDA, KDA/GSVD [76], KRDA [75], KPCA+LDA [73] and KPCA [57] algorithms, in comparison with one another and with that of the LDA algorithm using large sample size datasets. The execution platform used is Pentium 4, 2.8 GHz CPU, 1.0 GB RAM and WinXP operating system. Ten databases are used in our experiments. Among them, FERET [90], [91], YALE [93], AR [92] and ORL [97] are human face databases, and Dataset1 [94], Dataset2 and Dataset3 [95] are text document databases. The images of human face databases are preprocessed to move the faces to the centers of the images and to crop them to include mainly the face part. The other three, the spoken letter database, Isolet [96], the molecule conformation database, MUSK [96] and handwritten digital database, MNIST [98] are large sample size databases, where the small sample size problem does not occur. These databases are described briefly hereunder.

FERET face database contains 1564 sets of images for a total of 14,126 images and includes 1199 individuals and 365 duplicate sets of images (a duplicate set is a second set

of images of a person already in the database and was usually taken on a different day). A subset of the FERET database is used in our experiments. This subset includes 280 images of 28 individuals (each individual has ten images). The original images are cropped into 168 x 128 pixel images with 256 gray scales. In each run, 6 images from each class are randomly chosen for training and the remaining 4 images are used for testing.

YALE face database contains 165 images of 15 individuals (classes), each having 11 320 x 243 pixel 256 gray scale images with different facial expression and lighting conditions. The images contain variations with the following facial expressions or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image is manually cropped into a size of 92 x 112 pixels and is rearranged as a 10,304-dimensional vector. From each class, 5 images are randomly selected for training and the remaining 6 are used for testing.

AR face database contains over 4,000 images of 126 individuals (classes), all of which are frontal view faces with different facial expressions, illumination conditions, and occlusions like sun glasses and scarf. Our experiments involve 67 individuals each having 13 images. The original color images that are of 768 x 576 pixels and 24 bits of depth are converted and cropped into 140 x 126 pixel images with 256 gray scales. In each run, 15 of the 67 classes (individuals) are randomly drawn and 6 images from each class are randomly chosen for training and the remaining 7 images are used for testing.

ORL face database contains 40 persons/classes with each having 10 images. The images are taken at different times with varying lighting conditions, facial expressions, and facial details. All individuals are in an upright, frontal position (with tolerance for

some side movement). All these images are 92 × 112 pixel grey scale pictures and are expressed as 10304-dimension vectors in the input space. From each class, 6 images are randomly selected for training and the remaining 4 for testing.

Dataset1 is a text document database, derived from the TREC-5, TREC-6, and TREC-7 collections. It consists of 7 clusters of 30 documents, each document being arranged in a 7,454-dimensional vector space. From each cluster (class) 20 documents are randomly selected for training and the remaining 10 documents are used for testing.

Dataset2 is also a text document database, derived from the Reuters-21578 text categorization test collection Distribution 1.0. This dataset has 4 clusters each having 80 elements represented in a 2,887-dimensional vector space. One-half of the dataset is randomly chosen for training and the other half used for testing.

Dataset3 is also derived from the Reuters-21578 text categorization test collection Distribution 1.0, but contains 5 clusters each having 98 documents. Each document is represented as a vector of dimension 3,759. As in the case of Dataset2, one-half of the elements are randomly chosen for training and the other half for testing.

Isolet is a spoken letter database with 150 subjects speaking the name of each letter of the 26 alphabets (classes) twice. The number of training instances is 6238 and that of the test instances is 1559. Each instance is described by 617 attributes which are continuous, real valued, and scaled in the range -1.0 to 1.0. In the experiments, for each alphabet 25 instances are randomly chosen as training samples and 40 instances are chosen for testing.

MUSK is a two class database containing 6,598 molecule conformations, which are categorized into musks and non-musks. The conformations are described by 166 features.

From each category 250 conformations are randomly drawn for training and another 200 conformation are used for testing.

MNIST handwritten digit database includes 10 digits/classes from 0 to 9. It has 60,000 images and a test set of 10,000 images, each of which is a 28 × 28 pixel grey scale image and is represented by a 784-dimension vector. In the experiments, 50 images from each class are randomly drawn as training samples and the same number of images is chosen for testing.

Table 3.5 gives a summary of the databases used in our experiments. Appendix B gives the images of one subject (class) of each of the four human face databases in this chapter.

The two kernel functions used in our experiments are the Gaussian radial basis function (RBF) kernel, $k(\mathbf{x}_l, \mathbf{x}_h) = \exp(-\frac{\|\mathbf{x}_l - \mathbf{x}_h\|^2}{\sigma})$, where $\|\cdot\|$ denotes the Euclidean 2-norm and $\sigma > 0$, and the nonhomogeneous polynomial kernel, $k(\langle \mathbf{x}_l, \mathbf{x}_h \rangle) = (\langle \mathbf{x}_l, \mathbf{x}_h \rangle + 1)^d$, where d is an integer.

Except for GSVD-based LDA algorithm, for all the other linear and nonlinear algorithms, we specify a mechanism to determine the parameters. For the PCA+LDA and KPCA+LDA algorithms, the largest $N-1$ eigenvalues and the corresponding eigenvectors are used in the PCA stage, where N is the number of the classes. The parameters are estimated through cross validation by using a part of the training samples to carry out the actual training and the remaining for the estimation. The optimal regulation parameters for the RLDA and the optimal kernel parameters, d or σ, of all the kernelized algorithms are estimated using the κ-fold cross validation method, where $\kappa \geq 10$. The parameter

corresponding to the highest average recognition accuracy over the κ iterations is chosen as the optimal parameter. The KRDA algorithm has two parameters, perturbation and kernel, to be estimated; hence, a double cross-validation is needed.

Since our focus is on feature extraction, a simple classifier, the nearest neighbor classifier [3], is chosen to be used in all the algorithms so that the differences in the recognition accuracy of the various algorithms can be attributed to feature extraction process of the algorithms rather than the classifier employed. For each database, ten sets of samples are randomly drawn and each algorithm is run using one set at a time. The average recognition rate and execution time of an algorithm is determined as an average taken over ten runs of the algorithm using the ten data sets. Here, execution time includes both the training and testing times. Parameter estimation time is not explicitly given, but can be estimated as the product of κ, the number(s) of parameter candidates, and the execution time.

(a) Performance evaluation using small sample size databases

In this set of experiments, we assess the performance in terms of the recognition accuracy and the execution time of the six linear algorithms, MGSVD-LDA, MGSVD-OLDA, LDA/GSVD, RLDA, PCA+LDA and PCA, using small sample size databases, FERET, YALE, AR, ORL, Dataset1, Dateset2 and Dataset3. The results of the experiments are given in Table 3.6 and Table 3.7 from which we make the following observations:

1) For the four high-dimensional face databases, memory overflow occurs when the LDA/GSVD algorithm is used; whereas the RLDA, PCA+LDA, PCA and the proposed

MGSVD-LDA algorithm do not encounter this problem. Note that the GSVD-based algorithms are a special case of the RLDA algorithm with zero perturbation of the eigen values or in the values of the elements of the scatter matrix.

2) For the three text databases, the proposed MGSVD-LDA algorithm maintains the high level of recognition accuracy of the LDA/GSVD algorithm. However, the execution time of the former is significantly lower than that of the latter. Specifically, the execution times of the proposed linear algorithm are reduced from those of the LDA/GSVD algorithm by 99.5%, 94.6% and 92.4%for Dataset1, Dataset2 and Dataset3, respectively.

Table 3.5: Summary of databases

Database	Size of database	Dimension	Number of classes	Number of training samples	Number of test samples
FERET	14126	21504	28	168	112
YALE	165	10304	15	75	90
AR	4000	17640	15	90	105
ORL	400	10304	40	240	160
Dataset1	210	7454	7	140	70
Dataset2	320	2887	4	160	160
Dataset3	490	3759	5	245	245
Isolet	7797	617	26	650	1040
MUSK	6598	166	2	500	400
MNIST	70,000	784	10	500	500

Table 3.6: Recognition rate (%) and execution time (seconds) of linear algorithms with small samples size face datasets

Database	FERET		YALE		AR		ORL	
Algorithm	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
MGSVD-LDA	**97.43**	**1.20**	**96.92**	**1.28**	**95.71**	**0.71**	**96.37**	**1.44**
MGSVD-OLDA	**99.4**	**1.21**	**98.71**	**1.29**	**96.86**	**0.72**	**97.37**	**1.45**
LDA/GSVD	Memory overflow		Memory overflow		Memory overflow		Memory overflow	
RLDA	99.14	7.72	98.71	9.98	96.85	3.92	97.18	11.64
PCA+LDA	95.37	1.19	95.98	1.26	95.15	0.69	95.12	1.42
PCA	90.37	1.09	91.23	1.15	92.5	0.62	92.9	1.29

Table 3.7: Recognition rate (%) and execution time (seconds) of linear algorithms with small samples size text document datasets

Database	Dataset1		Dataset2		Dataset3	
Algorithm	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
MGSVD-LDA	**94.1**	**0.29**	**81.9**	**0.49**	**90.10**	**1.21**
MGSVD-OLDA	**96.4**	**0.30**	**83.54**	**0.52**	**91.79**	**0.11**
LDA/GSVD	94.0	29.54	81.9	9.32	90.10	15.72
RLDA	96.5	2.35	83.52	4.32	91.78	1.20
PCA+LDA	93.3	0.29	80.2	0.49	89.76	0.09
PCA	89.9	0.25	60.2	0.45	79.3	0.08

3) In some situations such as when the GSVD-based algorithms are employed to the FERET, YALE, AR and ORL databases, the orthogonalized algorithm significantly outperforms its original form. Overall, the orthogonalized algorithm is competitive to the other algorithms in terms of recognition accuracy and computational efficiency.

The above observations suggest that, compared to the LDA/GSVD algorithm, the proposed MGSVD-LDA algorithm overcomes the high computational complexity problem, works for patterns of large dimension without memory overflow. The over-fitting problem that sometime is encountered in the GSVD-based algorithms is effectively resolved with little effect on the computation load.

(b) Performance evaluation using large sample size datasets

This set of experiments is devoted to assessing the recognition accuracy and execution times of the four kernelized algorithms with the three large sample size databases, Isolet, MUSK and MNIST. Since in the case of a large sample size databases, the associated scatter matrix does not have the singularity problem, we also compare the recognition accuracy of the kernelized algorithms with that of the LDA algorithm, which is a linear algorithm not suitable for discriminant analysis of small sample size databases. The Table 3.8 gives the results on the recognition rate and execution time of the algorithms. From the results we observe the following:

1) Compared to the LDA algorithm, all the kernelized algorithms enhance the recognition accuracy significantly. Among all the algorithms considered, the proposed algorithms give the highest recognition accuracy.

2) Orthogonalization improves the recognition accuracy of the GSVD-based algorithms.

From the simulation results, we can see that kernelization of linear discriminant analysis algorithms is a necessary requirement for their applications to large sample size databases. This necessity arises from the fact that in large sample size databases, the samples are not linear separable, and therefore, linear algorithms cannot be effective for discriminant analysis. The process of kernelization facilitates the distribution of samples to be linearized and simplified in a high dimensional space. Over-fitting occurs to the GSVD-based algorithms but orthogonalization overcomes this problem.

Table 3.8: Recognition rate (%) and execution time (seconds) of kernelized algorithms with large samples size datasets

Database		Isolet		MUSK		MNIST	
Algorithm		Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
Linear	FLDA	85.9	3.83	82.67	0.09	85.85	7.06
Polyn.	**MGSVD-KDA**	**92.91**	**53.02**	**92.47**	**2.53**	**92.72**	**136.66**
	MGSVD-OKDA	**94.25**	**53.50**	**93.50**	**2.55**	**94.52**	**137.01**
	KDA-GSVD	92.90	53.91	92.44	2.70	92.71	136.65
	KRDA	93.16	370.64	92.33	279.28	93.83	1163.67
	KPCA+LDA	91.9	53.58	91.5	2.49	91.53	111.91
	KPCA	89.4	51.21	90.2	2.27	90.01	109.54
RBF	**MGSVD-KDA**	**93.5**	**73.91**	**93.34**	**3.65**	**94.2**	**198.0**
	MGSVD-OKDA	**95.23**	**74.51**	**94.22**	**3.71**	**95.1**	**198.7**
	KDA-GSVD	93.4	74.0	93.34	3.66	94.2	198.1
	KRDA	93.5	459.8	93.6	315.5	93.9	1532.5
	KPCA+LDA	92.3	73.92	92.9	3.21	92.2	197.5
	KPCA	91.5	72.52	91.8	2.82	91.4	180.2

3.7 Summary

In this chapter, the modified generalized singular value decomposition has been integrated with linear discriminant analysis resulting in the development of a new algorithm, the MGSVD-LDA algorithm, which successfully overcomes the singularity problem of the scatter matrices of the traditional FLDA methods and deals effectively with the computational complexity problem of the LDA/GSVD algorithm. In the new algorithm, the GSVD framework used in the LDA/GSVD algorithm has been modified by replacing the SVD of a high-dimensional matrix with the eigen-decomposition of a small size inner product matrix, thus circumventing the direct calculation of a high-dimensional singular vector matrix. A kernelized version of the proposed linear algorithm has been developed for the discriminant analysis of samples that are not linearly separable. An orthogonalization technique has been proposed to deal with the over-fitting problem of the GSVD-based algorithms. The main idea of this technique is to orthogonalize the basis of the discriminant subspace derived from the GSVD-based algorithms through eigen-decomposition.

The proposed MGSVD-LDA algorithm has been demonstrated to deal effectively with the computational problem associated with the high dimensionality of the patterns, when the LDA/GSVD algorithm completely fails. It has been shown that even in the case when the dimension of the patterns is not so high and the LDA/GSVD algorithm works, the proposed MGSVD-LDA algorithm provides a solution to the pattern recognition problem that is significantly less time consuming and more memory-space efficient and has an equally high recognition accuracy. It has also been shown that the orthogonalized algorithm, the MGSVD-OLDA algorithm, significantly outperforms its original form

with high recognition accuracy and low computational load when the over-fitting problem occurs.

Overall, the simulation results have shown that the MGSVD-based linear and kernel algorithms, especially their orthogonalized versions, provide high recognition accuracy with low computational load.

Chapter 4

Class Structure of Linearly Independent Patterns in an MGSVD- Derived Discriminant Subspace

4.1 Introduction

In Chapter 3, new GSVD-based linear and nonlinear algorithms have been developed for discriminant analysis. It has been shown that these algorithms deal effectively with the small sample size problem and are capable of accomplishing the task of feature extraction with low computational complexity and high recognition accuracy. In the MGSVD-LDA algorithm, the LDA/GSVD's operation of singular value decomposition of $\mathbf{C}$ is replaced by the eigen-decomposition of the inner product matrix $\mathbf{CC}^T$ in order to reduce its computational complexity. The accumulated round-off errors arising from forming the inner product matrix and that from carrying out its eigen-decomposition could become nontrivial when the samples have large dimensionality. Also, if the inner product matrix is ill-conditioned, its eigenvectors become sensitive to these errors [78], [79].

Although the eigen-decomposition of inner product matrices has been widely exploited in pattern recognition and mathematicians have studied its numerical stability [79], [80], few attempts have been made on investigating the implication of the numerical errors in

feature extraction techniques involving eigen-decomposition of inner product matrices. Therefore, it is important to study the effects of the numerical errors caused by the round-off errors introduced in carrying out the eigen-decomposition of the inner product matrices of the proposed algorithms. A question that also needs to be answered is as to how the accumulated computational errors influence the accuracy of the feature extraction in the implementation of the proposed algorithms.

The purpose of the proposed linear and nonlinear feature extraction methods, like other feature extraction techniques, has been to find a lower dimensional discriminant subspace, where different classes occupy compact and disjoint regions. However, in order to determine the impact of finite arithmetic in implementing the proposed algorithm on the accuracy of the feature extraction, one needs to have a better understanding of the class structure of the samples in the derived discriminant subspace.

The purpose of this chapter is first to study the class structure of the samples in the discriminant subspace derived from the proposed MGSVD algorithms and then to investigate whether this structure can be used to assess the numerical errors caused when the proposed algorithms are implemented.

In Section 4.2, a theorem is established to determine the class structure of linearly independent samples in the discriminant subspace derived from the proposed MGSVD algorithms. The numerical error incurred in the computational process of obtaining the inner product matrix and in carrying out its eigen-decomposition is investigated in Section 4.3. A scheme is described in this section for using the above theorem to estimate the numerical errors incurred in implementing the MGSVD algorithms and to adjust the kernel parameters to minimize the numerical errors in the implementation of the proposed

kernelized algorithm. In Section 4.4, experiments are designed to demonstrate the validity of the theorem presented to establish the class structure of linearly independent samples and to evaluate the effects of the numerical errors in implementing the proposed linear and kernelized algorithms.

4.2 Class Structure of Datasets with Linearly Independent Samples

The proposed MGSVD-LDA algorithm provides an optimal transformation matrix that projects the input samples onto a lower-dimensional discriminant subspace, where different class categories occupy compact and disjoint regions. The extent of separability between the samples belonging to different classes and the proximity of the samples belonging to the same class could provide an insight into the class structure of the samples in the subspace derived by a feature extraction algorithm. In this section, we first establish a theorem to gain an insight into the class structure of linearly independent samples in the discriminant subspace derived by the MGSVD-LDA algorithm. Then, it is shown that, in view of this theorem, a similar insight can be gained for nonlinearly distributed input samples.

Lemma 4.1: Given a set of m-dimensional linearly independent samples consisting of N classes with the ith class having n_i samples and sample size being n, we have

$$\text{rank}(\mathbf{H}_b) = N-1 \quad \text{and} \quad \text{rank}(\mathbf{C}) = \text{rank}(\mathbf{H}_b) + \text{rank}(\mathbf{H}_w).$$

Proof: Let us define the following three matrices

$$\mathbf{H}_b = \frac{1}{\sqrt{n}}\left[\sqrt{n_1}(\mathbf{c}^{(1)} - \mathbf{c}), \ldots, \sqrt{n_N}(\mathbf{c}^{(N)} - \mathbf{c})\right] \tag{4.1}$$

$$\mathbf{H}_w = \frac{1}{\sqrt{n}}\Big[(\mathbf{x}_1 - \mathbf{c}^{(1)}),\ldots,(\mathbf{x}_{n_1} - \mathbf{c}^{(1)}),(\mathbf{x}_{n_1+1} - \mathbf{c}^{(2)}),\ldots,(\mathbf{x}_{n_1+n_2} - \mathbf{c}^{(2)}),\ldots,$$

$$(\mathbf{x}_{n-n_N+1} - \mathbf{c}^{(N)}),\ldots,(\mathbf{x}_n - \mathbf{c}^{(N)})\Big] \tag{4.2}$$

$$\mathbf{H}_t = \frac{1}{\sqrt{n}}\big[(\mathbf{x}_1 - \mathbf{c}),\ldots,(\mathbf{x}_n - \mathbf{c})\big] \tag{4.3}$$

where the global centroid is given by

$$\mathbf{c} = \frac{1}{n}\sum_{l=1}^{n}\mathbf{x}_l \tag{4.4}$$

and the centroid of ith class is given by

$$\mathbf{c}^{(i)} = \frac{1}{n_i}\sum_{l=k_{i-1}+1}^{k_i}\mathbf{x}_l \tag{4.5}$$

where $k_i = n_1 + n_2 + \ \ldots\ + n_i$, and $\mathbf{x}_l$ is the lth ($l = 1,\ 2,\ \ldots,\ n$) sample vector, $n = \sum_{i=1}^{N} n_i$.

The between-class and within-class and total scatter matrices can, respectively, be defined as

$$\mathbf{S}_w = \mathbf{H}_w\mathbf{H}_w^T,\ \mathbf{S}_b = \mathbf{H}_b\mathbf{H}_b^T,\ \mathbf{S}_t = \mathbf{H}_t\mathbf{H}_t^T \tag{4.6}$$

It has been shown in [34] that

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = \mathbf{C}^T\mathbf{C} \tag{4.7}$$

We now prove by contradiction that $\text{rank}(\mathbf{H}_t) = n-1$.

We know that $\text{rank}(\mathbf{H}_t) \le n-1$. Assume that $\text{rank}(\mathbf{H}_t) \ne n-1$. Then, it follows that $\text{rank}(\mathbf{H}_t) < n-1$, and in this case the left most n − 1 columns of $\mathbf{H}_t$ must be linearly dependent. That is, there exists a set of numbers $\{a_j\}_{j=1,\ldots,n-1}$, not all zero, such that

$$\sum_{j=1}^{n-1} a_j \mathbf{x}_j = \sum_{j=1}^{n-1} a_j \mathbf{c} = \sum_{j=1}^{n} (\frac{1}{n} \sum_{i=1}^{n-1} a_i) \mathbf{x}_j$$

from which it follows that

$$\mathbf{x}_n = \sum_{j=1}^{n-1} (\frac{n a_j}{\sum_{i=1}^{n-1} a_i} - 1) \mathbf{x}_j$$

implying that the sample $\mathbf{x}_n$ is a linear combination of the other $n - 1$ samples. This contradicts the given condition that the samples are linearly independent. Consequently, the assumption that $\text{rank}(\mathbf{H}_t) < n-1$ is not valid. Thus, we have $\text{rank}(\mathbf{H}_t) = n-1$.

Since $\text{rank}(\mathbf{H}_w) \le n-N$, $\text{rank}(\mathbf{H}_b) \le N-1$, and $\text{rank}(\mathbf{H}_b) + \text{rank}(\mathbf{H}_w) \ge \text{rank}(\mathbf{H}_t)$, it follows that $\text{rank}(\mathbf{H}_w) = n-N$, $\text{rank}(\mathbf{H}_b) = N-1$, and $\text{rank}(\mathbf{H}_b) + \text{rank}(\mathbf{H}_w) = \text{rank}(\mathbf{H}_t)$. From (4.7), and the fact that $\text{rank}(\mathbf{H}_t) = \text{rank}(\mathbf{S}_t)$ and $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{C}^T\mathbf{C})$, we have $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{H}_t) = \text{rank}(\mathbf{H}_b) + \text{rank}(\mathbf{H}_w)$.

■

Lemma 4.2: Given a set of *m*-dimensional linearly independent samples that consisting of *N* classes with the *i*th class having n_i samples, then the discriminative transformation matrix **G** derived from the MGSVD-LDA algorithm satisfies the relations

$$\mathbf{G}^T\mathbf{S}_t\mathbf{G} = \mathbf{G}^T\mathbf{S}_b\mathbf{G} = \mathbf{I}_r \tag{4.8}$$

and

$$\mathbf{G}^T\mathbf{S}_w\mathbf{G} = \mathbf{0} \tag{4.9}$$

where $\mathbf{I}_r$ is an $r \times r$ identity matrix with $r = \text{rank}(\mathbf{H}_b) = N - 1$.

Proof: Given that the samples of a datasets are linearly independent, we know that rank(**C**) = rank ($\mathbf{H}_b$) + rank ($\mathbf{H}_w$). Using this result in (3.7), we have s = 0. Thus, $\mathbf{D}_b$ in

(3.5) and $\mathbf{D}_w$ in (3.6) vanish, and (3.20) and (3.21) become

$$\mathbf{Y}_k^T \mathbf{S}_b \mathbf{Y}_k = \mathbf{\Sigma}_b^2 = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{k \times k}$$

$$\mathbf{Y}_k^T \mathbf{S}_w \mathbf{Y}_k = \mathbf{\Sigma}_w^2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{k-r} \end{bmatrix}_{k \times k} \tag{4.10}$$

where $\mathbf{I}_{k-r}$ is a (k − r) x (k − r) identity matrix with k = rank ($\mathbf{H}_t$) = n − 1. Since $\mathbf{G}$ consists of the r left most columns of $\mathbf{Y}_k$ and $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$, (4.8) and (4.9) follow.

■

Theorem 4.1: Given a set of m-dimensional linearly independent samples consisting of N classes with the ith and jth classes having n_i and n_j samples, respectively, $i, j = 1, \ldots, N$, $\mathbf{x}_h$ and $\mathbf{x}_l$ are two samples from the ith and jth classes, respectively, the Euclidean distance between the two corresponding sample vectors, $\mathbf{G}\mathbf{x}_h$ and $\mathbf{G}\mathbf{x}_l$, in the discriminant subspace derived from the MGSVD-LDA algorithm, is given by

$$dist(\mathbf{G}^T\mathbf{x}_l, \mathbf{G}^T\mathbf{x}_h) = \begin{cases} 0, & \text{if } i = j \\ \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}, & \text{if } i \neq j \end{cases} \quad i, j = 1, 2, \ldots, N \tag{4.11}$$

where $\mathbf{G}$ is the transformation matrix derived from the proposed MGSVD-LDA algorithm.

Proof: Since $\mathbf{G}^T\mathbf{S}_w\mathbf{G}$ is a semi-positive definite matrix and according to Lemma 4.2 (see (4.9)), it is a zero matrix, we have $\mathbf{G}^T\mathbf{H}_w = 0$, that is,

$$\mathbf{G}^T(\mathbf{x}_l - \mathbf{c}^{(i)}) = \mathbf{G}^T(\mathbf{x}_h - \mathbf{c}^{(j)}) = 0, \text{ or}$$

$$\mathbf{G}^T\mathbf{x}_l = \mathbf{G}^T\mathbf{c}^{(i)}, \text{ and } \mathbf{G}^T\mathbf{x}_h = \mathbf{G}^T\mathbf{c}^{(j)} \tag{4.12}$$

Case (1): i=j

Since $\mathbf{G}^T\mathbf{x}_l = \mathbf{G}^T\mathbf{x}_h = \mathbf{G}^T\mathbf{c}^{(j)}$, we obtain

$$dist(\mathbf{G}^T\mathbf{x}_l, \mathbf{G}^T\mathbf{x}_h) = 0 \tag{4.13}$$

Case (2): $i \neq j$

Using (4.12), we have

$$\begin{aligned} & dist(\mathbf{G}^T\mathbf{x}_l, \mathbf{G}^T\mathbf{x}_h) \\ & = dist(\mathbf{G}^T(\mathbf{c}^{(i)} - \mathbf{c}), \mathbf{G}^T(\mathbf{c}^{(j)} - \mathbf{c})) \\ & = dist(\tfrac{1}{\sqrt{n_i}}\mathbf{G}^T\mathbf{g}_i, \tfrac{1}{\sqrt{n_j}}\mathbf{G}^T\mathbf{g}_j) \end{aligned} \tag{4.14}$$

where $\mathbf{g}_i$ and $\mathbf{g}_j$ are, respectively, the *i*th and *j*th columns of $\mathbf{H}_b$. Thus, from (3.13) and (4.10), we have

$$\mathbf{G}^T\mathbf{H}_b = [\mathbf{I}_{N-1}, \mathbf{0}]\mathbf{U}^T = \mathbf{U}_1^T \tag{4.15}$$

where $\mathbf{U}_1$ consists of the N-1columns of $\mathbf{U}$ leftmost , and $\mathbf{U} \in \Re^{N\times N}$ is the left singular vector matrix of $\mathbf{P}_{11}$. The Nth column of $\mathbf{U}$, $\mathbf{u}$, which is excluded from $\mathbf{U}_1$, corresponds to $\mathbf{P}_{11}$'s null space, i.e., $\mathrm{u}^{\mathrm{T}}\,\mathbf{P}_{11} = 0$. Since

$$\begin{bmatrix} \mathbf{P}_{11} \\ \\ \mathbf{P}_{12} \end{bmatrix} = \mathbf{P}_1 = \mathbf{C}\mathbf{Q}_1\mathbf{R}^{-1} = \begin{bmatrix} \mathbf{H}_b^T\mathbf{Q}_1\mathbf{R}^{-1} \\ \\ \mathbf{H}_w^T\mathbf{Q}_1\mathbf{R}^{-1} \end{bmatrix} \tag{4.16}$$

we have $\mathbf{P}_{11} = \mathbf{H}_b^T\mathbf{Q}_1\mathbf{R}^{-1}$. Therefore, $\mathbf{u}^T\mathbf{H}_b^T = \mathbf{0}$ or $\mathbf{H}_b\mathbf{u} = \mathbf{0}$. Solving this equation gives the components of $\mathbf{u}$ as $u_p = \sqrt{n_p / n}$, for $p = 1,\ldots, N$. Letting $\mathbf{d}_i$ and $\mathbf{d}_j$ to be, respectively, the *i*th and *j*th columns of $\mathbf{U}_1^T$, and concatenating $\mathbf{d}_i$ with $\mathbf{u}_i$, and $\mathbf{d}_j$ with $\mathbf{u}_j$, the resulting two vectors become two orthogonal unit vectors, which are the *i*th and *j*th columns of the

matrix U . Using these results in (4.14) , we have

$$
\begin{aligned}
& dist(\mathbf{G}^T\mathbf{x}_l, \mathbf{G}^T\mathbf{x}_h) \\
&= \left\| \frac{1}{\sqrt{n_i}}\mathbf{d}_i, \frac{1}{\sqrt{n_j}}\mathbf{d}_j \right\| \\
&= \sqrt{\frac{1}{n_i} + \frac{1}{n_j} - \left(\frac{\mathbf{u}_i}{\sqrt{n_i}} - \frac{\mathbf{u}_j}{\sqrt{n_j}}\right)^2} \\
&= \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}
\end{aligned}
$$

■

Theorem 4.1 provides the theoretical distance between two samples in the MGSVD-discriminant subspace given that all the samples are linearly independent. If the two samples belong to the same class in a given dataset, then according to this theorem, they are merged into a single point in the discriminant subspace. On the other hand, if they belong to two different classes, the distance between them is determined by the number of samples in the respective classes. When the samples are linearly independent, each class condenses into a distinct point in the discriminant subspace derived from the MGSVD-LDA algorithm. Thus, if the samples of a dataset are linearly independent, the class structure of the samples can be effectively captured through the linear operation of the MGSVD-LDA algorithm. Hence, for small sample size datasets in which the samples are linearly independent, the performance of a linear algorithm cannot be improved by subjecting it to the process of kernelization. On the other hand, if the samples of a dataset are not linearly independent, i.e., when the sample size of the dataset is larger than the sample dimension, the samples of a class cannot condense into a single point in the

derived MGSVD-LDA discriminant subspace. Divergence of class samples in the discriminant subspace leads to nonlinear class boundaries, and thus affects the recognition accuracy. In this case, the linear independence condition can be established by using the proposed kernelized MGSVD-KDA algorithm [54]. Thus, the class structure of the datasets with samples that are not linearly independent would be the same in the kernel discriminant subspace derived by using the MGSVD-KDA algorithm as the one provided by the above theorem.

4.3 Numerical Error Analysis of the Proposed MGSVD Algorithms

The proposed MGSVD-LDA algorithm includes the operations of inner product in computing the inner product matrix $\mathbf{C}\mathbf{C}^T$ and its eigen-decomposition. The round-off errors accumulated through these operations could be nontrivial. Normally, the eigenvalues of a symmetric matrix are well-conditioned. However, the sensitivity of the eigenvector or the subspace represented by a subset of the eigenvectors to the numerical error depends on the proximity between the associated eigenvalues and the rest of the eigenvalues [78] - [81].

In the proposed MGSVD-LDA algorithm, the sensitivity of the range represented by $\mathbf{P}_1$ depends on the minimum non-zero eigenvalue of $\mathbf{C}\mathbf{C}^T$. The smaller the minimum non-zero eigenvalue, the more sensitive the range to a perturbation. In such a case, the numerical error will amplify the deviation between the computed range $\hat{\mathbf{P}}_1$, and the actual range $\mathbf{P}_1$. This deviation, in turn, will result in an angular difference between $\hat{\mathbf{G}}$, the computed transformation matrix and $\mathbf{G}$, the actual transformation matrix. Consider the

distance between two samples in the discriminant subspace, which is generally used as a measure of similarity between the objects in recognition problems. The angular difference between $\hat{\mathbf{G}}$ and $\mathbf{G}$ will be reflected in the distance between the two samples. The error in the computed distance is the result of the accumulated computational errors, and hence, represents the degree to which the numerical errors influence the accuracy of the feature extraction algorithm. The problem of determining the numerical error of the proposed MGSVD-LDA algorithm finally boils down to finding the actual (theoretical) distance between two samples. Once such a distance is obtained, it can be compared with the computed one in order to evaluate the effects of the numerical errors on the performance of the algorithm.

It was shown in the previous section that according to Theorem 4.1, if the two samples belong to the same class in a given dataset, then according to this theorem, they are merged into a single point in the discriminative subspace, and the distances between the points depend only on the respective numbers of the samples in the corresponding classes. Note that distance between two samples in the computed discriminant subspace represented by $\hat{\mathbf{G}}$ will have an error term resulting from the computational errors accumulated in calculating the matrix $\hat{\mathbf{G}}$. The magnitude of this error term is the difference between the actual distance obtained from Theorem 4.1 and the computed one. If the maximum of such errors is much smaller than the minimum computed inter-class distance, the feature extraction can be considered reliable for a given database. The maximum of these differences, denoted as ρ, is used as a metric of numerical errors. In case ρ is significantly larger than the computed minimum inter-class distance, each class will not merge into a single point in the discriminative subspace. Thus, the derived

discriminative subspace cannot be considered reliable. A nonlinear kernel function has to be used to adjust the input matrix to build the linearly separable condition by adjusting the kernel parameter appropriately.

Similar to the linear case, the kernelized MGSVD-KDA algorithm should also be examined for the effects of numerical errors. As a matter of fact, the numerical errors of the kernelized algorithm is of greater concern, since the round-off errors introduced due to the inner product operations could be amplified by several folds by the involvement of the kernel function. Theorem 4.1 still provides a basis of analyzing the numerical errors of the kernelized algorithm. Recall that the results of Theorem 4.1 is applicable to situations where the samples are linearly independent, and since the kernelization establishes this independence for samples that are otherwise not so, the procedure of analyzing the numerical errors of the linear algorithm described earlier can also be applied to the samples in the high dimensional feature space. The linear independence for the mapped samples can be easily checked by examining whether or not the kernel matrix $\mathbf{K}$ is of full rank [54]. The numerical error metric ρ corresponding to the kernelized algorithm can be used to adjust the kernel parameters so as to minimize the effects of the numerical errors.

4.4 Experiments

In this section, simulations are carried out in support of the results established in this chapter. The first set of experiments is designed to illustrate that for datasets with linearly independent samples, linear algorithms are quite effective and that the process of kernelization does not help in improving their performance. For this purpose, six

kernelized algorithms, namely MGSVD-KDA, MGSVD-OKDA, KDA/GSVD, KRDA, KPCA+LDA, and KPCA, are run using small sample size datasets. The corresponding linear algorithms were run in Chapter 3 using the same linearly independent datasets. The second set of experiments is designed to evaluate the effects of the numerical errors arising from the implementation of the proposed MGSVD-LDA linear algorithm and that of its kernelized version, the MGSVD-KDA algorithm.

The execution platform used is Pentium 4, 2.8 GHz CPU, 1.0 GB RAM and WinXP operating system. As in the Section 3.4, the same ten databases are used in the experiments. Four face databases and three text document databases are small sample size databases, where the samples of each database are linearly independent. The other three, spoken letter database, Isolet, molecule conformation database, MUSK, and digital handwritten database, MNIST are large sample size databases, where the SSS problem does not occur.

The two kernel functions used in our experiments are the same as in the previous section. The Gaussian radial basis function (RBF) kernel, $k(\mathbf{x}_l, \mathbf{x}_h) = \exp(-\frac{\|\mathbf{x}_l - \mathbf{x}_h\|^2}{\sigma})$, where $\| \cdot \|$ denotes the Euclidean 2-norm and $\sigma > 0$, and the nonhomogeneous polynomial kernel, $k(\mathbf{x}_l, \mathbf{x}_h) = (\langle \mathbf{x}_l, \mathbf{x}_h \rangle + 1)^d$, where d is a positive integer. The above two kernel functions result in providing linear independence among the samples of large sample size databases which are otherwise not linearly independent. The linear independence is successfully established if the kernel matrix is confirmed to be of full rank. We use the same mechanism to determine the parameters of the linear and nonlinear algorithms.

(a) Performance evaluation of the kernelized algorithms using small sample size datasets

In this set of experiments, we examine the effectiveness of six linear algorithms for the feature extraction of linearly independent samples in terms of their recognition accuracy and execution time. Linear algorithms were run in Chapter 3. We present those results here again, along with the simulation results obtained by running their kernelized counterparts, namely MGSVD-KDA, MGSVD-OKDA, KDA/GSVD, KRLDA, KPCA+LDA and KPCA. The complete simulation results are given in Table 4.1 and Table 4.2, from which we make the following observations:

1) The kernelization has no significant effect on the performance of the algorithms except for the case of KDA/GSVD algorithm whose linear counterpart suffered from the memory overflow problem.

2) The recognition accuracies of GSVD and MGSVD are not always the same. Even though these two methods are theoretically equivalent, their computations are different, and thus their numerical errors are also different. These numerical errors result in a slight angular difference between the discrimination subspaces computed from these two algorithms. This angular difference influences the distance between a test sample and the point that represents a class. If a test sample has almost the same distance to two points representing two different classes, then a small error in finding the two distances may alter the classification results

The above observations suggest that kernelization does not have significant positive effect on the performance of the algorithms except for the case of KDA/GSVD algorithm

whose linear counterpart suffers from the memory overflow problem. For small sample size databases, the performance of the most of the linear algorithm cannot be improved through their kernelization. The fact that recognition accuracy of both the MGSVD-LDA and MGSVD-KDA algorithms are the same is consistent with the finding of Theorem 4.1, according to which the MGSVD-LDA algorithm effectively separates the classes of datasets with linearly independent samples.

(b) Analysis of the numerical errors of the MGSVD-LDA and MGSVD-KDA algorithm

In this set of experiments, we examine the numerical errors of the proposed linear MGSVD-LDA algorithm and that of its kernelized version, the MGSVD-KDA algorithm. For the purpose of this examination, we use the scheme proposed based on Theorem 4.1 and described in Section 4.2 for examining the sensitivity of the proposed algorithms to numerical errors. The small sample size databases, the face databases (FERET, YALE, AR and ORL) and the text databases (Dataset1, Dataset2 and Dataset3) are used to investigate the effects of the numerical errors of the linear algorithm, since the samples in these databases are linearly independent. On the other hand, the large sample size databases, the Isolet and MUSK databases are used for the kernelized algorithm. For a given database, using Theorem 4.1, the theoretical values of the intra-class and inter-class distances are obtained. Note that the theoretical value of the inter-class distance is always zero. The computed values of these distances are obtained by using the samples projected in the discriminative subspace obtained from the linear or kernelized algorithm. The differences between a theoretical and the corresponding computed distances are obtained.

Table 4.1: Recognition rate (%) and execution time (seconds) of kernelized algorithms with small samples size face datasets

Database		FERET		YALE		AR		ORL	
Algorithm		Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
Lin.	**MGSVD-LDA**	**97.43**	**1.20**	**96.92**	**1.28**	**95.71**	**0.71**	**96.37**	**1.44**
	MGSVD-OLDA	**99.4**	**1.21**	**98.71**	**1.29**	**96.86**	**0.72**	**97.37**	**1.45**
	LDA/GSVD	Memory overflow		Memory overflow		Memory overflow		Memory overflow	
	RLDA	99.14	7.72	98.71	9.98	96.85	3.92	97.18	11.64
	PCA+LDA	95.37	1.19	95.98	1.26	95.15	0.69	95.12	1.42
	PCA	90.37	1.09	91.23	1.15	92.5	0.62	92.9	1.29
Ployn	**MGSVD-KDA**	**97.44**	**1.65**	**96.94**	**1.96**	**95.89**	**1.34**	**96.40**	**2.11**
	MGSVD-OKDA	**99.41**	**1.66**	**98.91**	**1.97**	**97.01**	**1.35**	**97.40**	**2.12**
	KDA-GSVD	97.43	1.66	96.95	1.96	95.88	1.34	96.41	2.11
	KRDA	98.7	8.82	98.72	9.98	97.01	5.98	96.98	11.23
	KPCA+LDA	95.39	1.67	96.01	1.88	97.01	1.32	95.10	2.01
	KPCA	90.67	1.54	91.25	1.69	92.56	1.24	92.30	1.78
RBF	**MGSVD-KDA**	**97.47**	**9.21**	**96.99**	**11.20**	**95.57**	**8.23**	**96.46**	**12.88**
	MGSVD-OKDA	**99.51**	**9.22**	**99.01**	**11.22**	**97.00**	**8.23**	**97.51**	**12.89**
	KDA/GSVD	97.48	9.21	96.96	11.21	95.75	8.23	96.45	12.88
	KRDA	99.12	15.56	98.75	16.89	97.21	12.67	95.13	18.12
	KPCA+LDA	95.79	9.14	96.21	11.19	95.63	8.10	95.01	12.10
	KPCA	91.37	8.64	91.35	10.11	92.59	7.75	92.35	11.12

Table 4.2: Recognition rate (%) and execution time (seconds) of kernelized algorithms with small samples size text document datasets

Database		Dataset1		Dataset2		Dataset3	
Algorithm		Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
Linear	**MGSVD-LDA**	**94.1**	**0.29**	**81.90**	**0.49**	**90.10**	**0.10**
	MGSVD-OLDA	**96.4**	**0.30**	**83.54**	**0.52**	**91.79**	**0.11**
	LDA/GSVD	94.0	29.54	81.91	9.32	90.10	0.10
	RLDA	96.5	2.35	83.52	4.32	91.78	1.20
	PCA+LDA	93.3	0.29	80.2	0.49	89.76	0.09
	PCA	89.9	0.25	60.2	0.45	79.3	0.08
Polyn	**MGSVD-KDA**	**94.2**	**0.76**	**82.0**	**0.96**	**90.13**	**0.31**
	MGSVD-OKDA	**96.5**	**0.76**	**83.57**	**0.98**	**91.81**	**0.32**
	KDA-GSVD	94.0	0.77	81.9	0.96	90.12	0.31
	KRDA	96.3	5.56	83.01	8.92	90.98	2.18
	KPCA+LDA	93.5	0.74	80.2	0.97	89.98	0.31
	PCA	90.2	0.70	61.1	0.91	79.9	0.27
RBF	**MGSVD-KDA**	**94.1**	**3.66**	**81.95**	**4.57**	**90.19**	**1.50**
	MGSVD-OLDA	**96.7**	**3.67**	**83.62**	**4.59**	**91.79**	**1.52**
	KDA-GSVD	94.1	3.66	81.89	4.57	90.20	1.50
	KRDA	96.5	8.32	83.56	13.32	90.93	4.52
	KPCA+LDA	93.5	3.67	80.3	4.56	89.99	1.51
	KPCA	90.1	3.42	61.2	4.31	79.67	1.37

For the linear algorithm and a database, this process is repeated ten times by running the algorithm repeatedly, with the samples in each run taken randomly from the database. The final value of a distance difference is chosen as the one having the maximum value over the ten runs of the algorithm. The results are summarized in Table 4.3 – Table 4.4.

The minimum computed inter-class distance that consists of the theoretical inter-class distance minus the maximum computed error of the inter-class distance is given in this table in a pair of parentheses for each database. Note that each maximum computation error is negligibly small compared to the theoretical inter-class distance. The last row of the table gives the maximum computational error of the intra-class distance for each database. These results imply that the accumulated computational errors, including those generated from the computation of the inner products, have negligible influence on the accuracy of the feature extraction. The comparison of the value of the minimum computed inter-class distance with that of the maximum inter-class difference shows that the latter is negligibly smaller compared to the former for each database used. Hence, we conclude that the accumulated computational errors, including those generated from computation of the inner products, have no influence on the accuracy of the feature extraction performance of the algorithms for these databases. In our experiments, the proposed linear algorithm, when operating on databases with linearly independent samples, was found to produce only small values for ρ, thus indicating that for these databases the performance of the algorithm is not sensitive to numerical errors.

For the kernelized algorithm, the polynomial kernel and RBF kernel with different values of the kernel parameters are used. This process is repeated ten times with the samples in each run taken randomly from the databases. With every parameter the linear

independence condition in the kernel feature space is confirmed. The recognition accuracy as well as the numerical error metric ρ, associated with every value of the kernel parameters is obtained. The average accuracy rate and the maximum ρ of the ten runs as a function of the parameter value are depicted in Figure 4.1 and Figure 4.2 for the polynomial kernel and RBF kernel, respectively. Figure 4.1 shows that with the polynomial kernel the numerical errors are negligibly small for the entire range of the parameter values. But as the parameter value increases, recognition accuracy shows a slightly downward trend. This result suggests that once the linear independence is established and the numerical error metric ρ is sufficiently small, one should choose the lowest order of polynomial. For the RBF kernel, Figure 4.2 shows that with some kernel parameter values the numerical error metric ρ are very large and the corresponding classification has a low recognition accuracy or the recognition process totally fails. On the other hand, the parameter values associated with the minimum ρ corresponds to high recognition accuracy. This result has an important implication in that when empirical test is not conductible due to the lack of proper test samples, one can choose the RBF kernel parameter that leads to the minimum ρ.

Table 4.3: The maximum differences between the theoretical values and computed values of the inter- and intra-class distances in the discriminant subspace derived from the MGSVD-LDA algorithm using face databases *

Database	FERET	YALE	AR	ORL
Minimum. computed inter-class distance	(0.6324 – 3.66E-15)	0.6324 – 5.00E-15)	(0.6324 – 4.22E-15)	(0.6323 – 4.56E-15)
Max.diff(Intra-class)	8.73E-14	4.75E-14	4.69E-14	5.76E-15

* The precision of the experimental computer is approximately 1E-16.

Table 4.4: The maximum differences between the theoretical values and computed values of the inter- and intra-class distances in the discriminant subspace derived from the MGSVD-LDA algorithm using text document databases *

Database	Dataset1	Dataset2	Dataset3
Minimum. computed inter-class distance	0.5345 – 1.22E-15)	(0.2236 – 4.44E-16)	(0.4899 – 3.87E-16)
Max.diff(Intra-class)	2.76E-15	3.76E-14	5.98E-15

* The precision of the experimental computer is approximately 1E-16.

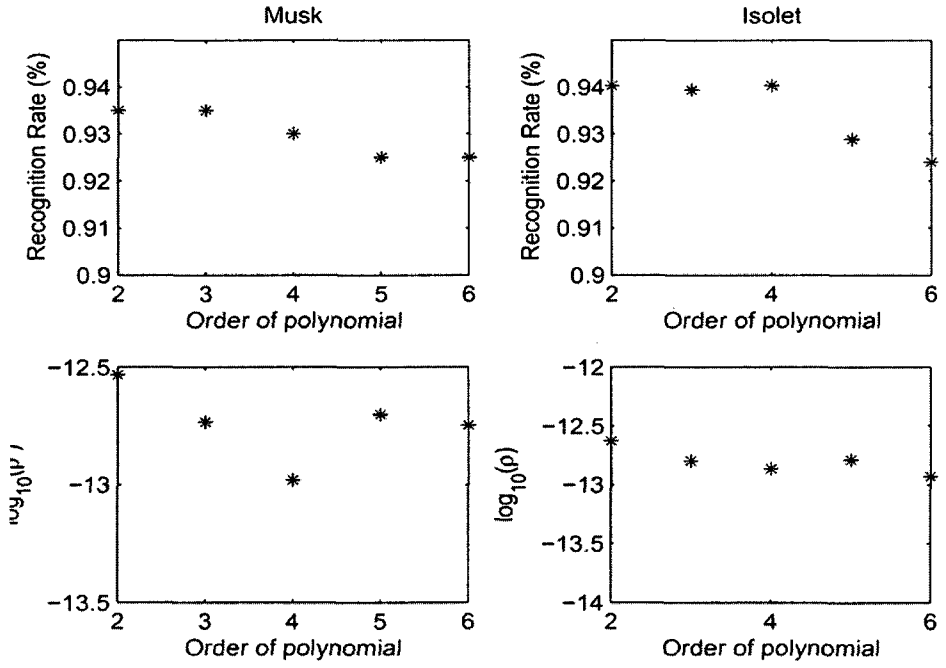

Figure 4.1: The recognition accuracy and numerical errors of MGSVD-KDA with respect to the order of the polynomial kernel function

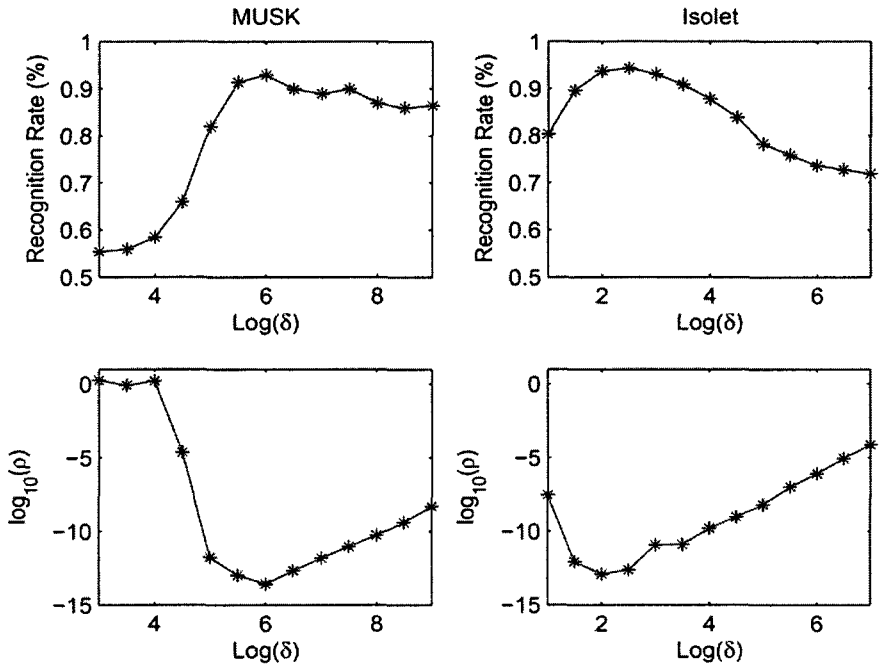

Figure 4.2: The recognition accuracy and numerical errors of MGSVD-KDA with respect to the parameter of the RBF function

4.5 Summary

In this chapter, a theorem has been established to determine the class structure of datasets with linearly independent samples in the discriminant subspace derived from the proposed MGSVD-LDA algorithm. According to this theorem, if the samples of datasets are linearly independent in the input space, all the samples of a class condense into a distinct single point of the discriminant subspace derived from the MGSVD-LDA algorithm, whereas a pair of samples belonging to different classes are separated in the discriminant subspace by a distance that is determined by the number of samples in each of the two classes. Thus, under the linear independence condition, the classes are linearly separable, that is, through the linear MGSVD-LDA algorithm the class structures of the

datasets can be successfully captured. It has been shown through simulation that for small sample size datasets, in which the samples are linearly independent, kernelization of the proposed linear algorithm, therefore, provides little improvement in the recognition accuracy.

The results of the above theorem have been used to develop a method to estimate the numerical errors in implementing the proposed linear and nonlinear algorithms. If the linear algorithm runs into a situation of datasets with linearly dependent samples, one necessarily needs to employ the kernelized algorithm. This estimate of numerical errors has also been used to devise a scheme to adjust the values of the kernel parameters to minimize the numerical errors in implementing the nonlinear algorithm and thus to improve the recognition accuracy.

Simulation results have shown that the proposed linear algorithm, when operating on datasets with linearly independent samples, produces only small values of numerical errors and the performance of the algorithm is not sensitive to numerical errors. For the polynomial kernel, the numerical errors in implementing the proposed kernelized algorithm are negligibly small for a wide range of values of the kernel parameter for the databases used in our experiments. For the RBF kernel, it has been shown that, by employing an estimate of the numerical errors, it is possible to adjust the kernel parameter so as to reduce the numerical errors and thus to increase the recognition accuracy.

Chapter 5

A Discriminant Model for the Feature Extraction of Linearly Independent Samples

5.1 Introduction

As mentioned in Chapter 2, a number of FLDA variants have been presented in the literature in order to overcome the singularity problem of the scatter matrices of the traditional FLDA algorithms. However, these variants, as also pointed out earlier, suffer from excessive computational load in dealing with the high dimensionality of patterns or lose some useful discriminant information in order to overcome the singularity problem in applying the Fisher criterion.

In Chapter 3, we proposed new GSVD-based linear and nonlinear algorithms for discriminant analysis. These algorithms provide an effective solution to the singularity problem of the Fisher criterion with low computational complexity and high recognition accuracy. In Chapter 4, we presented a theorem that essentially established the class structure for datasets with linearly independent samples in a specific discriminant subspace derived from the proposed MGSVD-LDA algorithm. Linearly independent samples are a very important category of patterns. For instance, face datasets, which are small sample size datasets with high dimensionality, are normally linearly independent. Samples in many other datasets, such as fingerprints, DNA data and iris data in biometric

applications, are also linearly independent. In view of the inspiration drawn from the results of Theorem 4.1, coupled with the practical significance of patterns with linearly independent samples, it is worth undertaking a deeper study of the feature extraction problem of datasets with linearly independent samples.

In this chapter, a discriminant model for a dataset with linearly independent samples in the input space is developed. It is shown that if the samples of a dataset that has N classes $C_i (i = 1, 2, \cdots N)$ are linearly independent, then there exist $N-1$ sets $S_l (l = 1, 2, \cdots, N-1)$ of mutually orthogonal hyperplanes in the input space, with each set S_l containing N parallel hyperplanes $P_i^l (i = 1, 2, \cdots, N)$ so that all the samples of the ith class $(i = 1, 2, \cdots, N)$ can be mapped onto the hyperplane P_i^l (i.e. onto the ith hyperplane of each of the $N-1$ sets of the mutually orthogonal hyperplanes). The common normal $\hat{\mathbf{g}}_l$ to all the parallel hyperplanes in S_l can then be selected as a discriminant vector and the collection of all such vectors $\{\hat{\mathbf{g}}_1, \cdots, \hat{\mathbf{g}}_j, \cdots, \hat{\mathbf{g}}_{N-1}\}$ as the discriminant subspace. Based on this model of datasets with linearly independent samples, some novel algorithms for discriminant analysis that do not run into the SSS problem are developed. Extensive simulations are carried out using benchmark datasets to examine the validity of the discriminant model presented and to demonstrate the effectiveness of the proposed algorithms, both in terms of the complexity and classification accuracy.

In Section 5.2, the discriminant model for the feature extraction of datasets with linearly independent samples is proposed. In Section 5.3, three new algorithms without encountering the SSS problem for obtaining the discriminant subspace of the proposed model are developed. In Section 5.4, a kernelized algorithm is also presented to deal with

the datasets in which the samples are not linearly independent. Section 5.5 presents the experimental results that are obtained by applying the proposed method on some benchmark datasets. The performance of the proposed algorithms is compared with that of several other well-known algorithms.

5.2 A Discriminant Model of Linearly Independent Samples

The objective of feature extraction in pattern recognition problems is to find appropriate features for representing the samples with enhanced discriminatory power for the purpose of classification. One of the commonly used feature extraction techniques is to transform the original sample space into a lower-dimensional discriminant subspace in which a sample of the dataset is more distinguishable in terms of the unique class to which it belongs. The idea is to find a transformation vector or a set of transformation vectors spanning over the discriminant subspace, on which the projections of the samples within each class condense into a compact region (ideally into a single point) separated from the regions corresponding to the other classes of the dataset. In this section, we show the existence of a discriminant model that allows the creation of a feature subspace in which the classes of linearly independent samples of a dataset can be efficiently discriminated.

Assume that a dataset consists of m-dimensional linearly independent samples each belonging to one of a total of N classes. If the ith class has n_i samples $(i = 1, 2, \cdots, N)$, the total number of samples in the dataset is $n = \sum_{i=1}^{N} n_i$. We define an $m \times n$ matrix

$$\mathbf{X} = [\mathbf{x}_{11}, \cdots, \mathbf{x}_{1n_1}; \mathbf{x}_{21}, \cdots, \mathbf{x}_{2n_2}; \cdots; \mathbf{x}_{i1}, \cdots, \mathbf{x}_{ij}, \cdots, \mathbf{x}_{in_i}; \cdots; \mathbf{x}_{N1} \cdots, \mathbf{x}_{Nn_N}] \tag{5.1}$$

where $\mathbf{x}_{ij}$ is the jth $(j = 1, 2, \cdots, n_i)$ sample in the ith class. Let $\mathbf{g}$ be an m-dimensional transformation vector, then the projection of the samples on $\mathbf{g}$ is given as

$$\mathbf{p} = \mathbf{g}^T\mathbf{X} = (p_{11}, \cdots, p_{1n_1}, p_{21}, \cdots, p_{2n_2}, \cdots, p_{i1}, \cdots, p_{ij}, \cdots, p_{in_i}, \cdots, p_{N1} \cdots, p_{Nn_N}) \tag{5.2}$$

where the scalar p_{ij} is the projection of the data sample $\mathbf{x}_{ij}$ on $\mathbf{g}$. This linear transformation can also be expressed as

$$\mathbf{X}^T\mathbf{g} = \mathbf{p}^T \tag{5.3}$$

where, using the terminology of linear systems, $\mathbf{X}^T$ is the coefficient matrix, $\mathbf{g}$ is the unknown vector, and $\mathbf{p}$ is a known constant vector of the nonhomogeneous linear system. Since the rows of $\mathbf{X}^T$ consists of n linearly independent samples, the rank of this matrix is n. With an arbitrary vector $\mathbf{p}^T$, the rank of the augmented matrix $\tilde{\mathbf{X}} = [\mathbf{X}^T \vdots \mathbf{p}^T]$ is also n. The linear independence also implies that the sample dimension m must be larger than or equal to the sample size n (i.e. $n \leq m$). As $\mathbf{X}$ and $\tilde{\mathbf{X}}$ have the same rank, namely n, the existence of solutions of the linear system given by (5.3) is guaranteed. If $n = m$, (5.3) has a unique solution for $\mathbf{g}$, otherwise it has infinitely many solutions. Thus, the existence of a solution for an arbitrary $\mathbf{p}^T$ implies that we can choose the elements of $\mathbf{p}$ such that

$$p_{i1} = p_{i2} = \cdots = p_{in_i} = q_i, \; i = 1, 2, \cdots, N \tag{5.4}$$

and

$$q_i \neq q_k, i, k = 1, \cdots, N; \; k \neq i \tag{5.5}$$

that is, q_i is a projection of the samples of the ith class on the transformation vector $\mathbf{g}$, different from those of the samples of the other classes. Thus, there always exists a transformation vector $\mathbf{g}$ such that all the samples belonging to a class are mapped onto a

unique single point of each transformation vector **g**.

A single-valued projection of all the samples within a class suggests that there exists an $(m-1)$-dimensional hyperplane in the original input space that is perpendicular to **g** and all the samples of this class lie on this hyperplane. There are N different projection points corresponding to the samples of the N different classes on a transformation vector **g**. This implies that there exist N parallel hyperplanes, each corresponding to one class and all the samples of one class belong to only one of these parallel hyperplanes. Thus, **g** is a vector normal to all the N hyperplanes.

As all the samples belonging to a class have the same projection point on each vector **g**, without loss of generality, any sample of the class can be used to represent all the samples of that class. From (5.2), using the constraints imposed by (5.4) and (5.5), we have the following equations:

$$\mathbf{g}^T(\mathbf{x}_{ir_1} - \mathbf{x}_{kr_2}) \neq 0, \quad i,k = 1,2,\cdots,N, \quad i \neq k, r_1 \in \{1,\cdots,n_i\}, r_2 \in \{1,\cdots,n_k\} \tag{5.6}$$

and

$$\mathbf{g}^T(\mathbf{x}_{ir_1} - \mathbf{x}_{ir_2}) = 0, \quad i = 1,2,\cdots N, r_1, r_2 \in \{1,\cdots,n_i\}, r_1 \neq r_2 \tag{5.7}$$

To facilitate the calculation of **g**, let us now construct the following two matrices:

$$\hat{\mathbf{A}}_b = [(\mathbf{x}_{21} - \mathbf{x}_{11}), \ \cdots, (\mathbf{x}_{N1} - \mathbf{x}_{11}), (\mathbf{x}_{31} - \mathbf{x}_{21}), \ \cdots, (\mathbf{x}_{N1} - \mathbf{x}_{21}), \ \cdots, (\mathbf{x}_{N1} - \mathbf{x}_{(N-1)1})] \tag{5.8}$$

$$\begin{aligned} \mathbf{A}_w = [&(\mathbf{x}_{12} - \mathbf{x}_{11}), \ \cdots, (\mathbf{x}_{1n_1} - \mathbf{x}_{11}), (\mathbf{x}_{22} - \mathbf{x}_{21}), \ \cdots, \\ &(\mathbf{x}_{2n_2} - \mathbf{x}_{21}), \ \cdots, (\mathbf{x}_{N2} - \mathbf{x}_{N1}), \ \cdots, (\mathbf{x}_{Nn_N} - \mathbf{x}_{N1})] \end{aligned} \tag{5.9}$$

Note that the columns of matrix $\mathbf{A}_w$ are formed by subtracting the first sample of each class from all the remaining samples of the same class. On the other hand, the columns of matrix $\hat{\mathbf{A}}_b$ are formed by subtracting the first samples of two different classes. It is

obvious that the columns of $\hat{\mathbf{A}}_b$ are linearly dependent, since they are formed through linear operations of the first samples of all the classes. We can obtain a matrix $\mathbf{A}_b$ by selecting $N-1$ linearly independent columns of $\hat{\mathbf{A}}_b$ as follows:

$$\mathbf{A}_b = [(\mathbf{x}_{21} - \mathbf{x}_{11}), (\mathbf{x}_{31} - \mathbf{x}_{11}), \ldots, (\mathbf{x}_{i1} - \mathbf{x}_{11}), \ldots, (\mathbf{x}_{N1} - \mathbf{x}_{11})] \tag{5.10}$$

Using (5.10) into (5.6) and (5.9) into (5.7) yields, respectively, the following equations:

$$\mathbf{g}^T \mathbf{A}_b = \boldsymbol{\alpha}^T \tag{5.11}$$

$$\mathbf{g}^T \mathbf{A}_w = \mathbf{0} \tag{5.12}$$

where $\boldsymbol{\alpha}$ is a column vector with all its elements being non-zero and unequal. Let $\mathbb{R}(\cdot)$ and $\mathbb{N}(\cdot)$ denote the range and the null space of the associated matrix, respectively. Equation (5.12) indicates that $\mathbf{g} \in \mathbb{N}(\mathbf{A}_w)$, and (5.11) implies that the columns of $\mathbf{A}_b$ have non-zero projections on $\mathbf{g}$. As stated earlier, there exists a set of vectors $\mathbf{g}$'s that satisfy (5.4) and (5.5), therefore it also satisfies (5.6) and (5.7) or (5.11) and (5.12) as long as the samples of a dataset are linearly independent. We now establish the following lemma and theorems to prove that the set of vectors $\mathbf{g}$'s consist of a subspace of $\mathbb{N}(\mathbf{A}_w)$ such that the columns of $\mathbf{A}_b$ have non-zero projections on this subspace.

Lemma 5.1: Assume that a dataset $\mathbf{X}$ consists of m-dimensional samples $\mathbf{x}_{ij}$'s $(i = 1, 2, \cdots, N; j = 1, 2, \cdots, n_i)$ such that the ith class C_i from the N classes has n_i samples. Given the matrices $\mathbf{A}_w$ and $\mathbf{A}_b$ as formed by the (5.9) and (5.10), the columns of the matrix $[\mathbf{A}_w \ \mathbf{A}_b]$ are linearly independent.

The proof of the lemma is straightforward and thus omitted.

Theorem 5.1: Given an N-class dataset whose samples $\mathbf{x}_{ij}$'s $(i=1,2,\cdots,N; j=1,2,\cdots,n_i)$ are linearly independent, and the matrices $\mathbf{A}_w$ and $\mathbf{A}_b$ as formed according to (5.9) and (5.10), respectively, the projection vectors of the columns of $\mathbf{A}_b$ on $\mathbb{N}(\mathbf{A}_w)$ are linearly independent.

Proof: Let $\boldsymbol{\varphi}_l$ $(l=1,\cdots,N-1)$ denote the N-1 columns of $\mathbf{A}_b$, and $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$, denote the projections of $\boldsymbol{\varphi}_l$ on $\mathbb{N}(\mathbf{A}_w)$. We prove by contradiction that the projection vectors of the columns of the matrix $\mathbf{A}_b$ on $\mathbb{N}(\mathbf{A}_w)$ are linearly independent. Assume that the vector $\boldsymbol{\delta}_l$'s are linearly dependent. Then there exists a set of constants $\{a_d\}_{d=1,\cdots,N-1}$, not all zero, such that $\sum_{l=1}^{N-1} a_l\boldsymbol{\delta}_l = \mathbf{0}$ is a zero (null) vector. It follows that the linear combination of the columns of $\mathbf{A}_b$, $\sum_{l=1}^{N-1} a_l\boldsymbol{\varphi}_l$, which is a non-zero vector, has zero projection on $\mathbb{N}(\mathbf{A}_w)$. Thus, $\sum_{l=1}^{N-1} a_l\boldsymbol{\varphi}_l$ must belong to $\mathbb{R}(\mathbf{A}_w)$. A linear combination of the columns of $\mathbf{A}_b$ can be expressed as a linear combination of that of $\mathbf{A}_w$, which contradicts Lemma 5.1. As a consequence, the projection vectors of $\mathbf{A}_b$ on $\mathbb{N}(\mathbf{A}_w)$, that is, $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$, are linearly independent.

■

As $\mathbf{A}_b$ has $N-1$ non-zero linearly independent projections on $\mathbb{N}(\mathbf{A}_w)$, the dimension of the subspace spanned by these non-zero projection vectors $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$ is $N-1$.

We now establish a relation between the subspace spanned by vector $\boldsymbol{\delta}_l$'s and the subspace spanned by $\mathbf{g}_l$'s.

Theorem 5.2: Given an N-class dataset whose samples $\mathbf{x}_{ij}$'s $(i=1,2,\cdots,N; j=1,2,\cdots,n_i)$ are linearly independent, all the samples of a class are projected into a distinct single point in the subspace spanned by $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$.

Proof: First, since according to (5.12), the projection of the difference between any two samples is a zero scalar, all the samples of a class project to the same point of $\mathbb{N}(\mathbf{A}_w)$, and hence, to the same point of the subspace spanned by $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$.

Now, we will prove by contradiction that a pair of projection points on the subspace spanned by $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$ corresponding to any two samples belonging to two different classes cannot be the same point. Assume that any two samples belonging to two different classes are projected to a single point in the subspace spanned by $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$. It follows that the projections of the columns of $\mathbf{A}_b$ have zero projections on the subspace, that is, for some l, there exists a zero-vector $\boldsymbol{\delta}_l$. This implies that the vectors $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$ are not linearly independent, that is, Theorem 5.1 is contradicted. Thus, the projections of the samples of a class are a distinct single point in the subspace spanned by $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$.

■

Since all the samples of a class are projected into a single distinct point in the subspace spanned by vectors $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$, this subspace can be used as the discriminant subspace (DS) and since the vectors $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$ are linearly

independent, the set $\{\boldsymbol{\delta}_1,\ldots\boldsymbol{\delta}_{N-1}\}$ constitutes a basis of this DS. Theorem 5.2 implies that the DS is actually the same set of vectors as that spanned by the vectors $\mathbf{g}$ satisfying (5.11) and (5.12). Hence, any vector in this DS can be treated as $\mathbf{g}$ that satisfies (5.11) and (5.12). Once the set of projection vectors $\boldsymbol{\delta}_l$ $(l=1,\cdots,N-1)$ is obtained, this DS is determined. Therefore, we can select $\mathbf{g}_l=\boldsymbol{\delta}_l$ $l=1,2,\ldots,N-1$, as a set of transformation vectors; any other vectors in this subspace can be represented as a linear combination of $\mathbf{g}_l$ $(l=1,\cdots,N-1)$. Without loss of generality, we can assume the basis of the discriminant subspace to consist of an orthonormalized set $\{\hat{\mathbf{g}}_1,\ldots\hat{\mathbf{g}}_{N-1}\}$ such that $\|\hat{\mathbf{g}}_k\|=1$ and $\hat{\mathbf{g}}_k^T\hat{\mathbf{g}}_l=0$ for k, l = 1, …, N−1 and $k\neq l$. Before closing this section, we can summarize the foregoing discussion and analyses as follows.

If a dataset $\mathbf{X}$ consisting of m-dimensional linearly independent samples $\mathbf{x}_{ij}$'s $(i=1,2,\cdots,N;j=1,2,\cdots,n_i)$ such that the ith class C_i from a total of N classes has n_i samples, then there exists a set of mutually orthonormal transformation vectors $\hat{\mathbf{g}}_l$ $(l=1,2,\cdots N-1)$. The set of $\{\hat{\mathbf{g}}_1,\ldots\hat{\mathbf{g}}_{N-1}\}$ forms the discriminant subspace in the sense that for each $\hat{\mathbf{g}}_l$, there exists a set of N parallel hyperplanes P_i^l $(i=1,2,\ldots,N)$ that are normal to $\hat{\mathbf{g}}_l$ such that all the samples $\mathbf{x}_{ij}$'s of C_i lie on the hyperplane P_i^l and have a distinct point on $\hat{\mathbf{g}}_l$ as their projection. The discriminant subspace is a subspace of the null space of $\mathbf{A}_w$, $\mathbb{N}(\mathbf{A}_w)$, on which $\mathbf{A}_b$ has non-zero projections.

5.3 Algorithms for Finding the Orthonormal Basis of the DS

The discriminant model developed in the previous section for a dataset with linearly independent samples shows that there exists a DS in which all the samples of a class merge into a unique point. It has also been shown that this DS is a subspace of $\mathbb{N}(\mathbf{A}_w)$ on which the projection of $\mathbf{A}_b$ is non-zero. In this section, we develop three algorithms referred to as Algorithm A, Algorithm B, and Algorithm C for finding the DS that comprises the set of vectors $\hat{\mathbf{g}}_l$ $(l = 1, 2, \cdots N-1)$ of the proposed model for a given dataset.

Among these three algorithms, the first one is a straight forward technique of finding the DS in that first, $\mathbb{N}(\mathbf{A}_w)$ is found by solving the linear equations $\mathbf{Z}^T\mathbf{A}_w = \mathbf{0}$. Then, a subset of the solutions is determined on which $\mathbf{A}_b$ has non-zero projections. In Algorithm B, by utilizing the range of $\mathbf{A}_w$, a direct solution of the linear equations $\mathbf{Z}^T\mathbf{A}_w = \mathbf{0}$ is avoided. Algorithm C is based on the same philosophy as that of deriving Algorithm B. However, here the range of $(\mathbf{A}_w \vdots \mathbf{A}_b)$ is utilized instead of that of $\mathbf{A}_w$. In Section 3.3 we will show that steps of this algorithm leading to the determination of the DS are very amenable to a kernelization of this algorithm.

Algorithm A

As the DS is a subspace of $\mathbb{N}(\mathbf{A}_w)$ on which $\mathbf{A}_b$ has non-zero projections, we propose an algorithm that first finds $\mathbb{N}(\mathbf{A}_w)$ and then projects $\mathbf{A}_b$ onto $\mathbb{N}(\mathbf{A}_w)$ to evaluate the basis of the DS. Consider the homogeneous linear system

$$\mathbf{Z}^T\mathbf{A}_w = \mathbf{0} \tag{5.13}$$

where $\mathbf{Z}$ represents a solution of the system. The set of solutions $\{\mathbf{Z}\}$ forms the null space of $\mathbf{A}_w$, i.e. $\mathbb{N}(\mathbf{A}_w)$. Then, the DS is obtained by projecting $\mathbf{A}_b$ onto $\mathbb{N}(\mathbf{A}_w)$ as

$$\mathbf{G} = \mathbf{Z}\mathbf{Z}^T\mathbf{A}_b \tag{5.14}$$

In order to find $\hat{\mathbf{G}}$, the transformation matrix with orthonormal basis, an eigen-decomposition of $\mathbf{G}^T\mathbf{G}$ is carried out to orthogonalize the columns of $\mathbf{G}$ as

$$\mathbf{G}^T\mathbf{G} = \hat{\mathbf{G}}^T\mathbf{R}^{-2}\hat{\mathbf{G}} \tag{5.15}$$

where $\mathbf{R}$ is a diagonal eigen-value matrix corresponding to $\mathbf{G}$.

Overall, the algorithm just presented can be summarized as having the steps given in Table 5.1.

Table 5.1: Algorithm A

Input: Training sample $\mathbf{x}_l$
Output: Transformation matrix $\mathbf{G}$
Step A1: Form the $\mathbf{A}_b$ and $\mathbf{A}_w$ using (5.9) and (5.10).
Step A2: Find a set of orthogonal solutions $\{\mathbf{Z}\}$ of the linear system given by (5.13).
Step A3: Evaluate $\mathbf{G}$ using (5.14)
Step A4: Find transformation matrix $\hat{\mathbf{G}}$ by applying eigen-decomposition of $\mathbf{G}$ using (5.15).

Although this algorithm is straight forward, its use is limited to determining the DS for low-dimensional datasets because of the high computational complexity involve with the

solution of the associated linear systems. The development of the following algorithms addresses this problem.

Algorithm B

Consider an orthonormal matrix $\mathbf{Q} \in \Re^{m \times m}$, partitioned as $[\mathbf{Q}_1 \vdots \mathbf{Q}_2]$, where the columns of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ consist of the bases of $\mathbb{R}(\mathbf{A}_w)$ and $\mathbb{N}(\mathbf{A}_w)$, respectively. Since $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, an identity matrix, we can write $\mathbf{A}_b$ as

$$\mathbf{A}_b = \mathbf{Q}\mathbf{Q}^T\mathbf{A}_b = [\mathbf{Q}_1 \ \mathbf{Q}_2]\begin{bmatrix}\mathbf{Q}_1^T \\ \\ \mathbf{Q}_2^T\end{bmatrix}\mathbf{A}_b = \mathbf{E} + \mathbf{E}' \tag{5.16}$$

where

$$\mathbf{E} = \mathbf{Q}_1\mathbf{Q}_1^T\mathbf{A}_b \tag{5.17}$$

and

$$\mathbf{E}' = \mathbf{Q}_2\mathbf{Q}_2^T\mathbf{A}_b \tag{5.18}$$

It is seen from the above equation that $\mathbf{E}$ and $\mathbf{E}'$ are the projections of $\mathbf{A}_b$ on $\mathbb{R}(\mathbf{A}_w)$ and $\mathbb{N}(\mathbf{A}_w)$, respectively. This equation indicates that $\mathbf{A}_b$ can be decomposed into two mutually orthogonal components, $\mathbf{E}$ and $\mathbf{E}'$. As $\mathbf{A}_w \in \Re^{m \times (n-N)}$ and $\mathbb{N}(\mathbf{A}_w) \in \Re^{m \times (m-n+N)}$, the size of $\mathbb{R}(\mathbf{A}_w)$ is smaller than that of $\mathbb{N}(\mathbf{A}_w)$ when the sample dimension is large. Consequently, the evaluation of $\mathbb{R}(\mathbf{A}_w)$ is computationally less expensive than that of $\mathbb{N}(\mathbf{A}_w)$. The orthonormal basis of $\mathbb{R}(\mathbf{A}_w)$, i.e. $\mathbf{Q}_1$, can be obtained by carrying out

eigen-decomposition of $\mathbf{A}_w^T\mathbf{A}_w$. Then, matrix $\mathbf{E}'$ can be obtained using (5.18) and $\mathbf{E}$ in turn can be used in (5.16) yielding

$$\mathbf{E} = \mathbf{A}_b - \mathbf{E}' \tag{5.19}$$

Finally, $\hat{\mathbf{G}}$, the orthonormal basis of the DS can be obtained through an eigen-decomposition of $\mathbf{E}^T\mathbf{E}$. This algorithm circumvents the direct calculation of $\mathbb{N}(\mathbf{A}_w)$ and thus it is time-efficient.

The steps of the algorithm thus developed are given in Table 5.2.

Table 5.2: Algorithm B

Input: Training sample $\mathbf{x}_l$
Output: Transformation matrix $\mathbf{G}$
Step B1: Form $\mathbf{A}_w$ and $\mathbf{A}_b$ using (5.9) and (5.10).
Step B2: Find $\mathbf{Q}_1$ by obtaining the eigen-decomposition of $\mathbf{A}_w^T\mathbf{A}_w$.
Step B3: Find $\mathbf{E}'$ using (5.18).
Step B4: Find $\mathbf{E}$ using (5.19).
Step B5: Find $\hat{\mathbf{G}}$ by finding the eigen-decomposition of $\mathbf{E}^T\mathbf{E}$.

Algorithm C

In this algorithm, we first construct an augmented matrix $\mathbf{A}_c = [\mathbf{A}_w\ \mathbf{A}_b]$ and then project the matrix $\mathbf{A}_w$ onto $\mathbb{R}(\mathbf{A}_c)$. The null space of $\mathbf{A}_w$ found within $\mathbb{R}(\mathbf{A}_c)$ is then the DS.

To find $\mathbb{R}(\mathbf{A}_c)$, one can conduct a singular value decomposition of $\mathbf{A}_c$. However, for high-dimensional datasets, it would result in a high computational load. An efficient alternate approach is to conduct an eigen-decomposition of the inner product matrix $\mathbf{A}_c^T\mathbf{A}_c$ as

$$\mathbf{A}_c^T\mathbf{A}_c = \mathbf{U}\Lambda\mathbf{U}^T \tag{5.20}$$

where $\Lambda \in \Re^{t\times t}$, with $t = \text{rank}(\mathbf{A}_c)$, is the diagonal matrix whose diagonal elements are the nonzero eigenvalues of $\mathbf{A}_c^T\mathbf{A}_c$, and $\mathbf{U} \in \Re^{n\times t}$ is the eigenvector matrix corresponding to $\mathbf{A}_c^T\mathbf{A}_c$. Then, the columns of $\hat{\mathbf{U}}$ given by

$$\hat{\mathbf{U}} = \mathbf{A}_c\mathbf{U}\Lambda^{-1/2} \tag{5.21}$$

constitute an orthonormal basis of $\mathbb{R}(\mathbf{A}_c)$. Projecting the columns of $\mathbf{A}_w$ onto $\mathbb{R}(\mathbf{A}_c)$ gives $\mathbf{B}_w = \hat{\mathbf{U}}^T\mathbf{A}_w$. The range and the null space of $\mathbf{B}_w$ can be obtained by conducting the eigen-decomposition of $\mathbf{B}_w$ as

$$\mathbf{B}_w = \mathbf{V}\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \Sigma\end{bmatrix}\mathbf{V}^T \tag{5.22}$$

where $\Sigma \in \Re^{d\times d}$, with $d = \text{rank}(\mathbf{A}_w)$, is a diagonal matrix whose diagonal elements are the nonzero eigenvalues of $\mathbf{B}_w$, and $\mathbf{V} \in \Re^{t\times t}$ is the eigenvector matrix corresponding to $\mathbf{B}_w$.

We partition $\mathbf{V}$ as $[\mathbf{V}_1\ \mathbf{V}_2]$, where $\mathbf{V}_1 \in \Re^{t\times(t-d)}$ consists of the eigenvectors corresponding to $\mathbb{R}(\mathbf{B}_w)$ and $\mathbf{V}_2 \in \Re^{t\times d}$ corresponds to $\mathbb{N}(\mathbf{B}_w)$. Finally, the orthonormal basis of the DS is obtained as

$$\mathbf{G} = \hat{\mathbf{U}}\mathbf{V}_2 = \mathbf{A}_c\mathbf{U}\Lambda^{-1/2}\mathbf{V}_2 \tag{5.23}$$

The steps of the algorithm just described are put together in Table 5.3.

Table 5.3: Algorithm C

Input: Training sample $\mathbf{x}_l$
Output: Transformation matrix $\mathbf{G}$
Step C1: Form $\mathbf{A}_w$ and $\mathbf{A}_b$ using (5.9) and (5.10), and set $\mathbf{A}_c = (\mathbf{A}_w \vdots \mathbf{A}_b)$.
Step C2: Find $\hat{\mathbf{U}}$ using (5.20) and (5.21).
Step C3: Evaluate the projection $\mathbf{B}_w = \hat{\mathbf{U}}^T\mathbf{A}_w$.
Step C4: Find $\mathbf{V}$ by eigen-decomposition of $\mathbf{B}_w$ given in (5.22).
Step C5: Find $\mathbf{V}_2$ by partitioning $\mathbf{V}$.
Step C6: Find the orthonormal basis of the DS using (5.23).

5.4 Kernel-Based Discriminant Subspace

In the preceding sections, a discriminant model for a dataset with linearly independent samples has been developed. It has been shown that there exists a discriminant subspace in which all the samples of a class merge into a distinct single point. Three algorithms have also been proposed based on this discriminant model. However, the linearly independent condition used for the development of the model may not be satisfied in many cases. For instance, when the sample size is large relative to the sample dimension, the samples of a dataset cannot be linearly dependent. In such a case, one can restore the

linear independence condition in a higher-dimensional space through a nonlinear kernel mapping of the input samples [48], [84].

A kernel is a nonlinear mapping $\mathbf{\Phi}$ that is designed to map the samples in the input space $\Re^m$ onto a higher-dimensional feature space $\mathbf{\Gamma}$

$$\begin{aligned} \mathbf{\Phi}: \ & \Re^m \rightarrow \mathbf{\Gamma} \\ & \mathbf{x}_{ij} \rightarrow \boldsymbol{\psi}_{ij} \end{aligned} \tag{5.24}$$

Correspondingly, sample $\mathbf{x}_{ij}$'s in the original input space $\Re^m$ is mapped into a potentially much higher-dimensional feature vector $\boldsymbol{\psi}_{ij}$'s in the feature space $\mathbf{\Gamma}$, in which samples become linearly independent, and hence, a linear technique for discriminant analysis can be applied. However, the high dimensionality of the derived feature space can make the overall process of the discriminant analysis computationally infeasible in practice. This problem is generally overcome by using the so called "kernel trick", in which the inner products of the mapped sample vectors in the feature space can be implicitly derived from the inner products between the input samples [54], [56] such that

$$\langle \boldsymbol{\psi}_l, \boldsymbol{\psi}_h \rangle = k\left(\langle \mathbf{x}_l, \mathbf{x}_h \rangle\right) = k_{lh} \tag{5.25}$$

where $\langle \bullet \rangle$ denotes the inner product of two vectors in the feature space, $k(\bullet)$ denotes a kernel function, and k_{lh} is a scalar. The key to a successful kernelization of a linear algorithm is in its ability to construct inner products in the input space and then to incorporate these in the feature space again in the form of inner products. The formulations of Algorithm A and Algorithm B developed in the previous section in their present forms lack suitable inner product representations for their kernelization. However,

the third algorithm, Algorithm C, has highly suitable format for its kernelization. Hence, we now kernelized this algorithm.

Define the following matrices similar to $\mathbf{A}_w$, $\mathbf{A}_b$ and $\mathbf{A}_c$ using samples in the higher-dimensional feature space defined by the mapping given by (5.24):

$$\mathbf{\Phi}_w = [(\psi_{12} - \psi_{11}),\ldots,(\psi_{1n_1} - \mathbf{x}_{11}),(\psi_{22} - \psi_{21}),\ldots,(\psi_{2n_2} - \psi_{21}),\ldots,$$

$$(\psi_{N2} - \psi_{N1}),\ldots,(\psi_{Nn_N} - \psi_{N1})]$$

$$\mathbf{\Phi}_b = [(\psi_{21} - \psi_{11}),(\psi_{31} - \psi_{11}),\ldots,(\psi_{i1} - \psi_{11}),\ldots,(\psi_{N1} - \psi_{11})]$$

$$\mathbf{\Phi}_c = \left[\mathbf{\Phi}_w \ \mathbf{\Phi}_b\right]$$

As for the linear algorithm (Algorithm C), form a symmetric matrix

$$\mathbf{\Phi}_c^T\mathbf{\Phi}_c = \begin{bmatrix} \mathbf{\Phi}_b^T\mathbf{\Phi}_b & \mathbf{\Phi}_b^T\mathbf{\Phi}_w \\ \mathbf{\Phi}_w^T\mathbf{\Phi}_b & \mathbf{\Phi}_w^T\mathbf{\Phi}_w \end{bmatrix} \tag{5.26}$$

In order to evaluate the above matrix, we first construct the following matrices:

$$\mathbf{\Phi}_1 = [\psi_{11}, \psi_{21},\ldots,\psi_{N1}] \tag{5.27}$$

$$\mathbf{\Phi}_2 = [\psi_{12},\ldots\psi_{1n_1},\psi_{22},\ldots,\psi_{2n_2},\ldots,\psi_{N2},\ldots,\psi_{Nn_N}] \tag{5.28}$$

$$\mathbf{C}_1 = \begin{bmatrix} (-\mathbf{1})_{1\times(N-1)} \\ \mathbf{I}_{N-1} \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} (\mathbf{1})_{1\times(n_1-1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (\mathbf{1})_{1\times(n_N-1)} \end{bmatrix} \tag{5.29}$$

Then, we can express the matrix $\mathbf{\Phi}_b$ and $\mathbf{\Phi}_w$ used in (5.26) as

$$\mathbf{\Phi}_b = \mathbf{\Phi}_1\mathbf{C}_1 \tag{5.30}$$

$$\mathbf{\Phi}_w = \mathbf{\Phi}_2 - \mathbf{\Phi}_1\mathbf{C}_2 \tag{5.31}$$

The four sub-matrices on the right side of (5.26) can be expressed as

$$\boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_b = \mathbf{C}_1^T\mathbf{K}_{11}\mathbf{C}_1$$

$$\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w = \mathbf{K}_{22} - \mathbf{C}_2^T\mathbf{K}_{12} - (\mathbf{C}_2^T\mathbf{K}_{12})^T + \mathbf{C}_2^T\mathbf{K}_{11}\mathbf{C}_2 \tag{5.32}$$

$$\boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_w = \mathbf{C}_1^T\mathbf{K}_{12} - \mathbf{C}_1^T\mathbf{K}_{11}\mathbf{C}_2$$

where $\mathbf{K}_{11} = \boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_1$, $\mathbf{K}_{22} = \boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_2$, and $\mathbf{K}_{12} = (\mathbf{K}_{21})^T = \boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_2$. Derivation of this set of formulas is presented in Appendix C. Carrying out an eigen-decomposition of $\boldsymbol{\Phi}_c^T\boldsymbol{\Phi}_c$, we have

$$\boldsymbol{\Phi}_c^T\boldsymbol{\Phi}_c = \tilde{\mathbf{U}}\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{U}}^T \tag{5.33}$$

where $\tilde{\boldsymbol{\Lambda}} \in \Re^{t\times t}$, with $t = \text{rank}(\boldsymbol{\Phi}_c)$, is a diagonal matrix whose diagonal elements are the nonzero eigenvalues of $\boldsymbol{\Phi}_c^T\boldsymbol{\Phi}_c$, and $\tilde{\mathbf{U}}$ is the eigenvector matrix corresponding to $\boldsymbol{\Phi}_c^T\boldsymbol{\Phi}_c$. Thus, $\tilde{\mathbf{U}} = \boldsymbol{\Phi}_c\tilde{\mathbf{U}}\tilde{\boldsymbol{\Lambda}}^{-1/2}$ is the eigenvector matrix of $\boldsymbol{\Phi}_c\boldsymbol{\Phi}_c^T$ with its orthonormal columns that span $\mathbb{R}(\boldsymbol{\Phi}_c)$. Projecting the columns of $\boldsymbol{\Phi}_w$ onto $\tilde{\mathbf{U}}$ yields

$$\tilde{\boldsymbol{\Phi}}_w = \tilde{\mathbf{U}}^T\boldsymbol{\Phi}_w = \tilde{\boldsymbol{\Lambda}}^{-1/2}\tilde{\mathbf{U}}^T\boldsymbol{\Phi}_c^T\boldsymbol{\Phi}_w = \tilde{\boldsymbol{\Lambda}}^{-1/2}\tilde{\mathbf{U}}^T\begin{bmatrix}\mathbf{K}_{22} - \mathbf{C}_2^T\mathbf{K}_{12} - (\mathbf{C}_2^T\mathbf{K}_{12})^T + \mathbf{C}_2^T\mathbf{K}_{11}\mathbf{C}_2 \\ \mathbf{C}_1^T\mathbf{K}_{12} - \mathbf{C}_1^T\mathbf{K}_{11}\mathbf{C}_2\end{bmatrix} \tag{5.34}$$

Now, another eigen-decomposition is carried out to find $\mathbb{N}(\tilde{\boldsymbol{\Phi}}_w)$ in $\mathbb{R}(\boldsymbol{\Phi}_c)$ as follows

$$\tilde{\boldsymbol{\Phi}}_w^T\tilde{\boldsymbol{\Phi}}_w = \tilde{\mathbf{V}}\begin{bmatrix}0 & 0 \\ 0 & \tilde{\boldsymbol{\Sigma}}^2\end{bmatrix}\tilde{\mathbf{V}}^T \tag{5.35}$$

where $\tilde{\boldsymbol{\Sigma}}$ is a diagonal matrix with nonzero eigenvalues of $\tilde{\boldsymbol{\Phi}}_w^T\tilde{\boldsymbol{\Phi}}_w$ on its diagonal and $\tilde{\mathbf{V}}$ is the eigenvector matrix corresponding to $\tilde{\boldsymbol{\Phi}}_w^T\tilde{\boldsymbol{\Phi}}_w$. Partitioning $\tilde{\mathbf{V}}$ as $\begin{bmatrix}\tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2\end{bmatrix}$, where $\tilde{\mathbf{V}}_1$ is the eigenvector matrix corresponding to the null space of $\tilde{\boldsymbol{\Phi}}_w^T\tilde{\boldsymbol{\Phi}}_w$, the transformation matrix can be obtained as

$$\tilde{\mathbf{G}} = \boldsymbol{\Phi}_c \tilde{\mathbf{U}} \tilde{\boldsymbol{\Lambda}}^{1/2} \tilde{\mathbf{V}}_1 = \boldsymbol{\Phi}_c \boldsymbol{\Psi} \tag{5.36}$$

where $\boldsymbol{\Psi} = \mathbf{U} \tilde{\boldsymbol{\Lambda}}^{-1/2} \tilde{\mathbf{V}}_1$.

Table 5.4: Algorithm KC

Training stage
1. Form the kernel matrices $\mathbf{K}_{11}$, $\mathbf{K}_{12}$ and $\mathbf{K}_{22}$ using a kernel function and (5.25), (5.27), (5.28), (5.29) and (5.32).
2. Evaluate the matrix $\boldsymbol{\Phi}_c^T \boldsymbol{\Phi}_c$ given in (5.26) by using (5.32).
3. Find $\tilde{\mathbf{U}}$ and $\tilde{\boldsymbol{\Lambda}}$ by eigen-decomposition of $\boldsymbol{\Phi}_c^T \boldsymbol{\Phi}_c$ given in (5.33).
4. Evaluate the projection of $\boldsymbol{\Phi}_w$ on $\mathbb{R}(\boldsymbol{\Phi}_c)$ using (5.34).
5. Find $\mathbf{V}$ through the eigen-decomposition given in (5.35) of $\tilde{\boldsymbol{\Phi}}_w^T \tilde{\boldsymbol{\Phi}}_w$.
6. $\tilde{\mathbf{V}}_2 \leftarrow \tilde{\mathbf{V}}(:, N : n-N)$.
7. Find the orthonormal basis of the DS using (5.36).
Classification stage
8. Form $\mathbf{k}_t$ using (5.38).
9. Find the projection $\boldsymbol{\psi}_t$ on feature vectors: $\boldsymbol{\pi} \leftarrow \boldsymbol{\Psi}^T \mathbf{k}_t$.

Finally, in order to determine as to the class to which a given test sample belongs to, the projection on the transformation vectors of its map $\boldsymbol{\psi}_t$ in the feature space has to be found. This projection using inner product can be express as

$$\boldsymbol{\pi} = \tilde{\mathbf{G}}^T \boldsymbol{\psi}_t = \boldsymbol{\Psi}^T (\boldsymbol{\Phi}_c^T \boldsymbol{\psi}_t) = \boldsymbol{\Psi}^T \mathbf{k}_t \tag{5.37}$$

where the column vector $\mathbf{k}_t$ can be evaluated as

$$\mathbf{k}_t = \begin{bmatrix} \mathbf{C}_2^T \boldsymbol{\Phi}_1^T \boldsymbol{\psi}_t - \boldsymbol{\Phi}_2^T \boldsymbol{\psi}_t \\ \\ \mathbf{C}_1^T \boldsymbol{\Phi}_1^T \boldsymbol{\psi}_t \end{bmatrix} \tag{5.38}$$

The kernel scheme just described can be summarized in the form of an algorithm, kernelized Algorithm C (Algorithm KC). This algorithm is summarized in Table 5.4.

5.5 Experiments

In this section, four different sets of experiments are carried out in order to illustrate the various ideas and schemes developed in this chapter. In the first set of experiments, the validity of the proposed discriminant model is examined. In the second set of experiments, the computational complexity of the three proposed algorithms is compared using three text document databases. The third set of experiments evaluates the performance of linear algorithms, Algorithm B and Algorithm C proposed in this chapter, and the PCA+LDA, MGSVD-LDA, MGSVD-OLDA and RLDA algorithms, using four human dace databases. In the last set of experiments, the performance of the proposed kernelized Algorithm C (Algorithm KC) and four other kernelized algorithms (KPCA+LDA, MGSVD-KDA, MGSVD-OKDA and KRDA algorithm) in comparison to that of the linear LDA algorithm is carried out using three large sample size databases. Ten databases are used in our experiments. Among them, FERET, YALE, ORL and AR are human face databases, and Dataset1, Daraset2 and Dataset3 are text document databases. Each of these six databases is small sample size database, in which samples are linearly independent. The remaining three databases, Isolet (a spoken letter database),

MUSK (a molecule conformation database), and MNIST (a handwritten digit database) are large sample size databases, in which samples are not linearly independent.

As discussed in the previous sections, if samples of a given database are linearly independent, they must be linearly separable. In the case of large sample size databases, samples are not linearly independent; kernelization is, therefore, necessary to establish the linear independence condition by using an appropriate kernel. Since the main objective of the kernelization is to achieve a linear independence among samples, the same simple kernel, a nonhomogeneous polynomial kernel [4], is chosen for all the nonlinear algorithms considered to establish the linear independence so that the differences in the recognition accuracy of the various algorithms can be attributed to feature extraction process of the algorithms. The nonhomogeneous polynomial kernel is given as

$$k(\langle \mathbf{x}_l, \mathbf{x}_h \rangle) = \left(\langle \mathbf{x}_l, \mathbf{x}_h \rangle + 1\right)^d \tag{5.39}$$

where d is a positive integer. In the classification stage, the nearest neighbor classifier [3] is used for all the algorithms.

For the PCA+LDA and KPCA+LDA algorithms, the largest N-1 eigenvalues, where N is the number of the classes, and the corresponding eigenvectors are used in the first stage for dimension reduction. The optimal regularization parameter for the RLDA and the optimal kernel parameter, d, for all the kernelized algorithms are estimated through the k-fold cross validation method, where $k \geq 10$, by using a part of the training samples for the actual training and the remaining for estimation. The parameter corresponding to the highest average recognition accuracy over the k iterations is chosen as the optimal parameter. The KRDA algorithm has two parameters, regularization and kernel, to be

estimated; hence, a double cross-validation is needed. For each database, ten sets of samples are randomly chosen and each algorithm is run using one set at a time. The average recognition rate and execution time of an algorithm is determined as an average taken over ten runs of the algorithm. Parameter estimation time is not explicitly given, but can be estimated as the product of the number of parameter candidates k, and the execution time. The execution platform used is dual core AMD opteron (tm), processor 180, 2.41 GHz, 2.0 GB RAM and Win XP operating system.

(a) Validation of the discriminant model

According to the proposed discriminant model, if the samples of a dataset are linearly independent, then there must exist a set of transformation vectors $\{\hat{\mathbf{g}}_1,\ldots\hat{\mathbf{g}}_{N-1}\}$ and a set of parallel hyperplanes normal to $\hat{\mathbf{g}}_l$ such that all the samples of a class of the dataset lie on one of hyperplanes and have a distinct point on $\hat{\mathbf{g}}_l$ as their projection. In this set of experiments, we validate this model experimentally.

The proposed algorithms, Algorithm A, Algorithm B, Algorithm C, and three other algorithms, namely MGSVD-LDA, RLDA and PCA+LDA, are applied on the small size face database AR as an example of a dataset of linearly independent samples, whereas the kernelized version of Algorithm C (that is, the KC algorithm) and the kernelized versions of the other three algorithms, MGSVD-KDA, KRDA and KPCA+LDA, are applied on the large sample size database Isolet as an example of linearly dependent dataset. Since the vector $\hat{\mathbf{g}}_l$'s are expected to be mutually orthogonal, for the purpose of illustration, any pair of arbitrarily chosen vectors can be used to form a plane in two dimensions.

Figure 5.1-Figure 5.4 depicted the projections of the samples on such two-dimensional planes, where only eight randomly selected classes are shown for the sake of simplicity. These figures show that if the samples are linearly independent or become linearly independent through nonlinear kernel mapping, the projections of the samples of each class condense to a distinct single point on the plane formed by any two transformation vectors that are derived from the proposed linear or kernelized algorithms. Thus, this example illustrates that, for the linearly independent samples, the proposed discriminant model holds. It is clear that the RLDA and PCA+LDA algorithms and their kernelized versions, the RLDA and KPCA+LDA algorithms, which are not designed on this discriminant model, are not as successful as the proposed algorithms in discriminating the classes.

(b) Computational efficiency of the proposed linear algorithms

As it has been pointed out in Section 5.4, the application of Algorithm A is limited to low-dimensional datasets because of its high computational complexity. Thus, it suffers from memory overflow problem when applied to patterns, such as human faces, that have a very high dimension. Hence, in order to compare the performance of the three proposed algorithms, Algorithms A, B, and C, the three text document databases are used in this experiment and their dimensions are reduced by one-half in order to avoid memory overflow of Algorithm A. The results of the experiment are given in Table 5.5. From these results, we can make the following observations:

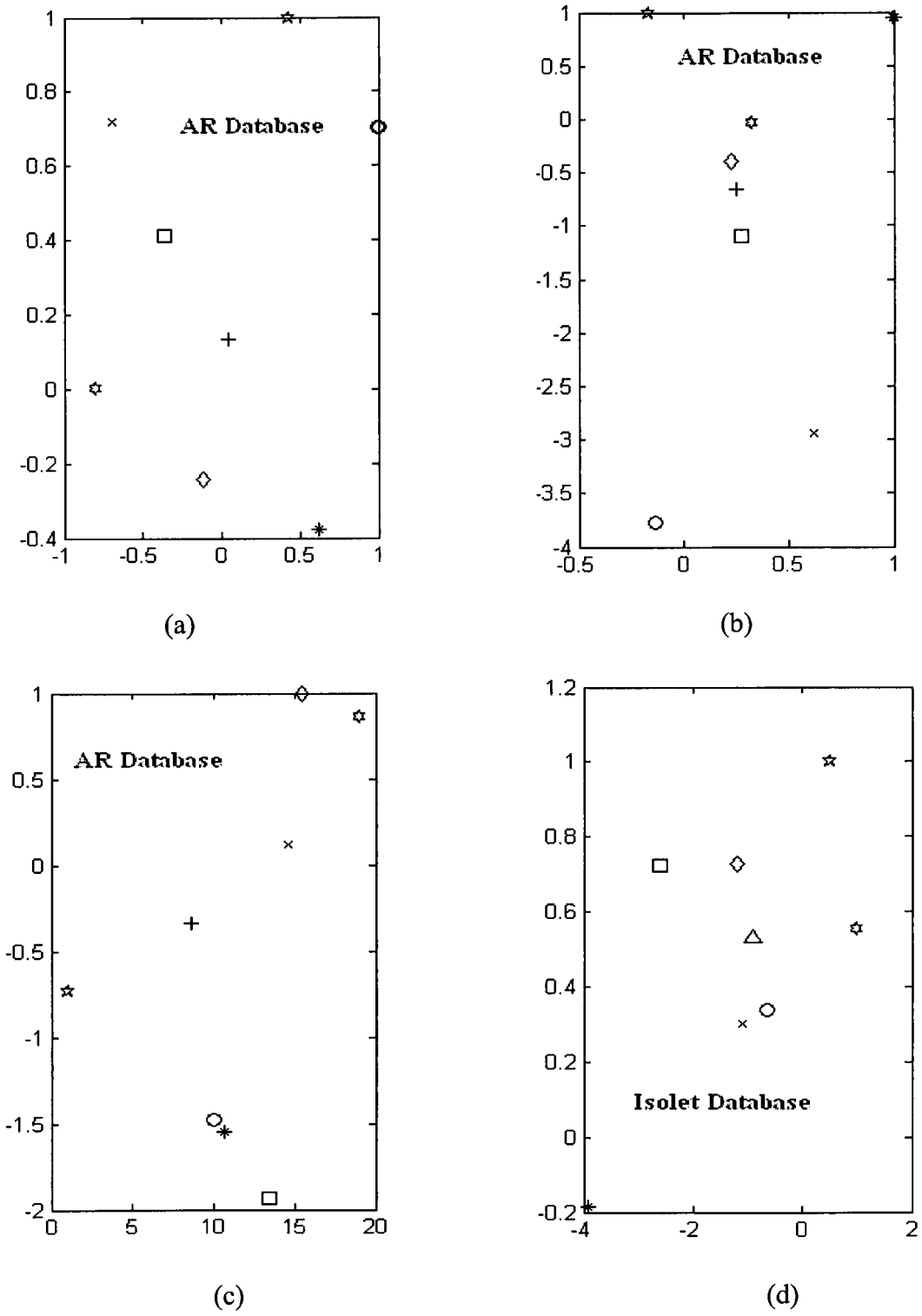

Figure 5.1: Sample projections in a two-dimensional discriminant subspace using algorithms (a) Algorithm A, (b) Algorithm B, (c) Algorithm C and (d) Algorithm KC

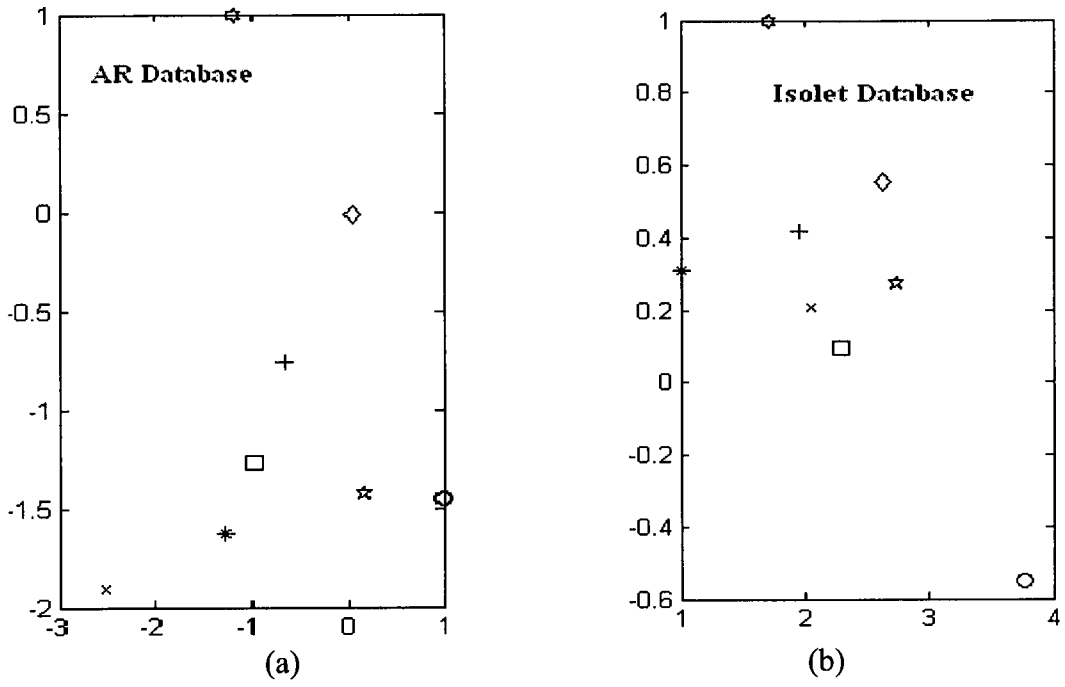

Figure 5.2: Sample projections in a two-dimensional discriminant subspace using algorithms (a) MGSVD-LDA and (b) MGSVD-KDA

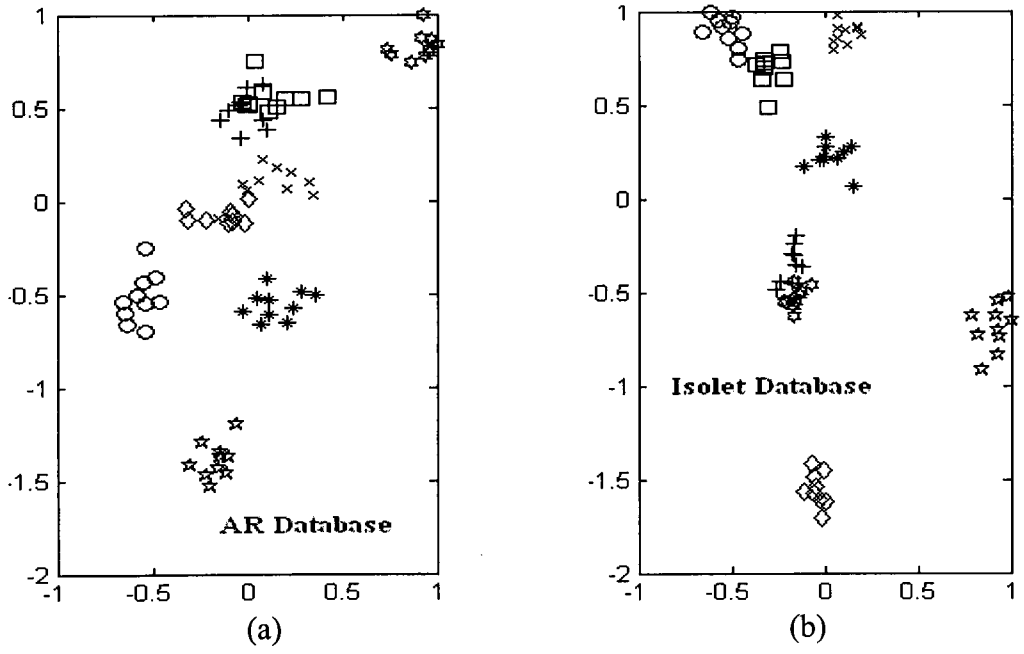

Figure 5.3: Sample projections in a two-dimensional discriminant subspace using algorithms (a) RLDA and (b) KRDA

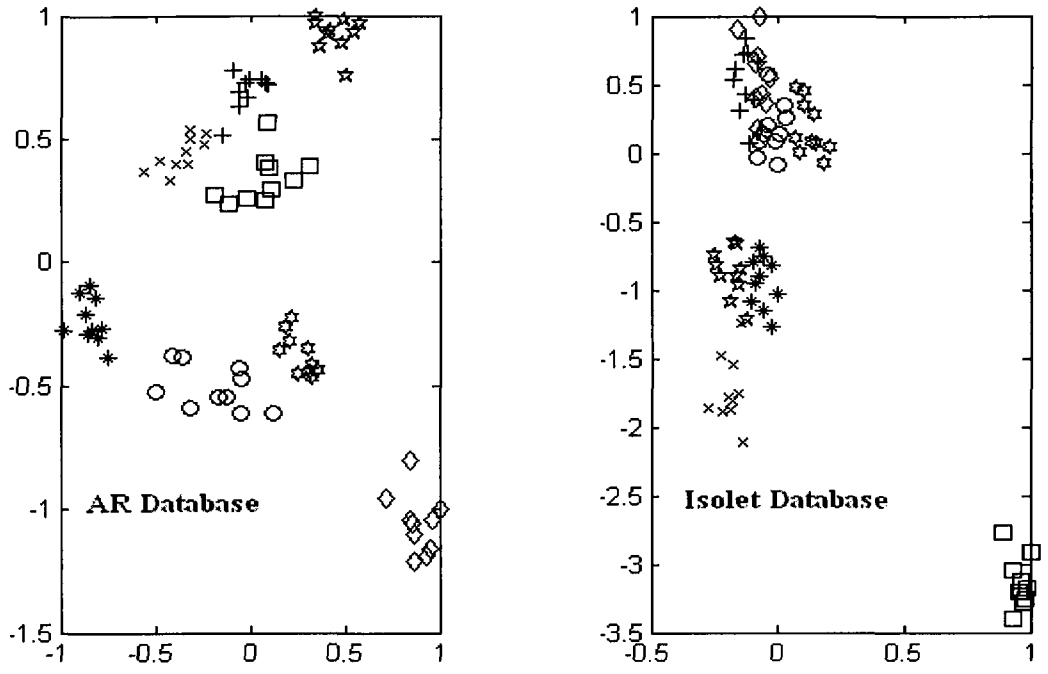

Figure 5.4: Sample projections in a two-dimensional discriminant subspace using algorithms (a) PCA+LDA and (b) KPCA+LDA

1) All the three proposed algorithms have the same recognition accuracy for the three text document databases.

2) The execution times of Algorithms B and C are significantly lower than that of Algorithm A.

3) For the three text document databases, the execution time of Algorithm B is lower than that of Algorithm C.

As expected, the three algorithms provide equally high recognition accuracy, since all the three are based on the same proposed discriminant model. However, their execution times are different. The computationally demanding of Algorithm A is the solution of the

linear equations $\mathbf{Z}^T\mathbf{A}_w = \mathbf{0}$ to obtain the null space of $\mathbf{A}_w$, $\mathbb{N}(\mathbf{A}_w)$. Since the dimension m is still much larger than the sample size n, the basis of $\mathbb{N}(\mathbf{A}_w)$, whose size is m x (m-n+N), requires a significant amount of computation. Implementation of neither Algorithm B nor Algorithm C involves computing the complete null space of $\mathbf{A}_w$; hence, these algorithms are computationally more efficient.

Table 5.5: Performance of the three proposed linear algorithms

Database	Dataset1		Dataset2		Dataset3	
Algorithm	Recognition Rate (%)	Execution Time in Seconds	Recognition Rate (%)	Execution Time in Seconds	Recognition Rate (%)	Execution Time in Seconds
Algorithm A	**96.4**	**16.23**	**83.54**	**11.58**	**91.79**	**3.71**
Algorithm B	**96.4**	**0.20**	**83.54**	**0.20**	**91.79**	**0.07**
Algorithm C	**96.4**	**0.29**	**83.54**	**0.49**	**91.79**	**0.09**

(c) Performance evaluation using datasets with linearly independent samples

In this set of experiments, the performance of the proposed linear algorithms, Algorithm B and Algorithm C, are evaluated and compared with that of the PCA+LDA, MGSVD-LDA and RLDA algorithms in terms of the recognition accuracy and execution

time using three human face databases. The three human face databases used for this evaluation are FERET, YALE, ORL and AR. The results of the experiment are given in Table 5.6. From the results, we have the following observations:

1) As in the case of low-dimension text databases, the two proposed algorithms provide high level of recognition accuracy and low execution time for large dimensional databases as well.

2) The recognition accuracy of Algorithm B and Algorithm C is higher than those of the MGSVD-LDA and PCA + LDA algorithms. The execution times of these four algorithms are about the same.

3) The proposed algorithms are competitive to the RLDA algorithm in terms of the recognition accuracy. However, the execution time of the RLDA algorithm is much larger than that of Algorithm B or Algorithm C due to the requirement of selection of the regulation parameter in the former.

Thus, taking into consideration both the recognition accuracy and execution time, the proposed Algorithm B and Algorithm C outperform the MGSVD-LDA, RLDA and PCA+LDA algorithms. The computational complexity of Algorithm C depends mainly on the operation of the eigen-decomposition of $\mathbf{A}_c$, whose size is $(n+N)$ x $(n+N)$. In contrast, the implementation of Algorithm B involves mainly the inner product computations and some arithmetic operations of vectors in obtaining the discriminant subspace. This difference in the computational complexities of Algorithm B and Algorithm C can be attributed to the lower execution time of the former.

(d) Performance Evaluation Using Datasets with Linearly Dependent Samples

In this set of experiments, the performance of the proposed kernelized algorithm, the Algorithm KC, and that of three other kernelized algorithms, the MGSVD-KDA, KRDA and KPCA+LDA algorithms, are assessed using three large sample size databases, MUSK, Isolet, and MNIST. We also compare the performance of the kernelized algorithms with that of the linear LDA algorithm. Table 5.7 shows the simulation results of this experiment. The purpose of including this linear algorithm is to illustrate the effectiveness of kernelization in each of the nonlinear algorithms.

1) Compared to the LDA algorithm, all the kernelized algorithms enhance the recognition accuracy significantly.

2) The proposed KC algorithm provides the highest recognition accuracy compared to the other kernelized algorithms.

3) The proposed KC algorithm requires a smaller execution time than its competitors except for the KPCA+LDA algorithm with the Isolet database.

The process of kernelization facilitates the samples to achieve a linear independence to capture the class structures of the databases in a high dimensional space. Compared to other kernelized algorithms, the proposed Algorithm KC provides the highest recognition accuracy with low computational load.

Table 5.6: Recognition rate (%) and execution time (seconds) of the proposed and some other linear algorithms with small sample size databases

Database	FERET		YALE		AR		ORL	
Algorithm	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
Algorithm B	**99.40**	**1.12**	**98.71**	**1.13**	**96.86**	**0.59**	**97.37**	**1.00**
Algorithm C	**99.40**	**1.20**	**96.92**	**1.21**	**96.86**	**0.69**	**97.37**	**1.28**
MGSVD-LDA	**97.43**	**1.20**	**96.92**	**1.28**	**95.71**	**0.71**	**96.37**	**1.44**
MGSVD-OLDA	**99.4**	**1.21**	**98.71**	**1.29**	**96.86**	**0.72**	**97.37**	**1.45**
RLDA	99.14	7.72	98.71	9.98	96.85	3.92	97.18	11.64
PCA+LDA	95.37	1.19	95.98	1.26	95.95	0.69	95.12	1.42
PCA	90.37	1.09	91.23	1.15	92.52	0.62	92.91	1.29

Table 5.7: Recognition rate (%) and execution time (seconds) of the proposed and some other kernelization algorithms with large sample size datasets

Database	Musk		Isolet		MNIST	
Algorithm	Recognition Rate	Execution Time	Recognition Rate	Execution Time	Recognition Rate	Execution Time
LDA	82.67	0.09	85.90	3.83	85.83	7.06
KC	**93.50**	**1.50**	**94.25**	**50.30**	**94.53**	**88.58**
MGSVD-KDA	**92.47**	**2.53**	**92.91**	**53.02**	**93.50**	**136.66**
MGSVD-OKDA	**93.50**	**2.55**	**94.25**	**53.50**	**94.52**	**137.01**
KRDA	92.33	279.28	93.16	370.64	93.83	1163.67
KPCA+LDA	91.50	2.49	91.90	53.58	91.53	111.91
KPCA	90.2	2.27	89.4	51.52	90.01	109.54

5.6 Summary

In this chapter, a systematic framework for the feature extraction of the linearly independent samples have been developed, which effectively addresses the SSS problem. Within this framework, first a discriminant model for the linearly independent samples has been established. If the samples of a dataset with N classes are linearly independent, then in accordance with this model, it has been shown that there exists a set of N-1 mutually orthogonal transformation vectors forming a discriminant subspace. For each of the transformation vectors, there exists a set of N parallel hyperplanes that are normal to this transformation vector. All the samples of one class lie totally on one of the N parallel hyperplanes and have a single distinct point on this transformation vector as their projections. Based on the proposed discriminant model, three algorithms have been developed. Whereas the first algorithm has been designed for low-dimensional datasets, the other two have been designed without such a restriction on the data dimensionality. Since the samples of a dataset are not linearly independent when the sample size is larger than its dimension, a kernelized algorithm has also been developed for the discriminant analysis of such datasets.

Extensive experiments have been conducted using benchmark database to demonstrate the validity of the proposed discriminant model for linearly independent samples. It has been shown that if the samples are linearly independent or made linearly independent through a nonlinear kernel mapping, the projections of the samples of each class condense into a distinct single point on any of the transformation vectors that are derived from the proposed linear or kernelized algorithms. It has been demonstrated that the three proposed linear algorithms provide a solution to the pattern recognition problem yielding

an equally high recognition accuracy with two of them consuming significantly less computational time. Simulation results using benchmark datasets have also shown that these two algorithms, in general, outperform the other existing linear algorithm in terms of recognition accuracy and execution time. Simulation results have also demonstrated that the proposed kernelized algorithm provides high recognition accuracy with low computational load compared to other kernelized algorithms.

Chapter 6

Conclusion

6.1 Concluding Remarks

Fisher's linear discriminant analysis provides an effective solution to many pattern recognition applications. However, it has a limitation in that it requires the within-class scatter matrix to be non-singular, which in practice is not the case when the small sample size problem occurs. Many FLDA variants that have been proposed in the past to address the small sample size problem either suffer from the high computational complexity due to the way the high-dimensionality of the input samples are dealt with or lose some useful discriminant information in dealing with the singularity problem of the scatter matrices of the traditional FLDA. This research has been concerned with an in-depth study of the discriminant analysis and development of feature extraction algorithms that can effectively deal with the small samples size problem. With this objective, the work of this research has been divided into two parts.

In the first part of this study, an algorithm, referred to as the MGSVD-LDA algorithm, which overcomes the small sample size problem, has been developed by solving the problem of generalized singular value decomposition through eigen-decomposition. The proposed algorithm has provided an efficient solution to the singularity problem of the within-class scatter matrix of the Fisher criterion with low computational complexity and

high recognition accuracy. A scheme has also been developed to kernelize the proposed linear algorithm yielding a nonlinear algorithm for discriminant analysis when the samples of a dataset are not linearly separable and a direct application of the linear algorithm fails to separate the classes of the dataset. An orthogonalization technique has been proposed to deal with the over-fitting problem through an eigen-decomposition of the basis of the discriminant subspace derived from the proposed MGSVD-LDA algorithm.

A theorem has been established to determine the class structure of linearly independent samples in the discriminant subspace derived by the proposed MGSVD-LDA algorithm. According to this theorem, if the samples of a dataset are linearly independent in the input space, then all the samples in a class condense into a distinct single point of the discriminative subspace, whereas a pair of samples belonging to two different classes are separated by a distance determined by the number of samples in each of the two classes. Thus, under the linear independence condition, the classes of a dataset are linearly separable, that is, through the linear MGSVD-LDA algorithm the class structures of the datasets with linearly independent samples can be successfully captured. It has been shown that for small sample size datasets, in which the samples are linearly independent, kernelization of the proposed linear algorithms provides little improvement in the recognition accuracy. The theorem that establishes the class structure of linearly independent samples has also been used to estimate the numerical errors of the proposed linear and nonlinear algorithms. This estimate of numerical errors has also be used to develop a scheme to adjust the kernel parameters to minimize the numerical errors in the implementation of the nonlinear algorithm and thus to improve the recognition accuracy.

In the second part of this thesis, a discriminant model for datasets with linearly independent samples has been established. If the samples of a dataset with N classes are linearly independent, then in accordance with this model, it has been shown that there exists a set of N-1 mutually orthogonal transformation vectors forming a discriminant subspace. For each of the transformation vectors, there exists a set of N parallel hyperplanes that are normal to this transformation vector. All the samples belonging to a single class lie totally on one of the N parallel hyperplanes and have a single distinct point on this transformation vector as their projections. Based on this discriminant model, three linear algorithms that effectively deal with the adverse effects of the SSS problem have been developed to determine the discriminant subspace for a given dataset with linearly independent samples. One of the three algorithms has been designed for low-dimensional datasets, whereas the other two have been designed to deal effectively with the computational problem associated with the high dimensionality of patterns. A scheme has also been developed to kernelize one of the three proposed linear algorithms for the discriminant analysis of datasets with linearly dependent samples.

Extensive experiments have been conducted throughout this research using benchmark databases to investigate the validity and effectiveness of the ideas developed therein. Simulation results have been used to demonstrate the validity of the schemes and the model presented. It has also been shown that the discriminant analysis algorithms proposed in this thesis provide superior performance in terms of the recognition accuracy and computational complexity.

6.2 Scope for Further Investigation

While the work of this thesis has focused on developing efficient techniques for feature extraction and developing a discriminant model for datasets with linearly independent samples, in the opinion of the author of this thesis, there are a number of problems related to the work of this thesis that needs to be further investigated.

1. The crux of the SSS problem is in the use of the Fisher criterion that involves the inversion of a scatter matrix, which cannot be accomplished when it is singular. Existing solutions dealing with this problem has been computationally expensive or lack of accuracy. Thus, there is a need to devise a new rational and objective optimality criterion, altogether different from the Fisher criterion that does not have to deal with the inversion of matrices.
2. There is a need to develop new feature extraction algorithms dealing with more complex databases having subjects with missing pixels, outliers or occlusions, or subjects corrupted by various types of noise.
3. In dealing with the feature extraction of datasets with samples that are not linearly separable, almost invariably linear algorithms have been kernelized. However, the computational complexity of the kernelized algorithms is very high. Hence, there is a need to develop directly low-complexity nonlinear algorithms without recourse to the linear ones.

References

1. A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, Jan. 2000.
2. I. T. Jolliffe, *Principle component analysis,* New York: Springer, 1986.
3. K.-L. Du and M.N.S. Swamy, *Neural networks in a softcomputing framework,* London: Springer, April 2006.
4. K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. New York: Academic Press, 1990.
5. R.A. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179-188, 1936.
6. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.
7. M. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol.24, pp.498-520, 1933.
8. C. R. Rao, "The use and interpretation of principle component analysis in applied research," *Sankhya* A, vol. 26, pp. 329-358, 1964.
9. W. L. Poston and D. J. Marchette, "Recursive dimensionality reduction using Fisher's linear discriminant," *Pattern Recognition*, vol. 31, no.7, pp. 881-888, 1998.
10. D.X. Sun, "Feature dimension reduction using reduced-rank maximum likelihood estimation for hidden markov model," in *Proc. International Conference on Spoken Language Processing*, Philadelphia, USA, pp. 244-247, 1996.

11. E. L. Bocchieri and J. G. Wilpon, "Discriminative analysis for feature reduction in automatic speech recognition," in *Proc. IEEE International Conference of Acoustics, Speech and Signal Processing*, San Francisco, vol. 1, pp. 501-554, 1992.

12. N. A. Campbell, "Shrunken estimators in discriminant and canonical variant analysis," *Applied Statistics*, vol. 29, no. 1, pp. 5-14, 1980.

13. R. P. W. Dubi and E. Backer, "Discriminant analysis in non-probabilistic context based on fuzzy labels," *Pattern Recognition and artificial intelligence*, pp. 229-235, 1988.

14. T. J. Hastie, R. Tibshirani, "Flexible discriminant analysis by optimal scoring," *AT&T Bell Labs Technical Report*, December, 1993.

15. Q. Tian, M. Barbero, Z.H. Gu, and S.H. Lee, "Image classification by the Foley-Sammon transform," *Optical Eng.*, vol. 25, no. 7, pp. 834-840, 1986.

16. Y.Q. Cheng, Y.M. Zhuang, and J.Y. Yang, "Optimal fisher discriminate analysis using the rank decomposition," *Pattern Recognition*, vol. 25, pp. 101-111, 1992.

17. G. Robertson, R.L. Kirlin, and W.-S. Lu, "A pseudo-inverse update algorithm for rank-reduced covariance matrices from 2-D data," *Signal Processing Letters*, vol. 4, pp. 230-231, Aug. 1997.

18. D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 831–836, Aug. 1996.

19. H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067-2070, 2001.

20. Y. Bing, J. Lianfu, and C. Ping, "A new LDA-based method for face recognition," in *Proc. 16th Int'l Conf. Pattern Recognition*, vol. 1, pp. 168-171, Aug. 2002.

21. D. Q. Dai and P.C. Yuen, "Regularized discriminant analysis and its application to face recognition," *Pattern Recognition*, vol. 36, pp. 845-847, 2003.

22. R. Huang, Q. Liu, H. Lu, and S. Ma, "Solving the small size problem of LDA," in *Proc. 16th Int'l Conf. Pattern Recognition*, vol. 3, pp. 29-32, Aug. 2002.

23. J. Yang and J. Y. Yang, "Why can LDA be performed in PCA transformed space?" *Pattern Recognition*, vol. 36, no. 2, pp. 563-566, 2003.

24. J. Yang and J. Y. Yang, "Optimal FLD algorithm for facial feature extraction," in *Proc. SPIE Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision*, pp. 438-444, Oct. 2001.

25. M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *Journal of Cognition. Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

26. J. Yang, D. Zhang, and J.-Y. Yang, "A generalized K-L expansion method which can deal with small sample size and high- dimensional problems," *Pattern Analysis & Applications*, vol. 6, pp. 47-54, April, 2003.

27. C. Lee and D. Landgrebe, "Analyzing high-dimensional multispectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 7, pp. 792–800, Jul. 1993.

28. H. Gao, and W. D. James, "Why direct LDA is not equivalent to LDA," *Pattern Recognition*, vol. 39, 2006, pp1002-1996

29. Z.-Q. Hong and J.-Y. Yang, "Optimal discriminant plane for a small number of samples and design method of classifier on the plane," *Pattern Recognition*, vol. 24, pp. 317–324, 1991.

30. W. Zhao, R. Chellappa, and A Krishna swamy, "Discriminant analysis of principal components for face recognition," in *Proc. Third IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 336-341, Apr. 1998.

31. J. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition," *Pattern Recognition Letters*, vol. 26, pp.181-191, 2005.

32. J. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE, Trans. on Neural Networks*, vol. 17, no. 1, pp.166-178, Jan. 2006.

33. J. Ye, T. Xiong, Q. Li, R. Janardan, J. Bi, V. Cherkassky and C. Kambhamettu, "Efficient model selection for regularized linear discriminant analysis," in *Proc. International Conference on Information and Knowledge Management*, Arlington, Virginia, USA, pp. 5-11, Nov. 2006.

34. P. Howland, M. Jeon, and H. Park, "Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition," *SIAM J. Matrix Anal. Appl.* vol. 25, no. 1, pp. 165-179, 2003.

35. J. Ye, R. Janardan, C. H. Park, and Haesun Park, "An optimization criterion for generalized discriminant analysis on under-sampled problems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 982-994, Aug. 2004.

36. C. H. Park and H. Park, "A comparison of generalized linear discriminant analysis algorithms," *Pattern Recognition*, vol. 41, pp. 1083-1097, 2008.

37. Wei Wu, M.O. Ahmad, and S. Samadi, "Discriminant analysis based on modified generalized singular value decomposition and its numerical error analysis," *IET Computer Vision*, vol.3, no.3, pp.159-173, September 2009.

38. Wei Wu and M.O. Ahmad, "Orthogonalized discriminant analysis based on generalized singular value decomposition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Taipei, Taiwan, pp.1833-1836, April 2009.

39. Wei Wu and M.O. Ahmad, "Orthogonalized linear discriminant analysis based on modified generalized singular value decomposition," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS),* Taipei, Taiwan, pp.1629 - 1632, May 2009.

40. Wei Wu and M.O. Ahmad, "A discriminant model for the pattern recognition of linearly independent samples," submitted to *IET Computer Vision.*

41. L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, pp. 1713–1726, 2000.

42. K. Liu, Y.-Q. Cheng, and J.-Y. Yang, "A generalized optimal set of discriminant vectors," *Pattern Recognition*, vol. 25, no. 7, pp. 731-739, 1992.

43. K. Liu, Y.-Q. Cheng, J.-Y. Yang, and X. Liu, "An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic method," in *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 6, no. 5, pp. 817-829, 1992.

44. W. Zhao, R. Chellappa, and J. Phillips, "Subspace linear discriminant analysis for face recognition," *Technical Report CS-TR4009*, Univ. of Maryland, 1999.

45. M. Kirby and L. Sirovich, "Application of the KL procedure for the characterization of human faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, Jan. 1990.

46. J. Yang, J.-y. Yang, and A.F. Frangi, "Combined Fisherfaces framework," *Image and Vision Computing*, vol. 21, no. 12, pp. 1037-1044, 2003.

47. V. N. Vapnik, *The nature of statistical learning theory*, 2nd ed. New York: Springer-Verlag, pp. 30–31, 1995.

48. H. Cevikalp, M. Neamtu and M. Wilkes, "Discriminative common vector method with kernels," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, p 1550-1565, November 2006.

49. B. Scholkopf; C. Burges; A. J. Smola, *Advances in kernel methods: support vector learning,* Cambridge, Mass.: MIT Press, 1999.

50. R. Herbrich, *Learning Kernel Classifiers* The MIT Press, 2002.

51. J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis,* Cambridge University Press, 2004.

52. N. Cristianini, J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning algorithms*, New York, Cambridge University Press, 2000.

53. S. Abe, *Support vector machines for pattern classification,* Springer, 2005.

54. C. Burges, "A tutorial on support vector machine for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

55. B. Sch"olkopf and A. J. Smola, *Learning with kernels - support vector machines, regularization, optimization, and beyond*, MIT, Cambridge, MA, 2002.

56. J. S. Taylor and N. Cristianini, *Kernel methods for pattern analysis,* Cambridge University Press, 2004.

57. B. Sch"olkopf, A. J. Smola and K. R. M"uller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

58. S. Mika, G. Rätsch, J.Weston, B. Schölkopf, and K.-R.Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX, IEEE*, New York, pp. 41–48, 1999.

59. G. Baudat and F. Fanouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385-2404, 2000.

60. G. Baudat, F. Anouar, "Generalized discriminant analysis, codes for Matlab 5", http://www.kernel-machines.org/index.html.

61. S. Mika, G. Rätsch, B. Schölkopf, A. Smola, J. Weston, and K.-R. Müller, "Invariant feature extraction and classification in kernel spaces," *Advances in Neural Information Processing Systems 12*, Cambridge, Mass.: MIT Press, 1999.

62. V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," *Advances in Neural Information Processing Systems*, S.A. Solla, T.K. Leen, and K.-R. Mueller, eds., vol. 12, pp. 568-574, MIT Press, 2000.

63. S. Mika, G. Ratsch, and K.-R. Mu¨ ller, "A mathematical programming approach to the kernel fisher algorithm," in *Advances in Neural Information Processing Systems 13*, T.K. Leen, T.G. Dietterich, and V. Tresp, eds., pp. 591-597, MIT Press, 2001.

64. S. Mika, A.J. Smola, and B. Scho¨ lkopf, "An improved training algorithm for kernel Fisher discriminants," in *Proc. Eighth Int'l Workshop Artificial Intelligence and Statistics,* T. Jaakkola and T. Richardson, eds., pp. 98-104, 2001.

65. S. Mika, G. Rätsch, J Weston, B. Schölkopf, A. Smola, and K.-R. Müller, "Constructing descriptive and discriminative nonlinear features: rayleigh coefficients in kernel feature spaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 623-628, May 2003.

66. M. H. Yang, "Kernel Eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods," in *Proc. Fifth IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 215-220, May 2002.

67. J. Xu, X. Zhang, and Y. Li, "Kernel MSE algorithm: a unified framework for KFD, LS-SVM, and KRR," in *Proc. Int'l Joint Conf. Neural Networks*, pp. 1486-1491, July 2001.

68. S. A. Billings and K.L Lee, "Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm," *Neural Networks*, vol. 15, no. 2, pp. 263-270, 2002.

69. T. V. Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vanderwalle, "Bayesian framework for least squares support vector machine classifiers, Gaussian processes and kernel Fisher discriminant analysis," *Neural Computation*, vol. 15, no. 5, pp. 1115-1148, May 2002.

70. G. C. Cawley and N.L.C. Talbot, "Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers," *Pattern Recognition*, vol. 36, no. 11, pp. 2585-2592, 2003.

71. N.D. Lawrence and B. Scho¨lkopf, "Estimating a kernel Fisher discriminant in the presence of label noise," in *Proc. 18th Int'l Conf. Machine Learning*, pp. 306-313, 2001.

72. J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos and J. Wang, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp.117-126, 2003.

73. J. Yang, A. F. Frangi, J. Yang, D. Zhang and Z. Jin, "KPCA plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp.230-244, February 2005.

74. J. Yang, A. F. Frangi, Z. Jin, and J.-Y. Yang, "Essence of kernel Fisher discriminant: KPCA plus LDA," *Pattern Recognition*, vol. 37, pp. 2097–2100, Oct. 2004.

75. J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos and J. Wang, "An efficient kernel discriminant analysis method," *Pattern Recognition*, vol. 38, no. 10, pp. 1788-1790, October 2005.

76. C. H. Park and H. Park, "Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition," *SIAM Journal on Matrix Analysis Application*, vol. 27, no. 1, pp. 87-102, 2005.

77. J. Yang, A.F. Frangi, and J.-Y. Yang, "A new kernel Fisher discriminant algorithm with application to face recognition," *Neurocomputing*, vol. 56, pp. 415-421, 2004.

78. J. H. Wilkson, *The algebraic eigenvalue problem*, Oxford: Clarendon Press, 1965.

79. B. N. Datta, *Numerical linear algebra and application,* ITP, 1995.

80. G. H. Golub and C. F. Van Loan, *Matrix Computation,* 3rd edition, The Johns Hopkins University Press, 1996.

81. P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, second ed. Orlando, Fla.: Academic Press, 1985.

82. C. F. Van Loan, "Generalizing the singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 13, no. 1, pp.76-83, March 1976.

83. C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398-405, June 1981.

84. Degang Chen, Qiang He; Xizhao Wang, "On linear separability of data sets in feature space", *Neurocomputing*, vol. 70, no. 13-15, pp. 2441-8, Aug. 2007.

85. C. Liu and H. Wechsler, "Enhanced Fisher linear discriminant models for face recognition," in *Proc. International Conference on Pattern Recognition, IEEE*, 1998.

86. J. Ye and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis," *Journal of Machine Learning Research*, vol. 7, pp. 1183-1204, 2006.

87. C. Burges, "A tutorial on support vector machine for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

88. H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 4–13, Jan. 2005.

89. H. Cevikalp, M. Neamtu, and M. Wilkes, "Discriminative common vector method with kernels," *IEEE Trans. Neural Networks*, Vol. 17, no. 6, Nov. 2006.

90. P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090-1104, Oct. 2000.

91. P.J. Phillips, "The facial recognition technology (FERET) database," http://www.itl.nist.gov/iad/humanid/feret/feret_master.html, 2004.

92. A. M. Martinez and R. Benavente, the AR Face Database, CVC Technical Report #24, June 1998. http://rvl1.ecn.purdue.edu/_aleix/aleix face DB.html

93. http://cvc.yale.edu/projects/yalefaces/yalefaces.html

94. "TREC", "Proc. text tetrieval conf.," http://trec.nist.gov, 1999.

95. D. D. Lewis, "Reuters-21578 text cagtegorization test collection distribution 1.0," http://www.research.att.com/lewis, 1999.

96. D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz (1998), UCI Repository of machine learning databases http://www.ics.uci.edu/ mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

97. AT&T Laboratories Cambridge, The ORL Database of Faces [Online]. Available: http://www.uk.research.att.com/facedatabase.html

98. MNIST database http://yann.lecun.com/exdb/mnist/

Appendix A

Derivation of the Expressions for $\Phi_b^T\Phi_b$, $\Phi_b^T\Phi_w$, and $\Phi_w^T\Phi_w$ in the MGSVD-KDA Algorithm

In this section, we give the expressions for $\Phi_b^T\Phi_b$, $\Phi_b^T\Phi_w$, and $\Phi_w^T\Phi_w$, which are the submatrices of $\mathbf{\Gamma\Gamma}^T$ given by (3.27).

1) *Derivation of* $\Phi_b^T\Phi_b$

$$\Phi_b^T\Phi_b$$

$$=\left[\sqrt{n_1}(\psi^{(1)}-\psi),\ldots,\sqrt{n_N}(\psi^{(N)}-\psi)\right]^T\left[\sqrt{n_1}(\psi^{(1)}-\psi),\ldots,\sqrt{n_N}(\psi^{(N)}-\psi)\right]$$

$$=\left(\sqrt{n_in_j}(\psi^{(i)}-\psi)^T(\psi^{(j)}-\psi)\right)_{\substack{i=1,\ldots,N\\ j=1,\ldots,N}}$$

$$=\left(\sqrt{n_in_j}(\psi^{(i)T}\psi^{(j)}-\psi^{(i)T}\psi-\psi^T\psi^{(j)}+\psi^T\psi\right)_{\substack{i=1,\ldots,N\\ j=1,\ldots,N}}$$

where

$$\psi^{(i)T}\psi^{(j)}=\frac{1}{n_in_j}\sum_{l=k_{i-1}+1}^{k_i}\psi_l^T\sum_{h=k_{j-1}+1}^{k_j}\psi_h$$

$$=\frac{1}{n_in_j}\sum_{l=k_{i-1}+1}^{k_i}\sum_{h=k_{j-1}+1}^{k_j}k_{lh}$$

$$\Rightarrow\left(\psi^{(i)T}\psi^{(j)}\right)_{\substack{i=1,\ldots,N\\ j=1,\ldots,N}}=\mathbf{B}^T\mathbf{K}\mathbf{B}$$

and $k_i=n_1+n_2+\ \ldots\ +n_i$ and $k_j=n_1+n_2+\ \ldots\ +n_j$

$$\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi} = \frac{1}{n_i n_j}\sum_{l=k_{i-1}+1}^{k_i}\sum_{h=1}^{n} k_{lh}$$

$$\Rightarrow \left(\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}\right)_{\substack{i=1,\ldots,N \\ j=1,\ldots,N}} = \mathbf{B}^T\mathbf{K}\mathbf{L}$$

$$\boldsymbol{\psi}^{T}\boldsymbol{\psi}^{(j)} = \frac{1}{n n_j}\sum_{l=1}^{n}\sum_{h=k_{j-1}+1}^{k_j} k_{lh}$$

$$\Rightarrow \left(\boldsymbol{\psi}^{T}\boldsymbol{\psi}^{(j)}\right)_{\substack{i=1,\ldots,N \\ j=1,\ldots,N}} = \mathbf{L}^T\mathbf{K}\mathbf{B}$$

$$\boldsymbol{\psi}^{T}\boldsymbol{\psi} = \frac{1}{n^2}\sum_{l=1}^{n}\boldsymbol{\psi}_l^T\sum_{h=1}^{n}\boldsymbol{\psi}_h$$

$$= \frac{1}{n^2}\sum_{l=1}^{n}\sum_{h=1}^{n} k_{lh}$$

$$\Rightarrow \left(\boldsymbol{\psi}^{T}\boldsymbol{\psi}\right)_{\substack{i=1,\ldots,N \\ j=1,\ldots,N}} = \mathbf{L}^T\mathbf{K}\mathbf{L}.$$

Hence, we have

$$\boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_b = \mathbf{D}(\mathbf{B}-\mathbf{L})^T\mathbf{K}(\mathbf{B}-\mathbf{L})\mathbf{D}.$$

2) *Expression for* $\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w$

$$\begin{aligned}
&\boldsymbol{\Phi}_w^T\boldsymbol{\Phi}_w \\
&= \left[(\boldsymbol{\psi}_1 - \boldsymbol{\psi}^{(1)}),\ldots,(\boldsymbol{\psi}_n - \boldsymbol{\psi}^{(N)})\right]^T\left[(\boldsymbol{\psi}_1 - \boldsymbol{\psi}^{(1)}),\ldots,(\boldsymbol{\psi}_n - \boldsymbol{\psi}^{(N)})\right] \\
&= \left((\boldsymbol{\psi}_l - \boldsymbol{\psi}^{(i)})^T(\boldsymbol{\psi}_h - \boldsymbol{\psi}^{(j)})\right)_{\substack{l\in(k_{i-1}+1,\ldots,k_i),h\in(k_{j-1}+1,\ldots,k_j) \\ i,j=1,\ldots,N}} \\
&= \left(\boldsymbol{\psi}_l^T\boldsymbol{\psi}_h - \boldsymbol{\psi}_l^T\boldsymbol{\psi}^{(i)} - \boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}_h + \boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}^{(j)}\right)_{\substack{l\in(k_{i-1}+1,\ldots,k_i),h\in(k_{j-1}+1,\ldots,k_j) \\ i,j=1,\ldots,N}}
\end{aligned}$$

where

$$\boldsymbol{\psi}_l^T \boldsymbol{\psi}_h = k_{lh}$$

$$\Rightarrow (\boldsymbol{\psi}_l^T \boldsymbol{\psi}_h)_{\substack{l\in(k_{i-1}+1,\ldots,k_i),h\in(k_{j-1}+1,\ldots,k_j) \\ i,j=1,\ldots,N}} = \mathbf{K}$$

$$\boldsymbol{\psi}_l^T \boldsymbol{\psi}^{(j)} = \frac{1}{n_j} \sum_{h=k_{j-1}+1}^{k_j} k_{lh}$$

$$\Rightarrow (\boldsymbol{\psi}_l^T \boldsymbol{\psi}^{(j)})_{\substack{l\in(k_{i-1}+1,\ldots,k_i),h\in(k_{j-1}+1,\ldots,k_j) \\ i,j=1,\ldots,N}} = \mathbf{KA}$$

$$\boldsymbol{\psi}^{(i)T} \boldsymbol{\psi}_h = \frac{1}{n_i} \sum_{l=k_{i-1}+1}^{k_i} k_{lh}$$

$$\Rightarrow (\boldsymbol{\psi}^{(i)T} \boldsymbol{\psi}_h)_{\substack{l\in(k_{i-1}+1,\ldots,k_i) \\ i,j=1,\ldots,N}} = \mathbf{AK}$$

$$\boldsymbol{\psi}^{(i)T} \boldsymbol{\psi}^{(j)} = \frac{1}{n_i n_j} \sum_{l=k_{i-1}+1}^{k_i} \sum_{h=k_{j-1}+1}^{k_j} k_{lh} k_{lh}$$

$$\Rightarrow (\boldsymbol{\psi}^{(i)T} \boldsymbol{\psi}^{(j)})_{\substack{l\in(k_{i-1}+1,\ldots,k_i),h\in(k_{j-1}+1,\ldots,k_j) \\ i,j=1,\ldots,N}} = \mathbf{AKA}.$$

Thus, the submatrix $\boldsymbol{\Phi}_w^T \boldsymbol{\Phi}_w$ can be expressed as

$$\boldsymbol{\Phi}_w^T \boldsymbol{\Phi}_w = (\mathbf{I} - \mathbf{A})^T \mathbf{K} (\mathbf{I} - \mathbf{A}).$$

3) *Expression for* $\mathbf{\Phi}_b^T\mathbf{\Phi}_w$

$$\mathbf{\Phi}_b^T\mathbf{\Phi}_w$$

$$=\left[\sqrt{\mathrm{n}_1}(\boldsymbol{\psi}^{(1)}-\boldsymbol{\psi}),\ldots,\sqrt{\mathrm{n}_\mathrm{N}}(\boldsymbol{\psi}^{(N)}-\boldsymbol{\psi})\right]\left[(\boldsymbol{\psi}_1-\boldsymbol{\psi}^{(1)}),\ldots,(\boldsymbol{\psi}_n-\boldsymbol{\psi}^{(N)})\right]$$

$$=\left(\sqrt{n_i}(\boldsymbol{\psi}^{(i)}-\boldsymbol{\psi})^T(\boldsymbol{\psi}_h-\boldsymbol{\psi}^{(j)})\right)_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}}$$

$$=\left(\sqrt{n_i}(\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}_h-\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}^{(j)}-\boldsymbol{\psi}^T\boldsymbol{\psi}_h+\boldsymbol{\psi}^T\boldsymbol{\psi}^{(j)}\right)_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}}$$

where

$$\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}_h=\frac{1}{n_i}\sum_{l=k_{i-1}+1}^{k_i}k_{lh}$$

$$\Rightarrow(\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}_h)_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}}=\mathbf{B}^T\mathbf{K}$$

$$\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}^{(j)}=\frac{1}{n_in_j}\sum_{l=k_{i-1}+1}^{k_i}\sum_{h=k_{j-1}+1}^{k_j}k_{lh}$$

$$\Rightarrow(\boldsymbol{\psi}^{(i)T}\boldsymbol{\psi}^{(j)})_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}}=\mathbf{B}^T\mathbf{K}\mathbf{A}$$

$$\boldsymbol{\psi}^T\boldsymbol{\psi}_h=\frac{1}{n}\sum_{l=1}^{n}k_{lh}$$

$$\Rightarrow(\boldsymbol{\psi}^T\boldsymbol{\psi}_h)_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}}=\mathbf{L}^T\mathbf{K}$$

$$\boldsymbol{\psi}^T\boldsymbol{\psi}^{(j)} = \frac{1}{nn_j}\sum_{l=1}^{n}\sum_{h=k_{j-1}+1}^{k_j} k_{lh}$$

$$\Rightarrow (\boldsymbol{\psi}^T\boldsymbol{\psi}^{(j)})_{\substack{h\in(k_{j-1}+1,\ldots,k_j)\\ i,j=1,\ldots,N}} = \mathbf{L}^T\mathbf{K}\mathbf{A}.$$

Thus, from the above, we have

$$\boldsymbol{\Phi}_b^T\boldsymbol{\Phi}_w = \mathbf{D}(\mathbf{B}-\mathbf{L})^T\mathbf{K}(\mathbf{I}-\mathbf{A}).$$

Appendix B

Examples of Some Typical Images in Face Databases

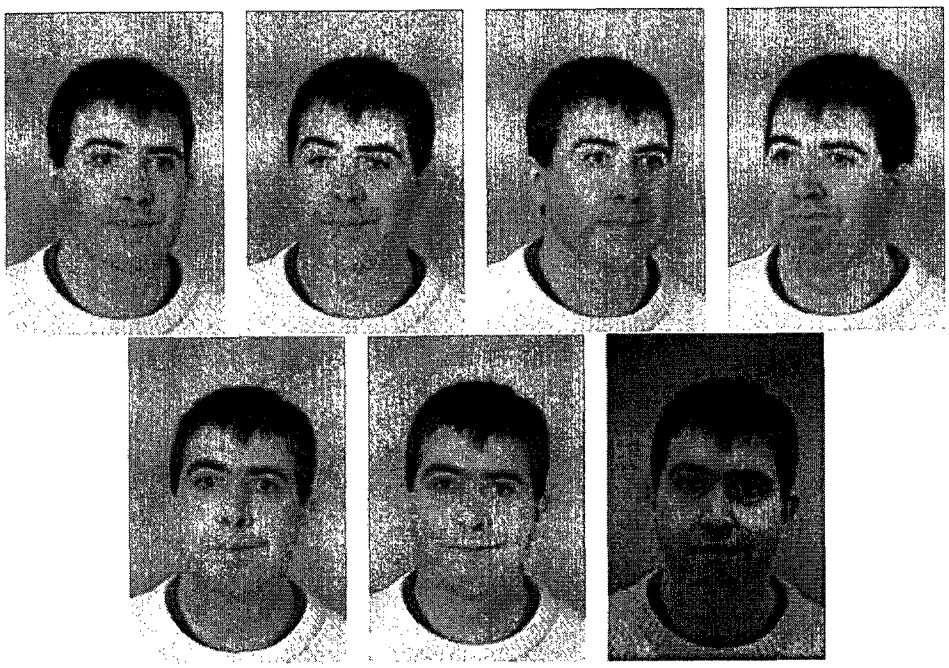

Figure B.1: Images of one subject in the FERET face database

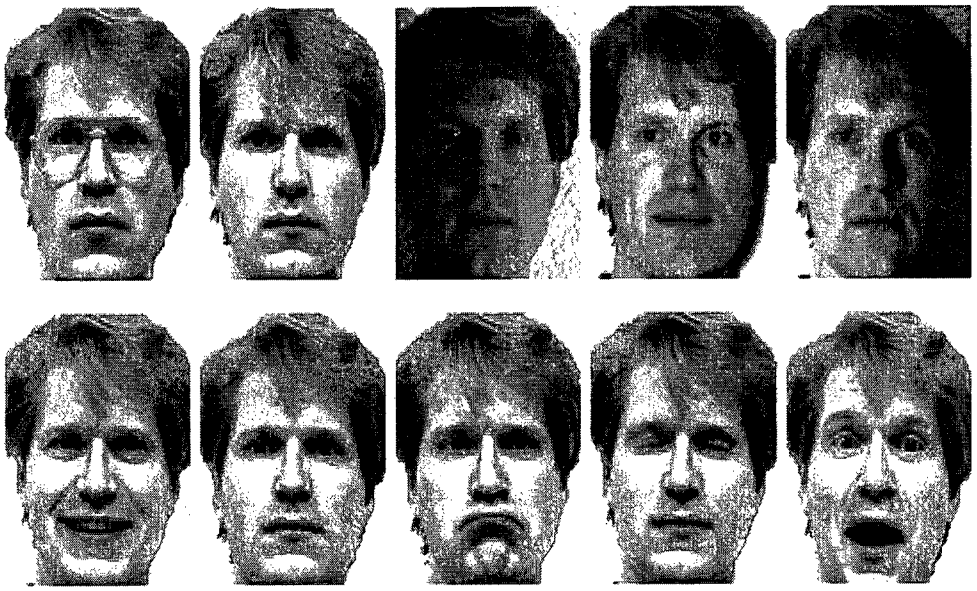

Figure B.2: Images of one subject in the YALE face database

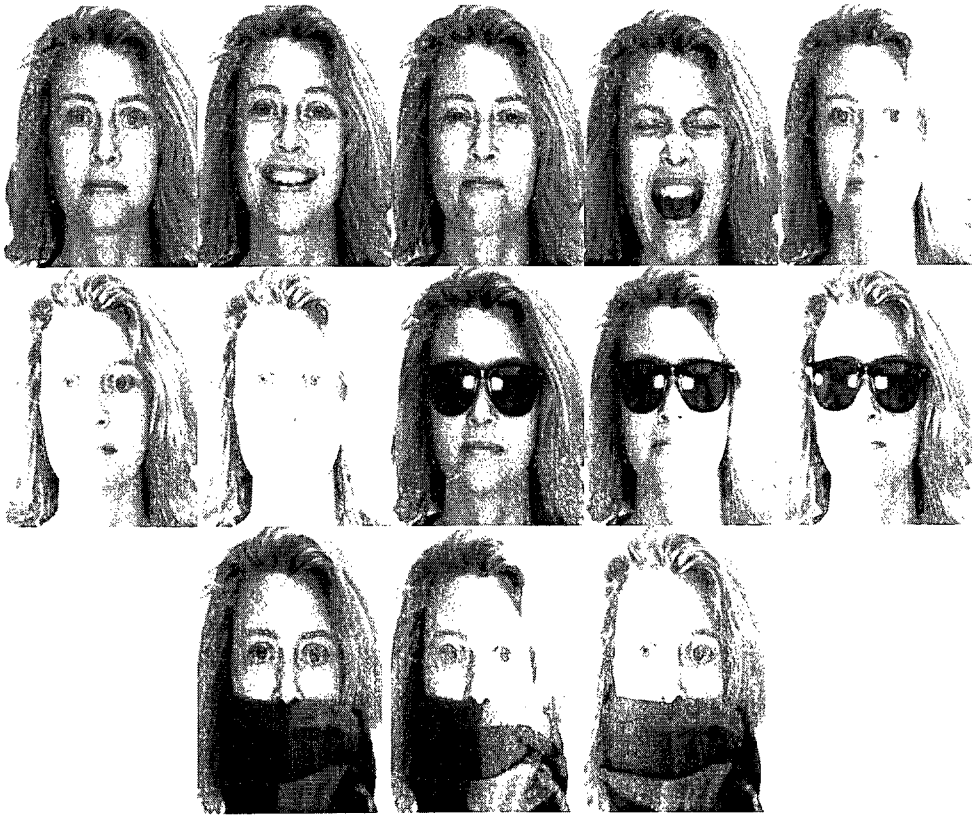

Figure B.3: Images of one subject in the AR face database.

Figure B.4: Images of one subject in the ORL face database.

Appendix C

Derivation of the Expressions for $\Phi_b^T\Phi_b$, $\Phi_b^T\Phi_w$, and $\Phi_w^T\Phi_w$ in Algorithm KC

In this section, we give the expressions for $\Phi_b^T\Phi_b$, $\Phi_b^T\Phi_w$, and $\Phi_w^T\Phi_w$, which are the submatrices of $\Phi_c^T\Phi_c$ given by (5.32).

1) *Derivation of* Φ_b

$$\Phi_b$$

$$= [(\psi_{21} - \psi_{11}), (\psi_{31} - \psi_{11}), \ldots, (\psi_{i1} - \psi_{11}), \ldots, (\psi_{N1} - \psi_{11})]$$

$$= [\psi_{11}, \psi_{21}, \ldots, \psi_{N1}] \begin{pmatrix} (-1)_{1\times(N-1)} \\ \mathbf{I}_{N-1} \end{pmatrix}$$

$$= \Phi_1 \mathbf{C}_1$$

where

$$\Phi_1 = [\psi_{11}, \psi_{21}, \ldots, \psi_{N1}],$$

$$\mathbf{C}_1 = \begin{pmatrix} (-1)_{1\times(N-1)} \\ \mathbf{I}_{N-1} \end{pmatrix}.$$

2) *Derivation of* $\mathbf{\Phi}_w$

$$\mathbf{\Phi}_w$$

$$= [(\psi_{12} - \psi_{11}), \ldots, (\psi_{1n_1} - \mathbf{x}_{11}), (\psi_{22} - \psi_{21}), \ldots, (\psi_{2n_2} - \psi_{21}), \ldots, (\psi_{N2} - \psi_{N1}), \ldots, (\psi_{Nn_N} - \psi_{N1})]$$

$$= [\psi_{12}, \ldots \psi_{1n_1}, \psi_{22}, \ldots, \psi_{2n_2}, \ldots, \psi_{N2}, \ldots, \psi_{Nn_N}] \begin{pmatrix} (\mathbf{1})_{1\times(n_1-1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (\mathbf{1})_{1\times(n_N-1)} \end{pmatrix}$$

$$= \mathbf{\Phi}_2 - \mathbf{\Phi}_1 \mathbf{C}_2$$

where

$$\mathbf{\Phi}_2 = [\psi_{12}, \ldots \psi_{1n_1}, \psi_{22}, \ldots, \psi_{2n_2}, \ldots, \psi_{N2}, \ldots, \psi_{Nn_N}],$$

$$\mathbf{C}_2 = \begin{pmatrix} (\mathbf{1})_{1\times(n_1-1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (\mathbf{1})_{1\times(n_N-1)} \end{pmatrix}.$$

1) *Derivation of* $\mathbf{\Phi}_b^T \mathbf{\Phi}_b$

$$\mathbf{\Phi}_b^T \mathbf{\Phi}_b$$

$$= (\mathbf{\Phi}_1 \mathbf{C}_1)^{\mathrm{T}} \ \mathbf{\Phi}_1 \mathbf{C}_1$$

$$= \mathbf{C}_1^T \mathbf{\Phi}_1^T \ \mathbf{\Phi}_1 \mathbf{C}_1$$

$$= \mathbf{C}_1^T \mathbf{K}_{11} \mathbf{C}_1$$

where $\mathbf{K}_{11} = \mathbf{\Phi}_1^T \mathbf{\Phi}_1$.

3) *Derivation of* $\mathbf{\Phi}_w^T \mathbf{\Phi}_w$

$$\mathbf{\Phi}_w^T \mathbf{\Phi}_w$$

$$= (\mathbf{\Phi}_2 - \mathbf{\Phi}_1 \mathbf{C}_2)^T (\mathbf{\Phi}_2 - \mathbf{\Phi}_1 \mathbf{C}_2)$$

$$= \mathbf{\Phi}_2^T \mathbf{\Phi}_2 - \mathbf{C}_2^T \mathbf{\Phi}_1^T \mathbf{\Phi}_2 - (\mathbf{C}_2^T \mathbf{\Phi}_1^T \mathbf{\Phi}_2)^T + \mathbf{C}_2^T \mathbf{\Phi}_1^T \mathbf{\Phi}_1 \mathbf{C}_2$$

$$= \mathbf{K}_{22} - \mathbf{C}_2^T \mathbf{K}_{12} - (\mathbf{C}_2^T \mathbf{K}_{12})^T + \mathbf{C}_2^T \mathbf{K}_{11} \mathbf{C}_2$$

where

$$\mathbf{K}_{11} = \mathbf{\Phi}_1^T \mathbf{\Phi}_1,$$
$$\mathbf{K}_{22} = \mathbf{\Phi}_2^T \mathbf{\Phi}_2,$$
$$\mathbf{K}_{12} = (\mathbf{K}_{21})^T = \mathbf{\Phi}_1^T \mathbf{\Phi}_2.$$

4) *Derivation of* $\mathbf{\Phi}_b^T \mathbf{\Phi}_w$

$$\mathbf{\Phi}_b^T \mathbf{\Phi}_w$$
$$= (\mathbf{\Phi}_1 \mathbf{C}_1)^T (\mathbf{\Phi}_2 - \mathbf{\Phi}_1 \mathbf{C}_2)$$

$$= \mathbf{C}_1^T \mathbf{\Phi}_1^T \mathbf{\Phi}_2 - \mathbf{C}_1^T \mathbf{\Phi}_1^T \mathbf{\Phi}_1 \mathbf{C}_2$$

$$= \mathbf{C}_1^T \mathbf{K}_{12} - \mathbf{C}_1^T \mathbf{K}_{11} \mathbf{C}_2$$